SEMESTER -3

DISCRETE MATHEMATICAL STRUCTURES

MAT 203	DISCRETE MATHEMATICAL STRUCTURES	CATEGORY	L	T	P	CREDITS
		BSC	3	1	0	4

Preamble:

The purpose of this course is to create awareness in students about the basic terminologies used in advanced courses in Computer Science and develop rigorous logical thinking for solving different kinds of problems in Computer Science. This course helps the learner to apply the theory and applications of elementary Counting Principles, Propositional Logic, Predicate Logic, Lattices, Generating Functions, Recurrence Relations and Algebraic Structures eventually in practical applications.

Prerequisite: A sound background in higher secondary school Mathematics

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Check the validity of predicates in Propositional and Quantified Propositional Logic using truth tables, deductive reasoning and inference theory on Propositional Logic **(Cognitive Knowledge Level: Apply)**
CO2	Solve counting problems by applying the elementary counting techniques - Rule of Sum, Rule of Product, Permutation, Combination, Binomial Theorem, Pigeonhole Principle and Principle of Inclusion and Exclusion **(Cognitive Knowledge Level: Apply)**
CO3	Classify binary relations into various types and illustrate an application for each type of binary relation, in Computer Science (**Cognitive Knowledge Level: Understand)**
CO4	Illustrate an application for Partially Ordered Sets and Complete Lattices, in Computer Science **(Cognitive Knowledge Level: Apply)**
CO5	Explain Generating Functions and solve First Order and Second Order Linear Recurrence Relations with Constant Coefficients **(Cognitive Knowledge Level: Apply)**
CO6	Illustrate the abstract algebraic systems - Semigroups, Monoids, Groups, Homomorphism and Isomorphism of Monoids and Groups **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓		✓						✓
CO4	✓	✓	✓	✓		✓						✓
CO5	✓	✓	✓	✓								✓
CO6	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of Series Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Fundamentals of Logic)

Mathematical logic - Basic connectives and truth table, Statements, Logical Connectives, Tautology, Contradiction. Logical Equivalence - The Laws of Logic, The Principle of duality, Substitution Rules . The implication - The Contrapositive, The Converse,The Inverse.

Logical Implication - Rules of Inference. The use of Quantifiers - Open Statement, Quantifier. Logically Equivalent – Contrapositive, Converse , Inverse , Logical equivalences and implications for quantified statement, Implications , Negation .

Module - 2 (Fundamentals of Counting Theory)

The Rule of Sum – Extension of Sum Rule . The Rule of Product - Extension of Product Rule . Permutations. Combinations. The Binomial Theorem (without proof). Combination with Repetition. The Pigeon hole Principle. The Principle of Inclusion and Exclusion Theorem (Without Proof) - Generalization of the Principle. Derangements.

Module - 3 (Relations and Functions)

Cartesian Product - Binary Relation. Function – domain , range-one to one function, Image-restriction. Properties of Relations- Reachability Relations, Reflexive Relations, Symmetric Relations, Transitive relations, Anti-symmetric Relations, Partial Order relations, Equivalence Relations, Irreflexive relations.

Partially ordered Set – Hasse Diagram, Maximal-Minimal Element, Least upper bound (lub), Greatest Lower bound(glb) (Topological sorting Algorithm- excluded). Equivalence Relations and Partitions - Equivalence Class.

Lattice - Dual Lattice , Sub lattice , Properties of glb and lub , Properties of Lattice , Special Lattice , Complete Lattice, Bounded Lattice, Completed Lattice , Distributive Lattice.

Module - 4 (Generating Functions and Recurrence Relations)

Generating Function - Definition and Examples , Calculation techniques, Exponential generating function. First order linear recurrence relations with constant coefficients – homogeneous, non-homogeneous Solution. Second order linear recurrence relations with constant coefficients, homogeneous, non-homogeneous Solution.

Module - 5 (Algebraic Structures)

Algebraic system-properties- Homomorphism and Isomorphism. Semi group and monoid – cyclic monoid , sub semi group and sub monoid, Homomorphism and Isomorphism of Semi group and monoids. Group- Elementary properties, subgroup, symmetric group on three symbols ,The direct product of two groups, Group Homomorphism, Isomorphism of groups, Cyclicgroup. Rightcosets - Leftcosets. Lagrange’s Theorem

Text Book

1. Discrete and Combinatorial Mathematics (An Applied Introduction), Ralph P Grimaldi, B V Ramana , 5th Edition, Pearson

Reference Books

1) Kenneth H. Rosen, Discrete Mathematics and Its Applications with Combinatorics and Graph Theory, Seventh Edition, MGH, 2011
2) Trembly J.P and Manohar R, “Discrete Mathematical Structures with Applications to Computer Science”, Tata Mc Graw Hill Pub. Co. Ltd., New Delhi, 2003.
3) Bernard Kolman, Robert C. Busby, Sharan Cutler Ross, “Discrete Mathematical Structures”, Pearson Education Pvt Ltd., New Delhi, 2003
4) Kenneth H .Rosen, “Discrete Mathematics and its Applications”, 5/e, Tata Mc Graw Hill Pub. Co. Ltd, New Delhi 2003
5) Richard Johnsonbaugh, “Discrete Mathematics”, 5/e, Pearson Education Asia, NewDelhi, 2002.
6) Joe L Mott, Abraham Kandel, Theodore P Baker,“Discrete Mathematics for Computer Scientists and Mathematicians”, 2/e, Prentice-Hall India, 2009.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Show that $R \vee M$, $\neg R \vee S$, $\neg M$, $\neg S$ cannot exist simultaneously (without using truth table)
2. Represent the following statement in symbolic form “Not every city in Canada is clean”.

Course Outcome 2 (CO2):

1. How many possible arrangements are there for the letters in MASSASAUGA in which 4 A’ s are together?
2. Find the number of integers between 1 and 1000 inclusive, which are not divisible by 5, 6 or 8

Course Outcome 3 (CO3):

1. If A = {1, 2, 3, 4}, give an example of a relation R that is reflexive and symmetric but not transitive.
2. Let Z be the set of integers. R is a relation called “Congruence Modulo 3 “ defined by R = { (x,y) /x ε Z,y ε Z, x - y is divisible by 3 } . Show that R is an equivalence relation.

Course Outcome 4 (CO4):

1. Assume A ={ a, b, c}. Let P(A) be its power set and ‘≤ ‘ be the subset relation on the power set. Draw the Hasse diagram of (P(A), ≤).
2. What is meant by Bounded Lattice ? Give an example.

Course Outcome 5 (CO5):

1. Solve $a_r - 3a_{r-1} - 4a_{r-2} = 3^r$ using Generating function method; Given $a_0 = 1$, $a_1 = 2$.
2. Find the generating function for the sequence 1, 3, 3^2, 3^3 …….

Course Outcome 6 (CO6):

1. Prove that the group { 1,-1, i, -i } is cyclic with generators i and –i.
2. State and prove Lagrange’s Theorem.

Model Question Paper

QP CODE:

Reg No:______________

Name :______________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

THIRD SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: MAT 203

Course Name: Discrete Mathematical Structures

Max.Marks :100 **Duration: 3 Hrs**

PART A

Answer all Questions. Each question carries 3 Marks

1. Show the following implication without constructing the truth table: (P ^ Q) => P→Q
2. Write the negation of the following statement. “ If I drive, then I will not walk”
3. What is pigeon hole principle? Explain. If you select any five numbers from 1 to 8 then prove that at least two of them will add up to 9 .
4. In how many ways can the letters of the word ALLAHABAD be arranged ?
5. Show that the divisibility relation ' / ' is a partial ordering on the set Z^{+}.
6. Consider the functions given by f(x) = 2x+3 and g(x) = x^2. Find (g o f) and (f o g).
7. What is meant by exponential generating function? Explain.
8. Provide one example of linear homogeneous recurrence relation. Mention the degree also.
9. What is a monoid ? Explain.
10. Let (A, .) be a group. Show that $(ab)^{-1} = b^{-1}a^{-1}$

(10 x 3 = 30 Marks)

PART B

(Answer any one Question from each Module. Each question carries 14 Marks)

11.

(a) Show that S ∨ R is tautologically implied by (PVQ) ∧ (P→R) ∧ (Q→S)

(6 marks)

(b) Show that from

(ii) $(\exists x)(F(x) \wedge S(x)) \rightarrow (y)(M(y) \rightarrow W(y))$.

(iii)$(\exists y)(M(y) \wedge \rceil W(y))$ the conclusion $(x)(F(x) \rightarrow \rceil S(x))$ follows.

(8 marks)

OR

12.

(a) Show that $(x)(P(x) \vee Q(x)) \Rightarrow ((x)P(x) \vee (\exists x) Q(x))$ using indirect method of proof .

(6 marks)

(b) Discuss indirect method of proof . Show that the following premises are inconsistent

(i) If Jack misses many classes through illness, then he fails high school.
(ii) If Jack fails high school, then he is uneducated.
(iii)If Jack reads a lot of books, then he is not uneducated.
(iv) Jack misses many classes through illness and reads a lot of books.

(8 marks)

13.

(a) Explain binomial theorem. Determine the coefficient of x^9y^3 in the expansion of $(x+y)^{12}$, $(x+2y)^{12}$ and $(2x- 3y)^{12}$ using binomial theorem.

(6 marks)

(b) How many 5 digit numbers can be formed from the digits 1,2,3,4,5 using the digits without repetition ?

(i) How many of them are even?
(ii) How many are even and greater than 30,000?

(8 marks)

OR

14.

(a) There are 8 guests in a party. Each guest brings a gift and receives another gift in return. No one is allowed to receive the gift they bought. How many ways are there to distribute the gifts?

(6 marks)

(b) Six papers are set in an examination of which two are mathematical. Only one examination will be conducted in a day. In how many different orders ,can the papers be arranged so that

(i) Two mathematical papers are consecutive?
(ii) Two mathematical papers are not consecutive?

(8 marks)

15.

(a) Let A = { 1,2,3,4,...11,12} and let R be the equivalence relation on A x A defined by (a,b) R (c,d) iff a+d = b+c. Prove that R is an equivalence relation and find the equivalence class of (2 ,5)

(8 marks)

(b) What is a chain lattice ? Explain. Also show that every chain is a distributive lattice.

(6 marks)

OR

16.

(a) Suppose f(x) = x+2 , g(x) = x-2, and h(x) = 3x for x ε R , where R is the set of real numbers. Find (g o f) , (f o g) , (f o f) and (g o g)

(8 marks)

(b) Let R and S be two relations on a set A . If R and S are symmetric, Prove that (R∩S) is also symmetric.

(6 marks)

17.

(a) Solve the recurrence relation $a_r - 7a_{r-1} + 10a_{r-2} = 0$ for $r \geq 2$; Given $a_0 = 0$; $a_1 = 41$ using generating functions

(8 marks)

(b) Solve the recurrence relation $a_r - 4a_{r-1} + 4a_{r-2} = (r+1)^2$ using generating function.

(6 marks)

OR

18.

(a) Solve $a_n - 3a_{n-1} + 2$; $a_0 = 1$ $n \geq 1$, using generating functions.

(8 marks)

(b) Use generating function to solve the following recurrence relation $a_n = 2a_{n-1} + 2^n$; with $a_0 = 2$.

(6 marks)

19.

(a) Prove that the set 'Q' of rational numbers other than 1 forms an abelian group with respect to the operation ' * ' defined by a * b = a+b -ab.

(8 Marks)

(b) Show that the direct product of two group is a group.

(6 Marks)

OR

20.

(a) Show that the subgroup of a cyclic group is cyclic.

(8 Marks)

(b) Let (A,*) be a group. Show that (A,*) is an abelian group if and only if $a^2 * b^2 = (a*b)^2$ for all 'a' and 'b' in A

(6 Marks)

TEACHING PLAN

No	Contents	No of Lecture Hrs
Module – 1 (Fundamentals of Logic) (9 hrs)		
1.1	Mathematical logic, Basic Connectives and Truth Table	1
1.2	Statements, Logical Connectives, Tautology, Contradiction	1
1.3	Logical Equivalence, The Laws of Logic	1
1.4	The Principle of duality, Substitution Rules	1
1.5	The implication, The Contrapositive, the Converse , the Inverse	1
1.6	Logical Implication,Rules of Inference, Logical Implication	1
1.7	The use of Quantifiers, Open Statement, Quantifier, Negation	1
1.8	Logically Equivalent, Contrapositive, The Converse, The Inverse	1
1.9	Logical Implications	1
Module - 2 (Fundamentals of Counting Theory) (9 hrs)		
2.1	The Pigeon-hole Principle	1
2.2	The Rule of Sum	1
2.3	Extension of Sum Rule	1
2.4	The Rule of Product	1
2.5	Extension of Product Rule , Permutations	1
2.6	Combinations, Combination with repetition	1
2.7	The Binomial Theorem	1
2.8	The Principle of Inclusion and Exclusion Theorem (Without Proof) Generalization of the Principle	1
2.9	Derangements	1
Module - 3 (Relations and Functions) (9 hrs)		
3.1	Cartesian Product, Binary Relation, Function, Domain, Range , One to One Function Image - Restriction	1
3.2	Properties, Reachability Relations, Reflexive Relations,Symmetric Relations,Transitive relations, Antisymmetric Relations.	1

3.3	Partial Order relations	1
3.4	Equivalence Relation, Irreflexive Relations.	1
3.5	Partially ordered Set, Hasse Diagram.	1
3.6	Maximal-Minimal Element, Least Upper bound, Greatest Lower Bound	1
3.7	Equivalence Relations and Partitions ,Equivalence Class	1
3.8	Lattice- Dual Lattice,sub lattice , Properties of glb and lub	1
3.9	Properties of Lattice , Special Lattice , Complete Lattice, Bounded Lattice, Completed Lattice, Distributive Lattice	1
Module - 4 (Generating Functions and Recurrence Relations) (9 hrs)		
4.1	Generating Function , Definition and Examples	1
4.2	Exponential Generating Function.	1
4.3	First Order Linear Recurrence Relations with Constant Coefficients (Lecture I)	1
4.4	First Order Linear Recurrence Relations with Constant Coefficients (Lecture II)	1
4.5	Homogeneous Solution	1
4.6	Non homogeneous Solution	1
4.7	Second order linear recurrence relations with constant coefficients	1
4.8	Homogeneous Solution	1
4.9	Non homogeneous Solution	1
Module - 5 (Algebraic Structures)(9 hrs)		
5.1	Algebraic System-Properties, Homomorphism and Isomorphism	1
5.2	Semi group , Monoid, Cyclic monoid	1

5.3	Sub semigroup and sub monoid	1
5.4	Homomorphism and Isomorphism of Semigroup, Monoids and Groups	1
5.5	Elementary Properties, Subgroup, Symmetric group on three symbols	1
5.6	The direct Product of two Groups	1
5.7	Group Homomorphism, Isomorphism, Cyclic group	1
5.8	Right coset, Left coset	1
5.9	Lagrange’s Theorem	1

CST 201	DATA STRUCTURES	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	3	1	0	4	2019

Preamble: This course aims at moulding the learner to understand the various data structures, their organization and operations. The course helps the learners to assess the applicability of different data structures and associated algorithms for solving real world problem which requires to compare and select appropriate data structures to solve the problem efficiently. This course introduces abstract concepts for data organization and manipulation using data structures such as stacks, queues, linked lists, binary trees, heaps and graphs for designing their own data structures to solve practical application problems in various fields of Computer Science.

Prerequisite: Topics covered under the course Programming in C (EST 102)

CO1	Design an algorithm for a computational task and calculate the time/space complexities of that algorithm **(Cognitive Knowledge Level: Apply)**
CO2	Identify the suitable data structure (array or linked list) to represent a data item required to be processed to solve a given computational problem and write an algorithm to find the solution of the computational problem **(Cognitive Knowledge Level: Apply)**
CO3	Write an algorithm to find the solution of a computational problem by selecting an appropriate data structure (binary tree/graph) to represent a data item to be processed **(Cognitive Knowledge Level: Apply)**
CO4	Store a given dataset using an appropriate Hash Function to enable efficient access of data in the given set (**Cognitive Knowledge Level: Apply**)
CO5	Select appropriate sorting algorithms to be used in specific circumstances (**Cognitive** Knowledge Level: **Analyze**)
CO6	Design and implement Data Structures for solving real world problems efficiently (**Cognitive Knowledge Level: Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓						✓
CO2	✓	✓	✓	✓		✓						✓
CO3	✓	✓	✓	✓		✓						✓
CO4	✓	✓	✓	✓		✓						✓
CO5	✓	✓	✓	✓		✓						✓
CO6	✓	✓	✓	✓		✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test1 (Percentage)	**Test2 (Percentage)**	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40

Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

SYLLABUS

Module 1

Basic Concepts of Data Structures

System Life Cycle, Algorithms, Performance Analysis, Space Complexity, Time Complexity, Asymptotic Notation, Complexity Calculation of Simple Algorithms

Module 2

Arrays and Searching

Polynomial representation using Arrays, Sparse matrix, Stacks, Queues-Circular Queues, Priority Queues, Double Ended Queues, Evaluation of Expressions

Linear Search and Binary Search

Module 3

Linked List and Memory Management

Self Referential Structures, Dynamic Memory Allocation, Singly Linked List-Operations on Linked List. Doubly Linked List, Circular Linked List, Stacks and Queues using Linked List, Polynomial representation using Linked List

Memory allocation and de-allocation-First-fit, Best-fit and Worst-fit allocation schemes

Module 4

Trees and Graphs

Trees, Binary Trees-Tree Operations, Binary Tree Representation, Tree Traversals, Binary Search Trees- Binary Search Tree Operations

Graphs, Representation of Graphs, Depth First Search and Breadth First Search on Graphs, Applications of Graphs

Module 5

Sorting and Hashing

Sorting Techniques – Selection Sort, Insertion Sort, Quick Sort, Merge Sort and Heap Sort

Hashing- Hashing Techniques, Collision Resolution, Overflow handling, Hashing functions – Mid square, Division, Folding, Digit Analysis

Text Book

1. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Universities Press, Fundamentals of Data Structures in C

Reference Books

1. Samanta D., Classic Data Structures, Prentice Hall India.
2. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C, 2/e, Cengage Learning.
3. Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication.
4. Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill.
5. Peter Brass, Advanced Data Structures, Cambridge University Press.
6. Lipschuts S., Theory and Problems of Data Structures, Schaum's Series.
7. Wirth N., Algorithms + Data Structures = Programs, Prentice Hall.
8. Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI.
9. Martin Barrett, Clifford Wagner, C And Unix: Tools For Software Design, John Wiley.

Sample Course Level Assessment Questions

Course Outcome1(CO1): Write an algorithm for matrix multiplication and calculate its time complexity.

Course Outcome 2(CO2): How a linked list can be used to represent the polynomial $5x^4y^6+24x^3y^4-17x^2y^3+15xy^2+45$.Write an algorithm to add two Bivariate polynomials represented using linked list.

Course Outcome 3(CO3): Create a Binary search Tree with node representing the following sequence 14, 15, 4, 18, 9, 16, 20, 17, 3, 7, 5, 2 and perform inorder, preorder and postorder traversals on the above tree and print the output.

Course Outcome 4(CO4): The size of a hash table is 7. The index of the hash table varies from 0 to 6. Consider the keys 89, 18, 49, 58, 25 in the order. Show how the keys are stored in the hash table using Linear probing.

Course Outcome 5(CO5): In what circumstances does Quick Sort perform over Merge sort.

Course Outcome 6(CO6): Design a reservation system for railways that include waiting list. If the reservation is full "Display reservation full" and put the passenger in in waiting list and give a waiting list number. If a passenger cancels the ticket, then the seat should be automatically allocated to the first passenger in the waiting list.

Model Question Paper

QP CODE: PAGES:3

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY THIRD SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 201

Course Name: DATA STRUCTURES

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Calculate the frequency count of the statement x = x+1; in the following code segment

```
for (i  = 0; i< n; i++)
for (j = 0; j< n; j*=2)
    x = x + 1;
```

2. What is the relevance of verification in System Life Cycle?
3. Write an algorithm to insert a new element in a particular position of an array.

4. Convert the expression ((A/(B-D+E))*(F-G)*H) to postfix form. Show each step in the conversion including the stack contents

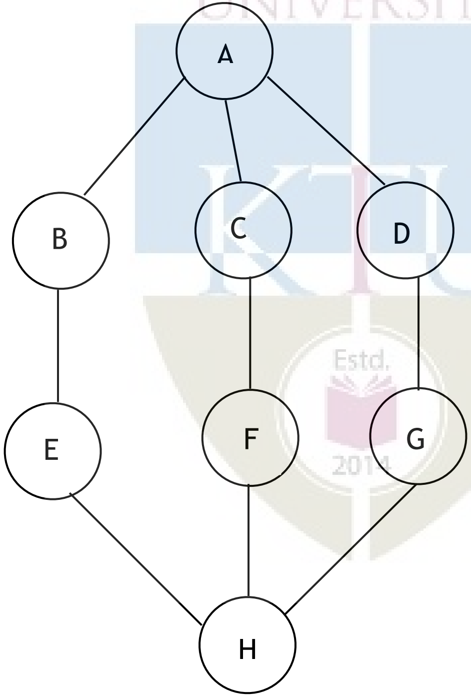
5. Write an algorithm to count the number of occurrences of a character in a linked list (each node contains only one character)

6. Write an algorithm for best-fit method of memory allocation

7. Draw the binary tree whose sequential representation is given below

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	-	D	E	-	-	-	-	F	G	-	-	-

8. Find the Depth First Search of the following Graph



9. Write an algorithm to arrange n numbers in nonincreasing order.

10. Let the size of a hash table is 10. The index of the hash table varies from 0 to 9. Assume the keys 73, 54, 15, 48, 89, 66, 37, 18, 41, 22, 62 are mapped using modulo operator. Show how the keys are distributed using chaining method.

Part B

Answer any one Question from each module. Each question carries 14 Marks

11. a) Explain the System Life Cycle in detail (10)

b) How the performance of an algorithm is evaluated? (4)

OR

12. a) Write algorithms for Linear Search and Binary Search and Compare their time complexities (10)

b) Between O(nlogn) and O(logn) which one is better and why? (4)

13. a) Write algorithms to insert and delete elements from a double ended queue. Demonstrate with examples (10)

b) Compare and contrast Circular Queue with a Normal Queue (4)

OR

14. a) Write an algorithm to insert and delete elements from a Priority Queue (8)

b) Discuss an algorithm to convert an infix expression to a prefix expression (6)

15. a) Write an algorithm to multiply two polynomials represented using linked list (10)

b) How doubly linked list can be used to find palindromes ? (4)

OR

16. a) How is memory compaction (de-allocation) done in memory management ? (8)

b) Discuss the advantages and disadvantages of First-fit, Best-fit and Worst-fit allocation schemes (6)

17. a) List the properties of Binary Search Tree. Write an algorithm to search an element from a Binary Search Tree (10)

b) Write an iterative algorithm for in-order traversal of a Binary Tree (4)

OR

18. a) Give algorithms for DFS and BFS of a graph and explain with examples (8)

b) How graphs can be represented in a Computer? (6)

19. a) Write algorithms for Merge sort and Quick Sort. (10)

b) Illustrate the working of Quick sort on the following input 38, 8, 0, 28, 45, -12, 89, 66, 42 (4)

OR

20. a) With examples discuss the different hash functions used for hashing (10)

b) Apply the hash function h(x) = x mod 7 for linear probing on the data 2341, 4234, 2839, 430, 22, 397, 3920 and show the resulting hash table (4)

Teaching Plan		
Module 1 :Basic Concepts of Data Structures		(5 hours)
1.1	System Life Cycle,	1 hour
1.2	Algorithms , Performance Analysis	1 hour
1.3	Space Complexity, Time Complexity	1 hour
1.4	Asymptotic Notation (Big O Notation)	1 hour
1.5	Complexity Calculation of Simple Algorithms	1hour
Module 2 :**Arrays and Searching**		(10 hours)
2.1	Polynomial representation using Arrays	1 hour
2.2	Sparse matrix (Lecture 1)	1 hour
2.3	Sparse matrix (Lecture 2)	1 hour

2.4	Stacks	1 hour
2.5	Queues, Circular Queues	1 hour
2.6	Priority Queues,	1 hour
2.7	Double Ended Queues,	1 hour
2.8	Conversion and Evaluation of Expressions (Lecture 1)	1 hour
2.9	Conversion and Evaluation of Expressions (Lecture 2)	1 hour
2.10	Linear Search and Binary Search	1 hour
Module 3 : **Linked List and Memory Management**		(12 hours)
3.1	Self Referential Structures	1 hour
3.2	Dynamic Memory Allocation	1 hour
3.3	Singly Linked List-Operations on Linked List,	1 hour
3.4	Doubly Linked List	1 hour
3.5	Circular Linked List	1 hour
3.6	Stacks using Linked List	1 hour
3.7	Queues using Linked List	1 hour
3.8	Polynomial representation using Linked List (Lecture 1)	1 hour
3.9	Polynomial representation using Linked List (Lecture2)	1 hour
3.10	Memory de-allocation	1 hour
3.11	Memory allocation-First-fit	1 hour
3.12	Best-fit and Worst-fit allocation schemes	1hour
Module 4 :**Trees and Graphs**		(8 hours)
4.1	Trees, Binary Trees	1hour
4.2	Tree Operations, Binary Tree Representation,	1hour
4.3	Tree Traversals	1hour
4.4	Binary Search Trees	1hour
4.5	Binary Search Tree Operations	1hour
4.6	Graphs, Representation of Graphs	1hour

4.7	Depth First Search and Breadth First Search on Graphs	1hour
4.8	Applications of Graphs	1hour
Module 5 : **Sorting and Hashing**		(10 hours)
5.1	Sorting Techniques – Selection Sort	1hour
5.2	Insertion Sort	1hour
5.3	Quick Sort	1hour
5.4	Merge Sort	1hour
5.5	Heap Sort	1hour
5.6	Hashing- Hashing Techniques	1hour
5.7	Collision Resolution	1hour
5.8	Overflow handling	1hour
5.9	Hashing functions – Mid square and Division methods	1hour
5.10	Folding and Digit Analysis methods	1hour

CST 203	**Logic System Design**	**Category**	**L**	**T**	**P**	**Credit**	**Year of Introduction**
		PCC	**3**	**1**	**0**	**4**	**2019**

Preamble: The objective of the course is to familiarize learners with the basic concepts of Boolean algebra and digital systems. This course covers the design of simple combinational and sequential logic circuits, representation and arithmetic algorithms for Binary, BCD (Binary Coded Decimal) and Floating point numbers which in turn are helpful in understanding organization & design of a computer system and understanding how patterns of ones and zeros can be used to store information on computers, including multimedia data.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO#	**CO**
CO1	Illustrate decimal, binary, octal, hexadecimal and BCD number systems, perform conversions among them and do the operations - complementation, addition, subtraction, multiplication and division on binary numbers (**Cognitive Knowledge level: Understand**)
CO2	Simplify a given Boolean Function and design a combinational circuit to implement the simplified function using Digital Logic Gates (**Cognitive Knowledge level: Apply**)
CO3	Design combinational circuits - Adders, Code Convertors, Decoders, Magnitude Comparators, Parity Generator/Checker and design the Programmable Logic Devices - ROM and PLA. **(Cognitive Knowledge level: Apply)**
CO4	Design sequential circuits - Registers, Counters and Shift Registers. **(Cognitive Knowledge level: Apply)**
CO5	Use algorithms to perform addition and subtraction on binary, BCD and floating point numbers **(Cognitive Knowledge level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓	✓		✓						✓
CO3	✓	✓	✓	✓		✓						✓
CO4	✓	✓	✓	✓		✓						✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern:

Bloom's Category	**Test 1 (%)**	**Test 2 (%)**	**End Semester Examination Marks (%)**
Remember	20	20	20
Understand	35	35	35
Apply	45	45	45
Analyse			
Evaluate			
Create			

Mark Distribution:

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

SYLLABUS

Module I

Number systems, Operations & Codes

Decimal, Binary, Octal and Hexadecimal Number Systems- Number Base Conversions. Addition, Subtraction, Multiplication and Division of binary numbers. Representation of negative numbers- Complements, Subtraction with complements. Addition and subtraction of BCD, Octal and Hexadecimal numbers. Binary codes- Decimal codes, Error detection codes, Reflected code, Character coding schemes – ASCII, EBCDIC.

Module II

Boolean Algebra

Postulates of Boolean Algebra. Basic theorems and Properties of Boolean Algebra. Boolean Functions - Canonical and Standard forms. Simplification of Boolean Functions- Using Karnaugh- Map Method (upto five variables), Don’t care conditions, Product of sums simplification, Tabulation Method. Digital Logic Gates- Implementation of Boolean functions using basic and universal gates.

Module III

Combinational Logic Circuits

Design Procedure & Implementation of combinational logic circuits- Binary adders and subtractors, Binary Parallel adder, Carry look ahead adder, BCD adder, Code converter, Magnitude comparator, Decoder, Demultiplexer, Encoder, Multiplexer, Parity generator/ Checker.

Module IV

Sequential logic circuits:

Flip-flops- SR, JK, T and D. Triggering of flip-flops- Master slave flip- flops, Edge- triggered flip- flops. Excitation table and characteristic equation. Registers- register with parallel load. Counter design: Asynchronous counters- Binary and BCD counters, timing sequences and state diagrams. Synchronous counters- Binary Up- down counter, BCD counter.

Module V

Shift registers

Shift registers – Serial In Serial Out, Serial In Parallel Out, Bidirectional Shift Register with Parallel load. Ring counter. Johnson counter- timing sequences and state diagrams.

Arithmetic algorithms

Algorithms for addition and subtraction of binary numbers in signed magnitude and 2's complement representations. Algorithm for addition and subtraction of BCD numbers. Representation of floating point numbers, Algorithm for addition and subtraction of floating point numbers.

Programmable Logic devices

ROM. Programmable Logic Array(PLA)- Implementation of simple circuits using PLA.

Text Books:

1. M. Morris Mano, Digital Logic & Computer Design, 4/e, Pearson Education, 2013
2. Thomas L Floyd, Digital Fundamentals, 10/e, Pearson Education, 2009.
3. M. Morris Mano, Computer System Architecture, 3/e, Pearson Education, 2007.

Reference Books:

1. M. Morris Mano, Michael D Ciletti , Digital Design With An Introduction to the Verilog HDL, 5/e, Pearson Education, 2013.
2. Donald D Givone, Digital Principles and Design, Tata McGraw Hill, 2003

Sample Course Level Assessment Questions

Course Outcome1(CO1): Perform the following number base conversions:

a) $(250.55)_{10}$ to Hexadecimal b) $(357)_8$ to Decimal

Course Outcome 2(CO2): Given a Boolean function F and don't care conditions D, using Karnaugh map obtain the simplified expression in (i) SOP and (ii) POS:

$$F(A,B,C,D) = A'B'D' + A'CD + A'BC$$
$$D(A,B,C,D) = A'BC'D + ACD + AB'D$$

Course Outcome 3(CO3): Design a BCD to Excess-3 Code Convertor.

Course Outcome 4(CO4): Design a 4- bit binary ripple counter.

Course Outcome 5(CO5): Demonstrate floating-point addition algorithm.

Model Question Paper

QP CODE: PAGES: 2

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY THIRD SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 203

Course name : LOGIC SYSTEM DESIGN

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Represent the decimal numbers $(459)_{10}$ and $(859)_{10}$ in hexadecimal and perform addition of these hexadecimal numbers.
2. Subtract $(1101)_2$ from $(11010)_2$ using: i) $2's$ complement and ii) $1's$ complement arithmetic.
3. Find the dual and complement of the boolean function $F = AB' + B(A + B')$.
4. Using K-map, reduce the expression: $AB + ABC + ABC + BC$.
5. Design a half subtractor with NAND gates only.
6. Design a combinational circuit that multiplies an input decimal digit by 5 represented in BCD. The output is also in BCD. Show that the outputs can be obtained from the input lines without using any logic gates.
7. Differentiate between ripple counter and synchronous counter.
8. Construct D flip- flop using NAND gates. Also give its truth table.
9. Explain how a shift register is used for serial data transfer?
10. Write short notes on ROM.

PART-B

(Answer any one full question from each module) (14X5=70)

11. (a) Perform the following operations using 2's complement arithmetic: (8)

(i) $88_{10} + (-37)_{10}$ (ii) $(-20)_{10} + (-12)_{10}$

(b) Perform the following base conversions: (i) $(101011.11)_2$ to octal (6)
(ii) $(3F9B)_{16}$ to binary (iii) $(121)_{10}$ to binary (iv) $(3077)_8$ to binary

OR

12. (a) Find the 12 bit $2's$ complement representation of the following decimal numbers. (6)

(i) – 97 (ii) – 224 (iii) - 197.5

(b) Perform the following operations (8)

(i) $(520)_8 + (488)_8$ (ii) $(520)_{16} - (488)_{16}$

13. (a) Prove that (i) $AB + A(B + C) + B(B + C) = B + AC$ (4)
(ii) $AB + A(B + C) + B(B + D) = A$

(b) Using K-map, simplify the Boolean function F in sum of products form, using the don't care conditions d: (10

$$F(w, x, y, z) = w'(x'y + x'y' + xyz) + x'z'(y + w)$$
$$d(w, x, y, z) = w'x(y'z + yz') + wyz$$

OR

14. (a) Simplify the following expressions using Karnaugh- map method. (8)

(i) $F = \Sigma(0,2,4,6,9,11,13,15,17,21,25,27,29,31)$
(ii) $F = \Pi(0,2,5,7)$

(b) Convert the following to the other canonical form: (6)

(i) $F(x, y, z, a) = \sum(1,3,7)$

(ii) $F(x, y, z) = \Pi(0,3,6,7)$

(iii) $F(A, B, C, D) = \Pi(0,1,2,3,4,6,12)$

15. (a) Implement Full adder circuit using NAND gate only. (4)

(b) Design a code converter for converting BCD to Excess 3 code (10)

OR

16. (a) With a neat diagram explain 4-bit carry look-ahead adder. (6)

(b) Design a Gray to binary code converter using a 4x1 MUX. Draw the circuit diagram and explain. (8)

17. (a) Design a counter that count the states 0,3,5,6,0… using T flip- flops. (10)

(b) Write the characteristics equation, excitation table of JK, T and D flipflop. (4)

OR

18. (a) Explain race around condition and how it can be avoided. (6)

(b) Design a synchronous Binary Up-Down Counter. (8)

19. (a) With a neat diagram explain universal shift register. (8)

(b) Explain Johnson Counter with timing diagram. (6)

OR

20. (a) Write algorithm for floating point addition and subtraction. (8)

(b) Implement the functions $Y_1 = AB'C' + AB'C + ABC$ and $Y_2 = BC + AC$ using minimum gates Programmable Logic Array. (6)

Teaching Plan

Module 1: Number systems, Operations & Codes (No algorithms)		**(7 hours)**
1.1	**Number Systems**: Decimal, Binary, Octal and Hexadecimal number systems, Number Base Conversions.	**1 hour**
1.2	**Binary Arithmetic:** Addition, Subtraction, Multiplication & Division of Binary Numbers. (Lecture 1)	**1 hour**
1.3	Addition, Subtraction, Multiplication & Division of Binary Numbers. (Lecture 2)	**1 hour**
1.4	Representation of Negative Numbers- Complements, subtraction with complements.	**1 hour**
1.5	**BCD Arithmetic:** Addition and Subtraction of BCD Numbers	**1 hour**
1.6	**Octal and Hexadecimal Arithmetic**: Addition & Subtraction of Octal and Hexadecimal Numbers.	**1 hour**

1.7	**Binary Codes:** Decimal Codes, Error detection codes, Reflected code, Character Coding Schemes-ASCII, EBCDIC	**1 hour**
Module 2: Boolean Algebra		**(9 hours)**
2.1	**Introduction to Boolean Algebra:** Postulates of Boolean Algebra	**1 hour**
2.2	Basic theorems and Properties of Boolean Algebra	**1 hour**
2.3	**Boolean Functions:** Canonical and Standard Forms	**1 hour**
2.4	**Simplification of Boolean Functions**: Karnaugh -Map Method (upto five variables), Don't care conditions (Lecture 1)	**1 hour**
2.5	**Simplification of Boolean Functions**: Karnaugh -Map Method (upto five variables), Don't care conditions (Lecture 2)	**1 hour**
2.6	**Product of sums simplification**	**1 hour**
2.7	**Tabulation method**	**1 hour**
2.8	**Digital Logic Gates:** AND, OR, NOT, NAND, NOR, XOR, XNOR, Implementation of Boolean functions using basic and universal gates. (Lecture 1)	**1 hour**
2.9	**Digital Logic Gates:** AND, OR, NOT, NAND, NOR, XOR, XNOR, Implementation of Boolean functions using basic and universal gates. (Lecture 2)	**1 hour**
Module 3: Combinational Logic Circuits		**(9 hours)**
3.1	Design Procedure & Implementation of Combinational Circuits	**1 hour**
3.2	**Binary Adders:** Implementation of Half Adder, Full Adder	**1 hour**
3.3	**Binary Subtractors:** Implementation of Half Subtractor, Full Subtractor	**1 hour**
3.4	Implementation of Binary Parallel Adder ,Carry look ahead Adder, BCD Adder (Lecture 1)	**1 hour**
3.5	Implementation of Binary Parallel Adder ,Carry look ahead Adder, BCD Adder (Lecture 2)	**1 hour**

3.6	**Implementation of Various Combinational Circuits:** Code Converters, Magnitude Comparator	**1 hour**
3.7	Implementation of Decoder, Demultiplexer	**1 hour**
3.8	Implementation of Encoder, Multiplexer	**1 hour**
3.9	Implementation of Parity Generator/Checker	**1 hour**
Module 4: Sequential logic circuits:		**(9 hours)**
4.1	**Flip flops:** SR, JK, T and D flip- flops (Lecture 1)	**1 hour**
4.2	SR, JK, T and D flip- flops (Lecture 2)	**1 hour**
4.3	Triggering of flip-flops- Master slave flip- flop, Edge- triggered flip-flops (Lecture 1)	**1 hour**
4.4	Triggering of flip-flops- Master slave flip- flop, Edge- triggered flip-flops (Lecture 2)	**1 hour**
4.5	Excitation table and characteristic equations of flip- flops	**1 hour**
4.6	**Registers-** Register with parallel load	**1 hour**
4.7	**Counter Design:** Asynchronous counters- Binary and BCD counters- timing sequences and state diagrams. (Lecture 1)	**1 hour**
4.8	Asynchronous counters- Binary and BCD counters- timing sequences and state diagrams. (Lecture 2)	**1 hour**
4.9	Synchronous counters- Binary Up- down counter, BCD counter	**1 hour**
Module 5: Shift registers, Arithmetic algorithms & PLD’s		**(11 hours)**
5.1	**Shift Registers** - Serial In Serial Out, Serial In Parallel Out.	**1 hour**
5.2	Bidirectional Shift Register with Parallel load	**1 hour**

5.3	**Shift register counters** - Ring Counter, Johnson Counter- timing sequences and state diagrams	**1 hour**
5.4	**Arithmetic Algorithms:** Algorithm for addition and subtraction of binary numbers in Signed magnitude and 2's complement representations (Lecture 1)	**1 hour**
5.5	Algorithm for addition and subtraction of binary numbers in Signed magnitude and 2's complement representations (Lecture 2)	**1 hour**
5.6	Algorithm for addition and subtraction of BCD numbers	**1 hour**
5.7	Representation of floating point numbers (IEEE Standard representations).	**1 hour**
5.8	Algorithms for floating point addition and subtraction	**1 hour**
5.9	**Programmable Logic devices** - ROM	**1 hour**
5.10	PLA, Implementation of simple circuits using PLA(Lecture 1)	**1 hour**
5.11	PLA, Implementation of simple circuits using PLA(Lecture 2)	**1 hour**

CST 205	**OBJECT ORIENTED PROGRAMMING USING JAVA**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**	**YEAR OF INTRODUCTION**
		PCC	**3**	**1**	**0**	**4**	**2019**

Preamble: The purpose of this course is to enable learners to solve problems by breaking it down to object level while designing software and to implement it using Java. This course covers Object Oriented Principles, Object Oriented Programming in Java, Inheritance, Exception handling, Event handling, multithreaded programming and working with window-based graphics. This course helps the learners to develop Desktop GUI Applications, Mobile applications, Enterprise Applications, Scientific Applications and Web based Applications.

Prerequisite: Topics covered under the course PROGRAMMING IN C (EST 102)

Course Outcomes: After the completion of the course the student will be able to

CO1	Write Java programs using the object oriented concepts - classes, objects, constructors, data hiding, inheritance and polymorphism (Cognitive Knowledge Level: **Apply**)
CO2	Utilise datatypes, operators, control statements, built in packages & interfaces, Input/ Output Streams and Files in Java to develop programs (Cognitive Knowledge Level: **Apply**)
CO3	Illustrate how robust programs can be written in Java using exception handling mechanism (Cognitive Knowledge Level: **Understand**)
CO4	Write application programs in Java using multithreading and database connectivity (Cognitive Knowledge Level: **Apply**)
CO5	Write Graphical User Interface based application programs by utilising event handling features and Swing in Java (Cognitive Knowledge Level: **Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓						✓		✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test1 (Marks %)	**Test2 (Marks %)**	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

SYLLABUS

Object Oriented Programming Using Java

Module 1

Introduction:

Approaches to Software Design - Functional Oriented Design, Object Oriented Design, Case Study of Automated Fire Alarm System.

Object Modeling Using Unified Modeling Language (UML) – Basic Object Oriented concepts, UML diagrams, Use case model, Class diagram, Interaction diagram, Activity diagram, State chart diagram.

Introduction to Java **-** Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. Java Virtual Machine (JVM), Java compiler, Bytecode, Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues.

Module 2

Core Java Fundamentals:

Primitive Data types - Integers, Floating Point Types, Characters, Boolean. Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.

Operators **-** Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.

Control Statements - Selection Statements, Iteration Statements and Jump Statements.

Object Oriented Programming in Java - Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods, Constructors, ***this*** Keyword, Method Overloading, Using Objects as Parameters, Returning Objects, Recursion, Access Control, Static Members, Final Variables, Inner Classes, Command Line Arguments, Variable Length Arguments.

Inheritance - Super Class, Sub Class, The Keyword ***super***, protected Members, Calling Order of Constructors, Method Overriding, the Object class, Abstract Classes and Methods, using ***final*** with Inheritance.

Module 3

More features of Java:

Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages, Interfaces.

Exception Handling **-** Checked Exceptions, Unchecked Exceptions, ***try*** Block and ***catch*** Clause, Multiple ***catch*** Clauses, Nested ***try*** Statements, ***throw***, ***throws*** and ***finally***.

Input/Output - I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class, Object Streams and Serialization, Working with Files.

Module 4

Advanced features of Java:

Java Library - String Handling – String Constructors, String Length, Special String Operations - Character Extraction, String Comparison, Searching Strings, Modifying Strings, using valueOf(), Comparison of StringBuffer and String.

Collections framework **-** Collections overview, Collections Interfaces- Collection Interface, List Interface.

Collections Class – ArrayList class. Accessing a Collection via an Iterator.

Event handling **-** Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Model.

Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread, Creating Multiple Threads, Synchronization, Suspending, Resuming and Stopping Threads.

Module 5

Graphical User Interface and Database support of Java:

Swings fundamentals - Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Swing Packages, Event Handling in Swings, Swing Layout Managers, Exploring Swings –JFrame, JLabel, The Swing Buttons, JTextField.

Java DataBase Connectivity (JDBC) **-** JDBC overview, Creating and Executing Queries – create table, delete, insert, select.

Text Books:

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.
2. Rajib Mall, Fundamentals of Software Engineering, 4th edition, PHI, 2014.
3. Paul Deitel, Harvey Deitel, Java How to Program, Early Objects 11th Edition, Pearson, 2018.

Reference Books:

1. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.
2. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.
3. Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
4. Barclay K., J. Savage, Object Oriented Design with UML and Java, Elsevier, 2004.
5. Sierra K., Head First Java, 2/e, O'Reilly, 2005.
6. Balagurusamy E., Programming JAVA a Primer, 5/e, McGraw Hill, 2014.

Sample Course Level Assessment Questions

Course Outcome1(CO1): For the following passage develop UML diagrams and then implement it as a Java program in accordance with your UML design.

Passage: College Office collects semester fee and college bus fee for each student. A clerk at the college office collects the fees from each student. The bus fee is calculated depending on the distance of the corresponding bus stop from the college. The semester fee varies depending upon the semester as well as branch of each student. Students are supposed to pay the fees in full. Economically backward students are eligible for 50% discount in semester fee. The consolidated fees receipt is issued to each student by the clerk, which contains the student name, admission number, semester and branch of student along with details of fees collected. Students can log in and view the details of fees remitted and dues if any. The system allows students and clerk level login to the system. Clerk is able to view reports of each class showing status of fees payment of each student.

Course Outcome 2(CO2): Write a Java program to evaluate a post fix expression containing two operands and a single operator using stack. Stack should be implemented as a separate entity so as to reflect OOP concepts.

Course Outcome 3(CO3): Write a program to demonstrate the start, run, sleep and join methods in Thread class.

Course Outcome 4(CO4): Write a GUI based program with separate buttons to add, delete and display student details i.e. name, student ID, current semester and branch of study based on student ID.

Course Outcome 5(CO5): Using Swing create a JFrame with a JLabel and two JButtons. Set the texts of JButtons as “Yes” and “No” respectively. Set the JLabel’s text to the text of the button currently being pressed. Initially the JLabel’s text is blank.

Model Question Paper

QP CODE: PAGES:3

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

THIRD SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 205

Course Name: Object Oriented Programming using Java

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Briefly explain the portable, secure and robust features of Java.

2. Describe the concepts of object and class with a suitable Java program.

3. Explain the concept of method overriding with an example.

4. What is the use of the keyword ***final*** in Java?

5. Explain the concept of streams.

6. Explain any two applications of Serialization.

7. Distinguish the usage of "==" and ***equals()*** method when comparing String type?

8. What are Collections in Java? Explain any one Collection interface in Java.

9. Explain any two properties of Swing components in Java.

10. Explain JLabel component. With suitable examples explain any two of its constructors.

Part B

Answer any one question completely from each module

11\.

(a) Describe in detail any three Object Oriented Programming principles. Illustrate with suitable examples.

(9)

(b) What is Java Runtime Environment? What is the role of Java Virtual Machine in it?

(5)

OR

12.

(a) Compare and contrast Java standard edition and Java enterprise edition.

(5)

(b) Why is Java considered to be platform independent? What is the role of Bytecode in making Java platform independent?

(9)

13.

(a) Explain in detail the primitive data types in Java.

(8)

(b) Explain automatic type conversion in Java with an example. What are the two conditions required for it?

(6)

OR

14.

(a) Using a suitable Java program explain the difference between ***private*** and ***public*** members in the context of inheritance.

(8)

(b) Is it possible to use the keyword ***super*** within a static method? Give justification for your answer.

(6)

15.

(a) Explain in detail about byte streams and character streams with suitable code samples.

(6)

(b) Describe in detail about exception handling, ***try*** block and ***catch*** clause with the help of a suitable Java program.

(8)

OR

16.

(a) Explain object streams in Java. Explain the role of Serializable interface with a suitable code sample.

(8)

(b) Explain ***throw***, ***throws*** and ***finally*** constructs with the help of a Java program.

(6)

17.

(a) Describe in detail the creation of a thread using the Runnable interface and the Thread class with suitable examples.

(10)

(b) Explain List Interface. Mention any two exceptions thrown by its methods.

(4)

OR

18.

(a) Explain in detail the Delegation Event model for event handling in Java.

(7)

(b) Write a simple program by extending appropriate class to demonstrate the working of threads in java.

(7)

19.

(a) Write a Java program to demonstrate the use of JLabel and JButton by adding them to JFrame.

(7)

(b) Explain step-by-step procedure of using Java DataBase Connectivity in Java programs.

(7)

OR

20.

(a) Explain the class hierarchy of Java Swing components.

(7)

(b) Write a Java Program to create a student table and to add student details to it using JDBC.

(7)

Teaching Plan		
Module 1 : Introduction		(8 hours)
1.1	Approaches to Software Design- Functional Oriented Design, Object-Oriented Design, Case Study of Automated Fire Alarm System.	1 hour
1.2	Object Modeling Using UML – Basic object oriented concepts	1 hour
1.3	Basic object oriented concepts	1 hour
1.4	UML diagrams, Use case model	1hour
1.5	Class diagram, Interaction diagram	1hour
1.6	Activity diagram, State chart diagram	1hour
1.7	Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. JVM, Java compiler, Bytecode	1hour
1.8	Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues	1hour
Module 2: Core Java Fundamentals		(11 hours)
2.1	Core Java Fundamentals**:** Primitive Data types, Integers, Floating Point Types, Characters, Boolean	1 hour
2.2	Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.	1 hour
2.3	Operators: Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.	1 hour
2.4	Control Statements: Selection Statements, Iteration Statements and Jump Statements.	1 hour
2.5	Object Oriented Programming in Java**:** Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods	1 hour
2.6	Constructors, ***this*** Keyword, Method Overloading, Using Objects as Parameters	1 hour
2.7	Returning Objects, Recursion, Access Control, static Members	1 hour

2.8	Final Variables, Inner Classes, Command-Line Arguments, Variable Length Arguments	1 hour
2.9	Inheritance : Super class, Sub class, the keywords ***super***, ***protected*** Members,	1 hour
2.10	Calling Order of Constructors, Method Overriding, the Object class,	1 hour
2.11	Abstract Classes and Methods, Using ***final*** with Inheritance	1 hour
Module 3: More features of Java		(8 hours)
3.1	Packages and Interfaces: Defining Package, CLASSPATH, Access Protection, Importing Packages	1 hour
3.2	Interfaces	1 hour
3.3	Input / Output: I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class	1 hour
3.4	Object Streams and Serialization	1 hour
3.5	Working with Files	1 hour
3.6	Exception Handling**:** Checked Exceptions, Unchecked Exceptions, ***try*** Block and ***catch*** Clause	1 hour
3.7	Multiple ***catch*** Clauses, Nested ***try*** Statements	1 hour
3.8	***throw***, ***throws*** and ***finally***	1 hour
Module 4:Advanced features of Java		(10 hours)
4.1	Java Library: String Handling – String Constructors, String Length, Special String Operations	1hour
4.2	Character Extraction, String Comparison, Searching Strings, Modifying Strings Using valueOf( ), Comparison of String Buffer and String.	1hour
4.3	Collections framework – Collections overview, Collections Interfaces- Collection Interface	1hour
4.4	List Interface, Collections Class – ArrayList Class	1hour
4.5	Accessing Collections via an Iterator.	1hour
4.6	Event handling**:** Event Handling Mechanisms, Delegation Event Model	1hour
4.7	Delegation Event Model, Event Classes	1hour

4.8	Sources of Events, Event Listener Interfaces, Using the Delegation Model	1hour
4.9	Multithreaded Programming: The Java Thread Model, The Main Thread, Creating Thread	1hour
4.10	Creating Multiple Threads, Synchronization, Suspending, Resuming and Stopping Threads.	1hour
Module 5: Graphical User Interface and Database support of Java		(8 hours)
5.1	Swings fundamentals, Swing Key Features	1hour
5.2	MVC, Swing Controls, Components and Containers	1hour
5.3	Swing Packages, Event Handling in Swings.	1 hour
5.4	Swing Layout Managers	1hour
5.5	Exploring Swings –JFrame, JLabel, The Swing Buttons, JTextField.	1 hour
5.6	JDBC overview, Creating and Executing Queries – create table, delete, insert, select (Basics only, DBMS course is not a prerequisite).	1hour
5.7	Creating and Executing Queries – create table, delete, insert, select.	1 hour
5.8	Creating and Executing Queries – create table, delete, insert, select.	1 hour

CSL 201	DATA STRUCTURES LAB	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	0	0	3	2	2019

Preamble: The aim of the Course is to give hands-on experience for Learners on creating and using different Data Structures. Data Structures are used to process data and arrange data in different formats for many applications. The most commonly performed operations on data structures are traversing, searching, inserting, deleting and few special operations like merging and sorting.

Prerequisite: Topics covered under the course Programming in C (EST 102)

CO1	Write a time/space efficient program using arrays/linked lists/trees/graphs to provide necessary functionalities meeting a given set of user requirements **(Cognitive Knowledge Level: Analyse)**
CO2	Write a time/space efficient program to sort a list of records based on a given key in the record (**Cognitive Knowledge Level: Apply**)
CO3	Examine a given Data Structure to determine its space complexity and time complexities of operations on it (**Cognitive Knowledge Level: Apply**)
CO4	Design and implement an efficient data structure to represent given data (**Cognitive Knowledge Level: Apply**)
CO5	Write a time/space efficient program to convert an arithmetic expression from one notation to another (**Cognitive Knowledge Level: Apply**)
CO6	Write a program using linked lists to simulate Memory Allocation and Garbage Collection (**Cognitive Knowledge Level: Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓		✓		✓		✓
CO2	✓	✓	✓	✓				✓		✓		✓
CO3	✓	✓	✓	✓				✓		✓		✓
CO4	✓	✓	✓	✓				✓		✓		✓
CO5	✓	✓	✓					✓		✓		✓
CO6	✓	✓	✓					✓		✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Test (Internal Exam)*Percentage*	End Semester Examination *Percentage*
Remember	20	20
Understand	20	20
Apply	60	60
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**75**	**75**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva-voce	: 15 marks

Internal Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Operating System to Use in Lab **:** Linux

Compiler/Software to Use in Lab **:** gcc

Programming Language to Use in Lab **:** Ansi C

Fair Lab Record:

All Students attending the Data Structures Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Data Structure used and the operations performed on them, Details of Experiment including algorithm and Result of Experiment. The left hand page should contain a print out of the code used for the experiment and sample output obtained for a set of input.

SYLLABUS

1. Implementation of Polynomials and Sparse matrices using arrays**
2. Implementation of Stack , Queues, Priority Queues, DEQUEUE and Circular Queues using arrays**
3. Application problems using stacks: Conversion of expression from one notation to another notation . **
4. Implementation of various linked list operations. **
5. Implementation of stack, queue and their applications using linked list.pression
6. Implementation of trees using linked list
7. Representation of polynomials using linked list, addition and multiplication of polynomials. **
8. Implementation of binary trees using linked lists and arrays- creations, insertion, deletion and traversal. **
9. Implementation of binary search trees – creation, insertion, deletion, search
10. Any application programs using trees
11. Implementation of sorting algorithms – bubble, insertion, selection, quick, merge sort

and heap sort.**

12. Implementation of searching algorithms – linear search, binary search.**

13. Representation of graphs and computing various parameters (in degree, out degree etc.) - adjacency list, adjacency matrix.

14. Implementation of BFS and DFS for each graph representations.**

15. Implementation of hash table using your own mapping functions and observe collisions and overflow resolving schemes.**

16. Simulation of first-fit, best-fit and worst-fit allocations.

17. Simulation of a basic memory allocator and garbage collector using doubly linked list.
** mandatory.

DATA STRUCTURES LAB - PRACTICE QUESTIONS

1. Write a program to read two polynomials and store them in an array. Calculate the sum of the two polynomials and display the first polynomial, second polynomial and the resultant polynomial.

2. C Write a program to enter two matrices in normal form . Write a function to convert two matrices to tuple form and display it. Also find the transpose of the two matrices represented in tuple form and display it. Find the sum of the two matrices in tuple form and display the sum in tuple form.

3. Write a program to enter two matrices in normal form . Write a function to convert two matrices to tuple form and display it. Also find the transpose of the two matrices represented in tuple form and display it. Find the sum of the two matrices in tuple form and display the sum in tuple form.

4. Implement a circular queue using arrays with the operations:
    4.1. Insert an element to the queue.
    4.2. Delete an elements from the queue.
    4.3. Display the contents of the queue after each operation.

5. Implement a Queue using arrays with the operations:

   5.1. Insert elements to the Queue.
   5.2. Delete elements from the Queue.
   5.3. Display the contents of the Queue after each operation.

6. Implement a Stack using arrays with the operations:
   6.1. Pushing elements to the Stack.
   6.2. Popping elements from the Stack
   6.3. Display the contents of the Stack after each operation.

7. Implement a Priority Queue using arrays with the operations:
   7.1. Insert elements to the Priority Queue.
   7.2. Delete elements from the Priority Queue.
   7.3. Display the contents of the Priority Queue after each operation.

8. Implement a Double-Ended Queue (DEQUEUE) with the operations:
   8.1. Insert elements to the Front of the queue.
   8.2. Insert elements to the Rear of the queue
   8.3. Delete elements from the Front of the queue.
   8.4. Delete elements from the Rear of the queue.
   8.5. Display the queue after each operation.

9. Using stack convert an infix expression to a postfix expression and evaluate the postfix expression.

10. Write a program to convert an infix expression to a prefix expression using stacks.

11. Convert an infix expression to a postfix expression without using a stack

12. Write a menu driven program for performing the following operations on a Linked List:
   12.1. Display
   12.2. Insert at Beginning
   12.3. Insert at End
   12.4. Insert at a specified Position
   12.5. Delete from Beginning
   12.6. Delete from End
   12.7. Delete from a specified Position

13. Implement a stack using linked list with the operations:
   13.1. Push elements to the queue.
   13.2. Pop elements from the queue.
   13.3. Display the queue after each operation.

14. Implement a Queue using linked list with the operations:

14.1.Insert an elements to the queue.
14.2.Delete an elements from the queue.
14.3.Display the queue after each operation.

15. Write a program to reverse the content of queue using stack

16. Write a program to read two polynomials and store them using linked list. Calculate the sum of the two polynomials and display the first polynomial, second polynomial and the resultant polynomial.

17. Write a program to read two polynomials and store them using linked list. Find the product of two polynomials and store the result using linked list. Display the resultant polynomial.

18. Write a program for addition of polynomials containing two variables using linked list.

19. The details of students(number, name, total-mark) are to be stored in a linked list. Write functions for the following operations:
19.1.Insert
19.2.Delete
19.3.Search
19.4.Sort on the basis of number
19.5.Display the resultant list after every operation

20. Create a Doubly Linked List from a string taking each character from the string. Check if the given string is palindrome in an efficient method.

21. Create a binary tree with the following operations
21.1.Insert a new node
21.2.Inorder traversal.
21.3.Preorder traversal.
21.4.Postorder traversal.
21.5.Delete a node.

22. Write a program to create a binary search tree and find the number of leaf nodes

23. Create a binary search tree with the following operations:
23.1.Insert a new node .
23.2.Inorder traversal.
23.3.Preorder traversal.
23.4.Postorder traversal.
23.5.Delete a node.

24. Write a program to sort a set of numbers using a binary tree.

25. Represent any given graph and

    25.1. Perform a depth first search .

    25.2. Perform a breadth first search

26. Create a text file containing the name, height, weight of the students in a class. Perform Quick sort and Merge sort on this data and store the resultant data in two separate files. Also write the time taken by the two sorting methods into the respective files.

Eg.	Sony Mathew	5.5	60
	Arun Sajeev	5.7	58
	Rajesh Kumar	6.1	70

27. Write a program to sort a set of numbers using Heap sort and find a particular number from the sorted set using Binary Search.

28. Implement a Hash table using Chaining method. Let the size of hash table be 10 so that the index varies from 0 to 9.

29. Implement a Hash table that uses Linear Probing for collision resolution

CSL 203	OBJECT ORIENTED PROGRAMMING LAB (IN JAVA)	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	0	0	3	2	2019

Preamble: The aim of the course is to provide hands-on experience to the learners on various object oriented concepts in Java Programming. This course helps the learners to enhance the capability to design and implement various Java applications for real world problems.

Prerequisite: Topics covered under the course Programming in C (EST 102)

Course Outcomes:

At the end of the course, the student should be able to

CO1	Implement the Object Oriented concepts - constructors, inheritance, method overloading & overriding and polymorphism in Java (Cognitive Knowledge Level: **Apply**)
CO2	Implement programs in Java which use datatypes, operators, control statements, built in packages & interfaces, Input/Output streams and Files (Cognitive Knowledge Level: **Apply**)
CO3	Implement robust application programs in Java using exception handling (Cognitive Knowledge Level: **Apply**)
CO4	Implement application programs in Java using multithreading and database connectivity (Cognitive Knowledge Level: **Apply**)
CO5	Implement Graphical User Interface based application programs by utilizing event handling features and Swing in Java (Cognitive Knowledge Level: **Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓	✓			✓		✓		✓
CO2	✓	✓	✓	✓	✓			✓		✓		✓
CO3	✓	✓	✓	✓	✓			✓		✓		✓
CO4	✓	✓	✓	✓	✓			✓		✓		✓
CO5	✓	✓	✓	✓	✓			✓		✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Test - Internal Exam (Percentage)	End Semester Examination (Percentage)
Remember	20	20
Understand	20	20
Apply	60	60
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**75**	**75**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva-voce	: 15 marks

Internal Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Operating System to Use in Lab	**:** Linux
Compiler/Software to Use in Lab	**:** gcc, javac, jdk, jre, Eclipse, NetBeans, MySQL / PostgreSQL.
Programming Language to Use in Lab	**:** Java

Fair Lab Record:

All Students attending the Object Oriented Programming Lab (in Java) should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Operations Performed, Details of Experiment including algorithm and Result of Experiment. The left hand page should contain a print out of the code used for the experiment and sample output obtained for a set of input.

SYLLABUS

The syllabus contains six sessions (A, B, C, D, E, F). Each session consists of three concrete Java exercises, out of which at least two questions are mandatory.

(A) Basic programs using datatypes, operators, and control statements in Java.

1) Write a Java program that checks whether a given string is a palindrome or not.
   Ex: MALAYALAM is palindrome.
2) Write a Java Program to find the frequency of a given character in a string. **
3) Write a Java program to multiply two given matrices. **

(B) Object Oriented Programming Concepts: Problem on the use of constructors, inheritance, method overloading & overriding, polymorphism and garbage collection:

4) Write a Java program which creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'printSalary( )' which prints the salary of the Employee. Two classes 'Officer' and 'Manager' inherits the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same. (Exercise to understand inheritance). **
5) Write a java program to create an abstract class named Shape that contains an empty method named numberOfSides( ). Provide three classes named Rectangle, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the classes contains only the method numberOfSides( ) that shows the number of sides in the given geometrical structures. (Exercise to understand polymorphism). **
6) Write a Java program to demonstrate the use of garbage collector.

(C) Handling different types of files as well as input and output management methods:

7) Write a file handling program in Java with reader/writer.
8) Write a Java program that read from a file and write to file by handling all file related exceptions. **
9) Write a Java program that reads a line of integers, and then displays each integer, and the sum of all the integers (Use String Tokenizer class of java.util). **

(D) Exception handling and multi-threading applications:

10) Write a Java program that shows the usage of try, catch, throws and finally. **

11) Write a Java program that implements a multi-threaded program which has three threads. First thread generates a random integer every 1 second. If the value is even, second thread computes the square of the number and prints. If the value is odd the third thread will print the value of cube of the number.

12) Write a Java program that shows thread synchronization. **

(E) Graphics Programming:

13) Write a Java program that works as a simple calculator. Arrange Buttons for digits and the + - * % operations properly. Add a text field to display the result. Handle any possible exceptions like divide by zero. Use Java Swing. **

14) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on, and only one light can be on at a time. No light is on when the program starts. **

15) Write a Java program to display all records from a table using Java Database Connectivity (JDBC).

(F) Standard Searching and Sorting Algorithms using data structures and algorithms learned from course Data Structures (**CST 201**):

16) Write a Java program for the following: **
    1) Create a doubly linked list of elements.
    2) Delete a given element from the above list.
    3) Display the contents of the list after deletion.

17) Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending order. **

18) Write a Java program that implements the binary search algorithm.

** Mandatory

PRACTICE QUESTIONS

1) Write a Java program to reverse an given string.
2) Write a Java program to display the transpose of a given matrix.
3) Write a Java program to find the second smallest element in an array.
4) Write a Java program to check whether a given number is prime or not.
5) Write a Java program to calculate the area of different shapes namely circle, rectangle, and triangle using the concept of method overloading.
6) Write two Java classes Employee and Engineer. Engineer should inherit from Employee class. Employee class to have two methods display() and calcSalary(). Write a program to display the engineer salary and to display from Employee class using a single object instantiation (i.e., only one object creation is allowed).
    - display() only prints the name of the class and does not return any value. Ex. “ Name of class is Employee.”
    - calcSalary() in Employee displays “Salary of employee is 10000” and calcSalary() in Engineer displays “Salary of employee is 20000.”
7) Write a Java program to illustrate Interface inheritance.
8) Write a Java program that shows how to create a user-defined exception.
9) Write a Java program to create two threads: One for displaying all odd number between 1 and 100 and second thread for displaying all even numbers between 1 and 100.
10) Write a Java program that shows thread priorities.
11) Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
12) Write a Java program that displays the number of characters, lines and words in a text file.
13) Write a Java program for handling mouse events.
14) Write a Java program for handling key events using Adapter classes (general).
15) Write a Java program that allows the user to draw lines, rectangles and ovals.
16) Write a Java Swing program to print a wave form on the output screen.
17) Write a program to accept rollno, name, CGPA of “n” students and store the data to a database using JDBC connectivity. Display the list of students having CGPA greater than 7. (Use MySQL /PostgreSQL).
18) Write a Java program to implement Heap sort algorithm using array.

SEMESTER -3
MINOR

CST 281	OBJECT ORIENTED PROGRAMMING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		MINOR	3	1	0	4	2019

Preamble: This is the programming course for awarding B.Tech. Minor in Computer Science and Engineering with specialization in ***Software Engineering***. The purpose of this course is to enable learners to solve problems by breaking it down to object level while designing software and to implement it using Java. This course covers Object Oriented Principles, Object Oriented Programming in Java, Inheritance, Exception handling, Event handling, multithreaded programming and working with window-based graphics. This course helps the learners to develop Mobile applications, Enterprise Applications, Scientific Applications and Web based Applications.

Prerequisite: Topics covered under the course PROGRAMMING IN C (EST 102)

Course Outcomes: After the completion of the course the student will be able to

CO1	Write Java programs using the object oriented concepts - classes, objects, constructors, data hiding, inheritance and polymorphism (Cognitive Knowledge Level: **Apply**)
CO2	Utilise datatypes, operators, control statements, built in packages & interfaces, Input/ Output Streams and Files in Java to develop programs (Cognitive Knowledge Level: **Apply**)
CO3	Illustrate how robust programs can be written in Java using exception handling mechanism (Cognitive Knowledge Level: **Understand**)
CO4	Write application programs in Java using multithreading (Cognitive Knowledge Level: **Apply**)
CO5	Write Graphical User Interface based application programs by utilising event handling features and Swing in Java (Cognitive Knowledge Level: **Apply**)

Mapping of course outcomes with program outcome

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓						✓		✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test1 (Marks %)	Test2 (Marks %)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

SYLLABUS

Object Oriented Programming Using Java

Module 1

Introduction:

Approaches to Software Design - Functional Oriented Design, Object Oriented Design, Case Study of Automated Fire Alarm System.

Object Modeling Using UML – Basic Object Oriented concepts, UML (Unified Modeling Language) diagrams, Use case model, Class diagram, Interaction diagram, Activity diagram, State chart diagram.

Introduction to Java - Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. Java Virtual Machine (JVM), Java compiler, Bytecode, Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues.

Module 2

Core Java Fundamentals:

Primitive Data types - Integers, Floating Point Types, Characters, Boolean. Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.

Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.

Control Statements - Selection Statements, Iteration Statements and Jump Statements.

Object Oriented Programming in Java - Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods, Constructors, ***this*** Keyword, Method Overloading, Using Objects as Parameters, Returning Objects, Recursion, Access Control, Static Members, Final Variables, Inner Classes, Command-Line Arguments, Variable Length Arguments.

Module 3

More features of Java:

Inheritance - Super Class, Sub Class, The Keyword super, protected Members, Calling Order of Constructors, Method Overriding, the Object class, Abstract Classes and Methods, Using final with Inheritance.

Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages, Interfaces.

Exception Handling **-** Checked Exceptions, Unchecked Exceptions, ***try*** Block and ***catch*** Clause, Multiple ***catch*** Clauses, Nested ***try*** Statements, ***throw***, ***throws*** and ***finally***.

Module 4

Advanced features of Java:

Input/Output - I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class, Object Streams and Serialization, Reading and Writing Files.

Java Library - String Handling – String Constructors, String Length, Special String Operations - Character Extraction, String Comparison, Searching Strings, Modifying Strings, Using valueOf(), Comparison of StringBuffer and String.

Collections framework – Collections overview, Collections Class – ArrayList. Accessing Collections via an Iterator.

Module 5

GUI Programming, Event Handling and Multithreaded Programming:

Swing fundamentals - Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Exploring Swing - JFrame, JLabel, JButton, JTextField.

Event handling **-** Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Model.

Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread, Creating Multiple Threads, Suspending, Resuming and Stopping Threads.

Text Books:

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.
2. Rajib Mall, Fundamentals of Software Engineering, 4th edition, PHI, 2014.
3. Paul Deitel, Harvey Deitel, Java How to Program, Early Objects 11th Edition, Pearson, 2018.

Reference Books:

1. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.
2. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.
3. Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
4. Barclay K., J. Savage, Object Oriented Design with UML and Java, Elsevier, 2004.
5. Sierra K., Head First Java, 2/e, O'Reilly, 2005.
6. Balagurusamy E., Programming JAVA a Primer, 5/e, McGraw Hill, 2014.

Sample Course Level Assessment Questions

Course Outcome1(CO1): For the following passage develop UML diagrams and then implement it as a Java program in accordance with your UML design.

Passage: College Office collects semester fee and college bus fee for each student. A clerk at the college office collects the fees from each student. The bus fee is calculated depending on the distance of the corresponding bus stop from the college. The semester fee varies depending upon the semester as well as branch of each student. Students are supposed to pay the fees in full. Economically backward students are eligible for 50% discount in semester fee. The consolidated fees receipt is issued to each student by the clerk, which contains the student name, admission number, semester and branch of student along with details of fees collected. Students can log in and view the details of fees remitted and dues if any. The system allows students and clerk level login to the system. Clerk is able to view reports of each class showing status of fees payment of each student.

Course Outcome 2 (CO2): Write a Java program to prepare the rank list of students based one their performance in the first Semester B.Tech. Degree examination at APJ Abdul Kalam Technological University. The output should be stored in a file.

Course Outcome 3 (CO3): Write a program to demonstrate how event handling and exception handling are supported in Java..

Course Outcome 4 (CO4): Write a program to demonstrate the start, run, sleep and join methods in Thread class..

Model Question Paper

QP CODE: PAGES:3

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

THIRD SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 281

Course Name: Object Oriented Programming using Java

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Briefly explain why Java is considered to be secure and portable.
2. Describe the concept of association among classes with an example.
3. Explain the different arithmetic operators in Java.
4. Explain the use for command line arguments with a suitable Java program
5. Explain the use of CLASSPATH with an example.
6. What are the different types of exceptions?
7. Explain file handling features available in Java.
8. Write a simple program to read and print an integer value in Java.
9. Explain the concept of *main thread* in multi-threading.
10. Explain any two Event classes in Java.

Part B

Answer any one question completely from each module

11.

(a) Describe in detail polymorphism, abstraction and inheritance with suitable examples. (9)

(b) What is Java Virtual Machine? (5)

OR

12.

(a) Compare and contrast Functional Oriented and Object Oriented approach by considering a simple bus ticket reservation system. (5)

(b) What is a class diagram? Explain with an example. (9)

13.

(a) Explain primitive data types in Java. How are they different from other data types? (8)

(b) Explain variables and arrays in Java. (6)

OR

14.s

(a) Using a suitable Java program explain the concept of methods and constructors. (8)

(b) Explain the keyword ***super*** and its usage in Java. (6)

15.

(a) Using a table, explain the effect of access specifiers in inheritance. (6)

(b) Describe in detail about exception handling using **try** block and **catch** clause in Java with the help of a suitable Java program. (8)

OR

16.

(a) What is an interface in Java? Explain with a suitable example. (8)

(b) Explain ***throw***, ***throws*** and ***finally*** constructs with the help of a Java program. (6)

17.

(a) Explain ***ArrayList*** collections framework. Also explain the use of iterator in accessing collections.

(8)

(b) Bring out difference between "==" and ***equals()*** method with the help of a sample program

(6)

OR

18.

(a) Compare Byte Streams and Character Streams. Write a program to demonstrate the usage of the ***PrintWriter*** class.

(8)

(b) Explain any three String constructors with the help of sample code for each.

(6)

19.

(a) Explain in detail the Delegation Event model for event handling in Java.

(7)

(b) Describe in detail the creation of a thread using the Runnable interface.

(7)

OR

20.

(a) What are the differences between a process and a thread?

(4)

(b) Write a Graphical User Interface (GUI) based Java program to implement a simple calculator supporting the operations addition, subtraction, multiplication and division. Use Swing controls to implement GUI. There may be three text boxes, the first two for operands and the last for result. Add four buttons for the above operations. Write neat comments in your program to show how you handle events.

(10)

Teaching Plan		
Module 1 (Introduction)		**(8 hours)**
1.1	Approaches to Software Design- Functional Oriented Design, Object-Oriented Design, Case Study of Automated Fire Alarm System.	1 hour
1.2	Object Modeling Using UML – Basic object oriented concepts	1 hour
1.3	Basic object oriented concepts	1 hour
1.4	UML diagrams, Use case model	1hour
1.5	Class diagram, Interaction diagram	1hour
1.6	Activity diagram, State chart diagram	1hour
1.7	Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. JVM, Java compiler, Bytecode	1hour
1.8	Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues	1hour
Module 2 (Core Java Fundamentals)		**(12 hours)**
2.1	Primitive Data types - Integers, Floating Point Types, Characters, Boolean	1 hour
2.2	Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.	1 hour
2.3	Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.	1 hour
2.4	Control Statements - Selection Statements, Iteration Statements and Jump Statements.	1 hour
2.5	Object Oriented Programming in Java - Class Fundamentals, Declaring Objects	1 hour
2.6	Object Reference, Introduction to Methods	1 hour
2.7	Constructors, ***this*** Keyword	1 hour
2.8	Method Overloading, Using Objects as Parameters	1 hour

2.9	Returning Objects, Recursion	1 hour
2.10	Access Control, static Members	1 hour
2.11	Final Variables, Inner Classes	1 hour
2.12	Command-Line Arguments, Variable Length Arguments	1 hour
Module 3 (More features of Java)		**(8 hours)**
3.1	Inheritance - Super class, Sub class, the keyword super, protected Members,	1 hour
3.2	Calling Order of Constructors, Method Overriding, the Object class,	1 hour
3.3	Abstract Classes and Methods, Using final with Inheritance	1 hour
3.4	Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages	1 hour
3.5	Interfaces	1 hour
3.6	Exception Handling **-** Checked Exceptions, Unchecked Exceptions, ***try*** Block and ***catch*** Clause	1 hour
3.7	Multiple ***catch*** Clauses, Nested ***try*** Statements	1 hour
3.8	***throw***, ***throws*** and ***finally***	1 hour
Module 4 (Advanced features of Java)		**(8 hours)**
4.1	Input/Output - I/O Basics, Reading Console Input	1hour
4.2	Writing Console Output, PrintWriter Class	1hour
4.3	Object Streams and Serialization	1hour
4.4	Serialization, Working with Files	1hour
4.5	Working with Files	1hour
4.6	Java Library - String Handling – String Constructors, String Length, Special String Operations	1hour
4.7	Character Extraction, String Comparison, Searching Strings, Modifying Strings Using valueOf( ), Comparison of StringBuffer and String.	1hour
4.8	Collections framework – Collections overview, Collections Class – ArrayList. Accessing Collections via an Iterator.	1hour

Module 5 (GUI Programming, Event Handling and Multithreaded Programming)		(9 hours)
5.1	Swings fundamentals, Swing Key Features	
5.2	MVC, Swing Controls, Components and Containers	
5.3	Exploring Swing –JFrame, JLabel, JButton, JTextField.	
5.4	Event handling - Event Handling Mechanisms, Delegation Event Model	1hour
5.5	Delegation Event Model, Event Classes	1hour
5.6	Sources of Events, Event Listener Interfaces, Using the Delegation Model	1hour
5.7	Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread	1hour
5.8	Creating Multiple Threads	1hour
5.9	Suspending, Resuming and Stopping Threads.	1hour

CST 283	Python for Machine Learning	Category	L	T	P	Credit	Year of Introduction
		MINOR	3	1	0	4	2019

Preamble: This is a programming course for awarding B. Tech. Minor in Computer Science and Engineering with specialization in ***Machine Learning***. The objective of the course is to provide learners an insight into Python programming, and develop programming skills to manage the development of software systems. It covers programming environment, important instructions, data representations, intermediate level features, Object Oriented Programming and file data processing of Python. This course lays the foundation to develop web applications, Machine Learning, and Artificial Intelligence-based applications and tools, Data Science and Data Visualization applications.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO1	Write, test and debug Python programs **(Cognitive Knowledge level: Apply)**
CO2	Illustrate uses of conditional (if, if-else, if-elif-else and switch-case) and iterative (while and for) statements in Python programs **(Cognitive Knowledge level: Apply)**
CO3	Develop programs by utilizing the modules Lists, Tuples, Sets and Dictionaries in Python **(Cognitive Knowledge level: Apply)**
CO4	Implement Object Oriented programs with exception handling **(Cognitive Knowledge level: Apply)**
CO5	Write programs in Python to process data stored in files by utilizing the modules Numpy, Matplotlib, and Pandas **(Cognitive Knowledge level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✔	✔	✔		✔						✔	✔
CO2	✔	✔	✔		✔					✔		✔
CO3	✔	✔	✔		✔	✔	✔					✔
CO4	✔	✔	✔		✔		✔					✔
CO5	✔	✔	✔	✔	✔	✔						✔

Abstract POs defined by National Board of Accreditation

#PO	Broad PO	#PO	Broad PO
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Test 1 (*Marks in percentage*)**	**Test 2 (*Marks in percentage*)**	**End Semester Examination (*Marks in percentage*)**
Remember	20	20	20
Understand	35	35	35
Apply	45	45	45
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 sub-divisions and carries 14 marks.

SYLLABUS

Module I

Programming Environment and Python Basics:

Getting Started with Python Programming - Running code in the interactive shell, Editing, Saving, and Running a script. Using editors - IDLE, Jupyter. The software development process - Case Study.

Basic coding skills - Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions, Working with numeric data, Type conversions, Comments in the program. Input, Processing, and Output. Formatting output. How Python works. Detecting and correcting syntax errors. Using built in functions and modules in math module.

Module II

Building Python Programs:

Control statements - Selection structure (if-else, switch-case), Iteration structure(for, while), Testing the control statements, Lazy evaluation. Functions - Hiding redundancy and complexity, Arguments and return values, Variable scopes and parameter passing, Named arguments, Main function, Working with recursion, Lambda functions. Strings and number systems - String function, Handling numbers in various formats.

Module III

Data Representation:

Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension. Work with tuples. Sets. Work with dates and times. Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup. Case Study - Data Structure Selection.

Module IV

Object Oriented Programming:

Design with classes - Objects and Classes, Methods, Instance Variables, Constructor, Accessors and Mutators. Structuring classes with Inheritance and Polymorphism. Abstract Classes. Exceptions - Handle a single exception, handle multiple exceptions.

Module V

Data Processing:

The *os* and *sys* modules. Introduction to file I/O - Reading and writing text files, Manipulating binary files. NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Random numbers. Plotting and visualization. Matplotlib - Basic plot, Ticks, Labels, and Legends. Working with CSV files. – Pandas - Reading, Manipulating, and Processing Data.

Text Books:

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017

Reference Books:

1. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
2. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016
3. David M.Baezly, Python Essential Reference. Addison-Wesley Professional; 4/e, 2009.
4. Charles Severance. Python for Informatics: Exploring Information,
5. http://swcarpentry.github.io/python-novice-gapminder/

Sample Course Level Assessment Questions

Course Outcome1(CO1): What is type conversion? How is it done in Python?

Course Outcome 2(CO2): Write a Python program which takes a positive integer **n** as input and finds the sum of cubes all positive even numbers less than or equal to the number.

Course Outcome 3(CO3): Given is a list of of words, *wordlist*, and a string, *name.* Write a Python function which takes *wordlist* and *name* as input and returns a tuple. The first element of the output tuple is the number of words in the *wordlist* which have *name* as a substring in it. The second element of the tuple is a list showing the index at which the *name* occurs in each of the words of the *wordlist* and a 0 if it doesn't occur.

Course Outcome 4(CO4): Write a Python program to implement the addition, subtraction, and multiplication of complex numbers using classes. Use constructors to create objects. The input to the program consist of real and imaginary parts of the complex numbers.

Course Outcome 5(CO5): Given a file "auto.csv" of automobile data with the fields *index, company, body-style, wheel-base, length, engine-type, num-of-cylinders, horsepower, average-mileage,* and *price***,** write python code to

1) Clean and Update the CSV file
2) Print total cars of all companies
3) Find the average mileage of all companies
4) Find the highest priced car of all companies.

Model Question Paper

QP CODE: PAGES:

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
THIRD SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 283

Course name : PYTHON FOR MACHINE LEARNING

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Explain the basic data types available in Python, with examples.
2. Write a Python program to reverse a number and also find the sum of digits of the number. Prompt the user for input.
3. Explain the concept of scope and lifetime of variables in Python programming language, with a suitable example.
4. Discuss format specifiers and escape sequences with examples.
5. Discuss the relation between tuples, lists, and dictionaries in detail.
6. Discuss the following dictionary methods with an example.
   i. *get( )* ii. *Keys( )* iii. *pop( )* iv. *update( )* v. *values( )* vi. *items( )*
7. What is polymorphism? Give an example in the context of OOP in Python.
8. How is exception handling accomplished in Python programs?
9. Write a note on the **os** and **os.path** modules in Python. Also, discuss the *walk( )* and *getcwd( )* methods of the **os** module.
10. Describe the characteristics of the CSV format.

PART-B

(Answer any one full question from each module)

11. (a) Compare and contrast interpreted languages and compiled languages. How does it affect the quality of program development and execution of the program? (6)

(b) What are the possible errors in a Python program. Write a Python program to print the value of $2^{2n}+n+5$ for n provided by the user. (8)

OR

12. (a) Describe Arithmetic operators, Assignment operators, Comparison operators, Logical operators, and Bitwise operators in detail with examples. (6)

(b) Explain the software development process in detail. (8)

13. (a) Write a Python code to check whether a given year is a leap year or not [An year is a leap year if it's divisible by 4 but not divisible by 100 except for those divisible by 400]**.** (5)

(b) Input 4 integers (+ve and −ve). Write a Python code to find the sum of negative numbers, positive numbers, and print them. Also, find the averages of these two groups of numbers and print. (9)

OR

14. (a) Write a Python program to find the value for *sin(x)* up to *n* terms using the series (8)

$$\sin(x) - \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$ where ***x*** is in degrees

(b) Write a Python code to determine whether the given string is a Palindrome or not using slicing. Do not use any string function. (6)

15. (a) Write a Python code to create a function called *list_of_frequency* that takes a string and prints the letters in non-increasing order of the frequency of their occurrences. Use dictionaries. (5)

(b) Write a Python program to read a list of numbers and sort the list in a non-decreasing order without using any built in functions. Separate function should be written to sort the list wherein the name of the list is passed as the parameter. (9)

OR

16. (a) Illustrate the following Set methods with an example. (6)
i. *intersection( )* ii. *Union( )* iii. *Issubset( )* iv. *Difference( )* v. *update( )* vi. *discard( )*

(b) Write a Python program to check the validity of a password given by the user. (8)
The Password should satisfy the following criteria:
1. Contains at least one letter between **a** and **z**
2. Contains at least one number between **0** and **9**
3. Contains at least one letter between **A** and **Z**
4. Contains at least one special character from **$, #, @**
5. Minimum length of password: **6**

17. (a) How can a class be instantiated in Python? Write a Python program to express the instances as return values to define a class RECTANGLE with parameters *height, width, corner_x,* and *corner_y* and member functions to find center, area, and perimeter of an instance. (10)

(b) Explain inheritance in Python. Give examples for each type of inheritance. (4)

OR

18. (a) Write a Python class named ***Circle*** constructed by a radius and two methods which will compute the area and the perimeter of a given circle (6)

(b) Write Python program to create a class called as **Complex** and implement *__add__( )* method to add two complex numbers. Display the result by overloading the + Operator. (8)

19. (a) Write a Python program to add two matrices and also find the transpose of the resultant matrix. (8)

(b) Given a file “auto.csv” of automobile data with the fields *index, company, body-style, wheel-base, length, engine-type, num-of-cylinders, horsepower, average-mileage,* and *price***,** write Python codes using Pandas to (6)
1) Clean and Update the CSV file
2) Print total cars of all companies
3) Find the average mileage of all companies
4) Find the highest priced car of all companies.

OR

20. (a) Write Python program to write the data given below to a CSV file. (5)

SN	Name	Country	Contribution	Year
1	Linus Torvalds	Finland	Linux Kernel	1991
2	Tim Berners-Lee	England	World Wide Web	1990
3	Guido van Rossum	Netherlands	Python	1991

(b) Given the sales information of a company as CSV file with the following fields *month_number, facecream, facewash, toothpaste, bathingsoap, shampoo, moisturizer, total_units, total_profit*. Write Python codes to visualize the data as follows (9)

1) Toothpaste sales data of each month and show it using a scatter plot
2) Face cream and face wash product sales data and show it using the bar chart
3) Calculate total sale data for last year for each product and show it using a Pie chart.

(14X5=70)

Teaching Plan

Module 1: Programming Environment and Python Basics		**(10 hours)**
1.1	Getting Started with Python Programming: Running code in the interactive shell Editing, Saving, and Running a script	**1 hour**
1.2	Using editors: IDLE	**1 hour**
1.3	Jupyter	**1 hour**
1.4	The software development process: Case Study.	**1 hour**
1.5	Basic coding skills: Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions,	**1 hour**
1.6	Working with numeric data, Type conversions, Comments in the program	**1 hour**
1.7	Input, Processing, and Output, Formatting output – How Python works	**1 hour**
1.8	How Python works – Detecting and correcting syntax errors	**1 hour**
1.9	Using built in functions and modules: Case – Using math module	**1 hour**
1.10	Using built in functions and modules: Case – Using math module (Examples)	**1 hour**

Module 2: Building Python Programs		**(8 hours)**
2.1	Control statements: Selection structure (if-else, switch-case),	**1 hour**
2.2	Iteration structure(for, while), Testing the control statements, Lazy evaluation	**1 hour**
2.3	Functions: Hiding redundancy and complexity, Arguments and return values,	**1 hour**
2.4	Variable scopes and parameter passing	**1 hour**
2.5	Named arguments, Main function,	**1 hour**
2.6	Working with recursion, Lambda functions	**1 hour**
2.7	Strings and number systems: String function	**1 hour**
2.8	Handling numbers in various format	**1 hour**
Module 3: Data Representation		**(9 hours)**
3.1	Lists: Basic list Operations and functions, List of lists	**1 hour**
3.2	Slicing, Searching and sorting list	**1 hour**
3.3	List comprehension	**1 hour**
3.4	Work with tuples, Sets	**1 hour**
3.5	Work with dates and times	**1 hour**
3.6	Dictionaries: Dictionary functions,	**1 hour**
3.7	Dictionary literals, adding and removing keys, accessing & replacing values	**1 hour**
3.8	Traversing dictionaries, reverse lookup	**1 hour**
3.9	Case Study: Data Structure Selection	**1 hour**
Module 4: Object Oriented Programming		**(8 hours)**
4.1	Design with classes : Objects and Classes, Methods, Instance Variables	**1 hour**
4.2	Constructor, Accessors and Mutators	**1 hour**
4.3	Structuring classes with Inheritance	**1 hour**
4.4	Polymorphism	**1 hour**
4.5	Abstract Classes	**1 hour**
4.6	Abstract Classes	**1 hour**
4.7	Exceptions : Handle a single exception	**1 hour**

4.8	handle multiple exceptions	**1 hour**
Module 5: Data Processing		**(10 hours)**
5.1	The *os* and *sys* modules	**1 hour**
5.2	Introduction to file I/O: Reading and writing text files	**1 hour**
5.3	Manipulating binary files	**1 hour**
5.4	NumPy : Basics, Creating arrays, Arithmetic, Slicing	**1 hour**
5.5	Matrix Operations, Random numbers.	**1 hour**
5.6	Matplotlib : Basic plot	**1 hour**
5.7	Matplotlib - Ticks, Labels, and Legends	**1 hour**
5.8	Working with CSV files	**1 hour**
5.9	Pandas : Reading, Manipulating	**1 hour**
5.10	Pandas : Processing Data and Visualize.	**1 hour**

CST 285	DATA COMMUNICATION	Category	L	T	P	Credit	Year of Introduction
		MINOR	3	1	0	4	2019

Preamble: This is a basic course in communication for awarding B. Tech. Minor in Computer Science and Engineering with specialization in ***Networing***. The purpose of this course is to prepare learners to understand the communication entities and the associated issues in the field of Computer Science. This course covers fundamental concepts of data transmission & media, digital & analog transmissions, multiplexing & spread spectrum, error detection & correction and switching. Concepts in data communication help the learner to understand the concepts in networking and mobile communication.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO1	Describe the characteristics of signals used for Analog and Digital transmissions (**Cognitive knowledge: Understand**)
CO2	Discuss the features and issues in data transmission (**Cognitive knowledge: Understand**)
CO3	Select transmission media based on characteristics and propagation modes (**Cognitive knowledge: Apply**)
CO4	Use appropriate signal encoding techniques for a given scenario (**Cognitive knowledge: Apply**)
CO5	Illustrate multiplexing and spread spectrum technologies (**Cognitive knowledge: Understand**)
CO6	Explain error detection & correction techniques and switching techniques used in data communication (**Cognitive knowledge: Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓								✓		✓
CO2	✓	✓								✓		✓
CO3	✓											✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓						✓		✓
CO6	✓	✓	✓	✓						✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination (Marks in percentage)
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module 1

Data Transmission Basics

Communication model - Simplex, Half duplex, Full duplex transmission. Periodic Analog signals - Sine wave, Amplitude, Phase, Wavelength, Time and frequency domain, Bandwidth. Analog & digital data and signals. Transmission impairments - Attenuation, Delay distortion, Noise. Data rate limits - Noiseless channel, Nyquist bandwidth, Noisy channel, Shannon's capacity formula.

Module 2

Transmission Media

Guided Transmission Media - Twisted pair, Coaxial cable, Optical fiber. Unguided media - Radio waves, Terrestrial microwave, Satellite microwave, Infrared. Wireless Propagation - Ground wave propagation, Sky Wave propagation, Line-of-Sight (LoS) Propagation.

Module 3

Digital Transmission and Analog Transmission

Digital data to Digital signal – Non-Return-to-Zero (NRZ), Return-to-Zero (RZ), Multilevel binary, Biphase. Analog data to Digital signal - Sampling theorem, Pulse Code Modulation (PCM), Delta Modulation (DM). Digital data to Analog signal: Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), Phase Shift Keying (PSK). Analog data to Analog signal: Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM).

Module 4

Multiplexing and Spread Spectrum

Multiplexing - Frequency Division Multiplexing (FDM), Wave length Division Multiplexing (WDM), Time Division Multiplexing (TDM), Characteristics, Synchronous TDM, Statistical TDM. Spread Spectrum Techniques - Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), Code Division Multiplexing, Code Division Multiple Access (CDMA).

Module 5

Error Detection, Correction and Switching

Digital data communication techniques - Asynchronous transmission, Synchronous transmission. Detecting and correcting errors - Types of Errors, Parity check, Checksum, Cyclic Redundancy Check (CRC), Forward Error Correction (FEC), Hamming Distance, Hamming Code. Basic principles of Switching - Circuit Switching, Packet Switching, Message Switching.

Text Books

1. Forouzan B. A., Data Communications and Networking, 5/e, McGraw Hill, 2013.
2. William Stallings, Data and Computer Communication 9/e, Pearson Education, Inc.

Reference Books

1. Schiller J., Mobile Communications, 2/e, Pearson Education, 2009.
2. Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): What is a periodic analog signal? List the main properties of a periodic analog signal.

Course Outcome 2 (CO2): What is attenuation? How can it be handled?

Course Outcome 3 (CO3): How can interference be reduced using optical fiber?

Course Outcome 4(CO4): Encode the data sequence 101011100 using Multilevel binary and Biphase schemes.

Course Outcome 5(CO5): Explain direct sequence spread spectrum with a neat diagram.

Course Outcome 6(CO6): Using Cyclic Redundancy Check (CRC), given the data-word 11110000 and the divisor 10011, show the generation of the codeword at the sender and the checking of the codeword at the receiver.

Model Question Paper

QP CODE: PAGES: ___

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FOURTH SEMESTER B.TECH DEGREE (MINOR) EXAMINATION, MONTH & YEAR

Course Code: CST 285
Course name : DATA COMMUNICATION

Max Marks: 100 **Duration: 3 Hours**

PART-A
(Answer All Questions. Each question carries 3 marks)

1. What is bandwidth? Find the lowest frequency, if a periodic signal has a bandwidth of 20 Hz and the highest frequency is 60 Hz. Draw the Spectrum if the signal contains all frequencies of same amplitude.
2. Assume that a TV picture is to be transmitted over a channel with 4.5 MHz bandwidth and a 35 dB Signal-to-Noise-Ratio. Find the capacity of the channel.
3. What is the purpose of cladding in optical fibres?
4. Which wireless propagation is suitable for satellite communication? Justify your answer.
5. Explain the working of Delta Modulation with an example.
6. Illustrate the equivalent square wave pattern of the bit string 01001101 using Non-Return-to-Zero(NRZ) - Level and NRZ-Invert encoding schemes.
7. Distinguish between synchronous and statistical Time Division Multiplexing.
8. Apply Direct Sequence Spread Spectrum to the data 101 using the Barker sequence 10110111000. Show the encoding and decoding steps.
9. Find the minimum hamming distance for the following cases:
   a) Detection of two errors
   b) Correction of two errors
   c) Detection of 3 errors or correction of 2 errors
   d) Detection of 6 errors or correction of 2 errors
10. Find the parity bit for simple even parity check for the following.
    a) 1001010
    b) 0001100
    c) 1000000
    d) 1110111

PART-B

(Answer ANY one full question from each module. Each question carries 14 marks)

11. a) With the help of suitable figures, distinguish between time domain and frequency domain. (4)

b) Describe the different types of transmission impairments. (10)

OR

12. a) Calculate the bandwidth, if a periodic signal is decomposed into 4 sine waves with frequencies 50 Hz, 100 Hz, 150 Hz and 200Hz. Draw the spectrum, assuming all components having amplitude in the range 6-12 V and all are multiple of two in the increasing order. (6)

b) Distinguish between Nyquist bandwidth and Shannon capacity. Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with (i) Two signal levels and (ii) Four signal levels. Determine the maximum bit rate in both these cases. (8)

13. a) For a parabolic reflective antenna operating at 12 GHz with a diameter of 2 m, calculate the effective area and the antenna gain. (6)

b) List any four advantages and disadvantages of twisted pair, coaxial cable and fiber optic cable. (8)

OR

14. a) Compare the features of terrestrial microwave and satellite microwave. (6)

b) With the help of suitable diagrams, differentiate Multi-mode and Single-mode optical fibres. How the rays are propagated in Step-index and Graded-index Multi-mode fibres. (8)

15. a) Distinguish between data rate and signal rate. (4)

b) What is polar encoding? Encode the pattern 010011001110 using the two Biphase schemes.

(10)

OR

16. a) Show the equivalent analog sine wave pattern of the bit string 010011010 using Amplitude Shift Keying, Frequency Shift Keying and Phase Shift Keying.

(4)

b) State Sampling theorem. Explain Pulse Code Modulation with suitable figures.

(10)

17. a) Four channels are multiplexed using Time Division Multiplexing. If each channel sends 100 bytes/sec and we multiplex one byte per channel, determine the frame size, duration of a frame, frame rate and bit rate of the link.

(6)

b) With the help of an example, explain the working of Frequency Hopping Spread Spectrum.

(8)

OR

18. a) Explain the different techniques by which the disparity in input data rate is handled by Time Division Multiplexing.

(4)

b) Suppose Alice and Bob are communicating using Code Division Multiple Access. Alice uses the code [+1 +1] and Bob uses the code [+1 -1]. Alice sends the data bit 0 and Bob sends the data bit 1. Show the data in the channel and how they can detect what the other person has sent.

(10)

19. a) Explain parity check with examples.

(4)

b) Describe the need for a switch. What are the different phases in circuit switching?

(10)

OR

20. a) With the help of a suitable example, explain the virtual circuit approach of packet switching.

(6)

b) Find the Hamming code for the data-word 1011001. Assume odd parity.

(8)

Teaching Plan

Module 1 : Data Transmission Basics		**(8 Hours)**
1.1	Introduction, Communication model - Simplex, Half duplex, Full duplex transmission	1
1.2	Periodic Analog signals - Sine wave, Amplitude, Phase, Wavelength	1
1.3	Time and frequency domain, Bandwidth	1
1.4	Analog data and signals	1
1.5	Digital data and signals	1
1.6	Transmission impairments - Attenuation, Delay distortion, Noise	1
1.7	Data rate limits - Noiseless channel, Nyquist bandwidth	1
1.8	Noisy channel, Shannon's capacity formula	1
Module 2: Transmission media		**(7 Hours)**
2.1	Guided Transmission Media - Twisted pair, Coaxial cable	1
2.2	Optical fiber	1
2.3	Unguided media - Radio waves	1
2.4	Terrestrial microwave, Satellite microwave	1
2.5	Infrared	1
2.6	Wireless Propagation - Ground wave propagation	1
2.7	Wave propagation, Line-of-Sight (LoS) Propagation	1
Module 3: Digital Transmission and Analog Transmission		**(10 Hours)**
3.1	Digital data to Digital signal – Non-Return-to-Zero (NRZ)	1
3.2	Return-to-Zero (RZ), Multilevel binary	1

3.3	Biphase	1
3.4	Analog data to Digital signal - Sampling theorem	1
3.5	Pulse Code Modulation (PCM)	1
3.6	Delta Modulation (DM)	1
3.7	Digital data to Analog signal: Amplitude Shift Keying (ASK)	1
3.8	Frequency Shift Keying (FSK), Phase Shift Keying (PSK)	1
3.9	Analog data to Analog signal: Amplitude Modulation (AM)	1
3.10	Frequency Modulation (FM), Phase Modulation (PM)	1
Module 4: Multiplexing and Spread Spectrum		**(9 Hours)**
4.1	Multiplexing - Frequency Division Multiplexing (FDM)	1
4.2	Wave length Division Multiplexing (WDM), Time Division Multiplexing (TDM)	1
4.3	Synchronous TDM, Statistical TDM	1
4.4	Spread Spectrum Techniques	1
4.5	Direct Sequence Spread Spectrum (DSSS)	1
4.6	Frequency Hopping Spread Spectrum (FHSS)	1
4.7	Code Division Multiplexing	1
4.8	Code Division Multiple Access (CDMA)	1
4.9	CDMA	1
Module 5: Error Detection, Correction and Switching		**(11 Hours)**
5.1	Digital data communication techniques - Asynchronous & Synchronous transmission	1
5.2	Detecting and correcting errors - Types of Errors	1
5.3	Parity check, Checksum	1
5.4	Cyclic Redundancy Check (CRC)	1
5.5	CRC	1
5.6	Forward Error Correction (FEC)	1
5.7	Hamming Distance, Hamming Code	1
5.8	Hamming Code	1
5.9	Basic principles of Switching - Circuit Switching	1

5.10	Packet Switching	1
5.11	Message Switching	1

SEMESTER -4

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
MAT 206	**GRAPH THEORY**	**BSC**	**3**	**1**	**0**	**4**

Preamble: This course introduces fundamental concepts in Graph Theory, including properties and characterisation of graph/trees and graph theoretic algorithms, which are widely used in Mathematical modelling and has got applications across Computer Science and other branches in Engineering.

Prerequisite: The topics covered under the course Discrete Mathematical Structures (MAT 203)

Course Outcomes: After the completion of the course the student will be able to

CO 1	Explain vertices and their properties, types of paths, classification of graphs and trees & their properties. **(Cognitive Knowledge Level: Understand)**
CO 2	Demonstrate the fundamental theorems on Eulerian and Hamiltonian graphs. **(Cognitive Knowledge Level: Understand)**
CO 3	Illustrate the working of Prim's and Kruskal's algorithms for finding minimum cost spanning tree and Dijkstra's and Floyd-Warshall algorithms for finding shortest paths. **(Cognitive Knowledge Level: Apply)**
CO 4	Explain planar graphs, their properties and an application for planar graphs. **(Cognitive Knowledge Level: Apply)**
CO 5	Illustrate how one can represent a graph in a computer. **(Cognitive Knowledge Level: Apply)**
CO 6	Explain the Vertex Color problem in graphs and illustrate an example application for vertex coloring. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	√	√	√							√		√
CO 2	√	√	√	√						√		√
CO 3	√	√	√	√						√		√
CO 4	√	√	√	√						√		√
CO 5	√	√	√							√		√
CO 6	√	√	√			√				√		√

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests (%)**		**End Semester Examination (%)**
	1	**2**	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer anyone. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module 1

Introduction to Graphs : Introduction- Basic definition – Application of graphs – finite, infinite and bipartite graphs – Incidence and Degree – Isolated vertex, pendant vertex and Null graph. Paths and circuits – Isomorphism, sub graphs, walks, paths and circuits, connected graphs, disconnected graphs and components.

Module 2

Eulerian and Hamiltonian graphs : Euler graphs, Operations on graphs, Hamiltonian paths and circuits, Travelling salesman problem. Directed graphs – types of digraphs, Digraphs and binary relation, Directed paths, Fleury's algorithm.

Module 3

Trees and Graph Algorithms : Trees – properties, pendant vertex, Distance and centres in a tree - Rooted and binary trees, counting trees, spanning trees, Prim's algorithm and Kruskal's algorithm, Dijkstra's shortest path algorithm, Floyd-Warshall shortest path algorithm.

Module 4

Connectivity and Planar Graphs : Vertex Connectivity, Edge Connectivity, Cut set and Cut Vertices, Fundamental circuits, Planar graphs, Kuratowski's theorem (proof not required), Different representations of planar graphs, Euler's theorem, Geometric dual.

Module 5

Graph Representations and Vertex Colouring : Matrix representation of graphs- Adjacency matrix, Incidence Matrix, Circuit Matrix, Path Matrix. Coloring- Chromatic number, Chromatic polynomial, Matchings, Coverings, Four color problem and Five color problem. Greedy colouring algorithm.

Text book:

1. Narsingh Deo, Graph theory, PHI,1979

Reference Books:

1. R. Diestel, *Graph Theory,* free online edition, 2016: diestel-graph-theory.com/ basic.html.
2. Douglas B. West, Introduction to Graph Theory, Prentice Hall India Ltd.,2001
3. Robin J. Wilson, Introduction to Graph Theory, Longman Group Ltd.,2010
4. J.A. Bondy and U.S.R. Murty. Graph theory with Applications

Sample Course Level Assessment Questions.

Course Outcome 1 (CO1):

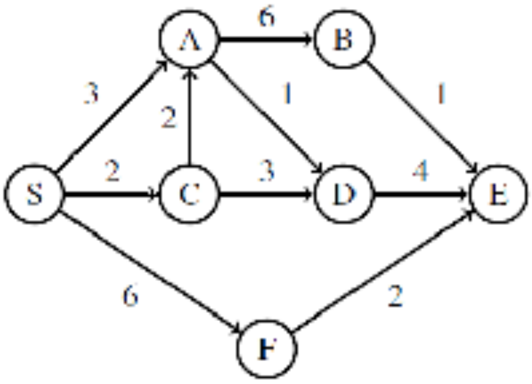
1. Differentiate a walk, path and circuit in a graph.
2. Is it possible to construct a graph with 12 vertices such that two of the vertices have degree 3 and the remaining vertices have degree 4? Justify
3. Prove that a simple graph with n vertices must be connected, if it has more than $\frac{(n-1)(n-2)}{2}$ edges.
4. Prove the statement: If a graph (connected or disconnected) has exactly two odd degree, then there must be a path joining these two vertices.

Course Outcome 2 (CO2):

1. Define Hamiltonian circuit and Euler graph. Give one example for each.
2. Define directed graphs. Differentiate between symmetric digraphs and asymmetric digraphs.
3. Prove that a connected graph G is an Euler graph if all vertices of G are of even degree.
4. Prove that a graph G of n vertices always has a Hamiltonian path if the sum of the degrees of every pair of vertices Vi, Vj in G satisfies the condition $d(Vi) + d(Vj) = n - 1$

Course Outcome 3 (CO3):

1. Discuss the centre of a tree with suitable example.
2. Define binary tree. Then prove that number of pendant vertices in a binary tree is $\frac{(n+1)}{2}$
3. Prove that a tree with n vertices has $n - 1$ edges.
4. Explain Floyd Warshall algorithm.
5. Run Dijkstra's algorithm on the following directed graph, starting at vertex ***S***.



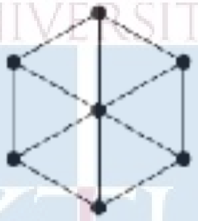
Course Outcome 4 (CO4):

1. Define edge connectivity, vertex connectivity and separable graphs. Give an example for each.

2. Prove that a connected graph with ***n*** vertices and e edges has $e - n + 2$ edges.

3. Prove the statement: Every cut set in a connected graph G must also contain at least one branch of every spanning tree of G.

4. Draw the geometrical dual ***(G*)*** of the graph given below, also check whether ***G*** and **G*** are self-duals or not, substantiate your answer clearly.



Course Outcome 5 (CO5):

1. Show that if ***A(G)*** is an incidence matrix of a connected graph ***G*** with ***n*** vertices, then rank of ***A(G)*** is ***n−1***.

2. Show that if ***B*** is a cycle matrix of a connected graph ***G*** with ***n*** vertices and ***m*** edges, then rank ***B*** = ***m−n+1***.

3. Derive the relations between the reduced incidence matrix, the fundamental cycle matrix, and the fundamental cut-set matrix of a graph ***G***.

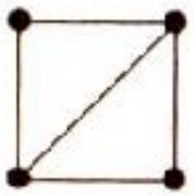
4. Characterize simple, self-dual graphs in terms of their cycle and cut-set matrices.

Course Outcome 6 (CO6):

1. Show that an n vertex graph is a tree iff its chromatic polynomial is $Pn(\lambda) = \lambda(\lambda - 1)^{n-1}$

2. Prove the statement: "A covering g of a graph is minimal if g contains no path of length three or more."

3. Find the chromatic polynomial of the graph



Model Question paper

QP Code : Total Pages: 4

Reg No.:_______________ Name:__________________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

IV SEMESTER B.TECH DEGREE EXAMINATION, MONTH and YEAR

Course Code: MAT 206

Course Name: GRAPH THEORY

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries3 marks. Marks

1 Construct a simple graph of 12 vertices with two of them having degree 1, three having degree 3 and the remaining seven having degree 10. (3)

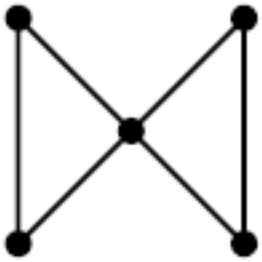
2 What is the largest number of vertices in a graph with 35 edges, if all vertices are of degree at least 3 ? (3)

3 Define a Euler graph. Give an example of Eulerian graph which is not Hamiltonian (3)

4 Give an example of a strongly connected simple digraph without a directed Hamiltonian path. (3)

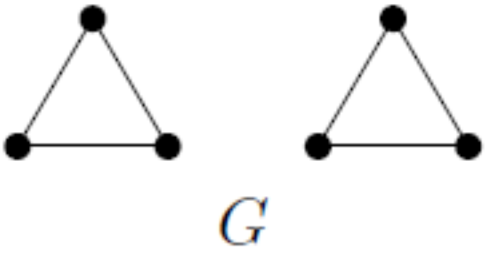
5 What is the sum of the degrees of any tree of ***n*** vertices? (3)

6 How many spanning trees are there for the following graph (3)



7 Show that in a simple connected planar graph ***G*** having ***V***-vertices, ***E***-edges, and no triangles ***E <= 3V - 6***. (3)

8 Let G be the following disconnected planar graph. Draw its dual G*, and the dual of the dual (G*)*. (3)



9 Consider the circuit matrix ***B*** and incidence matrix ***A*** of a simple connected graph whose columns are arranged using the same order of edges. Prove that every row of ***B*** is orthogonal to every row of ***A***? (3)

10 A graph is *critical* if the removal of any one of its vertices (and the edges adjacent to that vertex) results in a graph with a lower chromatic number. Show that K_n is critical for all ***n > 1***. (3)

PART B

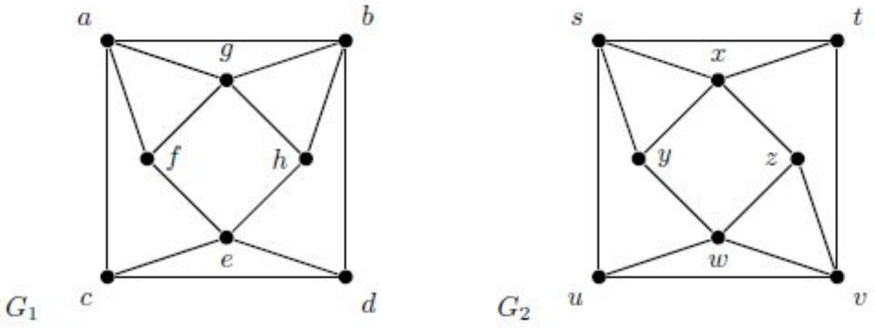
Answer any one Question from each module. Each question carries 14 Marks

11 a) Prove that for any simple graph with at least two vertices has two vertices of the same degree. (6)

b) Prove that in a complete graph with ***n*** vertices there are **(n-1)/2** edge disjoint Hamiltonian circuits and ***n >= 3*** (8)

OR

12 a) Determine whether the following graphs ***$G_1 = (V_1, E_1)$*** and ***$G_2 = (V_2, E_2)$*** are isomorphic or not. Give justification. (6)



b) Prove that a simple graph with ***n*** vertices and ***k*** components can have at most ***(n-k) (n-k+1)/2*** edges (8)

13 a) Let ***S*** be a set of 5 elements. Construct a graph ***G*** whose vertices are subsets of ***S*** of size 2 and two such subsets are adjacent in ***G*** if they are disjoint. (8)

i. Draw the graph ***G***.

ii. How many edges must be added to ***G*** in order for ***G*** to have a Hamiltonian cycle?

b) Let ***G*** be a graph with exactly two connected components, both being Eulerian. What is the minimum number of edges that need to be added to ***G*** to obtain an Eulerian graph? (6)

OR

14 a) Show that a ***k***-connected graph with no hamiltonian cycle has an independent set of size ***k + 1***. (8)

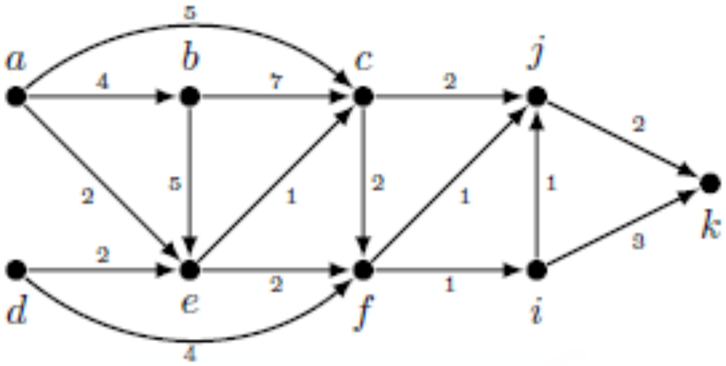
b) i. Let ***G*** be a graph that has exactly two connected components, both being Hamiltonian graphs. Find the minimum number of edges that one needs to add to ***G*** to obtain a Hamiltonian graph. (6)

ii. For which values of ***n*** the graph ***Q_n*** (hyper-cube on ***n*** vertices) is Eulerian.

15 a) A tree ***T*** has at least one vertex ***v*** of degree 4, and at least one vertex ***w*** of degree 3. Prove that ***T*** has at least 5 leaves. (5)

b) Write Dijkstra’s shortest path algorithm. (9)

Consider the following weighted directed graph ***G***.



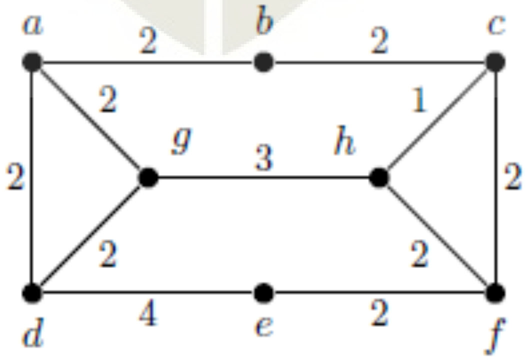
Find the shortest path between ***a*** and every other vertices in ***G*** using Dijkstra’s shortest path algorithm.

OR

16 a) Define pendent vertices in a binary tree? Prove that the number of pendent vertices in a binary tree with ***n*** vertices is ***(n+1)/2.*** (5)

b) Write Prim’s algorithm for finding minimum spanning tree. (9)

Find a minimum spanning tree in the following weighted graph, using Prim's algorithm.



Determine the number of minimum spanning trees for the given graph.

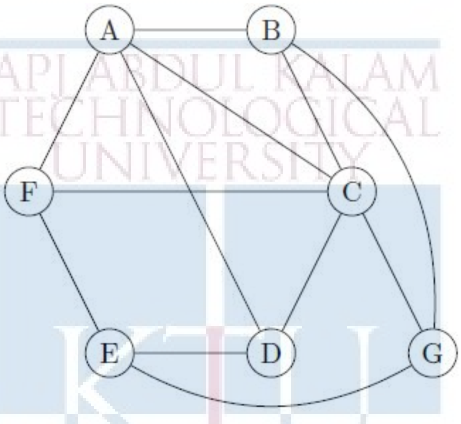
17 a) i. State and prove Euler's Theorem relating the number of faces, edges and vertices for a planar graph. (9)

ii. If ***G*** is a 5-regular simple graph and $|V| = 10$, prove that ***G*** is non-planar.

b) Let ***G*** be a connected graph and ***e*** an edge of ***G***. Show that ***e*** is a cut-edge if and only if ***e*** belongs to every spanning tree. (5)

OR

18 a) State Kuratowski's theorem, and use it to show that the graph G below is not planar. Draw G on the plane without edges crossing. Your drawing should use the labelling of the vertices given. (9)



b) Let ***G*** be a connected graph and ***e*** an edge of ***G***. Show that ***e*** belongs to a loop if and only if ***e*** belongs to no spanning tree. (5)

19 a) Define the circuit matrix ***B(G)*** of a connected graph ***G*** with ***n*** vertices and ***e*** edges with an example. Prove that the rank of ***B(G)*** is ***e-n+1*** (7)

b) Give the definition of the chromatic polynomial $P_G(k)$. Directly from the definition, prove that the chromatic polynomials of W_n and C_n satisfy the identity $P_{W_n}(k) = k\, P_{C_{n-1}}(k - 1)$. (7)

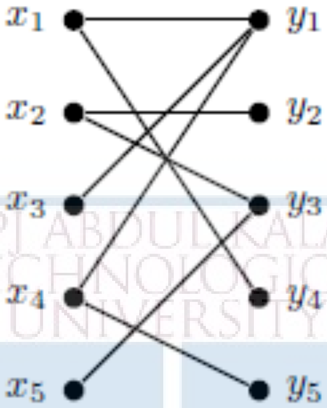
OR

20 a) Define the incidence matrix of a graph ***G*** with an example. Prove that the rank of an incidence matrix of a connected graph with ***n*** vertices is ***n-1.*** (4)

b) i. A graph ***G*** has chromatic polynomial ***$P_G(k)$ = k^4-4k^3+5k^2-2k***. How many vertices and edges does ***G*** have? Is ***G*** bipartite? Justify your answers.

ii. Find a maximum matching in the graph below and use Hall's theorem to show that it is indeed maximum.

(10)



****

Assignments

Assignment must include applications of the above theory in Computer Science.

Teaching Plan		
No	Topic	No. of Lectures
1	**Module-I (Introduction to Graphs)**	**(8)**
1.	Introduction- Basic definition – Application of graphs – finite and infinite graphs, bipartite graphs,	1
2.	Incidence and Degree – Isolated vertex, pendent vertex and Null graph	1
3.	Paths and circuits	1
4.	Isomorphism	1
5.	Sub graphs, walks	1
6.	Paths and circuits	1
7.	Connected graphs.	1
8.	Disconnected graphs and components	1
2	**Module-II (Eulerian and Hamiltonian graphs)**	**(8)**
1.	Euler graphs	1
2.	Operations on graphs	1
3.	Hamiltonian paths and circuits	1
4.	Hamiltonian paths circuits	1
5.	Travelling salesman problem	1
6.	Directed graphs – types of digraphs,	1
7.	Digraphs and binary relation, Directed paths	1
8.	Fleury’s algorithm	1
3	**Module-III (Trees and Graph Algorithms)**	(11)
1.	Trees – properties	1
2.	Trees – properties	1
3.	Trees – properties, pendent vertex	1
4.	Distance and centres in a tree	1

5.	Rooted and binary tree	1
6.	Counting trees	1
7.	Spanning trees, Fundamental circuits	1
8.	Prim’s algorithm	1
9.	Kruskal’s algorithm	1
10.	Dijkstra’s shortest path algorithm	1
11.	Floyd-Warshall shortest path algorithm	1
4	**Module-IV (Connectivity and Planar Graphs)**	(9)
1.	Vertex Connectivity, Edge Connectivity	1
2.	Cut set and Cut Vertices	1
3.	Fundamental circuits	1
4.	Fundamental circuits	1
5.	Planar graphs	1
6.	Kuratowski’s theorem	1
7.	Different representations of planar graphs	1
8.	Euler's theorem	1
9.	Geometric dual	1
5	**Module-V (Graph Representations and Vertex Colouring)**	(9)
1.	Matrix representation of graphs- Adjacency matrix, Incidence Matrix	1
2.	Circuit Matrix, Path Matrix	1
3.	Colouring- chromatic number,	1
4.	Chromatic polynomial	1
5.	Matching	1
6.	Covering	1
7.	Four colour problem and five colour problem	1

8.	Four colour problem and five colour problem	1
9.	Greedy colouring algorithm.	1

CST 202	Computer Organization and Architecture	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	3	1	0	4	2019

Preamble:

The course is prepared with the view of enabling the learners capable of understanding the fundamental architecture of a digital computer. Study of Computer Organization and Architecture is essential to understand the hardware behind the code and its execution at physical level by interacting with existing memory and I/O structure. It helps the learners to understand the fundamentals about computer system design so that they can extend the features of computer organization to detect and solve problems occurring in computer architecture.

Prerequisite : Topics covered under the course Logic System Design (CST 203)

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Recognize and express the relevance of basic components, I/O organization and pipelining schemes in a digital computer **(Cognitive knowledge: Understand)**
CO2	Explain the types of memory systems and mapping functions used in memory systems **(Cognitive Knowledge Level: Understand)**
CO3	Demonstrate the control signals required for the execution of a given instruction **(Cognitive Knowledge Level: Apply)**)
CO4	Illustrate the design of Arithmetic Logic Unit and explain the usage of registers in it (**Cognitive Knowledge Level: Apply**)
CO5	Explain the implementation aspects of arithmetic algorithms in a digital computer **(Cognitive Knowledge Level:Apply)**
CO6	Develop the control logic for a given arithmetic problem (**Cognitive Knowledge Level: Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓						✓		✓
CO3	✓	✓	✓	✓						✓		✓
CO4	✓	✓	✓	✓						✓		✓
CO5	✓	✓	✓							✓		✓
CO6	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test1 (%)	**Test2 (%)**	
Remember	20	20	30
Understand	40	40	30
Apply	40	40	40
Analyze			

Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module 1

Basic Structure of computers – functional units - basic operational concepts - bus structures. Memory locations and addresses - memory operations,Instructions and instruction sequencing , addressing modes.

Basic processing unit – fundamental concepts – instruction cycle – execution of a complete instruction - single bus and multiple bus organization

Module 2

Register transfer logic: inter register transfer – arithmetic, logic and shift micro operations.

Processor logic design: - processor organization – Arithmetic logic unit - design of arithmetic circuit - design of logic circuit - Design of arithmetic logic unit - status register – design of shifter - processor unit – design of accumulator.

Module 3

Arithmetic algorithms: Algorithms for multiplication and division (restoring method) of binary numbers. Array multiplier , Booth's multiplication algorithm.

Pipelining: Basic principles**,** classification of pipeline processors, instruction and arithmetic pipelines (Design examples not required), hazard detection and resolution.

Module 4

Control Logic Design: Control organization – Hard_wired control-microprogram control – control of processor unit - Microprogram sequencer,micro programmed CPU organization - horizontal and vertical micro instructions.

Module 5

I/O organization: accessing of I/O devices – interrupts, interrupt hardware -Direct memory access.

Memory system: basic concepts – semiconductor RAMs. memory system considerations – ROMs, Content addressable memory, cache memories - mapping functions.

Text Books

1. Hamacher C., Z. Vranesic and S. Zaky, Computer Organization ,5/e, McGraw Hill, 2011
2. Mano M. M., Digital Logic & Computer Design, PHI, 2004
3. KaiHwang, Faye Alye Briggs, Computer architecture and parallel processing McGraw-Hill, 1984

Reference Books

1. Mano M. M., Digital Logic & Computer Design, 3/e, Pearson Education, 2013.
2. Patterson D.A. and J. L. Hennessy, Computer Organization and Design, 5/e, Morgan Kaufmann Publishers, 2013.
3. William Stallings, Computer Organization and Architecture: Designing for Performance, Pearson, 9/e, 2013.
4. Chaudhuri P., Computer Organization and Design, 2/e, Prentice Hall, 2008.
5. Rajaraman V. and T. Radhakrishnan, Computer Organization and Architecture, Prentice Hall, 2011

Sample Course Level Assessment Questions

Course Outcome1(CO1): Which are the registers involved in a memory access operation and how are they involved in it?

Course Outcome 2(CO2): Explain the steps taken by the system to handle a write miss condition inside the cache memory.

Course Outcome 3(CO3): Generate the sequence of control signals required for the execution of the instruction MOV [R1],R2 in a threebus organization.

Course Outcome 4(CO4): Design a 4-bit combinational logic shifter with 2 control signals H0 and H1 that perform the following operations :

H1	H0	Operation
0	0	Transfer 1's to all output line
0	1	No shift operation
1	0	Shift left
1	1	Shift right

Course Outcome 5(CO5): Explain the restoring algorithm for binary division. Also trace the algorithm to divide $(1001)_2$ by $(11)_2$

Course Outcome 6(CO6): Design a software control logic based on microprogramed control to perform the addition of 2 signed numbers represented in sign magnitude form.

Model Question Paper

QP CODE: PAGES:2

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

THIRD SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 202

Course Name: Computer organization and architecture

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Give the significance of instruction cycle.
2. Distinguish between big endian and little endian notations. Also give the significance of these notations.
3. Compare I/O mapped I/O and memory mapped I/O.
4. Give the importance of interrupts in I/O interconnection.
5. Justify the significance of status register.
6. How does the arithmetic circuitry perform logical operations in an ALU.
7. Illustrate divide overflow with an example.
8. Write notes on arithmetic pipeline.
9. Briefly explain the role of micro program sequence.

10. Differentiate between horizontal and vertical micro instructions.

Part B

Answer any one Question from each module. Each question carries 14 Marks

11.

11.(a) What is the significance of addressing modes in computer architecture.

(4)

11.(b) Write the control sequence for the instruction DIV R1,[R2] in a three bus structure. (10)

OR

12. Explain the concept of a single bus organization with help of a diagram. Write the control sequence for the instruction ADD [R1],[R2].

(14)

13. Explain various register transfer logics.

(14)

OR

14.

14.(a) Design a 4 bit combinational logic shifter with 2 control signals H1 and H2 that perform the following operations (bit values given in parenthesis are the values of control variable H1 and H2 respectively.) : Transfer of 0's to S (00), shift right (01), shift left (10), no shift (11).

(5)

14.(b) Design an ALU unit which will perform arithmetic and logic operation with a given binary adder.

(9)

15.

15.(a) Give the logic used behind Booth's multiplication algorithm.

(4)

15.(b) Identify the appropriate algorithm available inside the system to perform the multiplication between -14 and -9. Also trace the algorithm for the above input.

(10)

OR

16.

16.(a) List and explain the different pipeline hazards and their possible solutions

(10)

16.(b) Design a combinational circuit for 3x2 multiplication.

(4)

17. Design a hardwared control unit used to perform addition/subtraction of 2 numbers represented in sign magnitude form.

(14)

OR

18. Give the structure of the micro program sequencer and its role in sequencing the micro instructions.

(14)

19.

19.(a) Explain the different ways in which interrupt priority schemes can be implemented

(10)

19.(b) Give the structure of SRAM cell.

(4)

OR

20.

20.(a) Explain the various mapping functions available in cache memory.

(9)

20.(b) Briefly explain content addressable memory.

(5)

TEACHING PLAN		
No	**Contents**	**No of Lecture Hrs**
	Module 1 : (Basic Structure of computers) (9 hours)	
1.1	Functional units,basic operational concepts,bus structures (introduction)	1
1.2	Memory locations and addresses , memory operations	1
1.3	Instructions and instruction sequencing	1
1.4	Addressing modes	1
1.5	Fundamental concepts of instruction execution, instruction cycle	1
1.6	Execution of a complete instruction - single bus organization (Lecture 1)	1
1.7	Execution of a complete instruction - single bus organization (Lecture 2)	1
1.8	Execution of a complete instruction - multiple bus organization (Lecture 1)	1
1.9	Execution of a complete instruction - multiple bus organization (Lecture 2)	1
	Module 2 :(Register transfer logic and Processor logic design) (10 hours)	
2.1	Inter register transfer – arithmetic micro operations	1
2.2	Inter register transfer – logic and shift micro operations	1
2.3	Processor organization	1
2.4	Design of arithmetic circuit	1
2.5	Design of logic circuit	1
2.6	Design of arithmetic logic unit	1
2.7	Design of status register	1
2.8	Design of shifter - processor unit	1

2.9	Design of accumulator (Lecture 1)	1
2.10	Design of accumulator (Lecture 2)	1
Module 3 : (Arithmetic algorithms and Pipelining) (9 hours)		
3.1	Algorithm for multiplication of binary numbers	1
3.2	Algorithm for division (restoring method) of binary numbers	1
3.3	Array multiplier	1
3.4	Booth’s multiplication algorithm	1
3.5	Pipelining: Basic principles	1
3.6	Classification of pipeline processors (Lecture 1)	1
3.7	Classification of pipeline processors (Lecture 2)	1
3.8	Instruction and arithmetic pipelines (Design examples not required)	1
3.9	Hazard detection and resolution	1
Module 4 :(Control Logic Design) (9 hours)		
4.1	Control organization –design of hardwired control logic (Lecture 1)	1
4.2	Control organization –design of hardwired control logic (Lecture 2)	1
4.3	Control organization –design of hardwired control logic (Lecture 3)	1
4.4	Design of microprogram control logic–control of processor unit (Lecture1)	1
4.5	Design of microprogram control logic–control of processor unit (Lecture2)	1
4.6	Design of microprogram control logic–control of processor unit (Lecture3)	1
4.7	Microprogram sequencer	1
4.8	Micro programmed CPU organization	1
4.9	Microinstructions –horizontal and vertical micro instructions	1
Module 5 : (Basic processing units, I/O and memory) (8 hours)		
5.1	Accessing of I/O devices –interrupts	1
5.2	Interrupt hardware	1

5.3	Direct memory access	1
5.4	Memory system: basic concepts –semiconductor RAMs	1
5.5	Memory system considerations – ROMs	1
5.6	Content addressable memory	1
5.7	Cache memories -mapping functions (Lecture 1)	1
5.8	Cache memories -mapping functions (Lecture 2)	1

CST 204	Database Management Systems	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	3	1	0	4	2019

Preamble: This course provides a clear understanding of fundamental principles of Database Management Systems (DBMS) with special focus on relational databases to the learners. The topics covered in this course are basic concepts of DBMS, Entity Relationship (ER) model, Relational Database principles, Relational Algebra, Structured Query Language (SQL), Physical Data Organization, Normalization and Transaction Processing Concepts. The course also gives a glimpse of the alternative data management model, NoSQL. This course helps the learners to manage data efficiently by identifying suitable structures to maintain data assets of organizations and to develop applications that utilize database technologies.

Prerequisite: Topics covered under the course Data Structures (CST 201), Exposure to a High Level Language like C/python.

Course Outcomes: After the completion of the course the student will be able to

CO1	Summarize and exemplify fundamental nature and characteristics of database systems **(Cognitive Knowledge Level: Understand)**
CO2	Model real word scenarios given as informal descriptions, using Entity Relationship diagrams. **(Cognitive Knowledge Level: Apply**)
CO3	Model and design solutions for efficiently representing and querying data using relational model (**Cognitive Knowledge Level: Analyze**)
CO4	Demonstrate the features of indexing and hashing in database applications **(Cognitive Knowledge Level: Apply**)
CO5	Discuss and compare the aspects of Concurrency Control and Recovery in Database systems **(Cognitive Knowledge Level: Apply**)
CO6	Explain various types of NoSQL databases (**Cognitive Knowledge Level: Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓							✓		✓
CO5	✓	✓	✓							✓		✓
CO6	✓	✓	✓		✓					✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test1 (%)	**Test2 (%)**	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module 1: Introduction & Entity Relationship (ER) Model

Concept & Overview of Database Management Systems (DBMS) - Characteristics of Database system, Database Users, structured, semi-structured and unstructured data. Data Models and Schema - Three Schema architecture. Database Languages, Database architectures and classification.

ER model **-** Basic concepts, entity set & attributes, notations, Relationships and constraints, cardinality, participation, notations, weak entities, relationships of degree 3.

Module 2: Relational Model

Structure of Relational Databases - Integrity Constraints, Synthesizing ER diagram to relational schema

Introduction to Relational Algebra - select, project, cartesian product operations, join - Equi-join, natural join. query examples, introduction to Structured Query Language (SQL), Data Definition Language (DDL), Table definitions and operations – CREATE, DROP, ALTER, INSERT, DELETE, UPDATE.

Module 3: SQL DML (Data Manipulation Language), Physical Data Organization

SQL DML (Data Manipulation Language) - SQL queries on single and multiple tables, Nested queries (correlated and non-correlated), Aggregation and grouping, Views, assertions, Triggers, SQL data types.

Physical Data Organization - Review of terms: physical and logical records, blocking factor, pinned and unpinned organization. Heap files, Indexing, Singe level indices, numerical examples, Multi-level-indices, numerical examples, B-Trees & B+-Trees (structure only, algorithms not required), Extendible Hashing, Indexing on multiple keys – grid files.

Module 4: Normalization

Different anomalies in designing a database, The idea of normalization, Functional dependency, Armstrong's Axioms (proofs not required), Closures and their computation, Equivalence of Functional Dependencies (FD), Minimal Cover (proofs not required). First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), Boyce Codd Normal Form (BCNF), Lossless join and dependency preserving decomposition, Algorithms for checking Lossless Join (LJ) and Dependency Preserving (DP) properties.

Module 5: Transactions, Concurrency and Recovery, Recent Topics

Transaction Processing Concepts - overview of concurrency control, Transaction Model, Significance of concurrency Control & Recovery, Transaction States, System Log, Desirable Properties of transactions.

Serial schedules, Concurrent and Serializable Schedules, Conflict equivalence and conflict serializability, Recoverable and cascade-less schedules, Locking, Two-phase locking and its variations. Log-based recovery, Deferred database modification, check-pointing.

Introduction to NoSQL Databases, Main characteristics of Key-value DB (examples from: Redis), Document DB (examples from: MongoDB)

Main characteristics of Column - Family DB (examples from: Cassandra) and Graph DB (examples from : ArangoDB)

Text Books

1. Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education, 2013.
2. Sliberschatz A., H. F. Korth and S. Sudarshan, Database System Concepts, 6/e, McGraw Hill, 2011.

Reference Books:

1. Adam Fowler, NoSQL for Dummies, John Wiley & Sons, 2015
2. NoSQL Data Models: Trends and Challenges (Computer Engineering: Databases and Big Data), Wiley, 2018
3. Web Resource: https://www.w3resource.com/redis/
4. web Resource: https://www.w3schools.in/category/mongodb/
5. Web Resource: https://www.tutorialspoint.com/cassandra/cassandra_introduction.htm
6. Web Resource : https://www.tutorialspoint.com/arangodb/index.htm

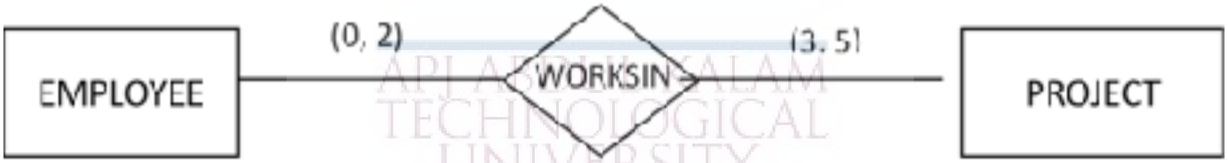
Sample Course Level Assessment Questions

Course Outcome1 (CO1):

1. List out any three salient features of database systems, which distinguish it from a file system.
2. Give one example each for logical and physical data independence.

Course Outcome 2(CO2):

1. What facts about the relationships between entities EMPLOYEE and PROJECT are conveyed by the following ER diagram?



1. Design an ER diagram for the following scenario:
   There is a set of teams, each team has an ID (unique identifier), name, main stadium, and to which city this team belongs. Each team has many players, and each player belongs to one team. Each player has a number (unique identifier), name, DoB, start year, and shirt number that he uses. Teams play matches, in each match there is a host team and a guest team.

Course Outcome 3(CO3):

1. For the SQL query, SELECT *A, B FROM R WHERE B='apple' AND C = 'orange'* on the table R(A, B, C, D), where A is a key, write any three equivalent relational algebra expressions.
2. Given the FDs P→Q, P→R, QR→S, Q→T, QR→U, PR→U, write the sequence of *Armstrong's Axioms* needed to arrive at the following FDs: (a) P → T (b) PR → S (c) QR → SU
3. Consider a relation PLAYER (PLAYER-NO, PLAYER-NAME, PLAYER-POSN, TEAM, TEAM-COLOR, COACH-NO, COACH-NAME, TEAM-CAPTAIN). Assume that PLAYER-NO is the *only* key of the relation and that the following dependencies hold:

   TEAM→{TEAM-COLOR, COACH-NO, TEAM-CAPTAIN}
   COACH-NO→COACH-NAME.

   i. Is the relation in 2NF? If not, decompose to 2NF.
   ii. Is the relation in 3NF? If not, decompose to 3NF.

4. In the following tables foreign keys have the same name as primary keys except DIRECTED-BY, which refers to the primary key ARTIST-ID. Consider only *single-director* movies.
MOVIES(MOVIE-ID, MNAME, GENRE, LENGTH, DIRECTED-BY)
ARTIST(ARTIST-ID, ANAME)
ACTING(ARTIST-ID, MOVIE-ID)
Write SQL expressions for the following queries:
    (a) *Name(s) and director name(s) of movie(s) acted by 'Jenny'.*
    (b) *Names of actors who have never acted with 'Rony'*
    (c) *Count of movies genre-wise.*
    (d) *Name(s) of movies with maximum length.*

Course Outcome 4(CO4):

1. Consider an EMPLOYEE file with 10000 records where each record is of size 80 bytes. The file is sorted on employee number (15 bytes long), which is the primary key. Assuming un-spanned organization, block size of 512 bytes and block pointer size of 5 bytes. Compute the number of block accesses needed for retrieving an employee record based on employee number if (i) No index is used (ii) Multi-level primary index is used.

Course Outcome 5(CO5):

1. Determine if the following schedule is *recoverable*. Is the schedule *cascade-less*? Justify your answer. *r1(X), r2(Z), r1(Z), r3(X), r3(Y), w1(X), c1, w3(Y), c3, r2(Y), w2(Z), w2(Y), c2.* (*Note: ri(X)/wi(X)* means transaction *Ti* issues read/write on item X; *ci* means transaction *Ti* commits.)
2. Two-phase locking protocol ensures serializability. Justify.

Course Outcome 6(CO6):

1. List out any three salient features of NoSQL databases. Give example of a document in MongoDB.

Model Question paper

QPCODE

Reg No:__________________

Name:____________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FOURTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 204

Course Name: **Database Management Systems**

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1 List out any three salient features of a database systems.

2 When is multi-valued composite attribute used in ER modelling?

3 For the SQL query, SELECT *A, B FROM R WHERE B='apple' AND C = 'orange'* on the table R(A, B, C, D), where A is a key, write any two equivalent relational algebra expressions.

4 Outline the concept of *theta*-join.

5 How is the purpose of *where* clause is different from that of having clause?

6 What is the use of a trigger?

7 When do you say that a relation is not in 1NF?

8 Given the FDs P→Q, P→R, QR→S, Q→T, QR→U, PR→U, write the sequence of Armstrong's Axioms needed to arrive at a. P → T b. PR → S

9 What is meant by the lost update problem?

10 What is meant by check pointing?

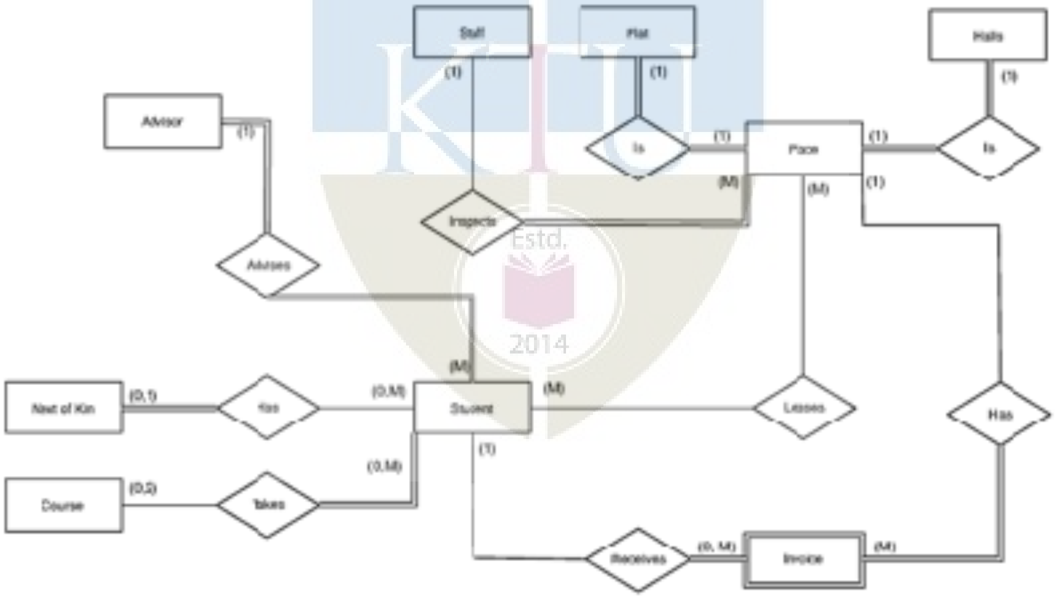
PART B

Answer any one Question from each module. Each question carries 14 Marks

11 a. Design an ER diagram for the following scenario: There is a set of teams, each team has an ID (unique identifier), name, main stadium, and to which city this team belongs. Each team has many players, and each player belongs to one team. Each player has a number (unique identifier), name, DoB, start year, and shirt number that he uses. Teams play matches, in each match there is a host team and a guest team. The match takes place in the stadium of the host team. For each match we need to keep track of the following: The date on which the game is played The final result of the match. The players participated in the match. For each player, how many goals he scored, whether or not he took yellow card, and whether or not he took red card. During the match, one player may substitute another player. We want to capture this substitution and the time at which it took place. Each match has exactly three referees. For each referee we have an ID (unique identifier), name, DoB, years of experience. One referee is the main referee and the other two are assistant referee. (14)

OR

12 a. Interpret the the following ER diagram. (8)



b. Distinguish between physical data independence and logical data independence with suitable examples. (6)

13 **EMPLOYEE(ENO, NAME, ADDRESS, DOB, AGE, GENDER, SALARY, DNUM, SUPERENO)** (14)
DEPARTMENT(DNO, DNAME, DLOCATION, DPHONE, MGRENO)
PROJECT(PNO, PNAME, PLOCATION, PCOST, CDNO)

DNUM is a foreign key that identifies the department to which an employee belongs. MGRENO is a foreign key identifying the employee who manages the department. CDNO is a foreign key identifying the department that controls the project. SUPERENO is a foreign key identifying the supervisor of each employee.

Write relational algebra expressions for the following queries:-

(a) Names of female employees whose salary is more than 20000.
(b) Salaries of employee from ‘Accounts’ department
(c) Names of employees along with his/her superviser’s name
(d) For each employee return name of the employee along with his department name and the names of projects in which he/she works
(e) Names of employees working in all the departments

OR

14 a.Write SQL DDL statements for the the following (Assume suitable domain types): (10)

i. Create the tables STUDENT(ROLLNO, NAME, CLASS, SEM, ADVISER), FACULTY(FID, NAME, SALARY, DEPT). Assume that ADVISER is a foreign key referring FACUTY table.

ii. Delete department with name ‘CS’ and all employees of the department.

iii. Increment salary of every faculty by 10%.

b.Illustrate foreign key constraint with a typical example. (4)

15 For the relation schema below, give an expression in SQL for each of the queries that follows: (14)

employee(employee-name, street, city)
works(employee-name, company-name, salary)
company(company-name, city)
manages(employee-name, manager-name)

a) Find the names, street address, and cities of residence for all employees who work for the Company 'RIL Inc.' and earn more than $10,000.

b) Find the names of all employees who live in the same cities as the companies for which they work.

c) Find the names of all employees who do not work for 'KYS Inc.'. Assume that all people work for exactly one company.

d) Find the names of all employees who earn more than every employee of 'SB Corporation'. Assume that all people work for at most one company.

e) List out number of employees company-wise in the decreasing order of number of employees.

OR

16 a. Consider an EMPLOYEE file with 10000 records where each record is of size 80 bytes. The file is sorted on employee number (15 bytes long), which is the primary key. Assuming un-spanned organization and block size of 512 bytes compute the number of block accesses needed for selecting records based on employee number if, (9)

i. No index is used
ii. Single level primary index is used
iii. Multi-level primary index is used

Assume a block pointer size of 6 bytes.

b. Illustrate correlated and non-correlated nested queries with real examples. (5)

17 a. Illstrate3NF and BCNF with suitable real examples. (6)

b. Given a relation R(A1,A2,A3,A4,A5) with functional dependencies A1→A2A4 and A4→A5, check if the decomposition R1(A1,A2,A3), R2(A1,A4), R3(A2,A4,A5) is lossless. (8)

OR

18 a. Consider the un-normalized relation R(A, B, C, D, E, F, G) with the FDs A→B , AC→G, AD→EF, EF→G, CDE→AB. Trace the normalization process to reach 3NF relations. (7)

b. Illustrate Lossless Join Decomposition and Dependency Preserving Decomposition with typical examples. (7)

19 a. Discuss the four ACID properties and their importance. (7)

b. Determine if the following schedule is conflict serializable. Is the schedule recoverable? Is the schedule cascade-less? Justify your answers. (7)

r1(X), r2(Z), r1(Z), r3(X), r3(Y), w1(X), c1, w3(Y), c3, r2(Y), w2(Z), w2(Y), c2

(Note: ri(X)/wi(X) means transaction Ti issues read/write on item X; ci means transaction Ti commits.)

OR

20 a. Discuss the main characteristics of Key-value DB and Graph DB. (7)

b. Illustrate two-phase locking with a schedule containing three transactions. Argue that 2PL ensures serializability. Also argue that 2Pl can lead to deadlock. (7)

Teaching Plan

	Course Name	Hours (48)
	Module 1: Introduction & ER Model	**8**
1.1	Concept & Overview of DBMS, Characteristics of DB system, Database Users.	1
1.2	Structured, semi-structured and unstructured data. Data Models and Schema	1
1.3	Three-Schema-architecture. Database Languages	1
1.4	Database architectures and classification	1
1.5	ER model: basic concepts, entity set & attributes, notations	1
1.6	Relationships and constraints – cardinality, participation, notations	1
1.7	Weak entities, relationships of degree 3	1
1.8	ER diagram – exercises	1
	Module 2: Relational Model	**7**
2.1	Structure of relational Databases, Integrity Constraints	1
2.2	Synthesizing ER diagram to relational schema, Introduction to relational algebra.	1
2.3	Relational algebra: select, project, Cartesian product operations	1
2.4	Relational Algebra: join - Equi-join, Natural join	1
2.5	Query examples	1
2.6	Introduction to SQL, important data types	1
2.7	DDL, Table definitions and operations – CREATE, DROP, ALTER, INSERT, DELETE, UPDATE	1
	Module 3: SQL DML, Physical Data Organization	**11**
3.1	SQL DML, SQL queries on single and multiple tables	1
3.2	Nested queries (correlated and non-correlated)	1
3.3	Aggregation and grouping	1

	Course Name	**Hours (48)**
3.4	Views, assertions (with examples)	1
3.5	Triggers (with examples), SQL data types	1
3.6	Review of terms: physical and logical records, blocking factor, pinned and unpinned organization. Heap files, Indexing	1
3.7	Singe level indices, numerical examples	1
3.8	Multi-level-indices, numerical examples	1
3.9	B-Trees and B+Trees (structure only, algorithms not required)	1
3.10	Extendible Hashing	1
3.11	Indexing on multiple keys – grid files	1
	Module 4: Normalization	**8**
4.1	Different anomalies in designing a database, The idea of normalization	1
4.2	Functional dependency, Armstrong's Axioms (proofs not required)	1
4.3	Closures and their computation, Equivalence of FDs, minimal Cover (proofs not required).	1
4.4	1NF, 2NF	1
4.5	3NF, BCNF	1
4.6	Lossless join and dependency preserving decomposition	1
4.7	Algorithms for checking Lossless Join and Dependency preserving properties (Lecture 1)	1
4.8	Algorithms for checking Lossless Join and Dependency preserving properties (Lecture 2)	1
	Module 5: Transactions, Concurrency and Recovery, Recent Topics	**14**
5.1	Transaction Processing Concepts: Transaction Model	1
5.2	Overview of concurrency control, Significance of concurrency Control & Recovery	1
5.3	Transaction States, System Log	1

	Course Name	Hours (48)
5.4	Desirable Properties of transactions, Serial schedules	1
5.5	Concurrent and Serializable Schedules	1
5.6	Conflict equivalence and conflict serializability	1
5.7	Recoverable and cascade-less schedules	1
5.8	Locking, Two-phase locking, strict 2PL.	1
5.9	Log-based recovery	1
5.10	Deferred database modification (serial schedule), example	1
5.11	Deferred database modification (concurrent schedule) example, check-pointing	1
5.12	Introduction to NoSQL Databases	1
5.13	Main characteristics of Key-value DB (examples from: Redis), Document DB (examples from: MongoDB) [detailed study not expected]	1
5.14	Main characteristics of Column-Family DB (examples from: Cassandra) and Graph DB (examples from : ArangoDB) [detailed study not expected]	1

CST 206	**OPERATING SYSTEMS**	**Category**	**L**	**T**	**P**	**Credit**	**Year of Introduction**
		PCC	**3**	**1**	**0**	**4**	**2019**

Preamble: Study of operating system is an essential to understand the overall working of computer system, tradeoffs between performance and functionality and the division of jobs between hardware and software. This course introduces the concepts of memory management, device management, process management, file management and security & protection mechanisms available in an operating system. The course helps the learner to understand the fundamentals about any operating system design so that they can extend the features of operating system to detect and solve many problems occurring in operating system and to manage the computer resources appropriately.

Prerequisite: Topics covered in the courses are **Data Structures (CST 201)** and **Programming in C (EST 102)**

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain the relevance, structure and functions of Operating Systems in computing devices. (**Cognitive knowledge: Understand**)
CO2	Illustrate the concepts of process management and process scheduling mechanisms employed in Operating Systems. (**Cognitive knowledge: Understand**)
CO3	Explain process synchronization in Operating Systems and illustrate process synchronization mechanisms using Mutex Locks, Semaphores and Monitors (**Cognitive knowledge: Understand**)
CO4	Explain any one method for detection, prevention, avoidance and recovery for managing deadlocks in Operating Systems. (**Cognitive knowledge: Understand**)
CO5	Explain the memory management algorithms in Operating Systems. (**Cognitive knowledge: Understand**)
CO6	Explain the security aspects and algorithms for file and storage management in Operating Systems. **(Cognitive knowledge: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓							✓		✓
CO2	✓	✓	✓	✓						✓		✓
CO3	✓	✓	✓	✓						✓		✓
CO4	✓	✓	✓	✓						✓		✓
CO5	✓	✓	✓	✓						✓		✓
CO6	✓	✓	✓	✓						✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	PO#	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	**Test 1 (Marks in percentage)**	**Test 2 (Marks in percentage)**	**End Semester Examination (Marks in percentage)**
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

<u>Syllabus</u>

Module I

Introduction: Operating system overview – Operations, Functions, Service – System calls, Types – Operating System structure - Simple structure, Layered approach, Microkernel, Modules – System boot process.

Module II

Processes - Process states, Process control block, threads, scheduling, Operations on processes - process creation and termination – Inter-process communication - shared memory systems, Message passing systems.

Process Scheduling – Basic concepts- Scheduling criteria -scheduling algorithms- First come First Served, Shortest Job Firs, Priority scheduling, Round robin scheduling

Module III

Process synchronization- Race conditions – Critical section problem – Peterson's solution, Synchronization hardware, Mutex Locks, Semaphores, Monitors – Synchronization problems - Producer Consumer, Dining Philosophers and Readers-Writers.

Deadlocks: Necessary conditions, Resource allocation graphs, Deadlock prevention, Deadlock avoidance – Banker's algorithms, Deadlock detection, Recovery from deadlock.

Module IV

Memory Management: Concept of address spaces, Swapping, Contiguous memory allocation, fixed and variable partitions, Segmentation, Paging. Virtual memory, Demand paging, Page replacement algorithms.

Module V

File System: File concept - Attributes, Operations, types, structure – Access methods, Protection. File-system implementation, Directory implementation. Allocation methods.
Storage Management: Magnetic disks, Solid-state disks, Disk Structure, Disk scheduling, Disk formatting.

Text Book

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, ' Operating System Concepts' 9th Edition, Wiley India 2015.

Reference Books:

1. Andrew S Tanenbaum, "Modern Operating Systems" , 4th Edition, Prentice Hall, 2015.
2. William Stallings, "Operating systems", 6th Edition, Pearson, Global Edition, 2015.
3. Garry Nutt, Nabendu Chaki, Sarmistha Neogy, "Operating Systems", 3rd Edition, Pearson Education.
4. D.M.Dhamdhere, "Operating Systems", 2nd Edition, Tata McGraw Hill, 2011.
5. Sibsankar Haldar, Alex A Aravind, "Operating Systems", Pearson Education.

Sample Course Level Assessment Questions

Course Outcome1 (CO1): What is the main advantage of the micro kernel approach to system design? How do user program and system program interact in a microkernel architecture?

Course Outcome 2 (CO2): Define process. With the help of a neat diagram explain different states of process.

Course Outcome 3 (CO3): What do you mean by binary semaphore and counting semaphore? With C, explain implementation of wait () and signal().

Course Outcome 4 (CO4): Describe resource allocation graph for the following. a) with a deadlock b) with a cycle but no deadlock.

Course Outcome 5 (CO5): Consider the following page reference string 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. Find out the number of page faults if there are 4 page frames, using the following page replacement algorithms. i) LRU ii) FIFO iii) Optimal

Course Outcome 6 (CO6): Explain the different file allocation methods with advantages and disadvantages.

Model Question Paper

QP CODE: PAGES: ___

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FOURTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 206

Course name : OPERATING SYSTEMS

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. How does hardware find the Operating System kernel after system switch-on?

2. What is the purpose of system call in operating system?

3. Why is context switching considered as an overhead to the system?

4. How is inter process communication implement using shared memory?

5. Describe resource allocation graph for the following.

   a) with a deadlock   b)with a cycle but no deadlock.

6. What is critical section? What requirement should be satisfied by a solution to the critical section problem?

7. Consider the reference string 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page faults occur while using FCFS for the following cases.

   a) frame=2   b)frame=3

8. Differentiate between internal and external fragmentations.

9. Compare sequential access and direct access methods of storage devices.

10. Define the terms (i) Disk bandwidth (ii) Seek time.

PART-B(Answer any one question from each module)

11. a) Explain the following structures of operating system (i) Monolithic systems (ii) Layered Systems (iii) Micro Kernel (iv) Modular approach. **(12)**

b) Under what circumstances would a user be better of using a time sharing system than a PC or a single user workstation? **(2)**

OR

12. a) What is the main advantage of the micro kernel approach to system design? How do user program and system program interact in a microkernel architecture? **(8)**

b) Describe the differences between symmetric and asymmetric multiprocessing? What are the advantages and disadvantages of multiprocessor systems? **(6)**

13. a) Define process. With the help of a neat diagram explain different states of process. **(8)**

b) Explain how a new process can be created in Unix using fork system call. **(6)**

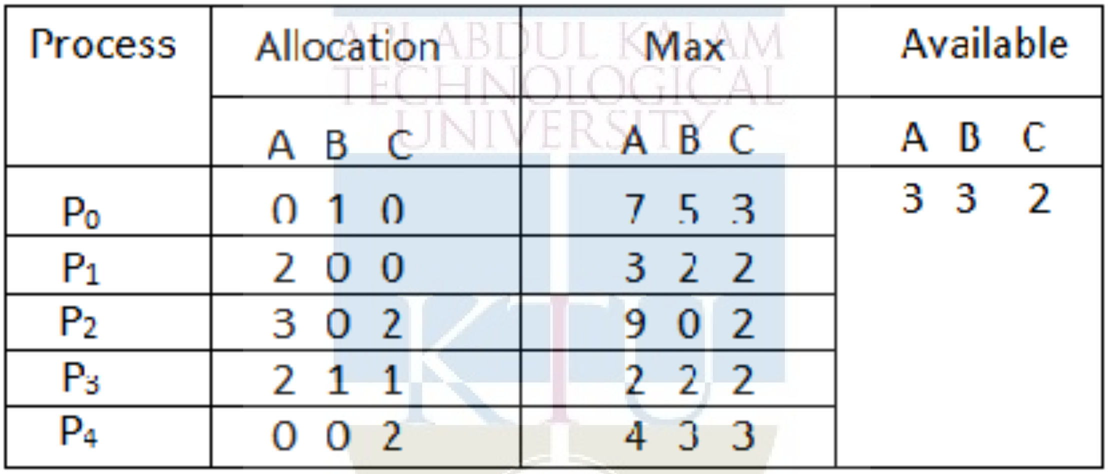
OR

14 a) Find the average waiting time and average turnaround time for the processes given in the table below using:- i) SRT scheduling algorithm ii) Priority scheduling algorithm **(9)**

Process	Arrival Time (ms)	CPU Burst Time (ms)	Priority
P1	0	5	3
P2	2	4	1
P3	3	1	2
P4	5	2	4

b) What is a Process Control Block? Explain the fields used in a Process Control Block. **(5)**

15. Consider a system with five processes P_0 through P_4 and three resources of type A, B, C. Resource type A has 10 instances, B has 5 instances and C has 7 instances. Suppose at time t_0 following snapshot of the system has been taken:



Process	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P_0	0	1	0	7	5	3	3	3	2
P_1	2	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	2	4	3	3			

i) What will be the content of the Need matrix? Is the system in a safe state? If Yes, then what is the safe sequence? **(8)**

iii)What will happen if process P_1 requests one additional instance of resource type A and two instances of resource type C? **(6)**

OR

16. a) State dining philosopher's problem and give a solution using semaphores. **(7)**

b) What do you mean by binary semaphore and counting semaphore? With C struct, explain implementation of wait () and signal() **(7)**

17. a) Consider the following page reference string 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. Find out the number of page faults if there are 4 page frames, using the following page replacement algorithms i) LRU ii) FIFO iii) Optimal **(9)**

b) Explain the steps involved in handling a page fault. **(5)**

OR

18. a) With a diagram, explain how paging is done with TLB. **(5)**

b) Memory partitions of sizes 100 kb, 500 kb, 200 kb, 300 kb, 600 kb are available, how would best ,worst and first fit algorithms place processes of size 212 kb, 417 kb, 112 kb, 426 kb in order. Rank the algorithms in terms of how efficiently they uses memory. **(9)**

19. a) Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. the drive currently services a request at cylinder 143, and the previous request was at cylinder 125. the queue of pending request in FIFO order is 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. Starting from the current position, what is the total distance (in cylinders) that the disk arm moves to satisfy all pending requests for each of the following algorithms

i) FCFS ii) SSFT iii) SCAN iv) LOOK v) C-SCAN **(10)**

b) What is the use of access matrix in protection mechanism? **(4)**

OR

20. a) Explain the different file allocation operations with advantages and disadvantages. **(8)**

b) Explain the following i) file types ii) file operation iii) file attributes **(6)**

Teaching Plan

	Module 1 - Introduction	**5 Hours**
1.1	Introduction to Operating System	1
1.2	Operating System operations, functions, service	1
1.3	System calls, Types	1
1.4	Operating System Structure: Simple, Layered, Microkernel, Modules	1
1.5	System Boot Process	1
	Module 2 – Processes and Process Scheduling	**9 Hours**
2.1	Processes, Process states	1
2.2	Process Control Block, Threads	1

2.3	Scheduling	1
2.4	Operations on processes: process creation and termination	1
2.5	Inter-process communication: Shared memory systems, Message Passing	1
2.6	Process Scheduling – Basic concepts, Scheduling Criteria	1
2.7	Scheduling algorithms - Basics	1
2.8	First come First Served, Shortest Job First	1
2.9	Priority scheduling, Round Robin Scheduling	1
	Module 3 - Process synchronization and Dead locks	**13 Hours**
3.1	Process synchronization, Race conditions	1
3.2	Critical Section problem, Peterson's solution	1
3.3	Synchronization hardware, Mutex Locks	1
3.4	Semaphores	1
3.5	Monitors	1
3.6	Synchronization problem examples (Lecture 1)	1
3.7	Synchronization problem examples (Lecture 2)	1
3.8	Deadlocks: Necessary conditions, Resource Allocation Graphs	1
3.9	Deadlock prevention	1
3.10	Deadlock avoidance	1
3.11	Banker's algorithm	1
3.12	Deadlock detection	1
3.13	Deadlock recovery	1
	Module 4 - Memory Management	**9 Hours**
4.1	Memory Management: Concept of Address spaces	1
4.2	Swapping	1
4.3	Contiguous memory allocation, fixed and variable partitions	1
4.4	Segmentation.	1
4.5	Paging (Lecture 1)	1
4.6	Paging (Lecture 2)	1
4.7	Virtual memory, Demand Paging	1

4.8	Page replacement algorithms (Lecture 1)	1
4.9	Page replacement algorithms (Lecture 2)	1
	Module 5 - File and Disk management	**9 Hours**
5.1	File concept, Attributes, Operations, types, structure	1
5.2	Access methods	1
5.3	Protection	1
5.4	File-System implementation	1
5.5	Directory implementation	1
5.6	Allocation methods	1
5.7	Magnetic disks, Solid-state disks, Disk structure	1
5.8	Disk scheduling	1
5.9	Disk formatting	1

CSL 202	DIGITAL LAB	CATEGORY	L	T	P	CREDIT
		PCC	0	0	3	2

Preamble: This course helps the learners to get familiarized with (i) Digital Logic Design through the implementation of Logic Circuits using ICs of basic logic gates & flip-flops and (ii) Hardware Description Language based Digital Design. This course helps the learners to design and implement hardware systems in areas such as games, music, digital filters, wireless communications and graphical displays.

Prerequisite: Topics covered under the course Logic System Design **(CST 203)**

Course Outcomes: After the completion of the course the student will be able to

CO 1	Design and implement combinational logic circuits using Logic Gates (Cognitive Knowledge Level: **Apply**)
CO 2	Design and implement sequential logic circuits using Integrated Circuits (Cognitive Knowledge Level: **Apply**)
CO 3	Simulate functioning of digital circuits using programs written in a Hardware Description Language (Cognitive Knowledge Level: **Apply**)
CO 4	Function effectively as an individual and in a team to accomplish a given task of designing and implementing digital circuits (Cognitive Knowledge Level: **Apply**)

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
CO 1	✓	✓	✓	✓				✓				✓
CO 2	✓	✓	✓	✓				✓				✓
CO 3	✓	✓	✓	✓	✓			✓				✓
CO 4	✓	✓	✓	✓				✓	✓			✓

Assessment Pattern

Bloom's Category	Continuous Assessment Test (Internal Exam) (Percentage)	End Semester Examination (Percentage)
Remember	20	20
Understand	20	20
Apply	60	60
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 15 marks
Continuous Evaluation in Lab	: 30 marks
Continuous Assessment Test	: 15 marks
Viva-voce	: 15 marks

Internal Examination Pattern: The marks will be distributed as Design/Algorithm 30 marks, Implementation/Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Design/Algorithm 30 marks, Implementation/Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Fair Lab Record:

All Students attending the Digital Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record, the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, and Aim of Experiment. The left hand page should contain components used, circuit design or a print out of the code used for the experiment and sample output obtained.

SYLLABUS

Conduct a minimum of **8** experiments from **Part A** and a minimum of **4** experiments from **Part B**. The starred experiments in Part A are mandatory. The lab work should be conducted in groups (maximum group size being 4). The performance of a student in the group should be assessed based on teamwork, integrity and cooperation.

Part A (Any 8 Experiments)

- A 2 hour session should be spent to make the students comfortable with the use of trainer kit/breadboard and ICs.
- The following experiments can be conducted on breadboard or trainer kits.
- Out of the 15 experiments listed below, a minimum of 8 experiments should be completed by a student, including the mandatory experiments (5).

1. Realization of functions using basic and universal gates (SOP and POS forms).
2. Design and realization of half adder, full adder, half subtractor and full subtractor using: a) basic gates (b) universal gates. *
3. Code converters: Design and implement BCD to Excess 3 and Binary to Gray code converters.
4. Design and implement 4 bit adder/subtractor circuit and BCD adder using IC7483.
5. Implementation of Flip Flops: SR, D, T, JK and Master Slave JK Flip Flops using basic gates.*
6. Asynchronous Counter: Design and implement 3 bit up/down counter.
7. Asynchronous Counter: Realization of Mod N counters (At least one up counter and one down counter to be implemented). *
8. Synchronous Counter: Realization of 4-bit up/down counter.
9. Synchronous Counter: Realization of Mod-N counters and sequence generators. (At least one mod N counter and one sequence generator to be implemented) *
10. Realization of Shift Register (Serial input left/right shift register), Ring counter and Johnson Counter using flipflops. *
11. Realization of counters using IC's (7490, 7492, 7493).
12. Design and implement BCD to Seven Segment Decoder.
13. Realization of Multiplexers and De-multiplexers using gates.
14. Realization of combinational circuits using MUX & DEMUX ICs (74150, 74154).
15. To design and set up a 2-bit magnitude comparator using basic gates.

PART B (Any 4 Experiments)

- The following experiments aim at training the students in digital circuit design with ***Verilog***. The experiments will lay a foundation for digital design with Hardware Description Languages.
- A 3 hour introductory session shall be spent to make the students aware of the fundamentals of development using Verilog
- Out of the 8 experiments listed below, a minimum of 4 experiments should be completed by a student

Experiment 1. Realization of Logic Gates and Familiarization of Verilog

(a) Familiarization of the basic syntax of Verilog

(b) Development of Verilog modules for basic gates and to verify truth tables.

(c) Design and simulate the HDL code to realize three and four variable Boolean functions

Experiment 2: Half adder and full adder

(a) Development of Verilog modules for half adder in 3 modeling styles (dataflow/ structural/behavioural).

(b) Development of Verilog modules for full adder in structural modeling using half adder.

Experiment 3: Design of code converters

Design and simulate the HDL code for

(a) 4- bit binary to gray code converter

(b) 4- bit gray to binary code converter

Experiment 4: Mux and Demux in Verilog

(a) Development of Verilog modules for a 4x1 MUX.

(b) Development of Verilog modules for a 1x4 DEMUX.

Experiment 5: Adder/Subtractor

(a) Write the Verilog modules for a 4-bit adder/subtractor

(b) Development of Verilog modules for a BCD adder

Experiment 6: Magnitude Comparator

Development of Verilog modules for a 4 bit magnitude comparator

Experiment 7: Flipflops and shiftregisters

(a) Development of Verilog modules for SR, JK, T and D flip flops.

(b) Development of Verilog modules for a Johnson/Ring counter

Experiment 8: Counters

(a) Development of Verilog modules for an asynchronous decade counter.

(b) Development of Verilog modules for a 3 bit synchronous up-down counter.

Practice Questions

PART A

1. Design a two bit parallel adder using gates and implement it using ICs of basic gates
2. A combinatorial circuit has 4 inputs and one output. The output is equal to 1 when (a) all inputs are 1, (b) none of the inputs are 1, (c) an odd number of inputs are equal to 1. Obtain the truth table and output function for this circuit and implement the same.
3. Design and implement a parallel subtractor.
4. Design and implement a digital circuit that converts Gray code to Binary.
5. Design a combinational logic circuit that will output the 1's compliment of a 4-bit input number.
6. Implement and test the logic function $f(A, B, C) = \sum m(0,1,3,6)$ using an 8:1 MUX IC
7. Design a circuit that will work as a ring counter or a Johnson counter based on a mode bit, M.
8. Design a 4-bit synchronous down counter.
9. Design a Counter to generate the binary sequence 0,1,3,7,6,4
10. Design an asynchronous mod 10 down counter
11. Design and implement a synchronous counter using JK flip flop ICs to generate the sequence: 0 - 1 -3 - 5 - 7 - 0.

PART B

1. Develop Verilog modules for a full subtractor in structural modeling using half subtractors.
2. Design a 4 bit parallel adder using Verilog.
3. Develop Verilog modules for a 4 bit synchronous down counter.
4. Write Verilog code for implementing a 8:1 multiplexer.
5. Develop Verilog modules for a circuit that converts Excess 3 code to binary.
6. Write the Verilog code for a JK Flip flop, and its test-bench. Use all possible combinations of inputs to test its working
7. Write the hardware description in Verilog of a 8-bit register with shift left and shift right modes of operations and test its functioning.
8. Write the hardware description in Verilog of a mod-N (N > 9) counter and test it.

CST 206	OPERATING SYSTEMS LAB	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	0	0	3	2	2019

Preamble: The course aims to offer students a hands-on experience on Operating System concepts using a constructivist approach and problem-oriented learning. Operating systems are the fundamental part of every computing device to run any type of software.

Prerequisite: Topics covered in the courses are **Data Structures (CST 201)** and **Programming in C (EST 102)**

Course Outcomes:

At the end of the course, the student should be able to

CO1	Illustrate the use of systems calls in Operating Systems. **(Cognitive knowledge: Understand)**
CO2	Implement Process Creation and Inter Process Communication in Operating Systems. **(Cognitive knowledge: Apply)**
CO3	Implement Fist Come First Served, Shortest Job First, Round Robin and Priority-based CPU Scheduling Algorithms. **(Cognitive knowledge: Apply)**
CO4	Illustrate the performance of First In First Out, Least Recently Used and Least Frequently Used Page Replacement Algorithms. **(Cognitive knowledge: Apply)**
CO5	Implement modules for Deadlock Detection and Deadlock Avoidance in Operating Systems. **(Cognitive knowledge: Apply)**
CO6	Implement modules for Storage Management and Disk Scheduling in Operating Systems. **(Cognitive knowledge: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓					✓		✓		✓
CO2	✓	✓	✓					✓		✓		✓
CO3	✓	✓	✓	✓				✓		✓		✓
CO4	✓	✓	✓	✓				✓		✓		✓
CO5	✓	✓	✓	✓				✓		✓		✓
CO6	✓	✓	✓	✓				✓		✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern:

Bloom's Category	**Continuous Assessment Test (Internal Exam) Marks in percentage**	**End Semester Examination Marks in percentage**
Remember	20	20
Understand	20	20
Apply	60	60
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	3 hours

Continuous Internal Evaluation Pattern:

Attendance	:	15 marks
Continuous Evaluation in Lab	:	30 marks
Continuous Assessment Test	:	15 marks
Viva Voce	:	15 marks

Internal Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The percentage of marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 75 marks.

Operating System to Use in Lab : Linux

Compiler/Software to Use in Lab : gcc

Progamming Language to Use in Lab : Ansi C

Fair Lab Record:

All Students attending the Operating System Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record, the right hand page should contain Experiment Heading, Experiment Number, Date of experiment, Aim of the Experiment and the operations performed on them, Details of experiment including algorithm and result of Experiment. The left hand page should contain a print out of the code used for experiment and sample output obtained for a set of input.

SYLLABUS

OPERATING SYSTEMS LAB

* mandatory

1. Basic Linux commands
2. Shell programming

    -Command syntax

    -Write simple functions with basic tests, loops, patterns

3. System calls of Linux operating system:*

    fork, exec, getpid, exit, wait, close, stat, opendir, readdir

4. Write programs using the I/O system calls of Linux operating system (open, read, write)
5. Implement programs for Inter Process Communication using Shared Memory *
6. Implement Semaphores*
7. Implementation of CPU scheduling algorithms. a) Round Robin b) SJF c) FCFS d) Priority *
8. Implementation of the Memory Allocation Methods for fixed partition*

    a) First Fit b) Worst Fit c) Best Fit

9. Implement l page replacement algorithms a) FIFO b) LRU c) LFU*
10. Implement the banker's algorithm for deadlock avoidance. *
11. Implementation of Deadlock detection algorithm
12. Simulate file allocation strategies.

    b) Sequential b) Indexed c) Linked

13. Simulate disk scheduling algorithms. *

    c) FCFS b)SCAN c) C-SCAN

OPERATING SYSTEMS LAB - PRACTICE QUESTIONS

1. Write a program to create a process in linux.
2. Write programs using the following system calls of Linux operating system:

    fork, exec, getpid, exit, wait, close, stat, opendir, readdir

3. Write programs using the I/O system calls of Linux operating system (open, read, write)

4. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time
5. Write a C program to simulate following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.

   a)FCFS b) SJF c) Round Robin (pre-emptive) d) Priority
6. Write a C program to simulate following contiguous memory allocation techniques

   a) Worst-fit b) Best-fit c) First-fit
7. Write a C program to simulate paging technique of memory management.
8. Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance.
9. Write a C program to simulate disk scheduling algorithms a) FCFS b) SCAN c) C-SCAN
10. Write a C program to simulate page replacement algorithms a) FIFO b) LRU c) LFU
11. Write a C program to simulate producer-consumer problem using semaphores.
12. Write a program for file manipulation for display a file and directory in memory.
13. Write a program to simulate algorithm for deadlock prevention.
14. Write a C program to simulate following file allocation strategies.

    a)Sequential b) Indexed c) Linked

SEMESTER -4

MINOR

CST 282	**Programming Methodologies**	**Category**	**L**	**T**	**P**	**CREDIT**	**YEAR OF INTRODUCTION**
		MINOR	**3**	**1**	**0**	**4**	**2019**

Preamble: This is the second course for awarding B.Tech Minor in Computer Science and Engineering with specialization in ***Software Engineering***. The course provides the learners a clear understanding of the main constructs of contemporary programming languages and the various systems of ideas that have been used to guide the design of programming languages. This course covers the concepts of Names, Bindings & Scope, Statement-Level Control Structures, Sub Programs, support for Object Oriented Programming, Exception Handling, Event Handling, Concurrency Control, Functional Programming and Logic Programming. This course helps the learners to equip with the knowledge necessary for the critical evaluation of existing and upcoming programming languages. It also enables the learner to choose the most appropriate language for a given programming task, apply that language's approach to structure or organize the code and classify programming languages based on their features.

Prerequisite:

1. Topics covered under the course Programming in C (EST 102)
2. Object Oriented Programming (CST 251)

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain the criteria for evaluating programming languages and compare Imperative, Functional and Logic programming languages (Cognitive Knowledge Level: **Understand**)
CO2	Explain the characteristics of data types and variables (Cognitive Knowledge Level: **Understand**)
CO3	Illustrate how control flow structures and subprograms help in developing the structure of a program to solve a computational problem (Cognitive Knowledge Level: **Apply**)
CO4	Explain the characteristics of Object Oriented Programming Languages (Cognitive Knowledge Level: **Understand**)
CO5	Compare concurrency constructs in different programming languages (Cognitive Knowledge Level: **Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓							✓		✓
CO2	✓	✓								✓		✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓								✓		✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination (Marks%)
	Test 1 (Marks%)	Test 2 (Marks%)	
Remember	30	30	30
Understand	50	50	50
Apply	20	20	20
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

SYLLABUS

Module 1

Introduction – Role of Programming Languages, Programming Domains, Language Evaluation Criteria, Influence on Language Design, Language Design Trade-offs, Implementation Methods. **Names, Bindings & Scope** – Names, Variables, Concept of Binding, Scope and Lifetime, Referencing Environments.

Module 2

Data Types – Primitive Data Types, Character String Types, User-Defined Ordinal Types, Array Types, Record Types, List Types, Pointer & Reference Types, Type Checking, Strong Typing, Type Equivalence. Expressions – Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation. Assignment - Assignment Statements, Mixed-mode Assignment.

Module 3

Statement-Level Control Structures – Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands. Subprograms – Design Issues of Subprograms, Local Referencing Environments, Parameter Passing Methods, Subprograms as Parameters, Overloaded Subprograms, Closures, Co-routines.

Module 4

Support for Object Oriented Programming – Inheritance, Dynamic Binding, Design Issues for Object Oriented Languages, Support for Object Oriented Programming in C++, Implementation of Object-Oriented Constructs. Exception Handling – Basic Concepts, Design Issues. Event Handling **-** Introduction to Event Handling.

Module 5

Concurrency – Subprogram Level Concurrency, Semaphores, Monitors, Message Passing. **Functional Programming Languages** – Introduction to LISP and Scheme, Comparison of Functional and Imperative Languages. Logic Programming Languages – Basic Elements of Prolog, Applications of Logic Programming.

Text Books

1. Robert W. Sebesta, Concepts of Programming Languages, 10th Edition, Pearson.

2. Scott M. L., Programming Language Pragmatics, 3rd Edn., Morgan Kaufmann Publishers.

Reference Books:

1. Kenneth C. Louden, Programming Languages: Principles and Practice, 2nd Edn., Cengage Learning**.**
2. Tucker A. B. and R. E. Noonan, Programming Languages: Principles and Paradigms, 2nd Edn. –TMH.
3. Ravi Sethi, Programming Languages: Concepts & Constructs, 2nd Edn., Pearson Education.
4. David A. Watt, Programming Language Design Concepts, Wiley Dreamtech

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Compare any three programming languages based on the language evaluation criteria. Prepare a list of characteristics that affect the language evaluation criteria. Identify the advantages and disadvantages of imperative, functional and logic programming languages.

Course Outcome 2 (CO2): Two most important design issues that are specific to character string types are (1) whether a string is simply a special kind of character array or a primitive type (2) whether strings have static or dynamic length. Justify your answer.

Course Outcome 3 (CO3):

1. Describe three situations where a combined counting and logical looping statement is needed.
2. Describe the ways that aliases can occur with pass-by-reference parameters.
3. Identify the two fundamental design considerations for parameter-passing methods.

Course Outcome 4 (CO4):

1. Describe the role of a virtual method table in implementing dynamic method binding.
2. Identify one disadvantage of inheritance.

Course Outcome 5 (CO5): Evaluate the use of semaphores and monitors for providing competition synchronization and cooperation synchronization.

Model Question paper

QP CODE: **PAGES:3**

Reg No:______________ **Name :______________**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FOURTH SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH & YEAR Course Code: CST 282

Course Name: Programming Methodologies

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Differentiate between readability and writability.
2. Define binding and binding time
3. What are the advantages of user-defined enumeration types?
4. Define narrowing and widening conversions.
5. Why **for** statement in C language is more flexible than that of older languages?
6. What are the advantages and disadvantages of dynamic local variables in subprograms?
7. Explain the concept of dynamic method binding with an example.
8. Is it mandatory to use constructors in object oriented languages? Justify your answer.
9. What are the applications of logical programming languages?
10. Explain the working of *let* and *let-rec* constructs in Scheme.

Part B

Answer any one Question from each module. Each question carries 14 Marks

11.

(a) Explain different criteria used for evaluating languages.

(7 marks)

(b) Explain the major methods of implementing programming languages.

(7 marks)

OR

12.

(a) Explain the meanings, purposes, advantages and disadvantages of four categories of scalar variables according to their storage bindings.

(7 marks)

(b) What is referencing environment of a statement? Show the referencing environment at the indicated program points (1), (2), (3) & (4) for the following program segment. Assume that the programming language used is statically scoped.

```
program  example;
    var  a, b : integer;
    procedure   sub1;
            var   x, y: integer;
            begin { sub1 }
            ………..                                   (1)
            end { sub1 }
    procedure   sub2;
            var   x : integer;
            ……..
             procedure  sub3;
                    var   x: integer;
                    begin { sub3 }
                    ………..                           (2)
                    end { sub3 }
            begin { sub2 }
            ………..                                   (3)
            end { sub2}
begin {example}
            ………..                                   (4)
end {example }
```

(7 Marks)

13.

(a) Explain any two problems associated with the pointer data types and also indicate how dangling pointer problem can be solved.

(7 marks)

(b) Describe the lazy and eager approaches for reclaiming garbage.

(7 marks)

OR

14.

(a) What is meant by *side effect* and illustrate the advantages of referential transparency?

(8 marks)

(b) Explain the terms: compound assignment operator, coercion and short circuit evaluation.

(6 marks)

15.

(a) Explain different categories of iteration control statements.

(8 marks)

(b) Explain techniques used for identifying correct referencing environment for a subprogram that was sent as a parameter.

(6 marks)

OR

16.

(a) Describe the implementation models of Parameter passing.

(10 Marks)

(b) Differentiate coroutines from conventional subprograms.

(4 marks)

17.

(a) What is an exception handler? Explain how exceptions are handled in object oriented language?

(7 Marks)

(b) What are the design issues in object oriented languages? (7 Marks)

OR

18. Explain the following object oriented features:

(i) Encapsulation

(ii) Inheritance

(iii) Constructors and Destructors

(iv) Operator Overloading

(v) Polymorphism

(14 Marks)

19.

(a) Compare functional and imperative programming languages.

(7 Marks)

(b) Explain the role of monitors in concurrency.

(7 Marks)

OR

20.

(a) Explain the searching strategies used in Prolog. Why backward chaining is preferred over forward chaining in Prolog?

(10 Marks)

(b) How does a binary semaphore differ from an ordinary semaphore?

(4 Marks)

Teaching Plan

Module 1 (Introduction)		**9 Hours**
1.1	Introduction : Reasons for studying Concepts of programming languages, Programming Domains	1 Hour
1.2	Language Evaluation Criteria (Lecture 1)	1 Hour
1.3	Language Evaluation Criteria (Lecture 2)	1 Hour
1.4	Influence on Language Design, Language Design Trade-offs	1 Hour
1.5	Implementation Methods	1 Hour
1.6	Names, Variables	1 Hour
1.7	Concept of Binding	1 Hour
1.8	Scope and Lifetime	1 Hour
1.9	Referencing Environments	1 Hour
Module 2 (Data Types, Expressions and Assignment Statements)		**8 Hours**
2.1	Primitive Data Types, Character String Types	1 Hour
2.2	User-Defined Ordinal Types, Array Types	1 Hour
2.3	Record Types, List Types, Pointer and Reference Types	1 Hour
2.4	Implementation of pointer and reference types, Type Checking, Strong Typing, Type Equivalence	1 Hour
2.5	Expressions and Assignment Statements, Arithmetic Expressions	1 Hour
2.6	Overloaded Operators, Type Conversions	1 Hour
2.7	Relational and Boolean Expressions, Short-Circuit Evaluation	1 Hour
2.8	Assignment Statements, Mixed-mode Assignment	1 Hour
Module 3 (Statement Level Control Structures, Subprograms)		**8 Hours**
3.1	Selection Statements, Iterative Statements	1 Hour
3.2	Unconditional Branching	1 Hour

3.3	Guarded Commands	1 Hour
3.4	Subprograms: Design Issues of Subprograms	1 Hour
3.5	Local Referencing Environments	1 Hour
3.6	Parameter Passing Methods	1 Hour
3.7	Subprograms as Parameters, Overloaded Subprograms	1 Hour
3.8	Closures, Co-routines	1 Hour
Module 4 (Support for Object Oriented Programming, Exception Handling, Event handling)		**10 Hours**
4.1	Inheritance	1 Hour
4.2	Dynamic Binding	1 Hour
4.3	Design Issues for Object Oriented Languages	1 Hour
4.4	Support for Object Oriented Programming in C++	1 Hour
4.5	Implementation of Object-Oriented Constructs (Lecture 1)	1 Hour
4.6	Implementation of Object-Oriented Constructs (Lecture 2)	1 Hour
4.7	Implementation of Object-Oriented Constructs (Lecture 3)	1 Hour
4.8	Basic Concepts	1 Hour
4.9	Exception Handling - Design Issues	1 Hour
4.10	Introduction to Event Handling	1 Hour
Module 5 (Concurrency, Functional Programming Languages, Logic Programming languages)		**10 Hours**
5.1	Subprogram Level Concurrency	1 Hour
5.2	Semaphores	1 Hour
5.3	Monitors	1 Hour
5.4	Message Passing	1 Hour
5.5	Introduction to LISP and Scheme (Lecture 1)	1 Hour
5.6	Introduction to LISP and Scheme (Lecture 2)	1 Hour
5.7	Comparison of Functional and Imperative Languages	1 Hour
5.8	Basic Elements of Prolog (Lecture 1)	1 Hour

5.9	Basic Elements of Prolog (Lecture 2)	1 Hour
5.10	Applications of Logic Programming	1 Hour

CODE CST 284	Mathematics for Machine Learning	CATEGORY	L	T	P	CREDIT
		MINOR	3	1	0	4

Preamble: This is a foundational course for awarding B. Tech. Minor in Computer Science and Engineering with specialization in ***Machine Learning***. The purpose of this course is to introduce mathematical foundations of basic Machine Learning concepts among learners, on which Machine Learning systems are built. This course covers Linear Algebra, Vector Calculus, Probability and Distributions, Optimization and Machine Learning problems. Concepts in this course help the learners to understand the mathematical principles in Machine Learning and aid in the creation of new Machine Learning solutions, understand & debug existing ones, and learn about the inherent assumptions & limitations of the current methodologies.

Prerequisite:

1. A sound background in higher secondary school Mathematics.
2. Python for Machine Learning (CST 253)

Course Outcomes: After the completion of the course the student will be able to

CO 1	Make use of the concepts, rules and results about linear equations, matrix algebra, vector spaces, eigenvalues & eigenvectors and orthogonality & diagonalization to solve computational problems (Cognitive Knowledge Level: **Apply**)
CO 2	Perform calculus operations on functions of several variables and matrices, including partial derivatives and gradients (Cognitive Knowledge Level: **Apply**)
CO 3	Utilize the concepts, rules and results about probability, random variables, additive & multiplicative rules, conditional probability, probability distributions and Bayes' theorem to find solutions of computational problems (Cognitive Knowledge Level: **Apply**)
CO 4	Train Machine Learning Models using unconstrained and constrained optimization methods (Cognitive Knowledge Level: **Apply**)
CO 5	Illustrate how the mathematical objects - linear algebra, probability, and calculus can be used to design machine learning algorithms (Cognitive Knowledge Level: **Understand**)

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	√	√	√	√								√
CO 2	√	√	√									√
CO 3	√	√	√	√								√
CO 4	√	√	√	√		√						√
CO 5	√	√	√	√	√	√				√		√

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	20%	20%	20%
Understand	40%	40%	40%
Apply	40%	40%	40%
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer anyone. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module 1

LINEAR ALGEBRA : Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence, Basis and Rank, Linear Mappings, Norms, - Inner Products - Lengths and Distances - Angles and Orthogonality - Orthonormal Basis - Orthogonal Complement - Orthogonal Projections. Matrix Decompositions - Determinant and Trace, Eigenvalues and Eigenvectors, Cholesky Decomposition, Eigen decomposition and Diagonalization, Singular Value Decomposition, Matrix Approximation.

Module 2

VECTOR CALCULUS : Differentiation of Univariate Functions - Partial Differentiation and Gradients, Gradients of Vector Valued Functions, Gradients of Matrices, Useful Identities for Computing Gradients. Back propagation and Automatic Differentiation - Higher Order Derivatives- Linearization and Multivariate Taylor Series.

Module 3

Probability and Distributions : Construction of a Probability Space - Discrete and Continuous Probabilities, Sum Rule, Product Rule, and Bayes' Theorem. Summary Statistics and Independence – Important Probability distributions - Conjugacy and the Exponential Family - Change of Variables/Inverse Transform.

Module 4

Optimization : Optimization Using Gradient Descent - Gradient Descent With Momentum, Stochastic Gradient Descent. Constrained Optimization and Lagrange Multipliers - Convex Optimization - Linear Programming **-** Quadratic Programming.

Module 5

CENTRAL MACHINE LEARNING PROBLEMS : Data and Learning Model- Empirical Risk Minimization - Parameter Estimation - Directed Graphical Models.

Linear Regression - Bayesian Linear Regression - Maximum Likelihood as Orthogonal Projection.

Dimensionality Reduction with Principal Component Analysis - Maximum Variance Perspective, Projection Perspective. Eigenvector Computation and Low Rank Approximations.

Density Estimation with Gaussian Mixture Models - Gaussian Mixture Model, Parameter Learning via Maximum Likelihood, EM Algorithm.

Classification with Support Vector Machines - Separating Hyperplanes, Primal Support Vector Machine, Dual Support Vector Machine, Kernels.

Text book:

1.Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong published by Cambridge University Press (freely available at https:// mml - book.github.io)

Reference books:

1. Linear Algebra and Its Applications, 4th Edition by Gilbert Strang
2. Linear Algebra Done Right by Axler, Sheldon, 2015 published by Springer
3. Introduction to Applied Linear Algebra by Stephen Boyd and Lieven Vandenberghe, 2018 published by Cambridge University Press
4. Convex Optimization by Stephen Boyd and Lieven Vandenberghe, 2004 published by Cambridge University Press
5. Pattern Recognition and Machine Learning by Christopher M Bishop, 2006, published by Springer
6. Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond by Bernhard Scholkopf and Smola, Alexander J Smola, 2002, bublished by MIT Press
7. Information Theory, Inference, and Learning Algorithms by David J. C MacKay, 2003 published by Cambridge University Press
8. Machine Learning: A Probabilistic Perspective by Kevin P Murphy, 2012 published by MIT Press.
9. The Nature of Statistical Learning Theory by Vladimir N Vapnik, 2000, published by Springer

Sample Course Level Assessment Questions.

Course Outcome 1 (CO1):

1. Find the set ***S*** of all solutions in ***x*** of the following inhomogeneous linear systems ***Ax* = *b***, where ***A*** and ***b*** are defined as follows:

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

2. Determine the inverses of the following matrix if possible

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

3. Are the following sets of vectors linearly independent?

$$x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

4. A set of ***n*** linearly independent vectors in $\boldsymbol{R^n}$ forms a basis. Does the set of vectors **(2, 4,−3)** , **(0, 1, 1)** , **(0, 1,−1**) form a basis for $\boldsymbol{R^3}$? Explain your reasons.

5. Consider the transformation ***T (x, y) = (x + y, x + 2y, 2x + 3y)***. Obtain ***ker T*** and use this to calculate the nullity. Also find the transformation matrix for ***T***.

6. Find the characteristic equation, eigenvalues, and eigenspaces corresponding to each eigenvalue of the following matrix

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

7. Diagonalize the following matrix, if possible

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}$$

8. Find the singular value decomposition (SVD) of the following matrix

$$\begin{bmatrix} 0 & 1 & 1 \\ \sqrt{2} & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Course Outcome 2 (CO2):

1. For a scalar function $f(x, y, z) = x^2 + 3y^2 + 2z^2$, find the gradient and its magnitude at the point **(1, 2, -1)**.
2. Find the maximum and minimum values of the function $f(x, y) = 4x + 4y - x^2 - y^2$ subject to the condition $x^2 + y^2 <= 2$.
3. Suppose you were trying to minimize $f(x, y) = x^2 + 2y + 2y^2$. Along what vector should you travel from (5, 12)?
4. Find the second order Taylor series expansion for $f(x, y) = (x + y)^2$ about **(0 , 0)**.
5. Find the critical points of $f(x, y) = x^2 - 3xy + 5x - 2y + 6y^2 + 8$.
6. Compute the gradient of the Rectified Linear Unit (ReLU) function ***ReLU(z) = max(0 , z)***.
7. Let $L = ||Ax - b||^2_2$, where A is a matrix and x and b are vectors. Derive dL in terms of dx.

Course Outcome 3 (CO3):

1. Let J and T be independent events, where $P(J)=0.4$ and $P(T)=0.7$.

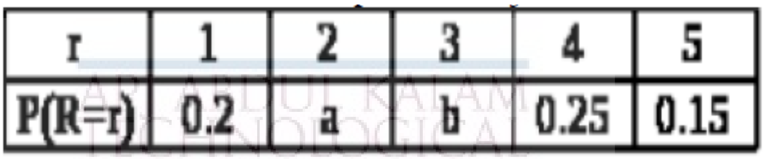
   i. Find $P(J\cap T)$

   ii. Find $P(J\cup T)$

   iii. Find $P(J\cap T')$

2. Let A and B be events such that $P(A)=0.45$, $P(B)=0.35$ and $P(A\cup B)=0.5$. Find $P(A|B)$.

3. A random variable **R** has the probability distribution as shown in the following table:



r	1	2	3	4	5
P(R=r)	0.2	a	b	0.25	0.15

   i. Given that $E(R)=2.85$, find a and b.

   ii. Find $P(R>2)$.

4. A biased coin (with probability of obtaining a head equal to $p > 0$) is tossed repeatedly and independently until the first head is observed. Compute the probability that the first head appears at an even numbered toss.

5. Two players A and B are competing at a trivia quiz game involving a series of questions. On any individual question, the probabilities that A and B give the correct answer are p and q respectively, for all questions, with outcomes for different questions being independent. The game finishes when a player wins by answering a question correctly. Compute the probability that A wins if

   i. A answers the first question,

   ii. B answers the first question.

6. A coin for which $P(heads) = p$ is tossed until two successive tails are obtained. Find the probability that the experiment is completed on the n^{th} toss.

7. You roll a fair dice twice. Let the random variable X be the product of the outcomes of the two rolls. What is the probability mass function of X? What are the expected value and the standard deviation of X?

8. While watching a game of Cricket, you observe someone who is clearly supporting Mumbai Indians. What is the probability that they were actually born within 25KM of Mumbai? Assume that:

   - the probability that a randomly selected person is born within 25KM of Mumbai is 1/20;
   - the chance that a person born within 25KMs of Mumbai actually supports MI is 7/10 ;
   - the probability that a person not born within 25KM of Mumbai supports MI with probability 1/10.

9. What is an exponential family? Why are exponential families useful?
10. Let $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$ be independent random variables each having the standard normal distribution. Define the random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ by $\boldsymbol{X = Z_1 + 3Z_2}$ and $Y = \boldsymbol{Z_1 + Z_2}$. Argue that the joint distribution of $\boldsymbol{(X, Y)}$ is a bivariate normal distribution. What are the parameters of this distribution?
11. Given a continuous random variable $\boldsymbol{x}$, with cumulative distribution function $\boldsymbol{F_x(x)}$, show that the random variable $\boldsymbol{y = F_x(x)}$ is uniformly distributed.
12. Explain Normal distribution, Binomial distribution and Poisson distribution in the exponential family form.

Course Outcome 4(CO4):

1. Find the extrema of $\boldsymbol{f(x, y) = x}$ subject to $\boldsymbol{g(x, y) = x^2 + 2y^2 = 3}$.
2. Maximize the function $\boldsymbol{f(x, y, z) = xy + yz + xz}$ on the unit sphere $\boldsymbol{g(x, y, z) = x^2 + y^2 + z^2 = 1}$.
3. Provide necessary and suffcient conditions under which a quadratic optimization problem be written as a linear least squares problem.
4. Consider the univariate function $\boldsymbol{f(x) = x^3 + 6x^2 - 3x - 5}$. Find its stationary points and indicate whether they are maximum, minimum, or saddle points.
5. Consider the update equation for stochastic gradient descent. Write down the update when we use a mini-batch size of one.

6. Consider the function

$$f(x) = (x_1 - x_2)^2 + \frac{1}{1 + x_1^2 + x_2^2}.$$

   i. Is ***f(x)*** a convex function? Justify your answer.

   ii. Is (1 , -1) a local/global minimum? Justify your answer.

7. Is the function ***f(x, y) = 2x² + y² + 6xy - x + 3y - 7*** convex, concave, or neither? Justify your answer.

8. Consider the following convex optimization problem

$$\text{minimize } \frac{x^2}{2} + x + 4y^2 - 2y$$

   Subject to the constraint ***x + y >= 4, x, y >= 1***.

   Derive an explicit form of the Lagrangian dual problem.

9. Solve the following LP problem with the simplex method.

$$max \;\; 5x_1 + 6x_2 + 9x_3 + 8x_4$$

   subject to the constraints

$$\begin{array}{ccccccccc} x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \leq & 5 \\ x_1 & + & x_2 & + & 2x_3 & + & 3x_4 & \leq & 3 \end{array}$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Course Outcome 5 (CO5):

1. What is a loss function? Give examples.
2. What are training/validation/test sets? What is cross-validation? Name one or two examples of cross-validation methods.
3. Explain generalization, overfitting, model selection, kernel trick, Bayesian learning

4. Distinguish between Maximum Likelihood Estimation (MLE) and Maximum A Posteriori Estimation (MAP)?

5. What is the link between structural risk minimization and regularization?

6. What is a kernel? What is a dot product? Give examples of kernels that are valid dot products.

7. What is ridge regression? How can one train a ridge regression linear model?

8. What is Principal Component Analysis (PCA)? Which eigen value indicates the direction of largest variance? In what sense is the representation obtained from a projection onto the eigen directions corresponding the the largest eigen values optimal for data reconstruction?

9. Suppose that you have a linear support vector machine (SVM) binary classifier. Consider a point that is currently classified correctly, and is far away from the decision boundary. If you remove the point from the training set, and re-train the classifier, will the decision boundary change or stay the same? Explain your answer in one sentence.

10. Suppose you have ***n*** independent and identically distributed (i.i.d) sample data points $\boldsymbol{x_1, \ldots , x_n}$. These data points come from a distribution where the probability of a given datapoint ***x*** is

$$P(x) = \frac{1}{\theta}e^{-\frac{1}{\theta}x}.$$

Prove that the MLE estimate of parameter is the sample mean.

11. Suppose the data set $\boldsymbol{y_1,\ldots,y_n}$ is a drawn from a random sample consisting of i.i.d. discrete uniform distributions with range 1 to *N*. Find the maximum likelihood estimate of ***N***.

12. Ram has two coins: one fair coin and one biased coin which lands heads with probability 3/4. He picks one coin at random (50-50) and flips it repeatedly until he gets a tails. Given that he observes 3 heads before the first tails, find the posterior probability that he picked each coin.

    i. What are the prior and posterior odds for the fair coin?

    ii. What are the prior and posterior predictive probabilities of heads on the next flip? Here prior predictive means prior to considering the data of the first four flips.

Model Question paper

QP Code : **Total Pages: 4**

Reg No.:_______________ Name:___________________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

IV SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH and YEAR

Course Code: CST 284

Course Name: MATHEMATICS FOR MACHINE LEARNING

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks. Marks

1 Show that with the usual operation of scalar multiplication but with addition on reals given by $x \# \ y = 2(x + y)$ is not a vector space.

2 Find the eigenvalues of the following matrix in terms of k. Can you find an eigenvector corresponding to each of the eigenvalues?

$$\begin{bmatrix} 1 & k \\ 2 & 1 \end{bmatrix}$$

3 Let $f(x, y, z) = xye^r$, where $r = x^2+z^2-5$. Calculate the gradient of f at the point **(1, 3, -2)**.

4 Compute the Taylor polynomials $T_n, n = 0 , ... , 5$ of $f(x) = sin(x) + cos(x)$ at $x_0 = 0$.

5 Let X be a continuous random variable with probability density function on $0 <= \ x <= 1$ defined by $f(x) = 3x^2$. Find the pdf of $Y = X^2$.

6 Show that if two events A and B are independent, then A and B' are independent.

7 Explain the principle of the gradient descent algorithm.

8 Briey explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.

9 What is the empirical risk? What is “empirical risk minimization”?

10 Explain the concept of a Kernel function in Support Vector Machines. Why are kernels so useful? What properties a kernel should posses to be used in an SVM?

PART B

Answer any one Question from each module. Each question carries 14 Marks

11 a) i. Find all solutions to the system of linear equations (6)

$$\begin{aligned} -4x + 5z &= -2 \\ -3x - 3y + 5z &= 3 \\ -x + 2y + 2z &= -1 \end{aligned}$$

ii. Prove that all vectors orthogonal to **$[2, -3, 1]^T$** forms a subspace ***W*** of ***R^3***. What is ***dim (W)*** and why?

b) Use the Gramm-Schmidt process to find an orthogonal basis for the column space of the following matrix (8)

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

OR

12 a) i. Let $\boldsymbol{L}$ be the line through the origin in $\boldsymbol{R^2}$ that is parallel to the vector **$[3, 4]^T$**. Find the standard matrix of the orthogonal projection onto L. Also find the point on $\boldsymbol{L}$ which is closest to the point **(7 , 1)** and find the point on $\boldsymbol{L}$ which is closest to the point ***(-3 , 5)***. (6)

ii. Find the rank-1 approximation of

$$\begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

b) i. Find an orthonormal basis of $\boldsymbol{R^3}$ consisting of eigenvectors for the following matrix (8)

$$\begin{bmatrix} 1 & 0 & -2 \\ 0 & 5 & 0 \\ -2 & 0 & 4 \end{bmatrix}$$

ii. Find a 3 × 3 orthogonal matrix $\boldsymbol{S}$ and a 3 × 3 diagonal matrix $\boldsymbol{D}$ such that $\boldsymbol{A = SDS^T}$.

13 a) A skier is on a mountain with equation $z = 100 - 0.4x^2 - 0.3y^2$, where z denotes height. (8)

i. The skier is located at the point with xy-coordinates **(1 , 1)**, and wants to ski downhill along the steepest possible path. In which direction (indicated by a vector **(a , b**) in the xy-plane) should the skier begin skiing.

ii. The skier begins skiing in the direction given by the xy-vector (**a , b)** you found in part (i), so the skier heads in a direction in space given by the vector **(a , b , c**). Find the value of **c**.

b) Find the linear approximation to the function ***f(x,y) = 2 - sin(-x - 3y)*** at the point **(0 , π)**, and then use your answer to estimate ***f(0.001 , π)***. (6)

OR

14 a) Let ***g*** be the function given by (8)

$$g(x,y) = \begin{cases} \dfrac{x^2y}{x^2+y^2} & \text{if } (x,y) \neq (0,0); \\ 0 & \text{if } (x,y) = (0,0). \end{cases}$$

i. Calculate the partial derivatives of ***g*** at **(0 , 0)**.

ii. Show that ***g*** is not differentiable at **(0 , 0)**.

b) Find the second order Taylor series expansion for ***f(x,y) = e***$^{-(x2+y2)}$ ***cos(xy)*** about **(0 , 0)**. (6)

15 a) There are two bags. The first bag contains four mangos and two apples; the second bag contains four mangos and four apples. We also have a biased coin, which shows "heads" with probability 0.6 and "tails" with probability 0.4. If the coin shows "heads". we pick a fruit at random from bag 1; otherwise we pick a fruit at random from bag 2. Your friend flips the coin (you cannot see the result), picks a fruit at random from the corresponding bag, and presents you a mango. (6)

What is the probability that the mango was picked from bag 2?

b) Suppose that one has written a computer program that sometimes compiles and sometimes not (code does not change). You decide to model the apparent stochasticity (success vs. no success) ***x*** of the compiler using a Bernoulli distribution with parameter μ: (8)

$$p(x \mid \mu) = \mu^x (1-\mu)^{1-x}, \quad x \in \{0,1\}$$

Choose a conjugate prior for the Bernoulli likelihood and compute the posterior distribution ***p(μ | x***$_1$ ***, ... , x***$_N$***)***.

OR

16 a) Consider a mixture of two Gaussian distributions (8)

$$0.4\mathcal{N}\left(\begin{bmatrix}10\\2\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right)+0.6\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}8.4&2.0\\2.0&1.7\end{bmatrix}\right)$$

i. Compute the marginal distributions for each dimension.

ii. Compute the mean, mode and median for each marginal distribution.

iii. Compute the mean and mode for the two-dimensional distribution.

b) Express the Binomial distribution as an exponential family distribution. Also express the Beta distribution is an exponential family distribution. Show that the product of the Beta and the Binomial distribution is also a member of the exponential family. (6)

17 a) Find the extrema of $f(x,y,z) = x - y + z$ subject to $g(x,y,z) = x^2 + y^2 + z^2 = 2$. (8)

b) Let

$$P=\begin{bmatrix}13&12&-2\\12&17&6\\-2&6&12\end{bmatrix},\ q=\begin{bmatrix}-22.0\\-14.5\\13.0\end{bmatrix},\ \text{and}\ r=1.$$

Show that $x^* = (1\ ,\ 1/2\ ,\ -1)$ is optimal for the optimization problem

$$\begin{array}{ll}\text{min} & \frac{1}{2}x^\mathsf{T}Px+q^\mathsf{T}x+r\\ \text{s.t.} & -1\le x_i\le 1,\ i=1,2,3.\end{array}$$

(6)

OR

18 a) Derive the gradient descent training rule assuming that the target function is represented as $o_d = w_0 + w_1x_1 + ... + w_nx_n$. Define explicitly the cost/ error function E, assuming that a set of training examples D is provided, where each training example $d \in D$ is associated with the target output t_d. (8)

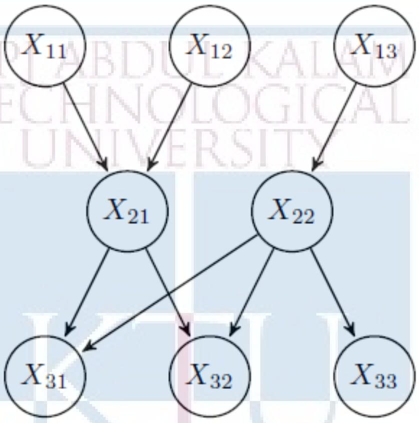
b) Find the maximum value of ***f(x,y,z) = xyz*** given that ***g(x,y,z) = x + y + z = 3*** and ***x,y,z >= 0***. (6)

19 a) Consider the following probability distribution (7)

$$P_\theta(x) = 2\theta x e^{-\theta x^2}$$

where $\boldsymbol{\theta}$ is a parameter and $\boldsymbol{x}$ is a positive real number. Suppose you get ***m*** i.i.d. samples $\boldsymbol{x_i}$ drawn from this distribution. Compute the maximum likelihood estimator for $\boldsymbol{\theta}$ based on these samples.

b) Consider the following Bayesian network with boolean variables. (7)



i. List variable(s) conditionally independent of X_{33} given X_{11} and X_{12}

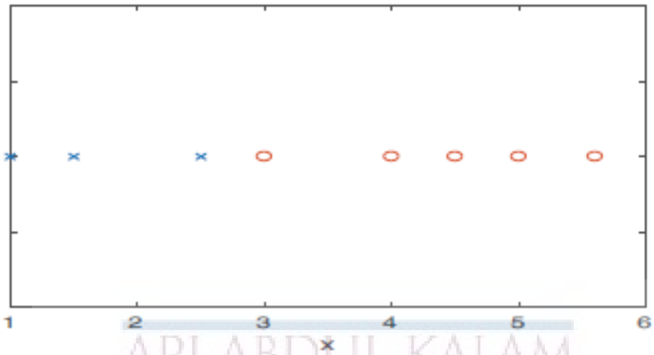
ii. List variable(s) conditionally independent of X_{33} and X_{22}

iii. Write the joint probability $P(X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{31}, X_{32}, X_{33})$ factored according to the Bayes net. How many parameters are necessary to define the conditional probability distributions for this Bayesian network?

iv. Write an expression for $P(X_{13} = 0, X_{22} = 1, X_{33} = 0)$ in terms of the conditional probability distributions given in your answer to part (iii). Justify your answer.

OR

20 a) Consider the following one dimensional training data set, ’x’ denotes negative examples and ’o’ positive examples. The exact data points and their labels are given in the table below. Suppose a SVM is used to classify this data. (6)



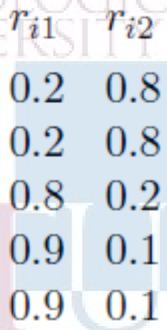
x	1	1.5	2.5	3	4	4.5	5	5.6
y	-1	-1	-1	1	1	1	1	1

i. Indicate which are the support vectors and mark the decision boundary.

ii. Give the value of the cost function and the model parameter after training.

b) Suppose that we are fitting a Gaussian mixture model for data items consisting of a single real value, $\boldsymbol{x}$, using $\boldsymbol{K = 2}$ components. We have $\boldsymbol{N = 5}$ training cases, in which the values of $\boldsymbol{x}$ are as **5, 15, 25, 30, 40**. Using the EM algorithm to find the maximum likeihood estimates for the model parameters, what are the mixing proportions for the two components, π_1 and π_2, and the means for the two components, μ_1 and μ_2. The standard deviations for the two components are fixed at 10. (8)

Suppose that at some point in the EM algorithm, the **E** step found that the responsibilities of the two components for the five data items were as follows:



r_{i1}	r_{i2}
0.2	0.8
0.2	0.8
0.8	0.2
0.9	0.1
0.9	0.1

What values for the parameters π_1, π_2 , μ_1, and μ_2 will be found in the next **M** step of the algorithm?

****

Teaching Plan

No	Topic	No. of Lectures (45)
1	**Module-I (LINEAR ALGEBRA)**	**8**
1.	Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence.	1
2.	Vector Spaces - Basis and Rank	1
3.	Linear Mappings	1
4.	Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Orthonormal Basis, Orthogonal Complement	1
5.	Orthogonal Projections, Matrix Decompositions, Determinant and Trace.	1
6.	Eigenvalues and Eigenvectors	1
7.	Cholesky Decomposition, Eigen decomposition and Diagonalization	1
8.	Singular Value Decomposition - Matrix Approximation	1
	Module-II (VECTOR CALCULUS)	**6**
1	Differentiation of Univariate Functions, Partial Differentiation and Gradients	1
2	Gradients of Vector Valued Functions, Gradients of Matrices	1
3	Useful Identities for Computing Gradients	1
4	Backpropagation and Automatic Differentiation	1
5	Higher Order Derivatives	1
6	Linearization and Multivariate Taylor Series	1
3	**Module-III (Probability and Distributions)**	**10**
1	Construction of a Probability Space - Discrete and Continuous Probabilities (Lecture 1)	1

2	Construction of a Probability Space - Discrete and Continuous Probabilities (Lecture 2)	1
3	Sum Rule, Product Rule	1
4	Bayes' Theorem	1
5	Summary Statistics and Independence	1
6	Important probability Distributions (Lecture 1)	1
7	Important probability Distributions (Lecture 2)	1
8	Conjugacy and the Exponential Family (Lecture 1)	1
9	Conjugacy and the Exponential Family (Lecture 2)	1
10	Change of Variables/Inverse Transform	1
4	**Module-IV (Optimization)**	**7**
1	Optimization Using Gradient Descent.	1
2	Gradient Descent With Momentum, Stochastic Gradient Descent	1
3	Constrained Optimization and Lagrange Multipliers (Lecture 1)	1
4	Constrained Optimization and Lagrange Multipliers (Lecture 2)	1
5	Convex Optimization	1
6.	Linear Programming	1
7.	Quadratic Programming	1
5	**Module-V (CENTRAL MACHINE LEARNING PROBLEMS)**	**14**
1.	Data and Learning models - Empirical Risk Minimization,	1
2.	Parameter Estimation	1
3.	Directed Graphical Models	1
4.	Linear Regression	1
5.	Bayesian Linear Regression	1
6.	Maximum Likelihood as Orthogonal Projection	1
7.	Dimensionality Reduction with Principal Component Analysis - Maximum Variance Perspective, Projection Perspective.	1
8.	Eigenvector Computation and Low Rank Approximations	1
9.	Density Estimation with Gaussian Mixture Models	1

10.	Parameter Learning via Maximum Likelihood	1
11.	EM Algorithm	1
12.	Classification with Support Vector Machines - Separating Hyperplanes	1
13.	Primal Support Vector Machines, Dual Support Vector Machines	1
14.	Kernels	1
*Assignments may include applications of the above theory. With respect to module V, programming assignments may be given.		

CST 286	INTRODUCTION TO COMPUTER NETWORKS	Category	L	T	P	Credit	Year of Introduction
		MINOR	3	1	0	4	2019

Preamble: This is the second course for awarding B. Tech. Minor in Computer Science and Engineering with specialization in ***Networking***. Study of this course provides the learners a clear understanding of how computer networks from local area networks to the massive and global Internet are built and how they allow the usage of computers to share information and communicate with one another. This course covers the layers of OSI Reference models and inter-networking. This course helps the learners to compare and analyze the existing network technologies and to choose a suitable network design for a given system.

Prerequisite: Data Communication **(CST 255)**

Course Outcomes: After the completion of the course the student will be able to

CO 1	Explain the features of computer networks, protocols and network design models **(Cognitive Knowledge : Understand)**
CO 2	Discuss the design issues of data link layer, data link layer protocols, bridges and switches **(Cognitive Knowledge : Understand)**
CO 3	Illustrate wired LAN protocols (IEEE 802.3/4/5) and wireless LAN protocols (IEEE 802.11a/b/g/n, 802.15) **(Cognitive Knowledge : Understand)**
CO 4	Select appropriate routing algorithms, congestion control techniques and Quality of Service requirements for a network **(Cognitive Knowledge : Apply)**
CO 5	Illustrate the functions and protocols of network layer, transport layer and application layer in inter-networking **(Cognitive Knowledge : Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓								✓		✓
CO2	✓	✓	✓							✓		✓
CO3	✓	✓	✓							✓		✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓			✓				✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	PO#	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Test 1 (Marks in percentage)**	**Test 2 (Marks in percentage)**	**End Semester Examination (Marks in percentage)**
Remember	40	30	30
Understand	60	50	50
Apply		20	20
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:
There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module 1

Introduction – Uses of Computer Networks, Network Hardware, Network Software, Reference Models – The OSI Reference Model, The TCP/IP Reference Model, Comparison of OSI and TCP/IP Reference models.

Module 2

The Data Link Layer - Data Link layer Design Issues, Error Detection and Correction, Elementary Data Link Protocols, Sliding Window Protocols, HDLC (High-Level Data Link Control) Protocol. The Medium Access Control (MAC) Sub layer – The Channel Allocation Problem, Multiple Access Protocols, Ethernet, Wireless LANs - 802.11 a/b/g/n, Bridges & Switches.

Module 3

Network Layer Design Issues. Routing Algorithms - The Optimality Principle, Shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Multicast Routing, Routing for Mobile Hosts. Congestion Control Algorithms, Quality of Service (QoS) - Requirements, Techniques for Achieving Good QoS.

Module 4

Network Layer in Internet – The IP Protocol, IP Addresses, Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (**RARP**), Bootstrap Protocol (**BOOTP**), Dynamic Host Configuration Protocol (DHCP). Open Shortest Path First (**OSPF**) Protocol, Border Gateway Protocol (**BGP**), Internet Multicasting, IPv6, ICMPv6.

Module 5

Transport Layer – The Transport Service – Services Provided to the Upper Layers, Transport Service Primitives. The User Datagram Protocol (UDP), Transmission Control Protocol (TCP) – Overview of TCP, TCP Segment Header, Connection Establishment & Release, Connection Management Modeling, TCP Retransmission Policy, TCP Congestion Control.

Application Layer – File Transfer Protocol (FTP), Domain Name System (DNS), Electronic mail, MIME, Simple Network Management Protocol (SNMP), World Wide Web – Architectural Overview.

Text Book

Andrew S. Tanenbaum, Computer Networks, 4/e, PHI (Prentice Hall India).

Reference Books

1. Behrouz A Forouzan, Data Communication and Networking, 4/e, Tata McGraw Hill
2. Larry L Peterson and Bruce S Dave, Computer Networks – A Systems Approach, 5/e, Morgan Kaufmann.
3. Fred Halsall, Computer Networking and the Internet, 5/e.
4. James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 6/e.
5. Keshav, An Engineering Approach to Computer Networks, Addison Wesley, 1998.
6. W. Richard Stevens. TCP/IP Illustrated volume 1, Addison-Wesley, 2005.
7. William Stallings, Computer Networking with Internet Protocols, Prentice-Hall, 2004.
8. Request for Comments (RFC) Pages - IETF -https://www.ietf.org/rfc.html

Sample Course Level Assessment Questions

CourseOutcome1 (CO1): Compare TCP/IP Reference model and OSI Reference model.

CourseOutcome2 (CO2): Distinguish between switches and bridges.

CourseOutcome3 (CO3): Draw and explain the frame format for Ethernet.

CourseOutcome5 (CO4): Discuss remedies for count to infinity problem in routing.

CourseOutcome4 (CO5): Subnet the Class C IP Address 206.16.2.0 so that you have 30 subnets. What is the subnet mask for the maximum number of hosts? How many hosts can each subnet have?

Model Question Paper

QP CODE: PAGES: ___

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FOURTH SEMESTER B.TECH DEGREE (MINOR) EXAMINATION, MONTH & YEAR

Course Code: CST 286

Course name : INTRODUCTION TO COMPUTER NETWORKS

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Why Layered Architecture is used in Computer Networks? Define the terms protocol and interface?
2. What are the different service primitives in Computer Networks?
3. Draw and explain Ethernet frame format.
4. What is the output string when the bit string 0111101111101111110 is subjected to bit stuffing?
5. Discuss the count to infinity problem in routing.
6. What is flooding? Describe any two situations where flooding is advantageous.
7. What is IP (Internet Protocol) subnetting? Illustrate with example.
8. How many octets does the smallest possible IPv6 (IP version 6) datagram contain?
9. Can TCP (Transmission Control Protocol) be used directly over a network (e.g. an Ethernet) without using IP? Justify your answer
10. What is the role of SNMP (Simple Network Management Protocol)?

(10x3=30)

Part B

(Answer any one Question from each module. Each question carries 14 Marks)

Module I

11. (a) With a neat diagram, explain the OSI (Open Systems Interconnection) reference Model. **(8)**

(b) Compare OSI Reference model and the TCP/IP model **(6)**

OR

12. (a) Consider two networks providing reliable connection-oriented service. One of them offers a reliable byte stream and the other offers a reliable message stream. Are they identical? Justify your answer. **(8)**

(b) Compare LAN (Local Area Networks), MAN (Metropolitan Area Networks) and WAN (Wide Area Networks). **(6)**

Module II

13. (a) Discuss the different strategies used to avoid collisions in CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) . **(8)**

(b) Briefly explain the working of HDLC (High-Level Data Link Control). **(6)**

OR

14. (a) Explain the working of IEEE 802.11. **(10)**

(b) Distinguish between Bridges and Switches. **(4)**

Module III

15. (a) Illustrate Distance Vector Routing Algorithm with an example. **(8)**

(b) Explain the characteristics of RIP (Routing Information Protocol). **(6)**

OR

16. (a) Explain an Interior Gateway protocol that uses a link state algorithm to propagate routing information. **(6)**

(b) Explain how routing is performed in a Mobile network. **(8)**

Module IV

17. (a) Explain address resolution problem and RARP (Reverse Address Resolution Protocol) with an example network. **(10)**

(b) How IGMP (Internet Group Management Protocol) supports internet multicasting? Explain. **(4)**

OR

18. (a) Subnet the class C IP address 195.1.1.0 so that you have 10 subnets with a maximum of 12 hosts in each subnet. **(6)**

(b) Draw IPv6 Datagram format and explain its features **(8)**

Module V

19. (a) Distinguish between TCP and UDP (User Datagram Protocol) header formats. **(8)**

(b) Explain the principal DNS (Domain Name System) resource record types for IPv4. **(6)**

OR

20. (a) What is the role of SMTP (Simple Mail Transfer Protocol) in E-mail? **(6)**

(b) With the help of a basic model explain the working of WWW (World Wide Web). **(8)**

Teaching Plan

	Module 1	**(8 Hours)**
1.1	Introduction – Uses of Computer Networks.	1
1.2	Network Hardware – Local Area Networks (LAN), Metropolitan Area Networks (MAN), Wide Area Networks (WAN).	1
1.3	Network Hardware – Wireless Networks, Home Networks, Internetworks	1
1.4	Network Software — Protocol Hierarchies.	1
1.5	Network Software — Design issues for the layers.	1
1.6	Network Software – Connection Oriented and Connectionless Services, Service Primitives, Relationship of Services to Protocols.	1
1.7	Reference Models – The OSI Reference Model	1

1.8	Reference Models – The TCP/IP Reference Model, Comparison of OSI and TCP/IP Reference models	1
Module 2		**(11 Hours)**
2.1	Data Link layer Design Issues.	1
2.2	Error Detection and Correction - Error Correcting Codes	1
2.3	Error Detection and Correction - Error Detecting Codes	1
2.4	Elementary Data link Protocols.	1
2.5	Sliding Window Protocols.	1
2.6	HDLC (High-Level Data Link Control) Protocol	1
2.7	The Medium Access Control (MAC) Sub layer – The Channel Allocation Problem, Multiple Access Protocols.	1
2.8	Ethernet - Ethernet Cabling, Manchester Encoding, The Ethernet MAC Sub layer Protocol, The Binary Exponential Backoff Algorithm.	1
2.9	Ethernet - Ethernet Performance, Switched Ethernet, Fast Ethernet, Gigabit Ethernet, IEEE 802.2: Logical Link Control.	1
2.10	Wireless LANs - 802.11 a/b/g/n.	1
2.11	Bridges & Switches.	1
Module 3		**(9 Hours)**
3.1	Network Layer Design Issues.	**1**
3.2	Routing Algorithms - The Optimality Principle, Shortest path routing, Flooding.	**1**
3.3	Distance Vector Routing, Link State Routing.	**1**
3.4	Link State Routing.	**1**
3.5	Multicast Routing, Routing for Mobile Hosts	**1**
3.6	Distance Vector Routing, Link State Routing	**1**

3.7	Congestion control algorithms - General Principles of Congestion Control, Congestion Prevention Policies, Congestion Control in Virtual-Circuit Subnets	**1**
3.8	Congestion control algorithms - Congestion Control in Datagram Subnets, Load Shedding, Jitter Control	**1**
3.9	Quality of Service – Requirements, Techniques for Achieving Good Quality of Service.	**1**
Module 4		**(9 Hours)**
4.1	Network layer in internet, IP Protocol	1
4.2	IP Addresses – Subnets, Classless Inter Domain Routing (CIDR)	1
4.3	IP Addresses - Network Address Translation (NAT)	1
4.4	Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP),	1
4.5	Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP)	1
4.6	Open Shortest Path First (**OSPF**) Protocol	1
4.7	Border Gateway Protocol (**BGP**)	1
4.8	Internet Multicasing.	1
4.9	IPv6, Internet Control Message Protocol version 6 (ICMPv6).	1
Module 5		**(8 Hours)**
5.1	The Transport Service – Services Provided to the Upper Layers, Transport Service Primitives. The User Datagram Protocol (UDP)	1
5.2	Transmission Control Protocol (TCP) – Overview of TCP, TCP Segment Header, Connection Establishment & Release, Connection Management Modeling.	1
5.3	TCP Retransmission Policy, TCP Congestion Control.	1
5.4	Application Layer – File Transfer Protocol (FTP).	1
5.5	Domain Name System (DNS).	1

5.6	Electronic Mail.	1
5.7	Simple Network Management Protocol (SNMP)	1
5.8	World Wide Web – Architectural Overview	1

SEMESTER -4

HONOURS

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT	Year of Introduction
CST 292	Number Theory	Honours	4	0	0	4	2019

Preamble: This is the foundational course for awarding B. Tech. Honours in Computer Science and Engineering with specialization in ***Security in Computing***. The purpose of this course is to create awareness among learners about the important areas of number theory used in computer science. This course covers Divisibility & Modular Arithmetic, Primes & Congruences, Euler's Function, Quadratic Residues and Arithmetic Functions, Sum of Squares and Continued fractions. Concepts in Number Theory help the learner to apply them eventually in practical applications in Computer organization & Security, Coding & Cryptography, Random number generation, Hash functions and Graphics.

Prerequisite: A sound background in Higher Secondary School Mathematics

Course Outcomes: After the completion of the course the student will be able to

CO1	Illustrate modular arithmetic operations, methods and techniques **(Cognitive Knowledge Level:Understand)**
CO2	Use the methods - Induction, Contraposition or Contradiction to verify the correctness of mathematical assertions **(Cognitive Knowledge Level: Apply)**
CO3	Utilize theorems and results about prime numbers, congruences, quadratic residues and integer factorization for ensuring security in computing systems **(Cognitive Knowledge Level: Analyse)**
CO4	Illustrate uses of Chinese Remainder Theorem & Euclidean algorithm in Cryptography and Security **(Cognitive Knowledge Level: Apply)**
CO5	Explain applications of arithmetic functions in Computer Science **(Cognitive Knowledge Level:Understand)**
CO6	Implement Number Theoretic Algorithms using a programming language **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓						✓		✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓		✓						✓
CO4	✓	✓	✓	✓		✓						✓
CO5	✓	✓	✓	✓						✓		✓
CO6	✓	✓	✓	✓	✓			✓				✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (Percentage)
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

SYLLABUS

Module 1

Divisibility and Modular Arithmetic:

Finite Fields – Groups, Rings and Fields.

Divisibility - Divisibility and Division Algorithms, Well ordering Principle,Bezout's Identity.

Modular Arithmetic- Properties, Euclid's algorithm for the greatest common divisor, Extended Euclid's Algorithm, Least Common multiple, Solving Linear Diophantine Equations, Modular Division.

Module 2

Primes and Congruences:

Prime Numbers-Prime Numbers andprime-powerfactorization, Fermat and Mersenne primes., Primality testing and factorization.

Congruences-Linear congruences, Simultaneous linear congruences, Chinese Remainder Theorem, Fermat’s little theorem, Wilson's theorem.

Module 3

Congruences with a Prime-Power Modulus&Euler's Function:

Congruences with a Prime-Power Modulus-Arithmetic modulo p, Pseudoprimes and Carmichael numbers, Solving congruences modulo prime powers.

Euler's Function-Euler’s Totient function, Applications of Euler’s Totient function, Traditional Cryptosystem, Limitations.

The Group of units- The group U_n,Primitive roots, Existence of primitive roots, Applications of primitive roots.

Module 4

Quadratic Residues & Arithmetic Functions :

Quadratic Residues- Quadratic Congruences, The group of Quadratic residues, Legendre symbol, Jacobi Symbol, Quadratic reciprocity.

Arithmetic Functions- Definition and examples, Perfect numbers, Mobius function and its properties, Mobius inversion formula, The Dirichlet Products.

Module 5

Sum of Squares and Continued Fractions:

Sum of Squares- Sum of two squares, The Gaussian Integers, Sum of three squares, Sum of four squares.

Continued Fractions -Finite continued fractions, Infinite continued fractions, Pell's Equation, Solution of Pell’s equation by continued fractions.

Text Books

1. G.A. Jones & J.M. Jones, Elementary Number Theory, Springer UTM, 2007.
2. Joseph Silverman, A Friendly introduction to Number Theory, Pearson Ed. 2009.

Reference Books

1. William Stallings, Cryptography and Network Security Principles and Practice, Pearson Ed.
2. Tom M.Apostol, ‘Introduction to Analytic Number Theory’, Narosa Publishing House Pvt. Ltd, New Delhi, (1996).
3. Neal Koblitz, A course in Number Theory and Cryptography, 2nd Edition, Springer ,2004.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Describe the properties of modular arithmetic and modulo operator.

Course Outcome 2 (CO2): Prove that the equation $y^2 = x^3 - 2$ has only the integer solution (3, ±5).

Course Outcome 3 (CO3): State the law of reciprocity for Jacobi symbols and use it to determine whether 888 is a quadratic residue or non residue of the prime 1999.

Course Outcome 4 (CO4): Using Chinese remainder theorem, solve the system of congruence $x \equiv 2 \pmod 3$, $x \equiv 3 \pmod 5$, $x \equiv 2 \pmod 7$

Course Outcome 5(CO5): State and prove Dirichlet product.

Course Outcome 6 (CO6): Use extended Euclid's algorithm to solve Diophantine equations efficiently. Given three numbers a>0, *b*>0, and *c*, the algorithm should return some *x* and *y* such that a x + b y = c.

Model Question Paper

QP CODE: PAGES: 03

RegNo : …............................... Name :………..……………

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FOURTH SEMESTER BTECH (HONOURS) DEGREE EXAMINATION, MONTH &YEAR

Course Code:CST 292 Course
Name: Number Theory

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks (10x3=30)

1. State and prove well ordering principle.
2. Find gcd d of x=525 and y=231 and express d as ax + by where a and b are integers.
3. Solve the congruence equation 103 x≡ 57 (mod 211).
4. Use Fermat's Little theorem to show that 91 is not a prime.
5. If m is relatively prime to n , show that *Φ(mn) = Φ(m) Φ(n)*.
6. Explain how public key cryptography can be used for digital signatures.
7. Define Mobius function and prove Mobius function is a multiplicative.
8. State and prove Dirichlet product.
9. Show that every prime of the form 4k+1 canbe represented uniquely as the sum of two squares.
10. Find the continued fraction representation of the rational number 55/89.

Part B

Answer any one Question from each module.
Each question carries 14 Marks

11. (a) State the Euclidean algorithm and its extension with an example. (7)

    (b) Find all the solutions of 24x + 34 y = 6. (7)

OR

12. (a) Describe the properties of modular arithmetic and modulo operator. (7)

    (b) Explain Extended Euclidean algorithm. Using the algorithm find the multiplicative inverse of 135 mod 61 (7)

13. (a) State and prove Wilson's theorem . (7)

(b) Explain Fermat's factorization method and use it to factor 809009 (7)

OR

14. (a) Using Chinese remainder theorem, solve the system of congruences, $x \equiv 2 (\text{mod } 3)$, $x \equiv 3 (\text{mod } 5)$, $x \equiv 2 (\text{mod } 7)$ (7)

(b) Define Fermat primes. Show that any two distinct Fermat numbers are Relatively prime. (7)

15. (a) Distinguish between public key and private key encryption techniques. Also point out the merits and demerits of both. (7)

(b) Define Carmichael number and show that a Carmichael number must be the product of at least three distinct primes. (7)

OR

16. (a)Define a pseudo prime to a base and find all non trivial bases for which 15 is a pseudo prime. (6)

(b) Find an element of

i) order 5 modulo 11 ii) order 4 modulo 13

iii) order 8 modulo 17 iv) order 6 modulo 19 (8)

17. (a) Determine the quadratic residues and non residues modulo 17. Also determine whether 219 is a quadratic residue or non residue of the prime 383. (8)

(b) State the law of quadratic reciprocity. Determine those odd primes p for which 3 is a quadratic residue and those for which it is a non residue. (6)

OR

18. (a) State and prove properties of Legendre's symbol. (7)

(b) State the law of reciprocity for Jacobi symbols and using it determine whether 888 is a quadratic residue or non residue of the prime 1999. (7)

19. (a) Prove that the equation $y^2 = x^3 - 2$ has only the integer solution $(3, \pm5)$. (7)

(b) Define a Gaussian integer. Factorize the Gaussian integer 440 − 55i. (7)

OR

20. (a) If *m*, and *n* can be expressed as sum of four squares, then show that *mn* can also be expressed the sum of four squares. (7)

(b) Find all the solutions of the Diophantine equation $x^2 – 6 y^2 =1$. (7)

Teaching Plan

Module 1: Divisibility and Euclidean Algorithm		**9 hours**
1.1	Finite Fields – Groups and Rings.	1 hour
1.2	Finite Fields – Fields.	1 hour
1.3	Divisibility and Division Algorithms, Well ordering Principle.	1 hour
1.4	Decimal Expansion of a positive Integer, Greatest Common Divisor, Bezout's Theorem.	1 hour
1.5	Modular Arithmetic- Properties of congruences, Modular Arithmetic Operations, Properties of Modular Arithmetic.	1 hour
1.6	Euclid's algorithm for the greatest common divisor, Extended Euclid's Algorithm.	1 hour
1.7	Solving Linear Diophantine Equations.	1 hour
1.8	Least Common multiple and Modular Division.	1 hour
1.9	Implementation of Euclid's algorithm, Extended Euclid's Algorithm and solution of Linear Diophantine Equations.	1 hour
Module 2: Primes and Congruences		**9 hours**
2.1	Prime Numbersand prime-powerFactorization.	1 hour
2.2	Fermat and Mersenne primes.	1 hour
2.3	Primality testing and factorization, Miller -Rabin Test for Primality.	1 hour
2.4	Pollard's Rho Method for Factorization, Fermat's Factorization.	1 hour

2.5	Linear congruences, Simultaneous linear congruences.	1 hour
2.6	Chinese Remainder Theorem.	1 hour
2.7	Implementation of Chinese Remainder Theorem.	1 hour
2.8	Fermat's little theorem.	1 hour
2.9	Wilson's theorem.	1 hour
Module 3: Congruences with a Prime-Power Modulus &Euler's Function		**9 hours**
3.1	Congruences with a Prime-Power Modulus, Arithmetic modulo p.	1 hour
3.2	Pseudo-primes and Carmichael numbers.	1 hour
3.3	Solving congruences modulo prime powers.	1 hour
3.4	Definition of Euler Totient function, Examples and properties.	1 hour
3.5	Multiplicativity of Euler's Totient function.	1 hour
3.6	Applications of Euler's function, Euler's Theorem.	1 hour
3.7	Traditional Cryptosystem, Limitations, Public Key Cryptography.	1 hour
3.8	The Group of Units, Primitive Roots.	1 hour
3.9	Existence of primitive roots for Primes, Applications of primitive roots.	1 hour
Module 4: Quadratic Residues and Arithmetic Functions		**9 hours**
4.1	Quadratic congruences, The group of Quadratic Residues.	1 hour
4.2	Legendre symbol, Jacobi Symbol.	1 hour
4.3	Quadratic reciprocity.	1 hour
4.4	Quadratic residues for prime-power moduli.	1 hour
4.5	Arithmetic Functions: Definition and examples.	1 hour

4.6	Perfect numbers, Definition and proposition.	1 hour
4.7	Mobius inversion formula., application of the Mobius inversion formula.	1 hour
4.8	Mobius function and its properties.	1 hour
4.9	The Dirichlet Product, Definition and proof.	1 hour
Module 5: Sum of Squares and Continued Fractions		**9 hours**
5.1	Sum of Squares, Sum of two squares.	1 hour
5.2	The Gaussian Integers.	1 hour
5.3	Sum of three squares.	1 hour
5.4	Sum of four squares.	1 hour
5.5	Continued Fractions, Finite continued fractions.	1 hour
5.6	Continued Fractions, Finite continued fractions.	1 hour
5.7	Infinite continued fractions.	1 hour
5.8	Pell's Equation, Definition.	1 hour
5.9	Solution of Pell's equation by continued fractions.	1 hour

CODE CST 294	Computational Fundamentals for Machine Learning	CATEGORY	L	T	P	CREDIT
		HONOURS	3	1	0	4

Preamble: This is the foundational course for awarding B. Tech. Honours in Computer Science and Engineering with specialization in ***Machine Learning***. The purpose of this course is to introduce mathematical foundations of basic Machine Learning concepts among learners, on which Machine Learning systems are built. This course covers Linear Algebra, Vector Calculus, Probability and Distributions, Optimization and Machine Learning problems. Concepts in this course help the learners to understand the mathematical principles in Machine Learning and aid in the creation of new Machine Learning solutions, understand & debug existing ones, and learn about the inherent assumptions & limitations of the current methodologies.

Prerequisite: A sound background in higher secondary school Mathematics.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Make use of the concepts, rules and results about linear equations, matrix algebra, vector spaces, eigenvalues & eigenvectors and orthogonality & diagonalization to solve computational problems (Cognitive Knowledge Level: **Apply**)
CO 2	Perform calculus operations on functions of several variables and matrices, including partial derivatives and gradients (Cognitive Knowledge Level: **Apply**)
CO 3	Utilize the concepts, rules and results about probability, random variables, additive & multiplicative rules, conditional probability, probability distributions and Bayes' theorem to find solutions of computational problems (Cognitive Knowledge Level: **Apply**)
CO 4	Train Machine Learning Models using unconstrained and constrained optimization methods (Cognitive Knowledge Level: **Apply**)
CO 5	Illustrate how the mathematical objects - linear algebra, probability, and calculus can be used to design machine learning algorithms (Cognitive Knowledge Level: **Understand**)

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	√	√	√	√								√
CO 2	√	√	√									√
CO 3	√	√	√	√								√
CO 4	√	√	√	√		√						√
CO 5	√	√	√	√	√	√				√		√

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	20%	20%	20%
Understand	40%	40%	40%
Apply	40%	40%	40%
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer anyone. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module 1

LINEAR ALGEBRA : Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence, Basis and Rank, Linear Mappings, Norms, - Inner Products - Lengths and Distances - Angles and Orthogonality - Orthonormal Basis - Orthogonal Complement - Orthogonal Projections. Matrix Decompositions - Determinant and Trace, Eigenvalues and Eigenvectors, Cholesky Decomposition, Eigen decomposition and Diagonalization, Singular Value Decomposition, Matrix Approximation.

Module 2

VECTOR CALCULUS : Differentiation of Univariate Functions - Partial Differentiation and Gradients, Gradients of Vector Valued Functions, Gradients of Matrices, Useful Identities for Computing Gradients. Back propagation and Automatic Differentiation - Higher Order Derivatives- Linearization and Multivariate Taylor Series.

Module 3

Probability and Distributions : Construction of a Probability Space - Discrete and Continuous Probabilities, Sum Rule, Product Rule, and Bayes' Theorem. Summary Statistics and Independence – Important Probability distributions - Conjugacy and the Exponential Family - Change of Variables/Inverse Transform.

Module 4

Optimization : Optimization Using Gradient Descent - Gradient Descent With Momentum, Stochastic Gradient Descent. Constrained Optimization and Lagrange Multipliers - Convex Optimization - Linear Programming - Quadratic Programming.

Module 5

CENTRAL MACHINE LEARNING PROBLEMS : Data and Learning Model- Empirical Risk Minimization - Parameter Estimation - Directed Graphical Models.

Linear Regression - Bayesian Linear Regression - Maximum Likelihood as Orthogonal Projection.

Dimensionality Reduction with Principal Component Analysis - Maximum Variance Perspective, Projection Perspective. Eigenvector Computation and Low Rank Approximations.

Density Estimation with Gaussian Mixture Models - Gaussian Mixture Model, Parameter Learning via Maximum Likelihood, EM Algorithm.

Classification with Support Vector Machines - Separating Hyperplanes, Primal Support Vector Machine, Dual Support Vector Machine, Kernels.

Text book:

1.Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong published by Cambridge University Press (freely available at https:// mml - book.github.io)

Reference books:

1. Linear Algebra and Its Applications, 4th Edition by Gilbert Strang

2. Linear Algebra Done Right by Axler, Sheldon, 2015 published by Springer

3. Introduction to Applied Linear Algebra by Stephen Boyd and Lieven Vandenberghe, 2018 published by Cambridge University Press

4. Convex Optimization by Stephen Boyd and Lieven Vandenberghe, 2004 published by Cambridge University Press

5. Pattern Recognition and Machine Learning by Christopher M Bishop, 2006, published by Springer

6. Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond by Bernhard Scholkopf and Smola, Alexander J Smola, 2002, bublished by MIT Press

7. Information Theory, Inference, and Learning Algorithms by David J. C MacKay, 2003 published by Cambridge University Press

8. Machine Learning: A Probabilistic Perspective by Kevin P Murphy, 2012 published by MIT Press.

9. The Nature of Statistical Learning Theory by Vladimir N Vapnik, 2000, published by Springer

Sample Course Level Assessment Questions.

Course Outcome 1 (CO1):

1. Find the set ***S*** of all solutions in ***x*** of the following inhomogeneous linear systems ***Ax*** = ***b***, where ***A*** and ***b*** are defined as follows:

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

2. Determine the inverses of the following matrix if possible

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

3. Are the following sets of vectors linearly independent?

$$\boldsymbol{x}_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

4. A set of ***n*** linearly independent vectors in $\boldsymbol{R^n}$ forms a basis. Does the set of vectors **(2, 4,−3)** , **(0, 1, 1)** , **(0, 1,−1**) form a basis for $\boldsymbol{R^3}$? Explain your reasons.

5. Consider the transformation ***T (x, y) = (x + y, x + 2y, 2x + 3y)***. Obtain ***ker T*** and use this to calculate the nullity. Also find the transformation matrix for ***T***.

6. Find the characteristic equation, eigenvalues, and eigenspaces corresponding to each eigenvalue of the following matrix

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

7. Diagonalize the following matrix, if possible

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}$$

8. Find the singular value decomposition (SVD) of the following matrix

$$\begin{bmatrix} 0 & 1 & 1 \\ \sqrt{2} & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Course Outcome 2 (CO2):

1. For a scalar function $f(x, y, z) = x^2 + 3y^2 + 2z^2$, find the gradient and its magnitude at the point **(1, 2, -1)**.
2. Find the maximum and minimum values of the function $f(x, y) = 4x + 4y - x^2 - y^2$ subject to the condition $x^2 + y^2 <= 2$.
3. Suppose you were trying to minimize $f(x, y) = x^2 + 2y + 2y^2$. Along what vector should you travel from (5, 12)?
4. Find the second order Taylor series expansion for $f(x, y) = (x + y)^2$ about **(0 , 0)**.
5. Find the critical points of $f(x, y) = x^2 - 3xy + 5x - 2y + 6y^2 + 8$.
6. Compute the gradient of the Rectified Linear Unit (ReLU) function $ReLU(z) = max(0 , z)$.
7. Let $L = ||Ax - b||^2_2$, where A is a matrix and x and b are vectors. Derive dL in terms of dx.

Course Outcome 3 (CO3):

1. Let J and T be independent events, where $P(J)=0.4$ and $P(T)=0.7$.

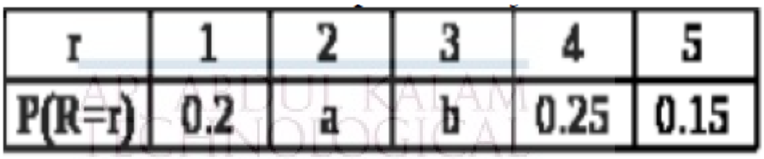
   i. Find $P(J \cap T)$

   ii. Find $P(J \cup T)$

   iii. Find $P(J \cap T')$

2. Let A and B be events such that $P(A)=0.45$, $P(B)=0.35$ and $P(A \cup B)=0.5$. Find $P(A|B)$.

3. A random variable **R** has the probability distribution as shown in the following table:



r	1	2	3	4	5
P(R=r)	0.2	a	b	0.25	0.15

   i. Given that $E(R)=2.85$, find a and b.

   ii. Find $P(R>2)$.

4. A biased coin (with probability of obtaining a head equal to $p > 0$) is tossed repeatedly and independently until the first head is observed. Compute the probability that the first head appears at an even numbered toss.

5. Two players A and B are competing at a trivia quiz game involving a series of questions. On any individual question, the probabilities that A and B give the correct answer are p and q respectively, for all questions, with outcomes for different questions being independent. The game finishes when a player wins by answering a question correctly. Compute the probability that A wins if

   i. A answers the first question,

   ii. B answers the first question.

6. A coin for which $P(heads) = p$ is tossed until two successive tails are obtained. Find the probability that the experiment is completed on the n^{th} toss.

7. You roll a fair dice twice. Let the random variable X be the product of the outcomes of the two rolls. What is the probability mass function of X? What are the expected value and the standard deviation of X?

8. While watching a game of Cricket, you observe someone who is clearly supporting Mumbai Indians. What is the probability that they were actually born within 25KM of Mumbai? Assume that:

   - the probability that a randomly selected person is born within 25KM of Mumbai is 1/20;
   - the chance that a person born within 25KMs of Mumbai actually supports MI is 7/10 ;
   - the probability that a person not born within 25KM of Mumbai supports MI with probability 1/10.

9. What is an exponential family? Why are exponential families useful?
10. Let Z_1 and Z_2 be independent random variables each having the standard normal distribution. Define the random variables X and Y by $X = Z_1 + 3Z_2$ and $Y = Z_1 + Z_2$. Argue that the joint distribution of (X, Y) is a bivariate normal distribution. What are the parameters of this distribution?
11. Given a continuous random variable x, with cumulative distribution function $F_x(x)$, show that the random variable $y = F_x(x)$ is uniformly distributed.
12. Explain Normal distribution, Binomial distribution and Poisson distribution in the exponential family form.

Course Outcome 4(CO4):

1. Find the extrema of $f(x, y) = x$ subject to $g(x, y) = x^2 + 2y^2 = 3$.
2. Maximize the function $f(x, y, z) = xy + yz + xz$ on the unit sphere $g(x, y, z) = x^2 + y^2 + z^2 = 1$.
3. Provide necessary and suffcient conditions under which a quadratic optimization problem be written as a linear least squares problem.
4. Consider the univariate function $f(x) = x^3 + 6x^2 - 3x - 5$. Find its stationary points and indicate whether they are maximum, minimum, or saddle points.
5. Consider the update equation for stochastic gradient descent. Write down the update when we use a mini-batch size of one.

6. Consider the function

$$f(x) = (x_1 - x_2)^2 + \frac{1}{1 + x_1^2 + x_2^2}.$$

   i. Is ***f(x)*** a convex function? Justify your answer.

   ii. Is (1 , -1) a local/global minimum? Justify your answer.

7. Is the function ***f(x, y) = 2x² + y² + 6xy - x + 3y - 7*** convex, concave, or neither? Justify your answer.

8. Consider the following convex optimization problem

$$\text{minimize } \frac{x^2}{2} + x + 4y^2 - 2y$$

   Subject to the constraint ***x + y >= 4, x, y >= 1***.

   Derive an explicit form of the Lagrangian dual problem.

9. Solve the following LP problem with the simplex method.

$$max\ \ 5x_1 + 6x_2 + 9x_3 + 8x_4$$

   subject to the constraints

$$\begin{array}{ccccccccc} x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \leq & 5 \\ x_1 & + & x_2 & + & 2x_3 & + & 3x_4 & \leq & 3 \end{array}$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Course Outcome 5 (CO5):

1. What is a loss function? Give examples.
2. What are training/validation/test sets? What is cross-validation? Name one or two examples of cross-validation methods.
3. Explain generalization, overfitting, model selection, kernel trick, Bayesian learning

4. Distinguish between Maximum Likelihood Estimation (MLE) and Maximum A Posteriori Estimation (MAP)?

5. What is the link between structural risk minimization and regularization?

6. What is a kernel? What is a dot product? Give examples of kernels that are valid dot products.

7. What is ridge regression? How can one train a ridge regression linear model?

8. What is Principal Component Analysis (PCA)? Which eigen value indicates the direction of largest variance? In what sense is the representation obtained from a projection onto the eigen directions corresponding the the largest eigen values optimal for data reconstruction?

9. Suppose that you have a linear support vector machine (SVM) binary classifier. Consider a point that is currently classified correctly, and is far away from the decision boundary. If you remove the point from the training set, and re-train the classifier, will the decision boundary change or stay the same? Explain your answer in one sentence.

10. Suppose you have ***n*** independent and identically distributed (i.i.d) sample data points $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}$. These data points come from a distribution where the probability of a given datapoint ***x*** is

$$P(x) = \frac{1}{\theta}e^{-\frac{1}{\theta}x}.$$

Prove that the MLE estimate of parameter is the sample mean.

11. Suppose the data set $\boldsymbol{y_1, \ldots, y_n}$ is a drawn from a random sample consisting of i.i.d. discrete uniform distributions with range 1 to *N*. Find the maximum likelihood estimate of ***N***.

12. Ram has two coins: one fair coin and one biased coin which lands heads with probability 3/4. He picks one coin at random (50-50) and flips it repeatedly until he gets a tails. Given that he observes 3 heads before the first tails, find the posterior probability that he picked each coin.

    i. What are the prior and posterior odds for the fair coin?

    ii. What are the prior and posterior predictive probabilities of heads on the next flip? Here prior predictive means prior to considering the data of the first four flips.

Model Question paper

QP Code : **Total Pages: 4**

Reg No.:_______________ Name:___________________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

IV SEMESTER B.TECH (HONOURS) DEGREE EXAMINATION, MONTH and YEAR

Course Code: CST 294

Course Name: COMPUTATIONAL FUNDAMENTALS FOR MACHINE LEARNING

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks. Marks

1 Show that with the usual operation of scalar multiplication but with addition on reals given by $x \# y = 2(x + y)$ is not a vector space.

2 Find the eigenvalues of the following matrix in terms of k. Can you find an eigenvector corresponding to each of the eigenvalues?

$$\begin{bmatrix} 1 & k \\ 2 & 1 \end{bmatrix}$$

3 Let $f(x, y, z) = xye^r$, where $r = x^2+z^2-5$. Calculate the gradient of f at the point **(1, 3, -2)**.

4 Compute the Taylor polynomials $T_n, n = 0 , \dots , 5$ of $f(x) = sin(x) + cos(x)$ at $x_0 = 0$.

5 Let X be a continuous random variable with probability density function on $0 <= x <= 1$ defined by $f(x) = 3x^2$. Find the pdf of $Y = X^2$.

6 Show that if two events A and B are independent, then A and B' are independent.

7 Explain the principle of the gradient descent algorithm.

8 Briey explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.

9 What is the empirical risk? What is “empirical risk minimization”?

10 Explain the concept of a Kernel function in Support Vector Machines. Why are kernels so useful? What properties a kernel should posses to be used in an SVM?

PART B

Answer any one Question from each module. Each question carries 14 Marks

11 a) i. Find all solutions to the system of linear equations (6)

$$\begin{aligned} -4x + 5z &= -2 \\ -3x - 3y + 5z &= 3 \\ -x + 2y + 2z &= -1 \end{aligned}$$

ii. Prove that all vectors orthogonal to **[2, −3, 1]^T** forms a subspace ***W*** of ***R***^{3}. What is ***dim (W)*** and why?

b) Use the Gramm-Schmidt process to find an orthogonal basis for the column space of the following matrix (8)

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

OR

12 a) i. Let $\boldsymbol{L}$ be the line through the origin in $\boldsymbol{R^2}$ that is parallel to the vector **$[3, 4]^T$**. Find the standard matrix of the orthogonal projection onto L. Also find the point on $\boldsymbol{L}$ which is closest to the point **(7 , 1)** and find the point on $\boldsymbol{L}$ which is closest to the point ***(-3 , 5)***. (6)

ii. Find the rank-1 approximation of

$$\begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

b) i. Find an orthonormal basis of $\boldsymbol{R^3}$ consisting of eigenvectors for the following matrix (8)

$$\begin{bmatrix} 1 & 0 & -2 \\ 0 & 5 & 0 \\ -2 & 0 & 4 \end{bmatrix}$$

ii. Find a 3 × 3 orthogonal matrix $\boldsymbol{S}$ and a 3 × 3 diagonal matrix $\boldsymbol{D}$ such that $\boldsymbol{A = SDS^T}$.

13 a) A skier is on a mountain with equation $z = 100 - 0.4x^2 - 0.3y^2$, where z denotes height. (8)

i. The skier is located at the point with xy-coordinates **(1 , 1)**, and wants to ski downhill along the steepest possible path. In which direction (indicated by a vector **(a , b**) in the xy-plane) should the skier begin skiing.

ii. The skier begins skiing in the direction given by the xy-vector (**a , b)** you found in part (i), so the skier heads in a direction in space given by the vector **(a , b , c**). Find the value of **c**.

b) Find the linear approximation to the function $f(x,y) = 2 - \sin(-x - 3y)$ at the point **(0 , π)**, and then use your answer to estimate $f(0.001 , \pi)$. (6)

OR

14 a) Let ***g*** be the function given by (8)

$$g(x,y) = \begin{cases} \dfrac{x^2 y}{x^2+y^2} & \text{if } (x,y) \neq (0,0); \\ 0 & \text{if } (x,y) = (0,0). \end{cases}$$

i. Calculate the partial derivatives of ***g*** at **(0 , 0)**.

ii. Show that ***g*** is not differentiable at **(0 , 0)**.

b) Find the second order Taylor series expansion for ***f(x,y) = e***$^{-(x2+y2)}$ ***cos(xy)*** about **(0 , 0)**. (6)

15 a) There are two bags. The first bag contains four mangos and two apples; the second bag contains four mangos and four apples. We also have a biased coin, which shows "heads" with probability 0.6 and "tails" with probability 0.4. If the coin shows "heads". we pick a fruit at random from bag 1; otherwise we pick a fruit at random from bag 2. Your friend flips the coin (you cannot see the result), picks a fruit at random from the corresponding bag, and presents you a mango. (6)

What is the probability that the mango was picked from bag 2?

b) Suppose that one has written a computer program that sometimes compiles and sometimes not (code does not change). You decide to model the apparent stochasticity (success vs. no success) ***x*** of the compiler using a Bernoulli distribution with parameter μ: (8)

$$p(x \mid \mu) = \mu^x (1-\mu)^{1-x}, \quad x \in \{0,1\}$$

Choose a conjugate prior for the Bernoulli likelihood and compute the posterior distribution ***p(μ | x***$_1$ ***, ... , x***$_N$***)***.

OR

16 a) Consider a mixture of two Gaussian distributions (8)

$$0.4\mathcal{N}\left(\begin{bmatrix}10\\2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right) + 0.6\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}8.4 & 2.0\\2.0 & 1.7\end{bmatrix}\right)$$

i. Compute the marginal distributions for each dimension.

ii. Compute the mean, mode and median for each marginal distribution.

iii. Compute the mean and mode for the two-dimensional distribution.

b) Express the Binomial distribution as an exponential family distribution. Also express the Beta distribution is an exponential family distribution. Show that the product of the Beta and the Binomial distribution is also a member of the exponential family. (6)

17 a) Find the extrema of $f(x,y,z) = x - y + z$ subject to $g(x,y,z) = x^2 + y^2 + z^2 = 2$. (8)

b) Let

$$P = \begin{bmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}, \quad q = \begin{bmatrix} -22.0 \\ -14.5 \\ 13.0 \end{bmatrix}, \quad \text{and} \quad r = 1.$$

Show that $x^* = (1, 1/2, -1)$ is optimal for the optimization problem

$$\begin{array}{ll} \text{min} & \frac{1}{2}x^{\mathsf{T}}Px + q^{\mathsf{T}}x + r \\ \text{s.t.} & -1 \leq x_i \leq 1, \; i = 1, 2, 3. \end{array}$$

(6)

OR

18 a) Derive the gradient descent training rule assuming that the target function is represented as $o_d = w_0 + w_1x_1 + ... + w_nx_n$. Define explicitly the cost/ error function E, assuming that a set of training examples D is provided, where each training example $d \in D$ is associated with the target output t_d. (8)

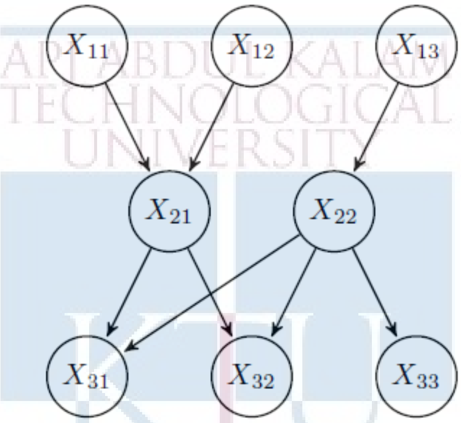
b) Find the maximum value of ***f(x,y,z) = xyz*** given that ***g(x,y,z) = x + y + z = 3*** and ***x,y,z >= 0***. (6)

19 a) Consider the following probability distribution (7)

$$P_\theta(x) = 2\theta x e^{-\theta x^2}$$

where $\boldsymbol{\theta}$ is a parameter and $\boldsymbol{x}$ is a positive real number. Suppose you get $\boldsymbol{m}$ i.i.d. samples $\boldsymbol{x_i}$ drawn from this distribution. Compute the maximum likelihood estimator for $\boldsymbol{\theta}$ based on these samples.

b) Consider the following Bayesian network with boolean variables. (7)



i. List variable(s) conditionally independent of X_{33} given X_{11} and X_{12}

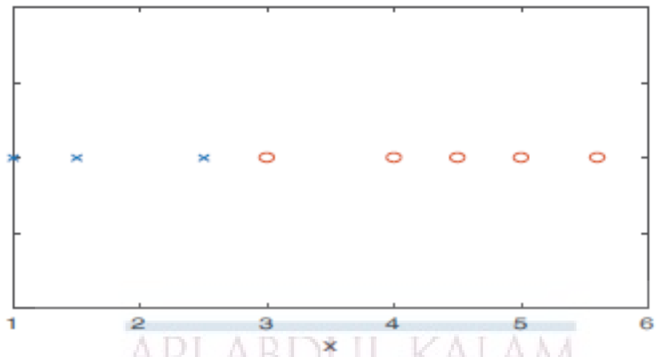
ii. List variable(s) conditionally independent of X_{33} and X_{22}

iii. Write the joint probability $P(X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{31}, X_{32}, X_{33})$ factored according to the Bayes net. How many parameters are necessary to define the conditional probability distributions for this Bayesian network?

iv. Write an expression for $P(X_{13} = 0, X_{22} = 1, X_{33} = 0)$ in terms of the conditional probability distributions given in your answer to part (iii). Justify your answer.

OR

20 a) Consider the following one dimensional training data set, ’x’ denotes negative examples and ’o’ positive examples. The exact data points and their labels are given in the table below. Suppose a SVM is used to classify this data. (6)



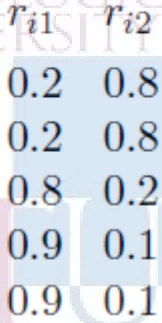
x	1	1.5	2.5	3	4	4.5	5	5.6
y	-1	-1	-1	1	1	1	1	1

i. Indicate which are the support vectors and mark the decision boundary.

ii. Give the value of the cost function and the model parameter after training.

b) Suppose that we are fitting a Gaussian mixture model for data items consisting of a single real value, ***x***, using ***K* = *2*** components. We have ***N* = *5*** training cases, in which the values of ***x*** are as **5, 15, 25, 30, 40**. Using the EM algorithm to find the maximum likeihood estimates for the model parameters, what are the mixing proportions for the two components, π_1 and π_2, and the means for the two components, μ_1 and μ_2. The standard deviations for the two components are fixed at 10. (8)

Suppose that at some point in the EM algorithm, the **E** step found that the responsibilities of the two components for the five data items were as follows:



r_{i1}	r_{i2}
0.2	0.8
0.2	0.8
0.8	0.2
0.9	0.1
0.9	0.1

What values for the parameters π_1, π_2 , μ_1, and μ_2 will be found in the next **M** step of the algorithm?

****

Teaching Plan

No	Topic	No. of Lectures (45)
1	**Module-I (LINEAR ALGEBRA)**	**8**
1.	Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence.	1
2.	Vector Spaces - Basis and Rank	1
3.	Linear Mappings	1
4.	Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Orthonormal Basis, Orthogonal Complement	1
5.	Orthogonal Projections, Matrix Decompositions, Determinant and Trace.	1
6.	Eigenvalues and Eigenvectors	1
7.	Cholesky Decomposition, Eigen decomposition and Diagonalization	1
8.	Singular Value Decomposition - Matrix Approximation	1
	Module-II (VECTOR CALCULUS)	**6**
1	Differentiation of Univariate Functions, Partial Differentiation and Gradients	1
2	Gradients of Vector Valued Functions, Gradients of Matrices	1
3	Useful Identities for Computing Gradients	1
4	Backpropagation and Automatic Differentiation	1
5	Higher Order Derivatives	1
6	Linearization and Multivariate Taylor Series	1
3	**Module-III (Probability and Distributions)**	**10**
1	Construction of a Probability Space - Discrete and Continuous Probabilities (Lecture 1)	1

2	Construction of a Probability Space - Discrete and Continuous Probabilities (Lecture 2)	1
3	Sum Rule, Product Rule	1
4	Bayes' Theorem	1
5	Summary Statistics and Independence	1
6	Important probability Distributions (Lecture 1)	1
7	Important probability Distributions (Lecture 2)	1
8	Conjugacy and the Exponential Family (Lecture 1)	1
9	Conjugacy and the Exponential Family (Lecture 2)	1
10	Change of Variables/Inverse Transform	1
4	**Module-IV (Optimization)**	**7**
1	Optimization Using Gradient Descent.	1
2	Gradient Descent With Momentum, Stochastic Gradient Descent	1
3	Constrained Optimization and Lagrange Multipliers (Lecture 1)	1
4	Constrained Optimization and Lagrange Multipliers (Lecture 2)	1
5	Convex Optimization	1
6.	Linear Programming	1
7.	Quadratic Programming	1
5	**Module-V (CENTRAL MACHINE LEARNING PROBLEMS)**	**14**
1.	Data and Learning models - Empirical Risk Minimization,	1
2.	Parameter Estimation	1
3.	Directed Graphical Models	1
4.	Linear Regression	1
5.	Bayesian Linear Regression	1
6.	Maximum Likelihood as Orthogonal Projection	1
7.	Dimensionality Reduction with Principal Component Analysis - Maximum Variance Perspective, Projection Perspective.	1
8.	Eigenvector Computation and Low Rank Approximations	1
9.	Density Estimation with Gaussian Mixture Models	1

10.	Parameter Learning via Maximum Likelihood	1
11.	EM Algorithm	1
12.	Classification with Support Vector Machines - Separating Hyperplanes	1
13.	Primal Support Vector Machines, Dual Support Vector Machines	1
14.	Kernels	1
*Assignments may include applications of the above theory. With respect to module V, programming assignments may be given.		

CST 296	Principles of Program Analysis and Verification	Category	L	T	P	CREDIT	YEAR OF INTRODUCTION
		HONOURS	3	1	0	4	2019

Preamble: This is the foundational course for awarding B. Tech. Honours in Computer Science and Engineering with specialization in ***Formal Methods***. Program Analysis and Program Verification are two important areas of study, discussing Methods, Technologies and Tools to ensure reliability and correctness of software systems. The syllabus for this course is prepared with the view of introducing the Foundational Concepts, Methods and Tools in Program Analysis and Program Verification.

Prerequisite: Topics covered in the course Discrete Mathematical Structures (MAT 203).

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain the concepts and results about Lattices, Chains, Fixed Points, Galois Connections, Monotone and Distributive Frameworks, Hoare Triples, Weakest Preconditions, Loop Invariants and Verification Conditions to perform Analysis and Verification of programs **(Cognitive knowledge level: Understand)**
CO2	Illustrate methods for doing intraprocedural/interprocedural Data flow Analysis for a given Program Analysis problem **(Cognitive knowledge level: Analyse)**
CO3	Formulate an Abstract Interpretation framework for a given Data flow Analysis problem and perform the analysis using the tool WALA **(Cognitive knowledge level: Analyse)**
CO4	Use Kildall's Algorithm to perform Abstract Interpretation of Programs and compare the results obtained by the Algorithm on Monotone and Distributive Frameworks **(Cognitive knowledge level: Apply)**
CO5	Explain the concept of Loop Invariants and use them in Hoare Triple based Weakest Precondition analysis to verify the total correctness of a code segment **(Cognitive knowledge level: Apply)**
CO6	Use the tool VCC to specify and verify the correctness of a C Program with respect to a given set of properties **(Cognitive knowledge level: Analyse)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓				✓		✓
CO2	✓	✓	✓	✓		✓				✓		✓
CO3	✓	✓	✓	✓	✓	✓						✓
CO4	✓	✓	✓	✓		✓						✓
CO5	✓	✓	✓	✓		✓				✓		✓
CO6	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern:

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Percentage)	Test 2 (Percentage)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : 10 Marks

Continuous Assessment Tests : 25 Marks

Assignment : 15 Marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions from Part A. Part B contains 2 questions from each module of which a student should answer any one, each question carries 14 marks. Each question in part B can have a maximum 2 sub-divisions.

SYLLABUS

Module 1

Mathematical Foundations – Partially Ordered Set, Complete Lattice, Construction of Complete Lattices, Chains, Fixed Points, Knaster-Tarski Fixed Point Theorem.

Module 2

Introduction to Program Analysis – The WHILE language, Reaching Definition Analysis, Data Flow Analysis, Abstract Interpretation, Algorithm to find the least solutions for the Data Flow Analysis problem.

Module 3

Intraprocedural DataFlow Analysis – Available Expressions Analysis, Reaching Definitions Analysis, Very Busy Expressions Analysis, Live Variable Analysis, Derived Data Flow Information, Monotone and Distributive Frameworks, Equation Solving - Maximal Fixed Point (MFP) and Meet Over all Paths (MOP) solutions.

Interprocedural Data Flow Analysis - Structural Operational Semantics, Intraprocedural versus Interprocedural Analysis, Making Context Explicit, Call Strings as Context, Flow Sensitivity versus Flow Insensitivity, Implementing Interprocedural Data-flow Analysis using the Tool WALA.

Module 4

Abstract Interpretation - A Mundane Approach to Correctness, Approximations of Fixed Points, Galois Connections, Systematic Design of Galois Connections, Induced Operations, Kildall's Algorithm for Abstract Interpretation.

Module 5

Program Verification - Why should we Specify and Verify Code, A framework for software verification - A core programming Language, Hoare Triples, Partial and Total Correctness, Program Variables and Logical Variables, Proof Calculus for Partial Correctness, Loop Invariants, Verifying code using the tool VCC (Verifier for Concurrent C).

Text Books

1. Flemming Nielson, Henne Nielson and Chris Kankin, Principles of Program Analysis, Springer (1998).
2. Michael Hutch and Mark Ryan, Logic in Computer Science - Modeling and Reasoning about Systems, Cambridge University Press, Second Edition.

References

1. Julian Dolby and Manu Sridharan, Core WALA Tutorial (PLDI 2010), available online at http://wala.sourceforge.net/files/PLDI_WALA_Tutorial.pdf

2. Ernie & Hillebrand, Mark & Tobies, Stephan (2012), Verifying C Programs: A VCC Tutorial.

Sample Course Level Assessment Questions

Course Outcome1 (CO1):

1. Find a lattice to represent the data states of a given program and propose a sound abstract interpretation framework to do a given analysis on the program.
2. When is an abstract interpretation framework said to be sound? Illustrate with an example.
3. When is an abstract interpretation framework said to be precise? Illustrate with an example.

Course Outcome2 (CO2):

1. Illustrate how one can do Intraprocedural Available Expression Analysis on a program.
2. Illustrate how one can do Intraprocedural Reaching Definition Analysis on a program.
3. Illustrate how one can do Intraprocedural Live Variable Analysis on a program.

Course Outcome3 (CO3):

1. Illustrate how one can do Interprocedural Data Flow Analysis using the tool WALA.

Course Outcome4 (CO4):

1. Illustrate the working of Kildall's algorithm to do Intraprocedural Available Expression Analysis on a program.
2. Compare the results obtained by applying Kildall's algorithms for Abstract Interpretation in Monotone and Distributive Frameworks.

Course Outcome5 (CO5):

1. Illustrate the process of obtaining verification conditions (VCs) using weakest precondition analysis.
2. Explain the concepts of partials and total correctness of programs.
3. Explain the necessity of obtaining loop invariants in verifying the total correctness of a program.

Course Outcome6 (CO6):

1. Using the tool VCC prove that a given code segment satisfies a given property.

Model Question paper

QP CODE: **PAGES:3**

Reg No:______________ **Name :______________**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

4th SEMESTER B.TECH DEGREE (HONOURS) EXAMINATION, MONTH & YEAR

Course Code: CST 296

Course Name: Principles of Program Analysis and Verification

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. What is a complete lattice? Give an example of a complete lattice.
2. Show that every chain is a lattice.
3. Write a program in *while* language to find the factorial of a number. Explain the statements of your program.
4. Consider a program that calculates x^y through repeated multiplications. Draw the flow graph of the program.
5. What is Available Expression (AE) analysis? Give an application for AE analysis.
6. What is Live variable (LV) analysis? Give an application for LV analysis.
7. Let P be a program analysis problem (like LV, AE etc.) and (A, F_A, γ_{AC}) and (B, F_B, γ_{BC}) be two abstract interpretations such that B is more abstract than A. Let α and γ be the abstraction and concretization functions between A and B. Then, what are the conditions required for α and γ to form a Galois Connection?
8. When is Kildall's algorithm for abstract interpretation guaranteed to terminate? Justify your answer.
9. Is it possible to verify total correctness of a program using Hoare Logic? If yes, how is it possible?
10. Define *loop invariant*. Show a simple loop with a *loop invariant*.

PART B

Answer any one Question from each module. Each question carries 14 Marks

11.

a. What is an infinite ascending chain in a lattice? Show an example lattice with an infinite ascending chain. Is it possible for a complete lattice to contain an infinite ascending chain? **(7 marks)**

b. State and prove Knaster-Tarski fixed point theorem. **(7 marks)**

OR

12.

a. Consider the lattice $(\mathbb{N}, \leq)$. Let $f : \mathbb{N} \rightarrow \mathbb{N}$, be a function defined as follows: when $x < 100, f(x) = x + 1$, when $x > 100, f(x) = x - 1$, otherwise $f(x) = x$. Then, show the following for f: (i) the set of all fixpoints, (ii) the set of all pre-fixpoints and (iii) the set of all post-fixpoints. **(7 marks)**

b. Let $(D, \leq)$ be a lattice with a least upper bound for each subset of D. Then, prove that every subset of D has a greatest lower bound. **(7 marks)**

13.

a. With a suitable example, explain the equational approach in Data Flow Analysis. **(7 marks)**

b. With a suitable example, explain how you obtain the collecting semantics of a program point. **(7 marks)**

OR

14.

a. With an example, explain the Constrained Based Approach in Data Flow Analysis. **(7 marks)**

b. Discuss the properties of an algorithm to solve the problem of computing the least solution to the program analysis problems in Data Flow Analysis. **(7 marks)**

15.

a. Using Intraprocedural Reaching Definition Analysis, find the assignments killed and generated by each of the blocks in the program

```
[x:=5]^1;
[y:=1]^2;
while [x>1]^3 do
        ([y:=x*y]^4; [x:=x-1]^5)
```

(7 marks)

b. Analyse the following program using Intraprocedural Very Busy Expression analysis

```
if [a>b]$^1$ then
       ([x: =b-a]$^2$ ; [y: =a-b]$^3$)
else
       ([y: =b-a]$^4$; [x: =a-b]$^5$)
```

(7 marks)

OR

16.

a. Find Maximal Fixed Point (MFP) solution for the program

```
[x: =a+b]$^1$;
[y: =a*b]$^2$ ;
while [y>a+b]$^3$ do
       ([a: =a+1]$^4$; [x: =a+b]$^5$)
```

(7 marks)

b. With examples, explain the difference between flow sensitive and flow insensitive analysis. **(7 marks)**

17.

a. Prove that (L, α, γ, M) is an adjunction if and only if (L, α, γ, M) is a Galois connection. **(7 marks)**

b. Prove that if $\alpha : L \to M$ is completely additive then there exists $\gamma : M \to L$ such that (L, α, γ, M) is a Galois connection. Similarly, if $\gamma : M \to L$ is completely multiplicative then there exists $\alpha : L \to M$ such that (L, α, γ, M) is a Galois connection. **(7 marks)**

OR

18.

a. Show that if $(L_i, \alpha_i, \gamma_i, M_i)$ are Galois connections and $\beta_i : V_i \rightarrow L_i$ are representation functions then

$$((\alpha_1 \circ \beta_1) \twoheadrightarrow (\alpha_2 \circ \beta_2))(\to) = \alpha_2 \circ ((\beta_1 \twoheadrightarrow \beta_2)(\to)) \circ \gamma_1$$

(7 marks)

b. Briefly explain Kildall's algorithm for abstract interpretation **(7 marks)**

19.

a. Briefly explain the need of specification and verification of code. **(7 marks)**

b. Argue that Hoare Logic is sound. When Hoare Logic is complete? Let {A}P{B} be a Hoare triple such that Hoare Logic is complete for the program P. Then, is it always possible to check the validity of the Hoare Triple? If not, what is the difficulty? **(7 marks)**

OR

20.

a. With suitable examples, show the difference between partial and total correctness. **(7 marks)**

b. With a suitable example, show how a basic program segment can be verified using the tool VCC. **(7 marks)**

Teaching Plan

Module 1 (Mathematical Foundations)		**6 Hours**
1.1	Partially Ordered Set	1 Hour
1.2	Complete Lattice, Construction of Complete Lattices	1 Hour
1.3	Chains	1 Hour
1.4	Fixed Points	1 Hour
1.5	Knaster-Tarski Fixed Point Theorem	1 Hour
1.6	Proof of Knaster-Tarski Fixed Point Theorem	1 Hour
Module 2 (Introduction to Program Analysis)		**5 Hours**
2.1	The WHILE language	1 Hour
2.2	Data Flow Analysis	1 Hour
2.3	Reaching Definition Analysis	1 Hour
2.4	Abstract Interpretation	1 Hour
2.5	Algorithm to find the least solutions for the Data Flow Analysis problem	1 Hour
Module 3 (Data flow Analysis)		**15 Hours**
3.1	Available Expressions Analysis, Reaching Definitions Analysis	1 Hour
3.2	Very Busy Expressions Analysis	1 Hour
3.3	Live Variable Analysis	1 Hour
3.4	Derived Data Flow Information	1 Hour
3.5	Monotone and Distributive Frameworks	1 Hour
3.6	Equation Solving - MFP Solution	1 Hour

3.7	Equation Solving - MOP Solution	1 Hour
3.8	Structural Operational Semantics (Lecture 1)	1 Hour
3.9	Structural Operational Semantics (Lecture 2)	1 Hour
3.10	Intraprocedural versus Interprocedural Analysis	1 Hour
3.11	Making Context Explicit	1 Hour
3.12	Call Strings as Context	1 Hour
3.13	Flow Sensitivity versus Flow Insensitivity	1 Hour
3.14	Implementing Interprocedural Dataflow Analysis using the Tool WALA (Lecture 1)	1 Hour
3.15	Implementing Interprocedural Dataflow Analysis using the Tool WALA (Lecture 2)	1 Hour
Module 4 (Abstract Interpretation)		**8 Hours**
4.1	A Mundane Approach to Correctness	1 Hour
4.2	Approximations of Fixed Points	1 Hour
4.3	Galois Connections,	1 Hour
4.4	Systematic Design of Galois Connections (Lecture 1)	1 Hour
4.5	Systematic Design of Galois Connections (Lecture 2)	1 Hour
4.6	Induced Operations	1 Hour
4.7	Kildall's Algorithm for Abstract Interpretation (Lecture 1)	1 Hour
4.8	Kildall's Algorithm for Abstract Interpretation (Lecture 2)	1 Hour
Module 5 (Program Verification)		**11 Hours**
5.1	Why should we Specify and Verify Code	1 Hour
5.2	A framework for software verification - A core programming Language	1 Hour

5.3	Hoare Triples (Lecture 1)	1 Hour
5.4	Hoare Triples (Lecture 2)	1 Hour
5.5	Partial and Total Correctness	1 Hour
5.6	Program Variables and Logical Variables	1 Hour
5.7	Proof Calculus for Partial Correctness	1 Hour
5.8	Loop Invariants	1 Hour
5.9	Verifying C programs using the tool VCC (Lecture 1)	1 Hour
5.10	Verifying C programs using the tool VCC (Lecture 2)	1 Hour
5.11	Verifying C programs using the tool VCC (Lecture 3)	1 Hour

B. Tech Computer Science and Engineering

SEMESTER V

SEMESTER VI

SEMESTER V

CST 301	FORMAL LANGUAGES AND AUTOMATA THEORY	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble: This is a core course in theoretical computer science. It covers automata and grammar representations for languages in Chomsky Hierarchy. For regular languages, it also covers representations using regular expression and Myhill-Nerode Relation. The topics covered in this course have applications in various domains including compiler design, decidability and complexity theory, software testing, formal modelling and verification of hardware and software.

Prerequisite: Basic knowledge about the following topic is assumed: sets, relations - equivalence relations, functions, proof by Principle of Mathematical Induction.

Course Outcomes: After the completion of the course the student will be able to

CO1	Classify a given formal language into Regular, Context-Free, Context Sensitive, Recursive or Recursively Enumerable. **[Cognitive knowledge level: Understand]**
CO2	Explain a formal representation of a given regular language as a finite state automaton, regular grammar, regular expression and Myhill-Nerode relation. [**Cognitive knowledge level: Understand**]
CO3	Design a Pushdown Automaton and a Context-Free Grammar for a given context-free language. [**Cognitive knowledge level** : **Apply**]
CO4	Design Turing machines as language acceptors or transducers. [**Cognitive knowledge level: Apply]**
CO5	Explain the notion of decidability. [**Cognitive knowledge level: Understand]**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓

CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test 1 (Marks)	**Test 2 (Marks)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment - Test	: **25 marks**
Continuous Assessment - Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

CST 301 Formal Languages and Automata Theory

Module - 1 (Introduction to Formal Language Theory and Regular Languages)

Introduction to formal language theory– Alphabets, Strings, Concatenation of strings, Languages.

Regular Languages - Deterministic Finite State Automata (DFA) (Proof of correctness of construction not required), Nondeterministic Finite State Automata (NFA), Equivalence of DFA and NFA**,** Regular Grammar (RG), Equivalence of RGs and DFA.

Module - 2 (More on Regular Languages)

Regular Expression (RE), Equivalence of REs and DFA, Homomorphisms, Necessary conditions for regular languages, Closure Properties of Regular Languages, DFA state minimization (No proof required).

Module - 3 (Myhill-Nerode Relations and Context Free Grammars)

Myhill-Nerode Relations (MNR)- MNR for regular languages, Myhill-Nerode Theorem (MNT) (No proof required), Applications of MNT.

Context Free Grammar (CFG)- CFG representation of Context Free Languages (proof of correctness is required), derivation trees and ambiguity, Normal forms for CFGs.

Module - 4 (More on Context-Free Languages)

Nondeterministic Pushdown Automata (PDA), Deterministic Pushdown Automata (DPDA), Equivalence of PDAs and CFGs (Proof not required), Pumping Lemma for Context-Free Languages (Proof not required), Closure Properties of Context Free Languages.

Module - 5 (Context Sensitive Languages, Turing Machines)

Context Sensitive Languages **-** Context Sensitive Grammar (CSG), Linear Bounded Automata.

Turing Machines - Standard Turing Machine, Robustness of Turing Machine, Universal Turing Machine, Halting Problem, Recursive and Recursively Enumerable Languages.

Chomsky classification of formal languages.

Text Book

1. Dexter C. Kozen, Automata and Computability, Springer (1999)

Reference Materials

1. John E Hopcroft, Rajeev Motwani and Jeffrey D Ullman, Introduction to Automata Theory, Languages, and Computation, 3/e, Pearson Education, 2007
2. Michael Sipser, Introduction To Theory of Computation, Cengage Publishers, 2013.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Identify the class of the following languages in Chomsky Hierarchy:

- $L_1 = \{a^p | pis\ a\ prime\ number\}$
- $L_2 =$

$\{x\{0,1\}^* | xis\ the\ binary\ representation\ of\ a\ decimal\ number\ which\ is\ a\ multiple\ of 5\}$

- $L_3 = \{a^n b^n c^n | n \geq 0\}$
- $L_4 = \{a^m b^n c^{m+n} | m > 0, n \geq 0\}$
- $L_5 = \{M\#x | Mhalts\ onx\}$. Here, M is a binary encoding of a Turing Machine and x is a binary input to the Turing Machine.

Course Outcome 2 (CO2):

(i) Design a DFA for the language $L = \{axb | x \in \{a, b\}^*\}$

(ii) Write a Regular Expression for the language:$L = \{x \in \{a, b\}^* | third\ last\ symbol\ in\ x\ is\ b\}$

(iii) Write a Regular Grammar for the language:$L = \{x \in \{0,1\}^* | there\ are\ no\ consecutive\ zeros\ inx\}$

(iv) Show the equivalence classes of the canonical Myhill-Nerode relation induced by the language: $L = \{x \in \{a, b\}^* | xcontains\ even\ number\ of a's\ and\ odd\ number\ of b's\}$.

Course Outcome 3 (CO3):

(i) Design a PDA for the language$L = \{ww^R | w \in \{a, b\}^*\}$. Here, the notation w^R represents the reverse of the string w.

(ii) Write a Context-Free Grammar for the language $L = \{a^n b^{2n} | n \geq 0\}$.

Course Outcome 4 (CO4):

(i) Design a Turing Machine for the language $L = \{a^n b^n c^n | n \geq 0\}$

(ii) Design a Turing Machine to compute the square of a natural number. Assume that the input is provided in unary representation.

Course Outcome 5 (CO5): Argue that it is undecidable to check whether a Turing Machine Menters a given state during the computation of a given input x.

Model Question paper

QP CODE: **PAGES:3**

Reg No:______________ **Name :______________**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST301

Course Name: Formal Languages and Automata Theory

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Design a DFA for the language $L = \{x \in \{a,b\}^* | aba\ is\ not\ a\ substring\ in\ x\}$.

2. Write a Regular Grammar for the language: $L = \{axb | x \in \{a,b\}^*\}$

3. Write a Regular Expression for the language:
$L = \{x \in \{0,1\}^* | there\ are\ no\ consecutive\ 1's\ in\ x\}$

4. Prove that the language $L_1 = \{a^{n!} | n \in N\}$ is not regular.

5. List out the applications of Myhill-Nerode Theorem.

6. Write a Context-Free Grammar for the language: $L = \{x \in \{a,b\}^* | \#_a(x) = \#_b(x)\}$. Here, the notation $\#_1(w)$ represents the number of occurrences of the symbol 1 in the string w.

7. Design a PDA for the language of odd length binary palindromes (no explanation is required, just list the transitions in the PDA).

8. Prove that Context Free Languages are closed under set union.

9. Write a Context Sensitive Grammar for the language $L = \{a^n b^n c^n | n \geq 0\}$ (no explanation is required, just write the set of productions in the grammar).

10. Differentiate between Recursive and Recursively Enumerable Languages.

(10x3=30)

Part B
(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Draw the state-transition diagram showing an NFA N for the following language L. Obtain the DFAD equivalent to N by applying the subset construction algorithm. **(7)**

$$L = \{x \in \{a, b\}^* | the\ second\ last\ symbol\ in\ x\ is\ b\}$$

(b) Draw the state-transition diagram showing a DFA for recognizing the following language: **(7)**

$L = \{x \in \{0,1\}^* | x\ is\ a\ binary\ representation\ of\ a\ natural$

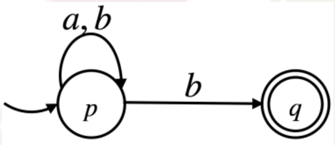
$\imath umber\ which\ is\ a$multiple of 5}

OR

12. (a) Write a Regular grammar G for the following language Ldefined as: $L = \{x \in \{a, b\}^* | xdoes\ not\ conatin\ consecutiveb's\}$. **(7)**

(b) Obtain the DFA A_G over the alphabet set $\Sigma = \{a, b\}$, equivalent to the regular grammar G with start symbol S and productions: $S \rightarrow aA$ and $A \rightarrow aA|bA|b$. **(7)**

13. (a) Using Kleen's construction, obtain the regular expression for the language represented by the following NFA

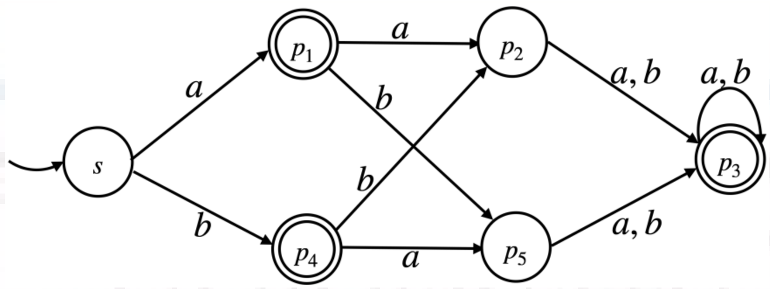


(8)

(b) Using pumping lemma for regular languages, prove that the language $L = \{a^n b^n | n \geq 0\}$ is not regular. **(7)**

OR

14. (a) Obtain the minimum-state DFA from the following DFA. **(8)**



(b) Using ultimate periodicity for regular languages, prove that the language $L = \{a^{n^2} | n \geq 0\}$ is not regular. **(6)**

15. (a) Show the equivalence classes of the canonical Myhill-Nerode relation for the language of binary strings with odd number of $1's$ and even number of 0s. **(7)**

(b) With an example, explain ambiguity in Context Free Grammar **(7)**

OR

16. (a) Convert the Context-Free Grammar with productions: $\{S \rightarrow aSb|\epsilon\}$ into Greibach Normal form. **(8)**

(b) Convert the Context-Free Grammar with productions: $\{S \rightarrow aSa|bSb|SS|\epsilon\}$ into Chomsky Normal form. **(6)**

17. (a) Design a PDA for the language $L = \{a^m b^n c^{m+n} | n \geq 0, m \geq 0\}$. Also illustrate the computation of the PDA on a string in the language **(7)**

(b) With an example illustrate how a multi-state PDA can be transformed into an equivalent single-state PDA. **(7)**

OR

18. (a) Using pumping lemma for context-free languages, prove that the language: $L = \{ww | w \in \{a, b\}^*\}$ is not a context-free language. **(6)**

(b) With an example illustrate how a CFG can be converted to a single-state PDA **(8)**

19. (a) Design a Turing machine to obtain the sum of two natural numbers a and b, both represented in unary on the alphabet set $\{1\}$. Assume that initially the tape contains $\vdash 1^a 0 1^b\ \not{b}^{\omega}$. The Turing Machine should halt with $\vdash 1^{a+b}\ \not{b}^{\omega}$ as the tape content. Also, illustrate the computation of your Turing Machine on the input $a = 3$and $b = 2$. **(7)**

(b) With an example illustrate how a CFG can be converted to a single-state PDA. **(7)**

OR

20. (a) Design a Turing machine to obtain the sum of two natural numbers $a$and b, both represented in unary on the alphabet set $\{1\}$. Assume that initially the tape contains $\vdash 1^a 0 1^b\ \not{b}^{\omega}$. The Turing Machine should halt with $\vdash 1^{a+b}\ \not{b}^{\omega}$ as the tape content. Also, illustrate the computation of your Turing Machine on the input $a = 3$and $b = 2$. **(7)**

(b) Write a context sensitive grammar for the language $L = \{a^n b^n c^n | n \geq 0\}$. Also illustrate how the the string $a^2 b^2 c^2$ can be derived from the start symbol of the proposed grammar. **(7)**

Teaching Plan

Sl. No	Topic	No. of Hours (45 hrs)
Module - 1 (Introduction to Formal Language Theory and Regular Languages)		**9 Hours**
1.1	Introduction to formal language theory – Alphabets, strings, concatenation of strings, Languages	1 Hour
1.2	Deterministic Finite State Automata (DFA) – Example DFA (Proof of correctness of construction not required)	1 Hour
1.3	Formal definition of DFA, Language accepted by the class of DFA	1 Hour
1.4	Nondeterministic Finite State Automata (NFA) – Example NFA	1 Hour
1.5	Formal definition of NFA, NFA with □ transitions - examples, formal definition	1 Hour
1.6	Equivalence of DFA and NFA with and without □ transitions - Subset construction	1 Hour
1.7	Regular Grammar (RG) – Example RGs, derivation of sentences	1 Hour
1.8	Formal definition of RG, Language represented by a RG	1 Hour
1.9	Equivalence of RG and DFA	1 Hour
Module - 2 (More on Regular Languages)		**9 Hours**
2.1	Regular Expression (RE) - Example REs and formal definition	1 Hour
2.2	Conversion of RE to NFA with □ transition	1 Hour
2.3	Conversion of NFA with □ transition to RE (Kleen's construction)	1 Hour
2.4	Homomorphisms	1 Hour
2.5	Pumping Lemma for regular languages	1 Hour
2.6	Ultimate periodicity	1 Hour
2.7	Closure Properties of Regular Languages (proof not required)	1 Hour

2.8	DFA state minimization - Quotient construction	1 Hour
2.9	State Minimization Algorithm - Example	1 Hour
Module - 3 (Myhill-Nerode Relations and Context Free Grammars)		**10 Hours**
3.1	Myhill-Nerode Relations (MNR) - Example, Properties of MyhillNerode Relation	1 Hour
3.2	Conversion of DFA to MNR (Proof of correctness not required)	1 Hour
3.3	Conversion of MNR to DFA(Proof of correctness not required)	1 Hour
3.4	Myhill-Nerode Theorem (MNT)	1 Hour
3.5	Applications of MNT	1 Hour
3.6	Context Free Grammar (CFG) - Example CFGs and formal definition	1 Hour
3.7	Proving correctness of CFGs	1 Hour
3.8	Derivation Trees and ambiguity	1 Hour
3.9	Chomsky Normal Form	1 Hour
3.10	Greibach Normal Form	1 Hour
Module - 4 (More on Context-Free Languages)		**8 Hours**
4.1	Nondeterministic Pushdown Automata (PDA) – Example PDAs, formal definition	1 Hour
4.2	Acceptance criteria - equivalence	1 Hour
4.3	Deterministic PDA	1 Hour
4.4	Conversion of CFG to PDA (No proof required)	1 Hour
4.5	Conversion of PDA to CGF - Part I (No proof required)	1 Hour
4.6	Conversion of PDA to CGF - Part II (No proof required)	1 Hour
4.7	Pumping Lemma for context-free languages (No proof required)	1 Hour
4.8	Closure Properties of Context Free Languages	1 Hour

Module - 5 (Context Sensitive Languages, Turing Machines)		**9 Hours**
5.1	Context Sensitive Grammar (CSG) - Examples, formal definition	1 Hour
5.2	Linear Bounded Automata (LBA) - Example LBA, formal definition	1 Hour
5.3	Turing Machine (TM) - TM as language acceptors - examples, formal definition	1 Hour
5.4	TM as transducers - examples	1 Hour
5.5	Robustness of the standard TM model - Multi-tape TMs, Nondeterministic TM	1 Hour
5.6	Universal Turing Machine	1 Hour
5.7	Halting Problem of TM - proof of its undecidability	1 Hour
5.8	Recursive and Recursively Enumerable Languages	1 Hour
5.9	Chomsky classification of formal languages	1 Hour

CST 303	COMPUTER NETWORKS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble: Study of this course provides the learners a clear understanding of how computer networks from local area networks to the massive and global Internet are built, how they allow computers to share information and communicate with one another. This course covers the physical aspects of computer networks, layers of OSI Reference model, and inter-networking. The course helps the learners to compare and analyze the existing network technologies and choose a suitable network design for a given system.

Prerequisite: Nil

Course Outcomes: After the completion of the course, the student will be able to

CO#	Course Outcomes
CO1	Explain the features of computer networks, protocols, and network design models **(Cognitive Knowledge: Understand)**
CO2	Describe the fundamental characteristics of the physical layer and identify the usage in network communication **(Cognitive Knowledge: Apply)**
CO3	Explain the design issues of data link layer, link layer protocols, bridges and switches **(Cognitive Knowledge: Understand)**
CO4	Illustrate wired LAN protocols (IEEE 802.3) and wireless LAN protocols (IEEE 802.11) **(Cognitive Knowledge: Understand)**
CO5	Select appropriate routing algorithms, congestion control techniques, and Quality of Service requirements for a network **(Cognitive Knowledge: Apply)**
CO6	Illustrate the functions and protocols of the network layer, transport layer, and application layer in inter-networking **(Cognitive Knowledge: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓	✓								✓
CO6	✓	✓	✓			✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and teamwork
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom’s Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination (Marks in percentage)
Remember	40	30	30

Understand	**50**	**50**	**50**
Apply	**10**	**20**	**20**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment Test	: **25 marks**
Continuous Assessment Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus. The second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction and Physical Layer)

Introduction – Uses of computer networks, Network hardware, Network software. Reference models – The OSI reference model, The TCP/IP reference model, Comparison of OSI and TCP/IP reference models.

Physical Layer – Modes of communication, Physical topologies, Signal encoding, Repeaters and hub, Transmission media overview. Performance indicators – Bandwidth, Throughput, Latency, Queuing time, Bandwidth–Delay product.

Module - 2 (Data Link Layer)

Data link layer - Data link layer design issues, Error detection and correction, Sliding window protocols, High-Level Data Link Control(HDLC)protocol. Medium Access Control (MAC) sublayer –Channel allocation problem, Multiple access protocols, Ethernet, Wireless LANs - 802.11, Bridges & switches - Bridges from 802.x to 802.y, Repeaters, Hubs, Bridges, Switches, Routers and Gateways.

Module - 3 (Network Layer)

Network layer design issues. Routing algorithms - The Optimality Principle, Shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Multicast routing, Routing for mobile hosts. Congestion control algorithms. Quality of Service (QoS) - requirements, Techniques for achieving good QoS.

Module - 4 (Network Layer in the Internet)

IP protocol, IP addresses, Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP). Open Shortest Path First(OSPF) Protocol, Border Gateway Protocol (BGP), Internet multicasting, IPv6, ICMPv6.

Module – 5 (Transport Layer and Application Layer)

Transport service – Services provided to the upper layers, Transport service primitives. User Datagram Protocol (UDP). Transmission Control Protocol (TCP) – Overview of TCP, TCP segment header, Connection establishment &release, Connection management modeling, TCP retransmission policy, TCP congestion control.

Application Layer –File Transfer Protocol (FTP), Domain Name System (DNS), Electronic mail, Multipurpose Internet Mail Extension (MIME), Simple Network Management Protocol

(SNMP), World Wide Web(WWW) – Architectural overview.

Text Books

1. Andrew S. Tanenbaum, Computer Networks, 4/e, PHI (Prentice Hall India).
2. Behrouz A Forouzan, Data Communication and Networking, 4/e, Tata McGraw Hill

Reference Books

1. Larry L Peterson and Bruce S Dave, Computer Networks – A Systems Approach, 5/e, Morgan Kaufmann.
2. Fred Halsall, Computer Networking and the Internet, 5/e.
3. James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 6/e.
4. Keshav, An Engineering Approach to Computer Networks, Addison Wesley, 1998.
5. W. Richard Stevens. TCP/IP Illustrated Volume 1, Addison-Wesley, 2005.
6. William Stallings, Computer Networking with Internet Protocols, Prentice-Hall, 2004.
7. Request for Comments (RFC) Pages - IETF -https://www.ietf.org/rfc.html

Course Level Assessment Questions

Course Outcome1 (CO1)

1. Compare TCP/IP and OSI reference model.
2. The purpose of physical layer is to transport a raw bit stream from one machine to another. Justify.

Course Outcome2 (CO2)

1. Write the physical and transmission characteristics of Optical Fibre Cable guided transmission media.
2. The distance between the sender and receiver systems is about 200 KM. The speed of transmission is 2GB/s. Find out the propagation time?

Course Outcome3 (CO3)

1. Ethernet frames must be at least 64 bytes long to ensure that the transmitter is still going in the event of a collision at the far end of the cable. Fast Ethernet has the same 64-byte minimum frame size but can get the bits out ten times faster. How is it possible to maintain the same minimum frame size?
2. What do you mean by bit stuffing?

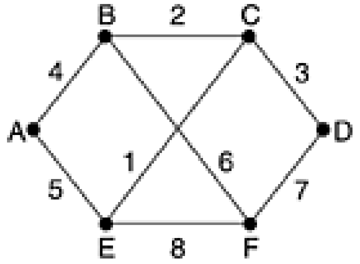
Course Outcome4 (CO4)

1. Draw and explain the frame format for Ethernet.
2. Give the differences between CSMA/CD and CSMA/CA protocol.

Course Outcome5 (CO5)

1. Consider the given subnet in which distance vector routing is used, and the vectors just come in to router C as follows: from B: (5, 0, 8, 12, 6, 2); from D: (16, 12, 6, 0, 9, 10);

and from E: (7, 6, 3, 9, 0, 4). The measured delays from C to B, D, and E, are 6, 3, and 5, respectively. What is C’s new routing table? Give both the outgoing line to use and the expected delay.



2. Illustrate the leaky bucket congestion control technique.

Course Outcome 6 (CO6)

1. How do you subnet the Class C IP Address 206.16.2.0 so as to have 30 subnets. What is the subnet mask for the maximum number of hosts? How many hosts can each subnet have?
2. Give the architecture of World Wide Web.

Model Question Paper

QP CODE: **PAGES:**

___

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 303

Course Name : Computer Networks

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. What does “negotiation" mean when discussing network protocols in a layered architecture? Give an example.

2. Define simplex, half-duplex, and full-duplex transmission modes. Give one example for each.

3. Data link protocols almost always put the CRC in a trailer rather than in a header. Why?

4. An 8-bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?

5. Illustrate the Count to Infinity problem in routing.

6. Describe two major differences between the warning bit method and the Random Early Detection (RED) method.

7. The Protocol field used in the IPv4 header is not present in the fixed IPv6 header. Why?

8. How many octets does the smallest possible IPv6 (IP version 6) datagram contain?

9. Can Transmission Control Protocol(TCP) be used directly over a network (e. g. an Ethernet) without using IP? Justify your answer.

10. When Web pages are sent out, they are prefixed by MIME headers. Why?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) With a neat diagram, explain Open Systems Interconnection (OSI) Reference Model. **(8)**

    (b) Compare Twisted Pair, Coaxial Cable and Optical Fibre guided transmission media. **(6)**

OR

12. (a) Consider two networks providing reliable connection-oriented service. One of them offers a reliable byte stream and the other offers a reliable message stream. Are they identical? Justify your answer. **(8)**

    (b) Sketch the waveform in Manchester and Differential Manchester Encoding for the bitstream 11000110010. **(6)**

13. (a) A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is $\square^{3} + 1$. Show the actual bit string transmitted. Suppose the third bit from the left is inverted during transmission. Show that this error is detected at the receiver's end. **(8)**

(b) Explain the working of High-Level Data Link Control (HDLC) protocol. **(6)**

OR

14. (a) Explain the working of IEEE 802.11 MAC sublayer. **(10)**

(b) Distinguish between Bridges and Switches. **(4)**

15. (a) Illustrate Distance Vector Routing algorithm with an example. **(8)**

(b) Explain the characteristics of Routing Information Protocol (RIP). **(6)**

OR

16. (a) A computer on a 6-Mbps network is regulated by a token bucket. The token bucket is filled at a rate of 1 Mbps. It is initially filled to capacity with 8 megabits. How long can the computer transmit at the full 6 Mbps? **(8)**

(b) Explain how routing is performed for mobile hosts. **(6)**

17. (a) Explain the address resolution problem using Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP)with an example network. **(10)**

(b) A network on the Internet has a subnet mask of 255.255.240.0. What is the maximum number of hosts it can handle? **(4)**

OR

18. (a) How do you subnet the Class C IP address 195.1.1.0 so as to have 10 subnets with a maximum of 12 hosts in each subnet. **(6)**

(b) Draw IPv6 Datagram format and explain its features. **(8)**

19. (a) Distinguish the header formats of Transmission Control protocol (TCP) and User Datagram Protocol (UDP). **(8)**

(b) Explain the principal Domain Name System (DNS) resource record types for **(6)**

IPv4.

OR

20. (a) What is the role of Simple Mail Transfer Protocol (SMTP) in E- mail? **(6)**

(b) With the help of a basic model, explain the working of World Wide Web (WWW). **(8)**

Teaching Plan

No	Contents	No of Lecture Hrs
Module – 1 (Introduction and Physical Layer) (10 hrs)		
1.1	Introduction, Uses of computer networks.	1 hour
1.2	Network Hardware, Local Area Networks (LAN), Metropolitan Area Networks (MAN), Wide Area Networks (WAN), Wireless networks, Home networks, Internetworks.	1 hour
1.3	Network Software, Protocol hierarchies, Design issues for the layers.	1 hour
1.4	Connection-oriented and Connectionless services, Service primitives, Relationship of services to protocols.	1 hour
1.5	Reference models, The OSI reference model.	1 hour
1.6	The TCP/IP reference model, Comparisonof OSI and TCP/IP reference models.	1 hour
1.7	Physical layer, Modes of communication, Simplex, Half-duplex, and Full-duplex, Physical topologies, Mesh, Star, Bus, Ring, Hybrid.	1 hour
1.8	Signal encoding, Manchester, Differential Manchester.	1 hour
1.9	Transmission media overview, Guided media (twisted pair, coaxial and fiber optic media), Unguided/wireless media (radio, microwave, and infrared).	1 hour
1.10	Performance indicators, Bandwidth (in Hertz and in Bits per Seconds),	1 hour

	Throughput, Latency (Delay), Queuing time, Bandwidth-Delay product.	
Module 2 – (Data Link Layer) (10 hrs)		
2.1	Data link layer design issues.	1 hour
2.2	Error detection and correction, Error correcting codes	1 hour
2.3	Error detecting codes.	1 hour
2.4	Sliding window protocols.	1 hour
2.5	High-Level Data Link Control(HDLC) protocol.	1 hour
2.6	Medium Access Control (MAC) sublayer, Channel allocation problem, Multiple access protocols.	1 hour
2.7	Ethernet, Ethernet cabling, Manchester encoding, Ethernet MAC sublayer protocol, Binary Exponential Backoff algorithm.	1 hour
2.8	Ethernet performance, Switched Ethernet, Fast Ethernet, Gigabit Ethernet, IEEE 802.2: Logical Link Control.	1 hour
2.9	Wireless LANs, 802.11 protocol stack, Physical layer, MAC Sublayer protocol, Frame structure.	1 hour
2.10	Bridges &switches, Bridges from 802.x to 802.y, Repeaters, Hubs, Bridges, Switches, Routers, and Gateways.	1 hour
Module 3 - (Network Layer) (8 hrs)		
3.1	Network layer design issues.	1 hour
3.2	Routing algorithms, The Optimality Principle, Shortest path routing, Flooding.	1 hour
3.3	Distance Vector Routing.	1 hour
3.4	Link State Routing.	1 hour
3.5	Multicast routing, Routing for mobile hosts.	1 hour

3.6	General principles of congestion control, Congestion prevention policies, Congestion control in virtual circuit subnets.	1 hour
3.7	Congestion control algorithms, Congestion control in Datagram subnets, Load shedding, Jitter control.	1 hour
3.8	Quality of Service, Requirements, Techniques for achieving good Quality of Service.	1 hour
Module 4 – (Network Layer in the Internet) (9 hrs)		
4.1	Network layer in the Internet, Internet Protocol (IP).	1 hour
4.2	IP Addresses, Subnets, Classless Inter-Domain Routing (CIDR).	1 hour
4.3	IP Addresses, Network Address Translation (NAT).	1 hour
4.4	Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP).	1 hour
4.5	Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP).	1 hour
4.6	Open Shortest Path First (OSPF) protocol.	1 hour
4.7	Border Gateway Protocol (BGP).	1 hour
4.8	Internet multicasting.	1 hour
4.9	IPv6, Header format, Extension headers, Internet Control Message Protocol version 6 (ICMPv6).	1 hour
Module 5 - (Transport Layer and Application Layer) (8 hrs)		
5.1	Transport Service, Services provided to the upper layers, Transport service primitives. User Datagram Protocol (UDP).	1 hour
5.2	Transmission Control Protocol (TCP), TCP segment header, Connection establishment &release, Connection management modeling.	1 hour
5.3	TCP retransmission policy, TCP congestion control.	1 hour
5.4	Application layer, File Transfer Protocol (FTP).	1 hour

5.5	Domain Name System (DNS).	1 hour
5.6	Electronic Mail, Multipurpose Internet Mail Extension (MIME).	1 hour
5.7	Simple Network Management Protocol (SNMP).	1 hour
5.8	World Wide Web, Architectural overview.	1 hour

CST 305	SYSTEM SOFTWARE	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble:

The purpose of this course is to create awareness about the low-level codes which are very close to the hardware and about the environment where programs can be developed and executed. This course helps the learner to understand the machine dependent and machine independent system software features and to design/implement system software like assembler, loader, linker, macroprocessor and device drivers. Study of system software develops ability to design interfaces between software applications and computer hardware**.**

Prerequisite: A sound knowledge in Data Structures, and Computer Organization

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Distinguish **softwares** into system and application software categories. **(Cognitive Knowledge Level: Understand)**
CO2	Identify **standard** and extended architectural features of machines. (**Cognitive Knowledge Level: Apply)**
CO3	Identify machine dependent features of system software (**Cognitive Knowledge Level: Apply)**
CO4	Identify machine independent features of system software. (**Cognitive Knowledge Level: Understand)**
CO5	Design **algorithms** for system softwares and analyze the effect of data structures. (**Cognitive Knowledge Level: Apply)**
CO6	Understand the features of device drivers and editing & debugging tools.(**Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓			✓							✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓									✓
CO4	✓	✓										✓
CO5	✓	✓	✓	✓								✓
CO6	✓	✓			✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks(%)
	Test 1 (%)	**Test 2 (%)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**
Continuous Assessment Test (Average of series Tests 1&2) : **25 marks**
Continuous Assessment Assignment **: 15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module-1 (Introduction)

System Software vs Application Software, Different System Software– Assembler, Linker, Loader, Macro Processor, Text Editor, Debugger, Device Driver, Compiler, Interpreter, Operating System (Basic Concepts only). SIC & SIC/XE Architecture, Addressing modes, SIC & SIC/XE Instruction set , Assembler Directives.

Module-2 (Assembly language programming and Assemblers)

SIC/XE Programming, Basic Functions of Assembler, Assembler Output Format – Header, Text and End Records. Assembler Data Structures, Two Pass Assembler Algorithm, Hand Assembly of SIC/XE Programs.

Module-3 (Assembler Features and Design Options)

Machine Dependent Assembler Features-Instruction Format and Addressing Modes, Program Relocation. Machine Independent Assembler Features –Literals, Symbol Defining Statements, Expressions, Program Blocks, Control Sections and Program Linking. Assembler Design Options- One Pass Assembler, Multi Pass Assembler. Implementation Example-MASM Assembler.

Module-4 (Loader and Linker)

Basic Loader Functions - Design of Absolute Loader, Simple Bootstrap Loader. Machine Dependent Loader Features- Relocation, Program Linking, Algorithm and Data Structures of Two Pass Linking Loader. Machine Independent Loader Features -Automatic Library Search, Loader Options. Loader Design Options.

Module-5 (Macro Preprocessor ,Device driver, Text Editor and Debuggers)

Macro Preprocessor - Macro Instruction Definition and Expansion, One pass Macro processor Algorithm and data structures, Machine Independent Macro Processor Features, Macro processor design options. Device drivers - Anatomy of a device driver, Character and block device drivers, General design of device drivers. Text Editors- Overview of Editing, User Interface, Editor

Structure. Debuggers - Debugging Functions and Capabilities, Relationship with other parts of the system, Debugging Methods- By Induction, Deduction and Backtracking.

Text book

1. Leland L. Beck, System Software: An Introduction to Systems Programming, 3/E, Pearson Education Asia

References

1. D.M. Dhamdhere, Systems Programming and Operating Systems, Second Revised Edition, Tata McGraw Hill.
2. John J. Donovan, Systems Programming, Tata McGraw Hill Edition 1991.
3. George Pajari, Writing UNIX Device Drivers, Addison Wesley Publications (Ebook : http://tocs.ulb.tu-darmstadt.de/197262074.pdf).
4. Peter Abel, IBM PC Assembly Language and Programming, Third Edition, Prentice Hall of India.
5. Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman, Linux Device Drivers, Third Edition, O.Reilly Books
6. M. Beck, H. Bohme, M. Dziadzka, et al., Linux Kernel Internals, Second Edition, Addison Wesley Publications,
7. J Nithyashri, System Software, Second Edition, Tata McGraw Hill.
8. The C Preprocessor http://gcc.gnu.org/onlinedocs/gcc-2.95.3/cpp_1.html -

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. List out two system software and two application software.

Course Outcome 2 (CO2):

1. How is upward compatibility between SIC and SIC/XE machines maintained?
2. Write a sequence of instructions for SIC/XE to divide BETA by GAMMA, setting ALPHA to the integer portion of the quotient and DELTA to the remainder. Use register-to-register instructions to make the calculation as efficient as possible.

Course Outcome 3 (CO3):

1. How do control sections and program blocks differ?
2. Can an assembler incorporating program blocks function using the same data structures as that of a normal two pass assembler? Justify your answer

Course Outcome 4 (CO4):

1. What are literals used for? Does the use of literals change the design of an assembler?

Course Outcome 5 (CO5):

1. Design an assembler that can assemble a source program with different control sections.

Course Outcome 6 (CO6):

1. Describe any one commonly used debugging method.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH. DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 305

Course Name: System Software

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Differentiate between system software and application software.

2. What are assembler directives? List out any five assembler directives in SIC.

3. Explain the different data structures used in the implementation of Assemblers.

4. List out the functions performed by an assembler.

5. What is a Literal? How is a literal handled by an assembler.

6. What are control sections? What is the advantage of using them?

7. Differentiate between linking loader and linkage editor? Which of these is preferable in a program development environment?

8. What is Automatic Library Search?

9. How should a programmer decide whether to use a macro or a subroutine to accomplish a given logical function?

10. Differentiate between character and block device drivers

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Differentiate between compilers and Interpreters. . **(4)**

(b) Explain the architecture and addressing modes of SIC machine. **(10)**

OR

12. (a) Explain the addressing modes supported by SIC/ XE machine with suitable illustrations. **(8)**

(b) Explain the difference between **(6)**

i) A1 RESW 3 and A1 WORD 3
ii) B BYTE C’23’ and B BYTE X’23’
iii) END and END LABEL

13. (a) Let NUMBERS be an array of 100 words. Write a sequence of SIC/XE instructions to find the maximum of these numbers. **(6)**

(b) Perform hand assembly of the above written program using two pass assembler and show the status of various data structures and object program create. **(8)**

OR

14. (a) Write down and explain the second pass of a two pass assembler algorithm. **(8)**

(b) What is a Program Block. What is its advantage? With suitable example, explain how Program Blocks are handled by SIC assembler. **(6)**

15. (a) What is a Program Block. What is its advantage? With suitable example, explain how Program Blocks are handled by SIC assembler. **(7)**

(b) What is a forward reference? With example, illustrate how forward references are handled by a single pass assembler? **(7)**

OR

16. (a) With suitable examples explain machine dependent assembler features. **(8)**

(b) Explain with examples, the need and working of multipass assembler. **(6)**

17. (a) With the data structures used, state and explain two pass algorithm for a linking loader. **(10)**

(b) Explain about bootstrap loader. **(4)**

OR

18. (a) Explain about machine independent loader features **(9)**

(b) What is Dynamic Linking? With example, illustrate how dynamic linking is performed. **(5)**

19. (a) Write down the single pass macro processor algorithm and with suitable example illustrate its working. **(10)**

(b) How are unique labels generated during Macro Expansion? **(4)**

OR

20. (a) Explain Text Editor structure in detail with a neat diagram. **(7)**

(b) Explain the different debugging methods in detail. **(7)**

Teaching Plan

No	Contents	**No: of** Lecture **Hours**
	Module -1 (Introduction) (9 hours)	
1.1	System Software Vs. Application Software , Different System Software– Assembler, Linker, Loader, Macro Processor	1 hour
1.2	Text Editor, Debugger, Device Driver, Compiler, Interpreter, Operating System(Basic Concepts only)	1 hour
1.3	SIC Architecture	1 hour
1.4	SIC Addressing modes	1 hour
1.5	SIC Instruction set & Assembler directives	1 hour
1.6	SIC/XE Architecture	1 hour
1.7	SIC/XE Instruction format	1 hour
1.8	SIC/XE Addressing modes	1 hour
1.9	SIC/XE Instruction set	1 hour
	Module -2 (Assembly language programming and Assemblers) (8 hours)	
2.1	SIC Programming	1 hour
2.2	SIC/XE Programming	1 hour
2.3	Basic Functions of Assembler	1 hour
2.4	Assembler output format – Header, Text and End Records	1 hour
2.5	Assembler data structures	1 hour
2.6	Pass 1 of two pass SIC assembler algorithm	1 hour
2.7	Pass 2 of two pass SIC assembler algorithm	1 hour
2.8	Hand assembly of SIC Program	1 Hour
	Module-3 (Assembler design options)(11 hours)	

3.1	Machine dependent assembler features-Instruction format and addressing modes, program relocation	1 hour
3.2	Hand assembly of SIC/XE program	1 Hour
3.3	Machine Independent assembler features – Literals	1 hour
3.4	Machine Independent assembler features – Symbol defining statements, expression	1 hour
3.5	Machine Independent assembler features – program blocks	1 hour
3.6	Machine Independent assembler features – program blocks illustration with examples	1 hour
3.7	Machine Independent assembler features – Control sections and program linking.	1 hour
3.8	Machine Independent assembler features – Control sections and program linking. Illustration with example	1 hour
3.9	Assembler design options- One Pass assembler	1 hour
3.10	Multi pass assembler	1 hour
3.11	Implementation example: MASM Assembler	1 hour
Module-4 (Linker and Loader) (8 hours)		
4.1	Basic Loader functions - Design of absolute loader	1 hour
4.2	Simple bootstrap Loader	1 hour
4.3	Machine dependent loader features- Relocation	1 hour
4.4	Machine dependent loader features- Program Linking algorithm and data structures of First pass of two pass Linking Loader	1 hour
4.5	Machine dependent loader features- Program Linking algorithm and data structures of Second pass of two pass Linking Loader	1 hour
4.6	Machine independent loader feature - Automatic library search	1 hour
4.7	Machine independent loader features - Loader options	1 hour
4.8	Loader Design Option- Linking Loader, Linkage Editor, Dynamic Linking	1 hour
Module –5 (Macro Preprocessor, Device drivers, Text Editors, Debuggers) (9 hours)		
5.1	Macro Preprocessor- Macro Instruction Definition and Expansion	1 hour

5.2	One pass Macro processor algorithm and data structures	1 hour
5.3	One pass Macro processor Algorithm and data structures illustration with example	1 hour
5.4	Machine Independent Macro Processor Features- generation of unique labels, Concatenation of macro parameter, Keyword macro parameters	1 hour
5.5	Machine Independent Macro Processor Features- Conditional Macro Expansion	1 hour
5.6	Macro processor design options	1 hour
5.7	Device drivers- Anatomy of a device driver, Character and block device drivers, General design of device drivers	1 hour
5.8	Text Editors- Overview of Editing, User Interface , Editor Structure	1 hour
5.9	Debuggers :- Debugging Functions and Capabilities, Debugging Methods- By Induction, Deduction and Backtracking.	1 hour

CST 307	MICROPROCESSORS AND MICROCONTROLLERS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble: The course enables the learners capable of understanding the fundamental architecture of microprocessors and micro controllers. This course focuses on the architecture, assembly language programming, interrupts, interfacing of microprocessors with peripheral devices and microcontrollers and its programming. It helps the learners to extend the study of latest processors and develop hardware based solutions.

Prerequisite : Sound knowledge in Logic System Design and Computer organization & architecture.

CO#	Course Outcomes
CO1	Illustrate the **architecture**, modes of operation and addressing modes of microprocessors **(Cognitive knowledge: Understand)**
CO2	Develop 8086 **assembly** language programs. **(Cognitive Knowledge Level: Apply**)
CO3	Demonstrate interrupts, its **handling** and programming in 8086. **(Cognitive Knowledge Level: Apply**))
CO4	Illustrate how different **peripherals** (8255,8254,8257) and memory are interfaced with microprocessors. (**Cognitive Knowledge Level: Understand**)
CO5	Outline features of **microcontrollers** and develop low level programs. **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test1 (%)	**Test2 (%)**	
Remember	**20**	**20**	**20**
Understand	**40**	**40**	**40**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	:	**10 marks**
Continuous Assessment Tests	:	**25 marks**
Continuous Assessment Assignment	:	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1(Evolution of microprocessors):

8085 microprocessor (-Basic Architecture only). 8086 microprocessor – Architecture and signals, Physical Memory organization, Minimum and maximum mode of 8086 system and timings. Comparison of 8086 and 8088.Machine language Instruction format.

Module-2 (Addressing modes and instructions):

Addressing Modes of 8086. Instruction set – data copy /transfer instructions, arithmetic instructions, logical instructions, string manipulation instructions, branch instructions, unconditional and conditional branch instruction, flag manipulation and processor control instructions. Assembler Directives and operators. Assembly Language Programming with 8086.

Module- 3 (Stack and interrupts):

Stack structure of 8086, programming using stack- Interrupts - Types of Interrupts and Interrupt Service Routine- Handling Interrupts in 8086- Interrupt programming. -

Programmable Interrupt Controller - 8259, Architecture (Just mention the control word, no need to memorize the control word)- Interfacing Memory with 8086.

Module- 4 (Interfacing chips):

Programmable Peripheral Input/output port 8255 - Architecture and modes of operation-Programmable interval timer 8254-Architecture and modes of operation- DMA controller 8257 Architecture (Just mention the control word, no need to memorize the control word of 8254 and 8257)

Module- 5 (Microcontrollers):

8051 Architecture- Register Organization- Memory and I/O addressing- Interrupts and Stack- 8051 Addressing Modes- Instruction Set- data transfer instructions, arithmetic instructions, logical instructions, Boolean instructions, control transfer instructions-Simple programs.

Text Books

1. Bhurchandi and Ray, Advanced Microprocessors and Peripherals, Third Edition McGraw Hill.
2. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, Pearson Education.
3. Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, Penram International Publishing Pvt. Ltd.

Reference Books

1. Barry B. Brey, The Intel Microprocessors – Architecture, Programming and Interfacing, Eighth Edition, Pearson Education.
2. A. NagoorKani, Microprocessors and Microcontrollers, Second Edition, Tata McGraw Hill
3. Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, Third Edition, McGrawHill Education.

Sample Course Level Assessment Questions

Course Outcome1 (CO1):

1) Describe how pipelining is implemented in 8086 microprocessor
2) Illustrate maximum mode signals in 8086.

Course Outcome 2(CO2):

1) Write an 8086 assembly language program for sorting a sequence of N, 8 bit numbers. Describe the modifications that can be done on the above program so that it will sort N, 16 bit numbers. Rewrite the program with those modifications also.

Course Outcome 3 (CO3):

1) Design an interface between 8086 CPU and two chips of 16 x 8 EPROM and

two chips of 32K x 8 RAM. Select the starting address of EPROM suitably.

The RAM address must start at 00000H.

2) Give the sequence of instructions for setting the IVT for interrupt type 23H. Assume the Interrupt Service Routine, is present in the code segment named CODE.

3) Describe the role of Interrupt Request register and In service register in 8259.

Course Outcome 4(CO4):

1) Show how to interface an 8255 with 8086 to work as an I/O port with the following specifications. Initialize port A as output, port B as input and port C as output. Port A address should be 05A0H. Write a program to sense switch positions SW 0 -SW 7 connected to port B. The sensed pattern is to be displayed on port A, to which 8 LED's are attached, while port C lower displays *number of off switches* out of total 8 switches.

2) Specify the importance of the DMA address register and Terminal count register in 8257.

Course Outcome 5(CO5):

1) Write an 8051 assembly language program to count the number of 1's and 0's in a given 8 bit number

2) Write an 8051 assembly language program for computing the square root of an 8 bit number.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH. DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 307

Course Name: Microprocessors and Microcontrollers

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Describe the functions of following signals in 8086
   a)NMI b)ALE

2. List any three differences between 8085 and 8086 microprocessors.

3. Assume AL register is having the value 7FH. What will be the content of AL after the following instructions are executed
   a)ROR AL,01 b)SAR AL,01

4. Specify the use of following assembler directives - EQU , EVEN

5. Differentiate between maskable and non maskable interrupts?

6. Define Interrupt Service Routine? How to find the address of the ISR corresponding to a given interrupt in 8086?

7. Give the purposes of the signals DRQ, TC and MARK in 8257?

8. How 8254 is used as a square wave generator?

9. Differentiate between indirect and indexed addressing modes in 8051.

10. Write the sequence of 8051 instructions to store any two numbers at two consecutive locations 70H and 71H, multiply them and store the result in location 72H. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Specify the significance of segmentation and how it is implemented in 8086 **(5)**

(b) Explain the maximum mode signals in 8086. **(9)**

OR

12. (a) Write down the differences between 8086 and 8088 processors **(4)**

(b) Explain the physical memory organization of 8086 with a neat diagram. How does the 8086 processor accesses a word from an odd memory location? How many memory cycles does it take? **(10)**

13. (a) Write an 8086 assembly language program for finding the sum of the squares of first N natural numbers. Calculate the squares of each number using a subroutine SQUARE. **(10)**

(b) Describe any four control transfer instructions in 8086. **(4)**

OR

14. (a) Write an 8086 assembly language program for printing the reverse of a given input string. **(5)**

(b) Explain the addressing modes for sequential control flow instructions in 8086. **(9)**

15. (a) Give the stack structure of 8086. **(5)**

(b) Explain the architecture of 8259 with diagram **(9)**

OR

16. (a) Interface 32Kx8 RAM using four numbers of 8Kx8 memory chips and 16Kx8 ROM using two numbers of 8Kx8 EPROM chips. The address map is given as RAM starts at 00000H and ROM ends at FFFFFH **(10)**

(b) Describe the predefined interrupts in 8086 **(4)**

17. (a) Explain the architecture of 8255 with a neat diagram **(10)**

(b) Identify the mode and I/O configuration for ports A, B and C of an 8255 after its control register is loaded with 86 H? **(4)**

OR

18. (a) Define Direct Memory Access (DMA)and illustrate the role of a DMA controller? Explain the register organization of 8257 and state how these registers are used during DMA transfer operations. **(8)**

(b) Explain the architecture of 8254 timer chip **(6)**

19. (a) Explain the architecture of 8051 microcontroller. **(9)**

(b) Write an 8051 assembly language program for adding two matrices whose elements are stored sequentially in some memory location. Assume suitable locations. **(5)**

OR

20. (a) Explain the internal data memory organization of 8051. **(9)**

(b) Describe the control transfer instructions of 8051microcontroller. **(5)**

Teaching Plan

No	Contents	No of Lecture Hrs
Module 1 : (Evolution of microprocessors) (9 hours)		
1.1	Overview of 8085 microprocessor	1 hour
1.2	Architecture of8085	1 hour
1.3	Architecture of 8086	1hour
1.4	Signals in 8086	1hour
1.5	Physical Memory organization	1hour
1.6	Minimum and maximum mode 8086 system and timings(Lecture 1)	1hour
1.7	Minimum and maximum mode 8086 system and timings(Lecture 2)	1hour
1.8	Comparison of 8086 and 8088	1hour
1.9	Machine language Instruction format	1hour
Module 2 :(programming of 8086) (9 hours)		
2.1	Addressing Modes of 8086	1 hour
2.2	Instruction set – data copy/transfer instructions	1hour
2.3	arithmetic instructions, logical instructions	1hour
2.4	string manipulation instructions, branch instructions	1hour
2.4	unconditional and conditional branch instruction	1hour
2.5	flag manipulation and processor control instructions	1hour
2.6	Assembler Directives and operators	1hour
2.7	Assembly Language Programming with 8086(Lecture 1)	1hour
2.8	Assembly Language Programming with 8086(Lecture 2)	1hour
2.9	Assembly Language Programming with 8086(Lecture 3)	1hour
Module 3 : (stack and Interrupts) (9 hours)		
3.1	Stack structure of 8086, programming using stack.	1hour
3.2	Types of Interrupts and Interrupt Service Routine.	1hour
3.3	Handling Interrupts in 8086(Lecture 1)	1hour
3.4	Handling Interrupts in 8086(Lecture 2)	1hour

3.5	Interrupt programming.	1hour
3.6	Programmable Interrupt Controller -8259 (Lecture 1)	1hour
3.7	Programmable Interrupt Controller -8259 (Lecture 2)	1hour
3.8	Interfacing Memory with 8086 (Lecture 1)	1hour
3.9	Interfacing Memory with 8086 (Lecture 2)	1hour
Module 4 :(Interfacing chips) (7 hours)		
4.1	Programmable Peripheral Input/output port- 8255 (Lecture 1)	1hour
4.2	Programmable Peripheral Input/output port- 8255 (Lecture 2)	1hour
4.3	Programmable Peripheral Input/output port- 8255 (Lecture 3)	1hour
4.4	Programmable interval timer 8254 (Lecture 1)	1hour
4.5	Programmable interval timer 8254 (Lecture 2)	1hour
4.6	DMA controller 8257 Architecture (Lecture 1)	1hour
4.7	DMA controller 8257 Architecture (Lecture 2)	1hour
Module 5 : (Microcontrollers) (11 hours)		
5.1	8051 Architecture (Lecture 1)	1hour
5.2	8051 Architecture (Lecture 2)	1hour
5.3	Register Organization, Memory and I/O addressing	1hour
5.4	Interrupts and Stack	1hour
5.5	Addressing Modes	1hour
5.6	Data transfer instructions, Arithmetic instructions	1hour
5.7	Logical instructions,	1hour
5.8	Boolean instructions	1hour
5.9	Control transfer instructions	1hour
5.10	Programming of 8051 (Lecture 1)	1hour
5.11	Programming of 8051(Lecture 2)	1hour

CST 309	MANAGEMENT OF SOFTWARE SYSTEMS	Category	L	T	P	Credit	Year of Introduction
		PCC	3	0	0	3	2019

Preamble: This course provides fundamental knowledge in the Software Development Process. It covers Software Development, Quality Assurance, Project Management concepts and technology trends. This course enables the learners to apply state of the art industry practices in Software development.

Prerequisite: Basic understanding of Object Oriented Design and Development.

Course Outcomes: After the completion of the course the student will be able to

CO1	Demonstrate Traditional and Agile Software Development approaches **(Cognitive Knowledge Level: Apply)**
CO2	Prepare Software Requirement Specification and Software Design for a given problem. **(Cognitive Knowledge Level**: **Apply**)
CO3	Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project. **(Cognitive Knowledge Level: Apply**)
CO4	Make use of software project management concepts while planning, estimation, scheduling, tracking and change management of a project, with a traditional/agile framework**. (Cognitive Knowledge Level: Apply**)
CO5	Utilize SQA practices, Process Improvement techniques and Technology advancements in cloud based software models and containers & microservices. **(Cognitive Knowledge Level: Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓						✓
CO2	✓	✓	✓	✓		✓				✓	✓	✓
CO3	✓	✓	✓	✓				✓		✓	✓	✓
CO4	✓	✓	✓	✓		✓			✓	✓	✓	✓
CO5	✓	✓	✓	✓		✓						✓

	Abstract POs defined by National Board of Accreditation		
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	**30**	**30**	**30**
Understand	**40**	**40**	**50**
Apply	**30**	**30**	**20**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks** (Each student shall identify a software development problem and prepare Requirements Specification, Design Document, Project Plan and Test case documents for the identified problem as the assignment.)

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus

Module 1 : Introduction to Software Engineering (7 hours)

Introduction to Software Engineering - Professional software development, Software engineering ethics. Software process models - The waterfall model, Incremental development. Process activities - Software specification, Software design and implementation, Software validation, Software evolution. Coping with change - Prototyping, Incremental delivery, Boehm's Spiral Model. Agile software development - Agile methods, agile manifesto - values and principles. Agile development techniques, Agile Project Management. Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care.

Module 2 : Requirement Analysis and Design (8 hours)

Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Personas, Scenarios, User stories, Feature identification. Design concepts - Design within the context of software engineering, Design Process, Design concepts, Design Model. Architectural Design - Software Architecture, Architectural Styles, Architectural considerations, Architectural Design Component level design - What is a component?, Designing Class-Based Components, Conducting Component level design, Component level design for web-apps. Template of a Design Document as per “IEEE Std 1016-2009 IEEE Standard for Information Technology Systems Design Software Design Descriptions”. Case study: The Ariane 5 launcher failure.

Module 3 : Implementation and Testing (9 hours)

Object-oriented design using the UML, Design patterns, Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD. Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. Informal Review, Formal Technical Reviews, Post-mortem evaluations. Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, White box testing, Path testing, Control Structure testing, Black box testing, Testing Documentation and Help facilities. Test automation, Test-driven development, Security testing. Overview of DevOps and Code Management - Code management, DevOps automation, Continuous Integration, Delivery, and Deployment (CI/CD/CD). Software Evolution - Evolution processes, Software maintenance.

Module 4 : Software Project Management (6 hours)

Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Software pricing, Plan-driven development, Project scheduling, Agile planning. Estimation techniques, COCOMO cost modeling. Configuration management, Version management, System building, Change management, Release management, Agile software management - SCRUM framework. Kanban methodology and lean approaches.

Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)

Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, SQA Tasks , Software measurement and metrics. Software Process Improvement(SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software. Cloud-based Software - Virtualisation and containers, Everything as a service(IaaS, PaaS), Software as a service. Microservices Architecture - Microservices, Microservices architecture, Microservice deployment.

Text Books

1. Book 1 - Ian Sommerville, Software Engineering, Pearson Education, Tenth edition, 2015.
2. Book 2 - Roger S. Pressman, Software Engineering : A practitioner's approach, McGraw Hill publication, Eighth edition, 2014
3. Book 3 - Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson Education, First Edition, 2020.

References

1. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements SpeciÞcations
2. IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design—Software Design Descriptions

3. David J. Anderson, Kanban, Blue Hole Press 2010
4. David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
5. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
6. Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
7. Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
8. Henrico Dolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition
9. Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
10. StarUML documentation - https://docs.staruml.io/
11. OpenProject documentation - https://docs.openproject.org/
12. BugZilla documentation - https://www.bugzilla.org/docs/
13. GitHub documentation - https://guides.github.com/
14. Jira documentation - https://www.atlassian.com/software/jira

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the advantages of an incremental development model over a waterfall model?
2. Illustrate how the process differs in agile software development and traditional software development with a socially relevant case study. (Assignment question)

Course Outcome 2 (CO2):

1. How to prepare a software requirement specification?
2. Differentiate between Architectural design and Component level design.
3. How does agile approaches help software developers to capture and define the user requirements effectively?
4. What is the relevance of the SRS specification in software development?
5. Prepare a use case diagram for a library management system.

Course Outcome 3 (CO3):

1. Differentiate between the different types of software testing strategies.
2. Justify the need for DevOps practices?
3. How do design patterns help software architects communicate the design of a complex system effectively?

4. What are the proactive approaches one can take to optimise efforts in the testing phase?

Course Outcome 4 (CO4):

1. Illustrate the activities involved in software project management for a socially relevant problem?
2. How do SCRUM, Kanban and Lean methodologies help software project management?
3. Is rolling level planning in software project management beneficial? Justify your answer.
4. How would you assess the risks in your software development project? Explain how you can manage identified risks?

Course Outcome 5 (CO5):

1. Justify the importance of Software Process improvement?
2. Explain the benefits of cloud based software development, containers and microservices.
3. Give the role of retrospectives in improving the software development process.
4. Illustrate the use of project history data as a prediction tool to plan future socially relevant projects.

Model Question Paper

QP CODE:

Reg No:______________

Name :______________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 309

Course Name: Management of Software Systems

Duration: 3 Hrs **Max. Marks :100**

PART A

Answer all Questions. Each question carries 3 marks

1. Why professional software that is developed for a customer is not simply the programs that have been developed and delivered.
2. Incremental software development could be very effectively used for customers who do not have a clear idea about the systems needed for their operations. Justify.
3. Identify any four types of requirements that may be defined for a software system
4. Describe software architecture
5. Differentiate between GPL and LGPL?
6. Compare white box testing and black box testing.
7. Specify the importance of risk management in software project management?
8. Describe COCOMO cost estimation model.
9. Discuss the software quality dilemma
10. List the levels of the CMMI model? **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Compare waterfall model and spiral model

(8)

(b) Explain Agile ceremonies and Agile manifesto **(6)**

12. (a) Illustrate software process activities with an example. **(8)**

(b) Explain Agile Development techniques and Agile Project Management **(6)**

13. (a) What are functional and nonfunctional requirements? Imagine that you are developing a library management software for your college, list eight functional requirements and four nonfunctional requirements. **(10)**

(b) List the components of a software requirement specification? **(4)**

OR

14. (a) Explain Personas, Scenarios, User stories and Feature identification? **(8)**

(b) Compare Software Architecture design and Component level design **(6)**

15. (a) Explain software testing strategies. **(8)**

(b) Describe the formal and informal review techniques. **(6)**

OR

16. (a) Explain Continuous Integration, Delivery, and Deployment CI/CD/CD) **(8)**

(b) Explain test driven development **(6)**

17. (a) What is a critical path and demonstrate its significance in a project schedule with the help of a sample project schedule. **(8)**

(b) Explain plan driven development and project scheduling. **(6)**

OR

18. (a) Explain elements of Software Quality Assurance and SQA Tasks. **(6)**

(b) What is algorithmic cost modeling? What problems does it suffer from when **(8)** compared with other approaches to cost estimation?

19. (a) Explain elements of Software Quality Assurance and SQA Tasks. **(8)**

(b) Illustrate SPI process with an example. **(6)**

OR

20. (a) Compare CMMI and ISO 9001:2000. **(8)**

(b) How can Software projects benefit from Container deployment and Micro service deployment? **(6)**

Teaching Plan

No	Contents	No of Lecture Hrs
Module 1 : Introduction to Software Engineering (7 hours)		
1.1	Introduction to Software Engineering.[Book 1, Chapter 1]	1 hour
1.2	Software process models [Book 1 - Chapter 2]	1 hour
1.3	Process activities [Book 1 - Chapter 2]	1 hour
1.4	Coping with change [Book 1 - Chapter 2, Book 2 - Chapter 4]	1 hour
1.5	Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care. [Book 1 - Chapter 1]	1 hour
1.6	Agile software development [Book 1 - Chapter 3]	1 hour
1.7	Agile development techniques, Agile Project Management.[Book 1 - Chapter 3]	1 hour
Module 2 : Requirement Analysis and Design (8 hours)		
2.1	Functional and non-functional requirements, Requirements engineering processes [Book 1 - Chapter 4]	1 hour
2.2	Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix [Book 1 - Chapter 4]	1 hour
2.3	Developing use cases, Software Requirements Specification Template [Book 2 - Chapter 8]	1 hour

2.4	Personas, Scenarios, User stories, Feature identification [Book 3 - Chapter 3]	1 hour
2.5	Design concepts [Book 2 - Chapter 12]	1 hour
2.6	Architectural Design [Book 2 - Chapter 13]	1 hour
2.7	Component level design [Book 2 - Chapter 14]	1 hour
2.8	Design Document Template. Case study: The Ariane 5 launcher failure. [Ref - 2, Book 2 - Chapter 16]	1 hour
Module 3 : Implementation and Testing (9 hours)		
3.1	Object-oriented design using the UML, Design patterns [Book 1 - Chapter 7]	1 hour
3.2	Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD [Book 1 - Chapter 7]	1 hour
3.3	Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. [Book 2 - Chapter 20]	1 hour
34	Informal Review, Formal Technical Reviews, Post-mortem evaluations. [Book 2 - Chapter 20]	1 hour
3.5	Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing and Debugging (basic concepts only). [Book 2 - Chapter 22]	1 hour
3.6	White box testing, Path testing, Control Structure testing, Black box testing. Test documentation [Book 2 - Chapter 23]	1 hour
3.7	Test automation, Test-driven development, Security testing. [Book 3 - Chapter 9]	1 hour
3.8	DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. [Book 3 - Chapter 10]	1 hour
3.9	Software Evolution - Evolution processes, Software maintenance. [Book 1 - Chapter 9]	1 hour
Module 4 : Software Project Management (6 hours)		
4.1	Software Project Management - Risk management, Managing people, Teamwork [Book 1 - Chapter 22]	1 hour
4.2	Project Planning - Software pricing, Plan-driven development, Project scheduling, Agile planning [Book 1 - Chapter 23]	1 hour
4.3	Estimation techniques [Book 1 - Chapter 23]	1 hour
4.4	Configuration management [Book 1 - Chapter 25]	1 hour

4.5	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.6	Kanban methodology and lean approaches.[Ref 9 - Chapter 2]	1 hour
Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)		
5.1	Software Quality, Software Quality Dilemma, Achieving Software Quality. [Book 2 - Chapter 19]	1 hour
5.2	Elements of Software Quality Assurance, SQA Tasks , Software measurement and metrics. [Book 3 - Chapter 21]	1 hour
5.3	Software Process Improvement (SPI), SPI Process [Book 2 - Chapter 37]	1 hour
5.4	CMMI process improvement framework, ISO 9001:2000 for Software. [Book 2 - Chapter 37]	1 hour
5.5	Cloud-based Software - Virtualisation and containers, IaaS, PaaS, SaaS.[Book 3 - Chapter 5]	1 hour
5.6	Microservices Architecture - Microservices, Microservices architecture, Microservice deployment [Book 3 - Chapter 6]	1 hour

CSL 331	SYSTEM SOFTWARE AND MICROPROCESSORS LAB	Category	L	T	P	Credit	Year of Introduction
		PCC	0	0	4	2	2019

Preamble: The aim of this course is to give hands-on experience in how microcontrollers, and microprocessors can be programmed. The course also aims to enable students to design and implement system software. The student should get familiar with assembly level programming of microprocessors and microcontrollers, interfacing of devices to microcontrollers, resource allocation algorithms in operating systems and design and implementation of system software.

Prerequisite: Sound knowledge in Operating systems

CO1	Develop 8086 programs and execute it using a microprocessor kit. **(Cognitive Knowledge Level: Apply) .**
CO2	Develop 8086 programs and, debug and execute it using MASM assemblers **(Cognitive Knowledge Level: Apply)**
CO3	Develop and execute programs to interface stepper motor, 8255, 8279 and digital to analog converters with 8086 trainer kit **(Cognitive Knowledge Level: Apply)**
CO4	Implement and execute different scheduling and paging algorithms in OS **(Cognitive Knowledge Level: Apply)**
CO5	Design and implement assemblers, Loaders and macroprocessors. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO 5	PO 6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓				✓		✓		✓
CO2	✓	✓	✓	✓				✓		✓		✓
CO3	✓	✓	✓	✓				✓		✓		✓
CO4	✓	✓	✓	✓				✓		✓		✓
CO5	✓	✓	✓	✓				✓		✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Test (Internal Exam)** ***Percentage***	**End Semester Examination** ***Percentage***
Remember	**20**	**20**
Understand	**20**	**20**
Apply	**60**	**60**
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**75**	**75**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	: **15 marks**
Continuous Evaluation in Lab	: **30 marks**
Continuous Assessment Test	: **15 marks**
Viva-voce	: **15 marks**

Internal Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Operating System to Use in Lab	**:** Linux
Compiler/Software to Use in Lab	**:** gcc
Programming Language to Use in Lab	**:** Ansi C

Any compatible assembler can be used for implementation of 8086 programs

Fair Lab Record:

All Students attending the System Software and Microprocessors Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Details of Experiment including algorithm and Result of Experiment. The left hand page should contain a print out of the code used for the experiment and sample output obtained for a set of input.

Syllabus

MICROPROCESSOR LAB

I. Assembly Language Programming Exercises/Experiments using 8086 Trainer kit
II. Exercises/Experiments using MASM (PC required)
III. Interfacing Exercises/Experiments with 8086 trainer kit through Assembly Language programming
IV. Exercises/Experiments using 8051 trainer kit

SYSTEM SOFTWARE LAB:

I. Experiments related to the operating system.
II. Exercises/Experiments related to the assemblers, loaders and macroprocessors

Text Books

1. Bhurchandi and Ray, Advanced Microprocessors and Peripherals, Third Edition McGraw Hill.
2. Andrew S Tanenbaum, “Modern Operating Systems” , 4th Edition, Prentice Hall, 2015.
3. Leland L. Beck, System Software: An Introduction to Systems Programming, 3/E, Pearson Education Asia, 1997.

Reference Books

1. A. NagoorKani, Microprocessors and Microcontrollers, Second Edition, Tata McGraw Hill
2. Douglas V. Hall, SSSP Rao, Microprocessors and Interfacing, Third Edition, McGrawHill Education.
3. William Stallings, “Operating systems”, 6th Edition, Pearson, Global Edition, 2015.
4. Garry Nutt, Nabendu Chaki, Sarmistha Neogy, “Operating Systems”, 3rd Edition, Pearson Education.
5. D.M. Dhamdhere, Systems Programming and Operating Systems, Second Revised Edition, Tata McGraw Hill.

Practice Questions

MICROPROCESSORS LAB : List of Exercises/ Experiments

(Minimum 10 Exercises (at least 2 questions from each part I, II, III & IV)) : 2 Hrs/week

I. Assembly Language Programming Exercises/Experiments using 8086 Trainer kit

1. Implementation of simple decimal arithmetic and bit manipulation operations.
2. Implementation of code conversion between BCD, Binary, Hexadecimal and ASCII.
3. Implementation of searching and sorting of 16-bit numbers.

II. Exercises/Experiments using MASM (PC Required)

4. Study of Assembler and Debugging commands.
5. Implementation of decimal arithmetic (16 and 32 bit) operations.
6. Implementation of String manipulations.
7. Implementation of searching and sorting of 16-bit numbers.

III. Interfacing Exercises/Experiments with 8086 trainer kit through Assembly Language Programming

8. Interfacing with stepper motor - Rotate through any given sequence.
9. Interfacing with 8255 (mode0 and mode1 only).
10. Interfacing with 8279 (Rolling message, 2 key lockout and N-key rollover implementation).

11. Interfacing with Digital-to-Analog Converter.

IV. Exercises/Experiments using 8051 trainer kit

12. Familiarization of 8051 trainer kit by executing simple Assembly Language programs such as decimal arithmetic and bit manipulation.
13. Implementation of Timer programming (in mode1).

SYSTEM SOFTWARE LAB: List of Exercises/ Experiments
(Minimum 8 Exercises (at least 3 and 5 questions from each part V and VI)) : 2 Hrs/week

V. Exercises/Experiments from operating system

1. Simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.
   a) FCFS b) SJF c) Round Robin (pre-emptive) d) Priority
2. Simulate the following file allocation strategies.
   a) Sequential b) Indexed c) Linked
3. Implement the different paging techniques of memory management.
4. Simulate the following file organization techniques
   a) Single level directory b) Two level directory c) Hierarchical
5. Implement the banker's algorithm for deadlock avoidance.
6. Simulate the following disk scheduling algorithms.
   a) FCFS b) SCAN c) C-SCAN
7. Simulate the following page replacement algorithms:
   a)FIFO b)LRU c) LFU

VI. Exercises/Experiments from assemblers, loaders and macroprocessor

1. Implement pass one of a two pass assembler.
2. Implement pass two of a two pass assembler.
3. Implement a single pass assembler.
4. Implement a two pass macro processor
5. Implement a single pass macro processor.
6. Implement an absolute loader.
7. Implement a relocating loader

CSL 333	**DATABASE MANAGEMENT SYSTEMS LAB**	**Category**	**L**	**T**	**P**	**Credits**	**Year of introduction**
		PCC	**0**	**0**	**4**	**2**	**2019**

Preamble:

The Database Management Systems course is intended to impart the elementary concepts of a database management system to students and equip them to design and implement a database application based on those concepts. This course helps the learners to get practical exposure on database creation, SQL queries creation, transaction processing and NoSQL & MongoDB based operations. The course enables the students to create, manage and administer the databases, develop necessary tools for the design and development of the databases, and to understand emerging technologies to handle Big Data.

Prerequisite: A sound knowledge of the basics of relational DBMS.

Course Outcomes: After the completion of the course the student will be able to

CO#	**Course Outcomes**
CO1	Design database schema for a given real world problem-domain using standard design and modeling approaches. **(Cognitive Knowledge Level: Apply)**
CO2	Construct queries using SQL for database creation, interaction, modification, and updation. **(Cognitive Knowledge Level: Apply)**
C03	Design and implement triggers and cursors. (Cognitive Knowledge Level: **Apply**)
C04	Implement procedures, functions, and control structures using PL/SQL**. (Cognitive Knowledge Level: Apply)**
CO5	Perform CRUD operations in NoSQL Databases. **(Cognitive Knowledge Level: Apply)**
C06	Develop database applications using front-end tools and back-end DBMS. **(Cognitive Knowledge Level: Create)**

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	✓	✓	✓		✓			✓		✓		✓
CO2	✓	✓	✓		✓			✓		✓		✓
CO3	✓	✓	✓	✓	✓			✓		✓		✓
CO4	✓	✓	✓	✓	✓			✓		✓		✓
CO5	✓	✓	✓		✓			✓		✓		✓
CO6	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern:

Bloom's Category	**Continuous Assessment Test (Internal Exam)Percentage**	**End Semester Examination Percentage**
Remember	**20**	**20**
Understand	**20**	**20**
Apply	**60**	**60**
Analyse		
Evaluate		
Create		

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: **15 marks**
Continuous Evaluation in Lab	: **30 marks**
Continuous Assessment Test	: **15 marks**
Viva-voce	: **15 marks**

Internal Examination Pattern : The marks will be distributed as Schema/Logic: 30 marks, Program/Queries: 20 marks, Output: 20 marks, and Viva: 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern:
The marks will be distributed as Schema/Logic: 30 marks,
Program/Queries: 20 marks, Output: 20 marks, and Viva: 30 marks. Total 100 marks will be converted out of 75 for the End Semester Examination.

DBMS software: Oracle, MySQL, SQL Server, PostgreSQL, MongoDB.
Front end Tool: Java

Fair Lab Record:
All Students attending the DBMS Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record, the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Schemas/Menu & Form Design, and Query questions. The left hand page should contain Queries and sample output(relations created, Form, and Menu Output) obtained for a set of input.

Syllabus

1. Design a database schema for an application with ER diagram from a problem description **.
2. Creation, modification, configuration, and deletion of databases using UI and SQL Commands **.
3. Creation of database schema - DDL (create tables, set constraints, enforce relationships, create indices, delete and modify tables). Export ER diagram from the database and verify relationships** (with the ER diagram designed in step 1).

4. Database initialization - Data insert, Data import to a database (bulk import using UI and SQL Commands)**.
5. Practice SQL commands for DML (insertion, updating, altering, deletion of data, and viewing/querying records based on condition in databases)**.
6. Implementation of built-in functions in RDBMS**.
7. Implementation of various aggregate functions in SQL**.
8. Implementation of Order By, Group By & Having clause **.
9. Implementation of set operators nested queries, and join queries **.
10. Implementation of queries using temp tables.
11. Practice of SQL TCL commands like Rollback, Commit, Savepoint **.
12. Practice of SQL DCL commands for granting and revoking user privileges **.
13. Practice of SQL commands for creation of views and assertions ** .
14. Implementation of various control structures like IF-THEN, IF-THEN-ELSE, IF-THEN-ELSIF, CASE, WHILE using PL/SQL **.
15. Creation of Procedures, Triggers and Functions**.
16. Creation of Packages **.
17. Creation of Cursors **.
18. Creation of PL/SQL blocks for exception handling **.
19. Database backup and restore using commands.
20. Query analysis using Query Plan/Show Plan.
21. Familiarization of NoSQL Databases and CRUD operations**.
22. Design a database application using any front end tool for any problem selected. The application constructed should have five or more tables**.

** mandatory

Text Books

1. Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education, 2013.
2. Sliberschatz A., H. F. Korth and S. Sudarshan, Database System Concepts, 6/e, McGraw Hill, 2011.

References

1. Adam Fowler, NoSQL for Dummies, John Wiley & Sons, 2015
2. NoSQL Data Models: Trends and Challenges (Computer Engineering: Databases and Big Data), Wiley, 2018

Practice Questions

Design a normalized database schema for the following requirement.

The requirement: A library wants to maintain the record of books, members, book issue, book return, and fines collected for late returns, in a database. The database can be loaded with book information. Students can register with the library to be a member. Books can be issued to students with a valid library membership. A student can keep an issued book with him/her for a maximum period of two weeks from the date of issue, beyond which a fine will be charged. Fine is calculated based on the delay in days of return. For 0-7 days: Rs 10, For 7 – 30 days: Rs 100, and for days above 30 days: Rs 10 will be charged per day.

Sample Database Design

BOOK (**Book_Id**, Title, Language_Id, MRP, Publisher_Id, Published_Date, Volume, Status) // Language_Id, Publisher_Id are FK (Foreign Key)

AUTHOR(Author_Id, Name, Email, Phone_Number, Status)

BOOK_AUTHOR(Book_Id, Author_Id) // many-to-many relationship, both columns are PKFK (Primary Key and Foreign Key)

PUBLISHER(Publisher_id, Name, Address)

MEMBER(Member_Id, Name, Branch_Code, Roll_Number, Phone_Number, Email_Id, Date_of_Join, Status)

BOOK_ISSUE(Issue_Id, Date_Of_Issue, Book_Id, Member_Id, Expected_Date_Of_Return, Status) // Book+Id and Member_Id are FKs

BOOK_RETURN(Issue_Id, Actual_Date_Of_Return, LateDays, LateFee) // Issue_Id is PK and FK

LANGUAGE(Language_id, Name) //Static Table for storing permanent data

LATE_FEE_RULE(FromDays, ToDays, Amount) // Composite Key

EXERCISES

1. Create a normalized database design with proper tables, columns, column types, and constraints
2. Create an ER diagram for the above database design.
3. Write SQL commands to
    a. Create a database by name *Library.* Drop the database and re-create it.
    b. Create DDL statements and create the tables and constraints (from the design) in the database created in step-a (*Library*)

Notes: [Create a script file and execute it. Create the script file in such a way that,,if the table exists, drop the tables and recreate)]

c. Create and execute DROP TABLE command in tables with and without FOREIGN KEY constraints.
d. Create and execute ALTER TABLE command in tables with data and without data.
e. Create and execute SQL commands to build indices on Member_Id and Book_Id on table Book_Issue.
f. Create and execute GRANT/REVOKE commands on tables.
g. Create and execute SQL commands to insert data into each of the tables designed
h. Learn and execute bulk import of data to tables from CSV files (insert 1000 records of books into the BOOK table from a CSV file).
i. Create and execute UPDATE/DELETE commands on tables. Try to update/delete rows with Primary and Foreign Keys. Try bulk updates or deletes using SQL UPDATE statement

4. Write SQLQuery to retrieve the following information
   a. Get the number of books written by a given author
   b. Get the list of publishers and the number of books published by each publisher
   c. Get the names of authors who jointly wrote more than one book.
   d. Get the list of books that are issued but not returned
   e. Get the list of students who reads only ‘Malayalam’ books
   f. Get the total fine collected for the current month and current quarter
   g. Get the list of students who have overdue (not returned the books even on due date)
   h. Calculate the fine (as of today) to be collected from each overdue book.
   i. Members who joined after Jan 1 2021 but has not taken any books
5. Book return should insert an entry into the Book_Return table and also update the status in Book_Issue table as ‘Returned’. Create a database *TRANSACTION* to do this operation (stored procedure).
6. Create a database view ‘Available_Books’, which will list out books that are currently available in the library
7. Create a database procedure to add, update and delete a book to the Library database (use parameters).
8. Use cursors and create a procedure to print Books Issue Register (page wise – 20 rows in a page)
9. Create a history table (you may use the same structure without any keys) for the MEMBER table and copy the original values of the row being updated to the history table using a TRIGGER.
10. NoSQL Exercise
    a. Practice Mongo DB CRUD operations. Refer: https://docs.mongodb.com/manual/crud/

b. You may use a MongoDB local installation or cloud MongoDB services like MongoDB Atlas for this exercise
c. For documentation: Refer: https://docs.mongodb.com/manual/introduction/

11.Application Development Problem examples:

1) Inventory Control System.
2) Material Requirement Processing.
3) Hospital Management System.
4) Railway Reservation System.
5) Personal Information System.
6) Web Based User Identification System.
7) Timetable Management System.
8) Hotel Management System.

SEMESTER V

MINOR

CST 381	CONCEPTS IN SOFTWARE ENGINEERING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble: This course provides fundamental knowledge in the Software Development Process. It covers Software Development, Quality Assurance and Project Management concepts. This course enables the learners to apply state of the art industry practices in Software development.
Prerequisite: Basic understanding of Object Oriented Design and Development.

Course Outcomes: After the completion of the course the student will be able to

CO1	Differentiate Traditional and Agile Software Development approaches **(Cognitive Knowledge Level: Understand)**
CO2	Prepare Software Requirement **Specification** and Software Design for a given problem. **(Cognitive Knowledge Level: Apply**)
CO3	Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project. **(Cognitive Knowledge Level: Apply**)
CO4	Make use of software project management concepts while planning, estimation, scheduling, tracking and change management of a project, with proper application of SCRUM, Kanban and Lean frameworks. (**Cognitive Knowledge Level: Apply**)
CO5	Utilize SQA practices, Process Improvement techniques and Technology improvements **namely** cloud based software model and containers & microservices in a Software Development Process. (Cognitive Knowledge Level: **Apply**)

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	✓	✓	✓	✓		✓						✓
CO2	✓	✓	✓	✓		✓				✓	✓	✓

CO3	✓	✓	✓	✓				✓		✓	✓	✓
CO4	✓	✓	✓	✓		✓			✓	✓	✓	✓
CO5	✓	✓	✓	✓		✓						✓

	Abstract POs defined by National Board of Accreditation		
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test1 (Percentage)	**Test2 (Percentage)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**

Apply	**40**	**40**	**40**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks** (Each student shall identify a software development problem and prepare Requirements Specification, Design Document, Project Plan and Test case documents for the identified problem as the assignment.)

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus

Module 1 : Introduction to Software Engineering (8 hours)

Introduction to Software Engineering - Professional software development, Software engineering ethics. Software process models - The waterfall model, Incremental development. Process activities - Software specification, Software design and implementation, Software validation, Software evolution. Coping with change - Prototyping, Incremental delivery, Boehm's Spiral Model. Agile software development - Agile methods, agile manifesto - values and principles. Agile development techniques, Agile Project Management. Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care.

Module 2 : Requirement Analysis and Design (10 hours)

Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Personas, Scenarios, User stories, Feature identification. Design concepts - Design within the context of software engineering, Design Process, Design concepts, Design Model. Architectural Design - Software Architecture, Architectural Styles, Architectural considerations, Architectural Design Component level design - What is a component?, Designing Class-Based Components, Conducting Component level design, Component level design for web-apps. Template of a Design Document as per “IEEE Std 1016-2009 IEEE Standard for Information Technology Systems Design Software Design Descriptions”. Case study: The Ariane 5 launcher failure.

Module 3 : Implementation and Testing (12 hours)

Object-oriented design using the UML, Design patterns, Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD. Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. Informal Review, Formal Technical Reviews, Post-mortem evaluations. Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, White box testing, Path testing, Control Structure testing, Black box testing, Testing Documentation and Help facilities. Test automation, Test-driven development, Security testing. Overview of DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. Software Evolution - Evolution processes, Software maintenance.

Module 4 : Software Project Management (8 hours)

Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Software pricing, Plan-driven development, Project scheduling, Agile planning. Estimation techniques, COCOMO cost modeling. Configuration management, Version management, System building, Change management, Release management, Agile software management - SCRUM framework. Kanban methodology and lean approaches.

Module 5 : Software Quality and Process Improvement (6 hours)

Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, SQA Tasks , Software measurement and metrics. Software Process Improvement(SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software.

Text Books

1. Book 1 - Ian Sommerville, Software Engineering, Pearson Education, Tenth edition, 2015.
2. Book 2 - Roger S. Pressman, Software Engineering : A practitioner's approach, McGraw Hill publication, Eighth edition, 2014
3. Book 3 - Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson Education, First Edition, 2020.

References

1. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
2. IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design—Software Design Descriptions
3. David J. Anderson, Kanban, Blue Hole Press 2010
4. David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
5. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
6. Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
7. Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
8. Henrico Dolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition
9. Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
10. StarUML documentation - https://docs.staruml.io/
11. OpenProject documentation - https://docs.openproject.org/

12. BugZilla documentation - https://www.bugzilla.org/docs/
13. GitHub documentation - https://guides.github.com/
14. Jira documentation - https://www.atlassian.com/software/jira

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the advantages of an incremental development model over a waterfall model?
2. Compare agile software development with traditional software development?

Course Outcome 2 (CO2):

1. How to prepare a software requirement specification?
2. Differentiate between Architectural design and Component level design.
3. How do agile approaches help software developers to capture and define the user requirements effectively?
4. What is the relevance of the SRS specification in software development?
5. Prepare a use case diagram for a library management system.

Course Outcome 3 (CO3):

1. Differentiate between the different types of software testing strategies.
2. What are the benefits of DevOps?
3. How do design patterns help software architects communicate the design of a complex system effectively?
4. What are the proactive approaches one can take to optimise efforts in the testing phase?

Course Outcome 4 (CO4):

1. What are the activities involved in software project management?
2. What is the need for SCRUM, Kanban and Lean methodologies?
3. What are the benefits of rolling level planning in software project management and how would you implement it?
4. How would you assess the risks in your software development project? How would you plan for risk mitigation and contingency?

Course Outcome 5 (CO5):

1. What is the importance of Software Process improvement?
2. How will retrospectives help in improving the software development process?
3. What are the important skills required for the SQA role?
4. How would you use project history data as a prediction tool to plan future projects?

Model Question Paper

QP CODE:

Reg No:______________

Name :______________

PAGES : 3

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 381

Course Name: Concepts in Software Engineering

Duration: 3 Hrs **Max. Marks : 100**

PART A

Answer all Questions. Each question carries 3 Marks

1. Explain why professional software that is developed for a customer is not simply the programs that have been developed and delivered

2. Incremental software development could be very effectively used for customers who do not have a clear idea about the systems needed for their operations. Discuss.

3. Identify and briefly describe four types of requirements that may be defined for a computer based system.

4. Describe software architecture in your own words.

5. What are the major differences between GPL and LGPL?

6. Compare between white box testing and black box testing.

7. What is the importance of risk management in software project management?

8. Explain COCOMO cost estimation model

9. Describe the software quality dilemma in your own words

10. Which are the levels of the CMMI model?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 marks)

11. (a) Compare between waterfall model and spiral model **(8)**

(b) Explain Agile methods and Agile manifesto **(6)**

OR

12. (a) Explain software process activities **(7)**

(b) Explain Agile Development techniques and Agile Project Management. **(7)**

13. (a) What are functional and nonfunctional requirements? Imagine that you are developing a library management software for your college, identify at least 8 functional requirements and 4 nonfunctional requirements. **(10)**

(b) What are the contents of a software requirement specification? **(4)**

OR

14. (a) Explain Personas, Scenarios, User stories and Feature identification? **(8)**

(b) Compare between Software Architecture design and Component level design **(6)**

15. (a) Describe the formal and informal review techniques in detail. **(6)**

(b) Explain various software testing strategies. **(8)**

OR

16. (a) Explain DevOps CI/CD/CD in detail. **(8)**

(b) Explain test driven development. **(6)**

17. (a) What is a critical path and demonstrate its significance in a project schedule with the help of a sample project schedule. **(6)**

(b) Explain plan driven development and project scheduling **(6)**

OR

18. (a) Explain the SCRUM framework. **(8)**

(b) What is algorithmic cost modeling? What problems does it suffer from when compared with other approaches to cost estimation? **(6)**

19. (a) Explain elements of Software Quality Assurance and SQA Tasks. **(8)**

(b) Explain the SPI process. **(6)**

OR

20. (a) Compare between CMMI and ISO 9001:2000 **(8)**

(b) Compare Quality Control and Quality Assurance. **(6)**

Teaching Plan [44 hours]		
	Module 1 : Introduction to Software Engineering (8 hours)	**Hours**
1.1	Introduction to Software Engineering. [Book 1, Chapter 1]	1 hour
1.2	Software process models [Book 1 - Chapter 2]	1 hour
1.3	Process activities [Book 1 - Chapter 2]	1 hour
1.4	Coping with change [Book 1 - Chapter 2, Book 2 - Chapter 4]	1 hour
1.5	Agile software development [Book 1 - Chapter 3]	1 hour
1.6	Agile development techniques [Book 1 - Chapter 3]	1 hour
1.7	Agile Project Management.[Book 1 - Chapter 3]	1 hour
1.8	Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care. [Book 1 - Chapter 1]	1 hour
	Module 2 : Requirement Analysis and Design (10 hours)	
2.1	Functional and non-functional requirements, Requirements engineering processes [Book 1 - Chapter 4]	1 hour

2.2	Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix [Book 1 - Chapter 4]	1 hour
2.3	Developing use cases, Software Requirements Specification Template [Book 2 - Chapter 8]	1 hour
2.4	Personas, Scenarios [Book 3 - Chapter 3]	1 hour
2.5	User stories, Feature identification [Book 3 - Chapter 3]	1 hour
2.6	Design concepts [Book 2 - Chapter 12]	1 hour
2.7	Architectural Design [Book 2 - Chapter 13]	1 hour
2.8	Component level design [Book 2 - Chapter 14]	1 hour
2.9	Component level design, Design Document Template. [Book 2 - Chapter 14, Ref - 2]	1 hour
2.10	Case study: The Ariane 5 launcher failure. [Book 2 - Chapter 16]	1 hour
Module 3 : Implementation and Testing (12 hours)		
3.1	Object-oriented design using the UML, Design patterns [Book 1 - Chapter 7]	1 hour
3.2	Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD [Book 1 - Chapter 7]	1 hour
3.3	Review Techniques - Cost impact of Software Defects, Code review. [Book 2 - Chapter 20]	1 hour
34	Informal Review, Formal Technical Reviews, Post-mortem evaluations. [Book 2 - Chapter 20]	1 hour
3.5	Software testing strategies [Book 2 - Chapter 22]	1 hour
3.6	Software testing strategies [Book 2 - Chapter 22]	1 hour
3.7	White box testing, Path testing, Control Structure testing [Book 2 - Chapter 23]	1 hour
3.8	Black box testing. Test documentation [Book 2 - Chapter 23]	1 hour
3.9	Test automation, Test-driven development [Book 3 - Chapter 9]	1 hour
3.10	Security testing. DevOps and Code Management [Book 3 - Chapter 9, Chapter 10]	1 hour
3.11	DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. [Book 3 - Chapter 10]	1 hour

3.12	Software Evolution - Evolution processes, Software maintenance. [Book 1 - Chapter 9]	1 hour
Module 4 : Software Project Management (8 hours)		
4.1	Software Project Management - Risk management, Managing people, Teamwork [Book 1 - Chapter 22]	1 hour
4.2	Project Planning - Software pricing, Plan-driven development, Project scheduling, Agile planning [Book 1 - Chapter 23]	1 hour
4.3	Estimation techniques [Book 1 - Chapter 23]	1 hour
4.4	Configuration management [Book 1 - Chapter 25]	1 hour
4.5	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.6	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.7	Kanban methodology and lean approaches. [Ref 9 - Chapter 2]	1 hour
4.8	Kanban methodology and lean approaches.[Ref 9 - Chapter 2]	1 hour
Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)		
5.1	Software Quality, Software Quality Dilemma, Achieving Software Quality. [Book 2 - Chapter 19]	1 hour
5.2	Elements of Software Quality Assurance, SQA Tasks [Book 3 - Chapter 21]	1 hour
5.3	Software measurement and metrics. [Book 3 - Chapter 21]	1 hour
5.4	Software Process Improvement(SPI), SPI Process[Book 2 - Chapter 37]	1 hour
5.5	Software Process Improvement(SPI), SPI Process[Book 2 - Chapter 37]	1 hour
5.6	CMMI process improvement framework, ISO 9001:2000 for Software. [Book 2 - Chapter 37]	1 hour

CST 383	CONCEPTS IN MACHINE LEARNING	Category	L	T	P	Credit	Year of introduction
		VAC	3	1	0	4	2019

Preamble: This course enables the learners to understand the fundamental concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning & the naive Bayes algorithm, support vector machines& kernels, basic clustering algorithms and dimensionality reduction methods. This course helps the students to provide machine learning based solutions to real world problems.

Prerequisite: Familiarity with basics in linear algebra, probability and Python programming.

	Course Outcomes
CO1	Illustrate Machine Learning concepts and basic parameter estimation methods.**(Cognitive Knowledge Level: Apply)**
CO2	Demonstrate supervised learning concepts (regression, linear classification). **(Cognitive Knowledge Level**: **Apply**)
CO3	Illustrate the concepts of Multilayer neural network and Support Vector Machine **(Cognitive Knowledge Level: Apply**)
CO4	Describe unsupervised learning concepts and dimensionality reduction techniques. (**Cognitive Knowledge Level**: **Apply**)
CO5	Solve real life problems using appropriate machine learning models and evaluate the performance measures (**Cognitive Knowledge Level: Apply**)

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	✓	✓	✓	✓	✓							✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓	✓							✓

CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test1 (Percentage)	**Test2 (Percentage)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module-1 (Overview of machine learning)

Machine learning paradigms-supervised, semi-supervised, unsupervised, reinforcement learning. Basics of parameter estimation - maximum likelihood estimation(MLE) and maximum a posteriori estimation(MAP). Introduction to Bayesian formulation.

Module-2 (Supervised Learning)

Regression - Linear regression with one variable, Linear regression with multiple variables, solution using gradient descent algorithm and matrix method, basic idea of overfitting in regression. Linear Methods for Classification- Logistic regression, Perceptron, Naive Bayes, Decision tree algorithm ID3.

Module-3 (Neural Networks (NN) and Support Vector Machines (SVM))

NN - Multilayer feed forward network, Activation functions (Sigmoid, ReLU, Tanh), Backpropagation algorithm.

SVM - Introduction, Maximum Margin Classification, Mathematics behind Maximum Margin Classification, Maximum Margin linear separators, soft margin SVM classifier, non-linear SVM, Kernels for learning non-linear functions, polynomial kernel, Radial Basis Function(RBF).

Module-4 (Unsupervised Learning)

Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, Expectation maximization (EM) for soft clustering. Dimensionality reduction – Principal Component Analysis, factor Analysis, Multidimensional scaling, Linear Discriminant Analysis.

Module-5 (Classification Assessment)

Classification Performance measures - Precision, Recall, Accuracy, F-Measure, Receiver Operating Characteristic Curve(ROC), Area Under Curve(AUC. Bootstrapping, Cross Validation, Ensemble methods, Bias-Variance decomposition. Case Study: Develop a classifier for face detection.

Text Book

1. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
2. Mohammed J. Zaki and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, First South Asia edition, 2016.
3. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 2016
4. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.

Reference Books

1. Christopher Bishop. Neural Networks for Pattern Recognition, Oxford University Press, 1995.
2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, MIT Press 2012.
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements Of Statistical Learning, Second edition Springer 2007.
4. P. Langley, Elements of Machine Learning, Morgan Kaufmann, 1995.
5. Richert and Coelho, Building Machine Learning Systems with Python.
6. Davy Cielen, Arno DB Meysman and Mohamed Ali. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Dreamtech Press 2016.

Sample Course Level Assessment Questions

Course Outcome1(CO1):

1. A coin is tossed 100 times and lands heads 62 times. What is the maximum likelihood estimate for ***θ,*** the probability of heads.
2. Suppose data $x_1, ..., x_n$are independent and identically distributed drawn from an exponential distribution ***exp(λ)***. Find the maximum likelihood for **λ**.
3. Suppose $x_1, ..., x_n$ are independent and identically distributed(iid) samples from a distribution with density

$$f_X(x \mid \theta) = \begin{cases} \frac{\theta x^{\theta-1}}{3^\theta}, & 0 \le x \le 3 \\ 0, & \text{otherwise} \end{cases}$$

   Find the maximum likelihood estimate(MLE) for ***θ***.
4. Find the maximum likelihood estimator (MLE) and maximum a posteriori (MAP) estimator for the mean of a univariate normal distribution. Assume that we have ***N*** samples, $x_1,..., x_N$ independently drawn from a normal distribution with known variance σ^2 and unknown mean ***μ*** and the prior distribution for the mean is itself a normal distribution with mean ***ν*** and variance β^2. What happens to the MLE and MAP estimators as the number of samples goes to infinity.

Course Outcome 2 (CO2):

1. Explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.
2. Suppose that you are asked to perform linear regression to learn the function that outputs ***y***, given the ***D***-dimensional input ***x***. You are given ***N*** independent data points, and that all the ***D*** attributes are linearly independent. Assuming that ***D*** is around 100, would you prefer the closed form solution or gradient descent to estimate the regressor?
3. Suppose you have a three class problem where class label $y \in$ 0, 1, 2and each training example ***X*** has 3 binary attributes $X_1, X_2, X_3 \in$ 0, 1. How many parameters (probability distribution) do you need to know to classify an example using the Naive Bayes classifier?

Course Outcome 3 (CO3):

1. What are support vectors and list any three properties of the support vector classifier solution?
2. Why do you use kernels to model a projection from attributes into a feature space, instead of simply projecting the dataset directly?
3. Describe how Support Vector Machines can be extended to make use of kernels. Illustrate with reference to the Gaussian kernel $K(x, y) = e^{-z}$,where $z = (x-y)^2$.

4. Briefly explain one way in which using tanh instead of logistic activations makes optimization easier.
5. ReLU activation functions are most used in neural networks instead of the tanh activation function. Draw both activation functions and give a) an advantage of the ReLU function compared to the tanh function. b) a disadvantage of the ReLU function compared to the tanh function.

Course Outcome 4(CO4):

1. Describe cluster analysis? Identify two applications where cluster analysis can be applied to multimedia data?
2. Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

(i) Compute the Euclidean distance between the two objects.

(ii) Compute the Manhattan distance between the two objects.

3. Use PCA to reduce the dimension from 2 to 1 for the design matrix ***X***.

$$X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$$

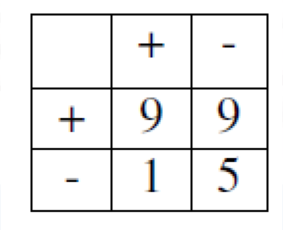
4. What is Principal Component Analysis (PCA)? Which eigen value indicates the direction of largest variance?
5. Suppose that one runs a principal component analysis on a data set and tells that the percentage of variance explained by the first 3 components is 80%. How is this percentage of variance explained?

Course Outcome 5 (CO5):

1. Suppose that you are contacted by a food processing company that wants you to develop a classifier that detects whether a rat is present in an image. You collect a large dataset of images by crawling the web, and have annotators determine which images contain rats. This set of images can then be used as the training set for your classifier.
   a. Suggest a machine learning method to use for this classification task and evaluate its performance.
   b. After you have delivered your solution to the company, they get back to you and complain that when they evaluate on a new test set, they get precision and recall values that are much lower than what you reported to them. Explain what might have gone wrong and propose remedial measures .
2. A real estate firm would like to build a system that predicts the sale prices of a house. They create a spreadsheet containing information about 1,500 house sales in the Kochi

area. In addition to the price, there are 10 features describing the house, such as number of bedrooms, total indoor area, lot area, a swimming pool, location, etc. Explain how you would implement a machine learning model that would solve this prediction task. Give all steps you would carry out when developing it. Explain why the model you built is probably useless in the long run.

3. For a classifier, the confusion matrix is given by:



	+	-
+	9	9
-	1	5

What is the precision, recall and accuracy of that classifier?

Model Question Paper

QP CODE: PAGES:3

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 383

Course Name: CONCEPTS IN MACHINE LEARNING

Max.Marks:100 Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Define supervised learning? Name special cases of supervised learning depending on whether the inputs/outputs are categorical, or continuous.
2. Differentiate between Maximum Likelihood estimation (MLE) and Maximum a Posteriori (MAP) estimation?
3. What is overfitting and why is it a problem? Give an example of a method to reduce the risk of overfitting.
4. Specify the basic principle of gradient descent algorithm.
5. Suppose that you have a linear support vector machine(SVM) binary classifier. Consider a point that is currently classified correctly, and is far away from the decision boundary. If you

remove the point from the training set, and re-train the classifier, will the decision boundary change or stay the same? Justify your answer.

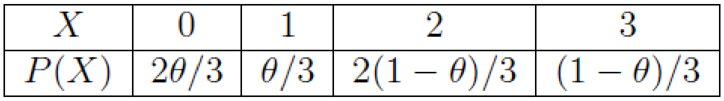
6. Mention the primary motivation for using the kernel trick in machine learning algorithms?
7. Expectation maximization (EM) is designed to find a maximum likelihood setting of the parameters of a model when some of the data is missing. Does the algorithm converge? If so, do you obtain a locally or globally optimal set of parameters?
8. Illustrate the strength and weakness of k-means algorithm.
9. Classifier A attains 100% accuracy on the training set and 70% accuracy on the test set. Classifier B attains 70% accuracy on the training set and 75% accuracy on the test set. Which one is a better classifier. Justify your answer.
10. How does bias and variance trade-off affect machine learning algorithms?

(10x3=30)

Part B

Answer any one Question from each module. Each question carries 14 Marks

11. a) Suppose that ***X*** is a discrete random variable with the following probability mass function: where ***0 ≤ θ ≤ 1*** is a parameter. The following 10 independent observations



X	0	1	2	3
$P(X)$	$2\theta/3$	$\theta/3$	$2(1-\theta)/3$	$(1-\theta)/3$

were taken from such a distribution: ***(3, 0, 2, 1, 3, 2, 1, 0, 2, 1)***. What is the maximum likelihood estimate of ***θ***. **(6)**

b) A gamma distribution with parameters ***α, β*** has the following density function, where ***Γ(t)*** is the gamma function.

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

If the posterior distribution is in the same family as the prior distribution, then we say that the prior distribution is the conjugate prior for the likelihood function. Using the Gamma distribution as a prior, show that the Exponential distribution is a conjugate prior of the Gamma distribution. Also, find the maximum a posteriori estimator for the parameter of the Exponential distribution as a function of ***α*** and ***β***. **(8)**

OR

12. a) Traffic between 8AM and 9AM at a certain place was measured by counting the number of vehicles that passed at that time. Suppose the counts follow a Poisson process. A random sample of 9 observations was collected, having observed the following number of vehicles: (95, 100, 80, 70, 110, 98, 97, 90, 70). Derive the maximum likelihood estimator for the

average number of vehicles that pass by that place between 8 AM and 9 AM, and compute the corresponding estimate using the given sample. **(7)**

b) Find the maximum a posteriori (MAP) estimator for the mean of a univariate normal distribution. Assume that we have ***N*** samples, $\boldsymbol{x_1,\ldots, x_N}$ independently drawn from a normal distribution with known variance $\boldsymbol{\sigma^2}$ and unknown mean $\boldsymbol{\mu}$ and the prior distribution for the mean is itself a normal distribution with mean $\boldsymbol{v}$ and variance $\boldsymbol{\beta^2}$. **(7)**

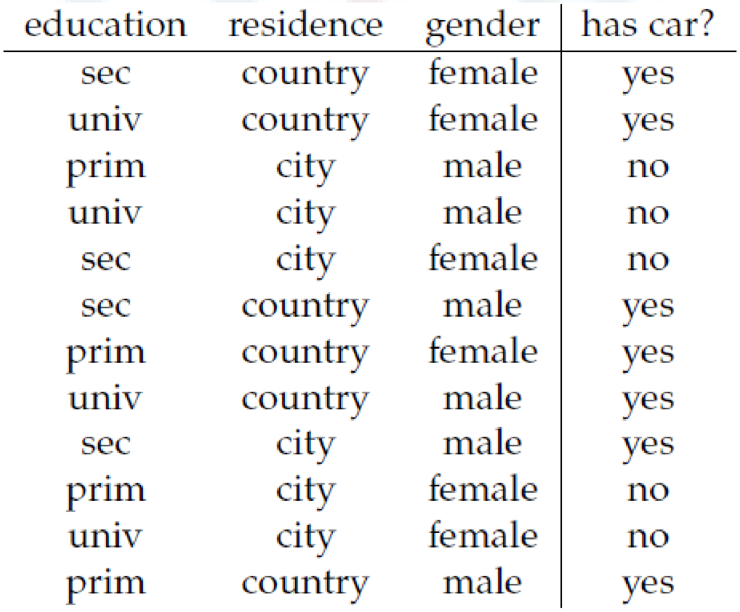
13.a) Derive the gradient descent training rule assuming for the target function$\boldsymbol{o_d = w_0 + w_1x_1 + \ldots + w_nx_n}$. Define explicitly the squared cost/error function ***E***, assuming that a set of training examples ***D*** is provided, where each training example $\boldsymbol{d \in D}$is associated with the target output $\boldsymbol{t_d}$. **(10)**

b) How can we interpret the output of a two-class logistic regression classifier as a probability? **(4)**

OR

14. a) In a two-class logistic regression model, the weight vector $\boldsymbol{w = [4, 3, 2, 1, 0]}$. We apply it to some object that we would like to classify; the vectorized feature representation of this object is $\boldsymbol{x = [-2, 0, -3, 0.5, 3]}$. What is the probability, according to the model, that this instance belongs to the positive class? **(6)**

b) The following dataset can be used to train a classifier that determines whether a given person is likely to own a car or not. There are three features: education level (primary, secondary, or university); residence (city or country); gender (female, male).



education	residence	gender	has car?
sec	country	female	yes
univ	country	female	yes
prim	city	male	no
univ	city	male	no
sec	city	female	no
sec	country	male	yes
prim	country	female	yes
univ	country	male	yes
sec	city	male	yes
prim	city	female	no
univ	city	female	no
prim	country	male	yes

Find the root attribute and justify your answer **(8)**

15. a) Consider a support vector machine whose input space is 2-D, and the inner products are computed by means of the kernel ***K(x, y) = (x.y + 1)²-1***, where ***x.y***denotes the ordinary inner product. Show that the mapping to feature space that is implicitly defined by this kernel is the mapping to 5-D given by **(10)**

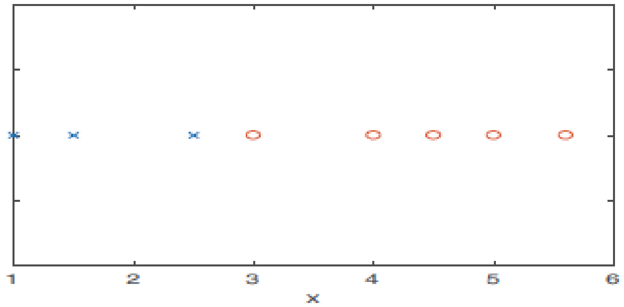
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2}\, x_1 \\ \sqrt{2}\, x_2 \end{bmatrix}.$$

b) What is the basic idea of a Support Vector Machine? **(4)**

OR

16. a) Explain how back propagation can be used to solve XOR problem which is not linearly separable. **(8)**

b) Consider the following one dimensional training data set, ’x’ denotes negative examples and ’o’ positive examples. The exact data points and their labels are given in the table. Suppose a SVM is used to classify this data. Indicate which are the support vectors and mark the decision boundary. Find the equation of the hyperplane. (6)



x	1	1.5	2.5	3	4	4.5	5	5.6
y	-1	-1	-1	1	1	1	1	1

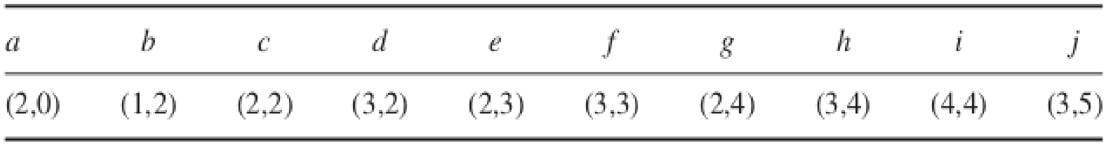
17. a)Suppose that we have the following data (one variable). Use single linkage Agglomerative clustering to identify the clusters.
Daa: (2, 5, 9, 15, 16, 18, 25, 33, 33, 45). **(8)**

b) Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):
(i) Compute the Euclidean distance between the two objects.
(ii) Compute the Manhattan distance between the two objects.

(iii) Compute the Minkowski distance between the two objects, using p = 3 **(6)**

OR

18. a) Suppose that we have the following data:



a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*
(2,0)	(1,2)	(2,2)	(3,2)	(2,3)	(3,3)	(2,4)	(3,4)	(4,4)	(3,5)

Identify the cluster by applying the k-means algorithm, with k = 2. Try using initial cluster centers as far apart as possible. **(10)**

b) List the steps involved in Principal Component Analysis. **(4)**

19. a) Suppose the dataset had 9700 cancer-free images from 10000 images from cancer patients. Find precision, recall and accuracy ? Is it a good classifier? Justify. **(8)**

Actual Class\ Predicted class	**cancer = yes**	**cancer = no**	**Total**
cancer = yes	90	210	300
cancer = no	140	9560	9700
Total	230	9770	10000

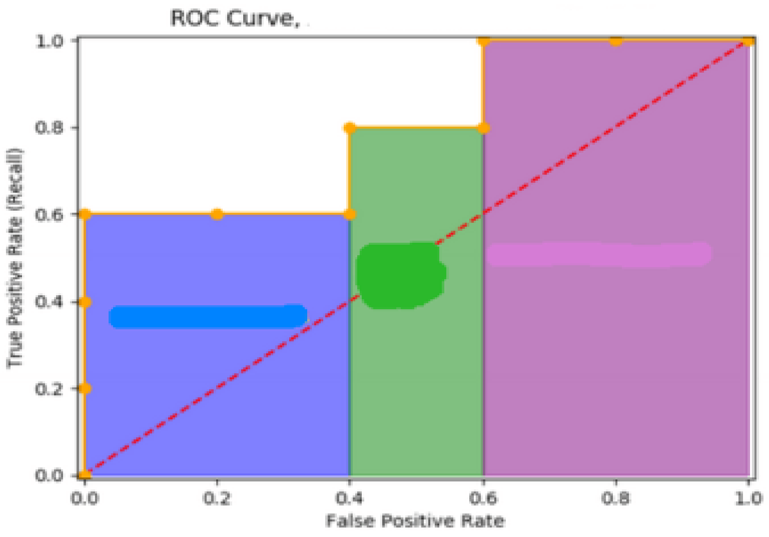
b) Suppose that you have a classification problem where our feature representation contains about 10,000,000 features. We would like to develop a classifier that can be deployed in a mobile phone, so preferably it should have a small memory footprint. Discuss one solution for how this can be done. **(6)**

OR

20. a) What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why? **(6)**

b)Suppose there are three classifiers A,B and C. The (FPR, TPR) measures of the three classifiers are as follows – A (0, 1), B (1, 1) , C (1,0.5). Which can be considered as a perfect classifier? Justify your answer. **(4)**

c) Given the following ROC Curve? Find the AUC? **(4)**



Teaching Plan

No	**Contents**	**No of Lecture Hrs**
	Module 1 :Overview of machine learning (7 hours)	
1.1	Supervised, semi-supervised, unsupervised learning, reinforcement learning (Text Book (TB) 1: Chapter 1)	1hour
1.2	Maximum likelihood estimation(MLE) (TB 1: Section 4.2)	1hour
1.3	Maximum likelihood estimation (MLE)- example (TB 1: Section 4.2)	1hour
1.4	Maximum a posteriori estimation(MAP) (TB 4: Section 6.2)	1hour
1.5	Maximum a posteriori estimation(MAP)-example (TB 4: Section 6.2)	1hour
1.6	Bayesian formulation (TB 1: Section 14.1, 14.2)	1hour
1.7	Bayesian formulation -example (TB 1: Section 14.1, 14.2)	1hour
	Module 2 : Supervised Learning (8 hours)	

2.1	Linear regression with one variable (TB 1: Section 2.6)	1hour
2.2	Multiple variables, Solution using gradient descent algorithm and matrix method (No derivation required) (TB 1: Section 5.8)	1hour
2.3	Overfitting in regression, Lasso and Ridge regularization	1hour
2.4	Logistic regression	1hour
2.5	Perceptron	1hour
2.6	Naive Bayes (TB 2: Section 18.2)	1hour
2.7	Decision trees (TB 2: Chapter 19)	1hour
2.8	Decision trees- ID3 algorithm (TB 2: Chapter 19)	1hour
Module 3 : Neural Networks and Support Vector Machines (TB 2: Chapter 21) (11 hours)		
3.1	Multilayer Feed forward Network, Activation Functions (Sigmoid, ReLU, Tanh)	1hour
3.2	Back Propagation Algorithm	1hour
3.3	Illustrative Example for Back Propagation	1hour
3.4	Introduction, Maximum Margin Hyperplane,	1hour
3.5	Mathematics behind Maximum Margin Classification	1hour
3.6	Formulation of maximum margin hyperplane and solution	1hour
3.7	Soft margin SVM	1hour
3.8	Solution of Soft margin SVM	1hour
3.9	Non-linear SVM	1hour
3.10	Kernels for learning non-linear functions and properties of kernel functions.	1hour
3.11	Example Kernels functions- Linear, RBF, Polynomial.	1hour
Module 4 : Unsupervised Learning (10 hours)		
4.1	Similarity measures- Minkowski distance measures(Manhattan, Euclidean), Cosine Similarity	1hour
4.2	Clustering - Hierarchical Clustering (TB 2: Chapter 14)	1hour
4.3	K-means partitional clustering (TB 2: Chapter 13)	1hour
4.4	Expectation maximization (EM) for soft clustering (TB 2: Chapter 13)	1hour
4.5	Expectation maximization (EM) for soft clustering (TB 2: Chapter 13)	1hour

4.6	Dimensionality reduction – Principal Component Analysis (TB 1: Section 6.3)	1hour
4.7	Dimensionality reduction – Principal Component Analysis (TB 1: Section 6.3)	1hour
4.8	Factor Analysis (TB 1: Section 6.4)	1hour
4.9	Multidimensional scaling (TB 1: Section 6.5)	1hour
4.10	Linear Discriminant Analysis (TB 1: Section 6.6)	1hour
Module 5 : Classification Assessment (8 hours)		
5.1	Performance measures - Precision, Recall, Accuracy, F-Measure, ROC, AUC. (TB 2: Chapter 22.1)	1hour
5.2	Boot strapping, Cross validation	1hour
5.3	Ensemble methods- bagging	1hour
5.4	Ensemble methods- boosting	1hour
5.5	Bias-Variance decomposition (TB 2: Chapter 22.3)	1hour
5.6	Bias-Variance decomposition (TB 2: Chapter 22.3)	1hour
5.7	Face detection (TB 3: Chapter 5 Section Application: A Face Detection Pipeline)	1hour
5.8	Face detection (TB 3: Chapter 5 Section Application: A Face Detection Pipeline)	1hour

CST 385	CLIENT SERVER SYSTEMS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

The syllabus is prepared with the view of preparing the Engineering Graduates to build effective Client/Server applications. This course aims at providing a foundation in decentralized computer systems, using the client/server model. The course content is decided to cover the essential fundamentals which can be taught within the given slots in the curriculum.

Prerequisite: **Basic knowledge in Computer**

Course Outcomes: After the completion of the course the student will be able to

	Course Outcomes
CO 1	Identify the basics of client/server systems and the driving force behind the development of client/server systems**(Cognitive Knowledge Level: Understand)**
CO 2	Outline the architecture and classifications of client/server systems**(Cognitive Knowledge Level: Understand)**
CO 3	Summarize the client/server network services for an application**(Cognitive Knowledge Level: Understand)**
CO 4	Identify management services and issues in network **(Cognitive Knowledge Level: Understand)**
CO 5	Outline the Client/Server technology in respect of databases and Client/Server database architecture **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓										✓
CO3	✓	✓										✓
CO4	✓											✓
CO5	✓	✓										✓

Abstract POs defined by National Board of Accreditation

PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test 1 (Percentage)	**Test 2 (Percentage)**	
Remember	**40**	**40**	**40**
Understand	**40**	**40**	**40**
Apply	**20**	**20**	**20**
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Test 1 (for theory, for 2 hrs) : **20 marks**

Continuous Assessment Test 2 (for lab, internal examination, for 2hrs) : **20 marks**

Internal Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 5 questions with 2 questions from each module (2.5 modules x 2 = 5), having 3 marks for each question. Students should answer all questions. Part B also contains 5 questions with 2 questions from each module (2.5 modules x 2 = 5), of which a student should answer any one. The questions should not have sub-divisions and each one carries 7 marks.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Computing in client server architecture over Mainframe architecture has certain advantages and disadvantages. Describe at least three advantages and disadvantages for each architecture.

Course Outcome 2 (CO2):

1. Explain the role of mainframe-centric model in Client/Server computing?

Course Outcome 3(CO3):

1. Describe the client server system development methodology? Explain different phases of System Integration Life-Cycle.

Course Outcome 4 (CO4):

1. Explain about network management and remote system management. How can security be provided to the network?

Course Outcome 5 (CO5):

1. Explain various types of Client/Server Database Architecture

Syllabus

Module – 1 (Introduction)

Introduction to Client/Server computing - Basic Client/Server Computing Model, Server for Every Client- File Server, Print Server, Application Server, Mail Server, Directory Services Server, Web Server, Database Server, Transaction Servers. Client/Server-Fat or Thin, Stateless or Stateful, Servers and Mainframes, Client/Server Functions. Driving Forces behind Client/Server Computing- Business Perspective, Technology Perspective.

Module -2 (Client/Server Classification)

Client/Server Types-Single Client/Single Server, Multiple Clients/Single Server, Multiple Clients/Multiple Servers, Integration With Distributed Computing, Alternatives To Client/Server Systems. Classification of Client/Server Systems- Two-Tier Computing, Middleware, Three-Tier Computing- Model View Controller (MVC), Principles behind Client/Server Systems.Client/Server Topologies. Existing Client/Server Architecture. Architecture for Business Information System.

Module -3 (Client/Server Application Components)

Client- Services, Request for services, RPC, Windows services, Print services, Remote boot services, other remote services, Utility Services. Server- Detailed server functionality, Network operating system, Available platforms, Server operating system. Organizational Expectations, Improving performance of client/server applications, Single system image, Downsizing and Rightsizing, Advantages and disadvantages of Client/Server computing, Applications of Client/Server.

Module -4 (Client/ Server Systems Services and Support)

Services and Support- System administration, Availability, Reliability, Scalability, Observability, Agility, Serviceability. Software Distribution, Performance, Network management. Remote Systems Management- RDP, Telnet, SSH, Security. LAN and Network Management issues.

Module -5(Client/Server Technology and Databases)

Client/Server Technology and Databases **-** Storing Data, Database System Architectures. Client/Server In Respect Of Databases- Client/Server Databases, Client/Server Database Computing, Database Computing Vs. Mainframe, PC/File Server Computing. Client/Server Database Architecture - Process-Per-Client Architecture, Multi-Threaded Architecture, Hybrid Architecture. Database Middleware Component - Application Programming Interface, Database Translator, Network Translator.

Text Book

1. Patrick Smith &Steve Guengerich, Client / Server Computing, PHI

2. Subhash Chandra Yadav, Sanjay Kumar Singh, An Introduction to Client/Server Computing, New Age International Publishers

Reference Books

1. Jeffrey D.Schank, "Novell's Guide to Client-Server Application & Architecture" Novell Press

2. Robert Orfali, Dan Harkey, Jeri Edwards, Client/Server Survival Guide, Wiley-India Edition, Third Edition

3. Dawna Travis Dewire, Client Server Computing — McGraw Hill

4. W.H.Inman, Developing Client Server Applications, BPB

Model Question Paper

QP CODE: **PAGES: ___**

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 385

Course Name : Client Server Systems

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Differentiate between Stateful and Stateless servers

2. List the different phases and activities of client/server system development methodology.

3. How does transmission protocol work in client/server applications?

4. List any six services in single system image environment.

5. Specify the role of the client in Client/Server computing and also list any six services provided by the client.

6. Why do most RPC system support call by value semantics for parameter passing?

7. What do you mean by a thin client network? List three advantages of the Thin

Client Network system.

8. How are connectivity and interoperability between .client/server achieved?

9. One disadvantage of the Client/Server system is lack of control in a Database Management environment. Justify.

10. Explain the DBMS concept in client/server architecture.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Differentiate between Transaction server and Data server system with examples. **(7)**

(b) Computing in client server architecture over Mainframe architecture has certain advantages and disadvantages. Describe at least three advantages and disadvantages for each architecture. **(7)**

OR

12. (a) Explain various Clients/Server system development tools. **(6)**

(b) Classify and describe the driving forces that drive the move to Client/Server computing. **(8)**

13. (a) Explain the role of mainframe-centric model in Client/Server computing? **(5)**

(b) Describe the three types of Client/Server systems in existence **(9)**

OR

14. (a) List and explain the general forces behind the architecture for business information systems **(7)**

(b) Explain the different distribution styles. **(7)**

15. (a) Illustrate the concept of rightsizing and downsizing in Client/Server Computing **(7)**

(b) What is client server system development methodology? Explain the **(7)** different phases of System Integration Life-Cycle.

OR

16. (a) In Client/Server computing, explain the following with examples **(10)**

    i. Dynamic Data Exchange
    ii. RPC, Remote Procedure Call
    iii. Remote Boot Service
    iv. Diskless Computer
    v. Object-linking and embedding

    (b) Explain the functions and features of Network Operating System **(4)**

17. (a) Explain about network management and remote system management. How can security be provided to the network ? **(10)**

    (b) In client server architecture, what do you mean by Availability, Reliability, Serviceability and Security? Explain with examples. **(4)**

OR

18. (a) Client server is modular infrastructure, this is intended to improve Usability, Flexibility, Interoperability and Scalability. Explain each term with an example, in each case how it helps to improve the functionality of client server architecture. **(7)**

    (b) Explain about network management and remote system management. How can security be provided to network? **(7)**

19. (a) Explain the different types of Client/Server Database Architecture **(9)**

    (b) List and explain the main components of Database middleware **(5)**

OR

20. (a) Discuss types of database utilities, tools and their functions **(7)**

    (b) Discuss about the role of traditional and web databases in handling client/server based applications. **(7)**

Teaching Plan

Module- 1(Introduction)		**(10 hours)**
1.1	Basic Client/Server Computing Model	1 hour
1.2	Server for Every Client- File Server, Print Server	1 hour
1.3	Application Server, Mail Server, Directory Services Server	1 hour
1.4	Web Server, Database Server	1 hour
1.5	Transaction Servers	1 hour
1.6	Client/Server-Fat or Thin	1 hour
1.7	Stateless or Stateful	1 hour
1.8	Servers and Mainframes	1 hour
1.9	Client/Server Functions	1 hour
1.10	Driving Forces behind Client/Server Computing- Business Perspective, Technology Perspective	1 hour
Module- 2 (Client/Server Classification)		**(10 hours)**
2.1	Client/Server Types-Single Client/Single Server	1 hour
2.2	Multiple Clients/Single Server, Multiple Clients/Multiple Servers	1 hour
2.3	Integration With Distributed Computing	1 hour
2.4	Alternatives To Client/Server Systems	1 hour
2.5	Classification of Client/Server Systems- Two-Tier Computing, Middleware	1 hour
2.6	Three-Tier Computing- Model View Controller (MVC)	1 hour
2.7	Principles behind Client/Server Systems.	1 hour
2.8	Client/Server Topologies	1 hour
2.9	Existing Client/Server Architecture	1 hour
2.10	Architecture for Business Information System	1 hour
Module -3 (Client/Server Application Components)		**(9 hours)**
3.1	The client: Services, Request for services, RPC	1 hour
3.2	Windows services, Print services, Remote boot services	1 hour

3.3	Utility Services & Other Services	1 hour
3.4	Server- Detailed server functionality, Network operating system	1 hour
3.5	Available platforms, Server operating system	1 hour
3.6	Organizational Expectations, Improving performance of client/server applications	1 hour
3.7	Single system image, Downsizing and Rightsizing	1 hour
3.8	Advantages and disadvantages of Client/Server computing	1 hour
3.9	Applications of Client/Server	1 hour
Module -4 (Client/ Server Systems Services and Support)		**(8 hours)**
4.1	Services and Support, System administration	1 hour
4.2	Availability, Reliability	1 hour
4.3	Scalability, Observability, Agility	1 hour
4.4	Serviceability, Software Distribution	1 hour
4.5	Performance	1 hour
4.6	Network management	1 hour
4.7	Remote Systems Management- RDP, Telnet, SSH	1 hour
4.8	Security, LAN and Network Management issues	1 hour
Module -5(Client/Server Technology and Databases)		**(8 hours)**
5.1	Client/Server Technology and Databases - Storing Data	1 hour
5.2	Database System Architectures	1 hour
5.3	Client/Server In Respect Of Databases- Client/Server Databases	1 hour
5.4	Client/Server Database Computing	1 hour
5.5	Database Computing Vs. Mainframe, PC/File Server Computing	1 hour
5.	Client/Server Database Architecture - Process-Per-Client Architecture	1 hour
5.7	Multi-Threaded Architecture, Hybrid Architecture	1 hour
5.8	Database Middleware Component - Application Programming Interface, Database Translator, Network Translator	1 hour

SEMESTER V

HONOURS

CST 393	CRYPTOGRAPHIC ALGORITHMS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

The course on Cryptographic Algorithms aims at exploring various algorithms deployed in offering confidentiality, integrity, authentication and non-repudiation services. This course covers classical encryption techniques, symmetric and public key crypto-system, key exchange and management, and authentication functions. The concepts covered in this course enable the learners in effective use of cryptographic algorithms for real life applications.

Prerequisite: A sound background in Number Theory**.**

Course Outcomes: After the completion of the course the student will be able to

CO1	Identify the security services provided for different types of security attacks. (**Cognitive Knowledge Level : Understand**)
CO2	Summarize the classical encryption techniques for information hiding. (**Cognitive Knowledge Level: Apply**)
CO3	Illustrate symmetric / asymmetric key cryptographic algorithms for secure communication.(**Cognitive Knowledge Level: Apply**)
CO4	Interpret key management techniques for secure communication.(**Cognitive Knowledge Level: Understand)**
CO5	Summarize message authentication functions in a secure communication scenario.(**Cognitive Knowledge Level: Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓

CO2	✓	✓	✓		✓	✓						✓
CO3	✓	✓	✓	✓	✓	✓						✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examinati on Marks**
	Test1 (Percentage)	**Test2 (Percent**	

		age)	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Introduction to the Concepts of Security)

Need for security, Security approaches, Principles of security, Types of attacks, OSI Security Architecture, Classical encryption techniques - Substitution techniques, Transposition techniques. Stream cipher, Block cipher, Public key cryptosystems vs. Symmetric key cryptosystems, Encrypting communication channels.

Module-2 (Symmetric Key Cryptosystems)

Overview of symmetric key cryptography, Block cipher principles, Data Encryption Standard (DES), Differential and Linear cryptanalysis, Double DES, Triple DES, International Data Encryption Algorithm (IDEA), Advanced Encryption Algorithm (AES),Block cipher modes of operation, Stream cipher, RC4.

Module-3 (Public Key Cryptosystems)

Principles of public key cryptosystems, RSA algorithm, RSA illustration, Attacks, ElGamal cryptographic system, Knapsack algorithm, Diffie-Hellman key exchange algorithm, Elliptical curve cryptosystems.

Module-4 (Key Management)

Symmetric key distribution using symmetric encryption, Symmetric key distribution using asymmetric encryption, Distribution of public keys, Generating keys, transferring keys, Verifying keys, Updating keys, Storing keys, Backup keys, Compromised keys, Public key infrastructure.

Module – 5 (Authentication)

Authentication requirements, Authentication functions, Message authentication codes (MAC), Hash functions, Security of Hash functions and MAC, Message Digest 5 (MD5), Secure Hash Algorithm (SHA)-512, Hash-based Message Authentication Code (HMAC), Cipher-based Message Authentication Code (CMAC), X.509 Authentication services.

Text Books

1. William Stallings, Cryptography and Network Security Principles and Practice, Pearson Edu, 6e.
2. Bruice Schneier, Applied Cryptography Protocols, Algorithms and source code in C, Wiley,2e.

References

1. Behrouz A. Forouzan, Cryptography and Network Security, McGraw Hill, 2e.

2. Johannes A. Buchmann, Introduction to Cryptography, Springer, 2e.

3. DouglasR. Stinson, Cryptography Theory and Practice, 3e,Chapman & Hall/CRC, 2006.

4. Bernard Menezes, Network Security and Cryptography, Cengage Learning, 2011.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Consider an automated teller machine (ATM) in which users provide a personal identification number (PIN) and a card for account access. Give examples of confidentiality, integrity, and availability requirements associated with the system and, in each case, indicate the degree of importance of the requirement.

2. Discuss the different security services provided for preventing security attacks.

Course Outcome 2 (CO2):

1. The encryption key in a transposition cipher is (3,2,6,1,5,4). Find the decryption key

2.Discuss the process of encryption in Vernam cipher

Course Outcome 3 (CO3):

1. Devise a meet-in-the-middle attack for a triple DES.

2. Write an algorithm for the InvSubBytes transformation and implement using python (**Assignment**)
3. Consider the following elliptic curve signature scheme. We have a global elliptic curve, prime *p*, and "generator" G. Alice picks a private signing key X_A and forms the public verifying $Y_A = X_AG$. To sign a message *M*:
   - Alice picks a value k
   - Alice sends Bob *M*, k and the signature $S = M - kX_AG$.
   - Bob verifies that $M=S+kY_A$.

   Show that the verification process produces an equality if the signature is valid.
4. Write an algorithm to add two points on an elliptic curve over GF(*p*) and implement using Python. **(Assignment)**
5. Write an algorithm for encryption using knapsack cryptosystem and implement using Java. **(Assignment)**

Course Outcome4 (CO4):

1. List four general categories of schemes for the distribution of public keys.
2. What are the essential ingredients of a public-key directory?

Course Outcome 5 (CO5):

1. State the value of the length field in SHA-512 if the length of the message is 1919 bits and 1920 bits.
2. Write an algorithm in pseudo code for HMAC and implement using Python (**Assignment**)

Model Question Paper

QP CODE:
Reg No:______________
Name :______________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(HONORS), MONTH & YEAR

Course Code: CST 393

Course Name: Cryptographic Algorithms

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. State the two approaches in attacking a cipher.

2. Define Substitution Cipher. Encrypt using one time pad M = HONORS and K = CIPHER.

3. Specify the purpose of S-Boxes in Data Encryption Standard (DES).

4. Differentiate between diffusion and confusion.

5. Perform encryption using RSA Algorithm for the following *p*=7; *q*=11; *e*=13; *M*=5.

6. Is Diffie-Hellman key exchange protocol vulnerable? Justify.

7. List the techniques for distribution of public keys.

8. Define a certificate authority and its relation to public key cryptography.

9. Distinguish between integrity and message authentication.

10. What types of attacks are addressed by message authentication?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) With a neat sketch, Explain OSI Security architecture model. **(8)**

(b) How does link encryption differ from end-to-end encryption? Explain. **(6)**

OR

12. (a) Encrypt the text “cryptography” using the Hill Cipher with the key $\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}$. Show the calculations. **(8)**

(b) Illustrate the steps involved in encrypting a plain text using playfair cipher with an example. **(6)**

13. (a) With a neat sketch, explain a single round in DES. **10**

(b) Explain encryption and decryption using 2 keys and 3 keys of triple DES. **(4)**

OR

14. (a) Explain the block cipher modes i) Cipher feedback mode ii) Output feedback mode. **(8)**

(b) Describe the four types of transformations in AES. **(6)**

15. (a) Write an algorithm for generating public and private key using Elliptical curve cryptography. **(10)**

(b) The equation $y^2=x^3+x+1$, the calculation is done modulo 13. Add two points R= P+Q, where P= (4,2) and Q= (10,6). **(4)**

OR

16. User A and B use the Diffie-Hellman key exchange technique with a common prime q=71 and primitive root alpha=7.

(a) If user A has private key X_A =3, What is A's public key Y_A? **(7)**

(b) If user B has private key X_B =6, What is A's public key Y_B? **(7)**

17. (a) Define a session key and show how a KDC can create can create a session key between Alice and Bob. **(7)**

(b) What are the requirements for the use of a public-key certificate scheme? **(7)**

OR

18. (a) What are the core components of a PKI? Briefly describe each component. **(8)**

(b) Describe the following (i) Updating keys (ii) Compromised Keys. **(6)**

19. (a) Describe how SHA-512 logic produce message digest **(10)**

(b) Distinguish between HMAC and CMAC **(4)**

OR

20. (a) Specify the format for X.509 certificate. Explain the steps required to obtain user's certificate. **(7)**

(b) With suitable block diagrams, explain the types of functions that may be used to produce an authenticator. **(8)**

Teaching Plan

No	Contents	No of Lecture Hrs
	Module - 1 (Introduction to the Concepts of Security) (9 hrs)	
1.1	Need for security, Security approaches	1 hour
1.2	Principles of security, Types of attacks	1 hour
1.3	OSI Security Architecture	1 hour
1.4	Classical encryption techniques: Substitution techniques(Caesar cipher, Monoalphabetic cipher, Playfair cipher)	1 hour
1.5	Classical encryption techniques: Substitution techniques (Hill cipher, Polyalphabetic cipher, One-time pad)	1 hour
1.6	Classical encryption techniques: Transposition techniques	1 hour
1.7	Stream cipher, Block cipher	1 hour
1.8	Public- key cryptosystems vs. Symmetric key cryptosystems	1 hour
1.9	Encrypting communication channels	1 hour
	Module - 2 (Symmetric key cryptosystems) (11 hrs)	
2.1	Overview of symmetric key cryptography	1 hour
2.2	Block cipher principles	1 hour
2.3	Data Encryption Standard (DES)	1 hour
2.4	DES design criteria	1 hour
2.5	Differential and Linear cryptanalysis	1 hour
2.6	Double DES, Triple DES	1 hour

2.7	IDEA	1 hour
2.8	Advanced Encryption Algorithm (AES structure)	1 hour
2.9	Advanced Encryption Algorithm (Transformations)	1 hour
2.10	Block cipher modes of operation	1 hour
2.11	Stream cipher, RC4	1 hour
Module - 3 (Public key cryptosystems) (8 hrs)		
3.1	Principles of public key cryptosystems	1 hour
3.2	RSA algorithm	1 hour
3.3	RSA illustration, Attacks	1 hour
3.4	ElGamal cryptographic system	1 hour
3.5	Knapsack algorithm	1 hour
3.6	Diffie-Hellman key exchange algorithm	1 hour
3.7	Elliptical curve cryptosystems(Elliptical curve arithmetic)	1 hour
3.8	Elliptical curve cryptosystems (Elliptical curve algorithm)	1 hour
Module - 4 (Key Management) (8 hrs) [Text book-2]		
4.1	Symmetric key distribution using symmetric encryption	1 hour
4.2	Symmetric key distribution using asymmetric encryption	1 hour
4.3	Distribution of public keys	1 hour
4.4	Generating keys, Transferring keys	1 hour

4.5	Verifying keys, Updating keys	1 hour
4.6	Storing keys, Backup keys	1 hour
4.7	Compromised keys	1 hour
4.8	Public key infrastructure	1 hour
	Module - 5 (Authentication) (9 hrs)	
5.1	Authentication requirements	1 hour
5.2	Authentication functions	1 hour
5.3	Message Authentication Codes (MAC)	1 hour
5.4	Hash functions	1 hour
5.5	Security of Hash functions and MAC	1 hour
5.6	MD5	1 hour
5.7	SHA-512	1 hour
5.8	HMAC, CMAC	1 hour
5.9	X.509 Authentication services	1 hour

CST 395	NEURAL NETWORKS AND DEEP LEARNING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

Neural networks is a biologically inspired programming paradigm which enables a computer to learn from observational data and deep learning is a powerful set of techniques for training neural networks. This course introduces the key concepts in neural networks, its architecture and learning paradigms, optimization techniques, basic concepts in deep learning, Convolutional Neural Networks and Recurrent Neural Networks. The students will be able to provide best solutions to real world problems in domains such as computer vision and natural language processing.

Prerequisite: A Sound knowledge in Computational fundamentals of machine learning

Course Outcomes: After the completion of the course the student will be able to

CO1	Demonstrate the basic concepts of machine learning models and performance measures. **(Cognitive Knowledge Level : Understand)**
CO2	Illustrate the basic concepts of neural networks and its practical issues**(Cognitive Knowledge Level : Apply)**
CO3	Outline the standard regularization and optimization techniques for deep neural networks (**Cognitive Knowledge Level : Understand)**
CO4	Build CNN and RNN models for different use cases. **(Cognitive Knowledge Level : Apply)**
CO5	Explain the concepts of modern RNNs like LSTM, GRU **(Cognitive Knowledge Level : Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (%)	Test2 (%)	
Remember	**30**	**30**	**30**
Understand	**40**	**40**	**40**
Apply	**30**	**30**	**30**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern

Attendance	**10 marks**
Continuous Assessment Tests	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum 2 subdivisions and carry 14 marks.

Syllabus

Module - 1 (Basics of Machine Learning)

Machine Learning basics - Learning algorithms - Supervised, Unsupervised, Reinforcement, Overfitting, Underfitting, Hyper parameters and Validation sets, Estimators -Bias and Variance. Challenges in machine learning. Simple Linear Regression, Logistic Regression, Performance measures - Confusion matrix, Accuracy, Precision, Recall, Sensitivity, Specificity, Receiver Operating Characteristic curve(ROC), Area Under Curve(AUC).

Module -2 (Neural Networks)

Introduction to neural networks -Single layer perceptrons, Multi Layer Perceptrons (MLPs), Representation Power of MLPs, Activation functions - Sigmoid, Tanh, ReLU, Softmax. Risk minimization, Loss function, Training MLPs with backpropagation, Practical issues in neural network training - The Problem of Overfitting, Vanishing and exploding gradient problems, Difficulties in convergence, Local and spurious Optima, Computational Challenges. Applications of neural networks.

Module 3 (Deep learning)

Introduction to deep learning, Deep feed forward network, Training deep models, Optimization techniques - Gradient Descent (GD), GD with momentum, Nesterov accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam. Regularization Techniques - L1 and L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout, Parameter initialization.

Module -4 (Convolutional Neural Network)

Convolutional Neural Networks – Convolution operation, Motivation, Pooling, Convolution and Pooling as an infinitely strong prior, Variants of convolution functions, Structured outputs, Data types, Efficient convolution algorithms. Practical use cases for CNNs, Case study - Building CNN model AlexNet with handwritten digit dataset MNIST.

Module- 5 (Recurrent Neural Network)

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs. Case study - Natural Language Processing.

Text Book

1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Neural Networks and Deep Learning, Aggarwal, Charu C., c Springer International Publishing AG, part of Springer Nature 2018
3. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms (1st. ed.). Nikhil Buduma and Nicholas Locascio. 2017. O'Reilly Media, Inc.

Reference Books

1. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Michael Nielsen, Neural Networks and Deep Learning, 2018

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Predict the price of a 1000 square feet house using the regression model generated from the following data.

No.	Square feet	Price(Lakhs)
1	500	5
2	900	10
3	1200	13
4	1500	18
5	2000	25
6	2500	32
7	2700	35

2. Consider a two-class classification problem of predicting whether a photograph contains a man or a woman. Suppose we have a test dataset of 10 records with expected outcomes and a set of predictions from our classification algorithm. Compute the confusion matrix, accuracy, precision, recall, sensitivity and specificity on the following data.

Sl.**No.**	Actual	Predicted
1	man	woman
2	man	man
3	woman	woman
4	man	man

5	man	woman
6	woman	woman
7	woman	man
8	man	man
9	man	woman
10	woman	woman

Course Outcome 2 (CO2):

1. Suppose you have a 3-dimensional input x = (x1, x2, x3) = (2, 2, 1) fully connected with weights (0.5, 0.3, 0.2) to one neuron which is in the hidden layer with sigmoid activation function. Calculate the output of the hidden layer neuron.
2. Consider the case of the XOR function in which the two points {(0, 0),(1, 1)} belong to one class, and the other two points {(1, 0),(0, 1)} belong to the other class. Design a multilayer perceptron for this binary classification problem.

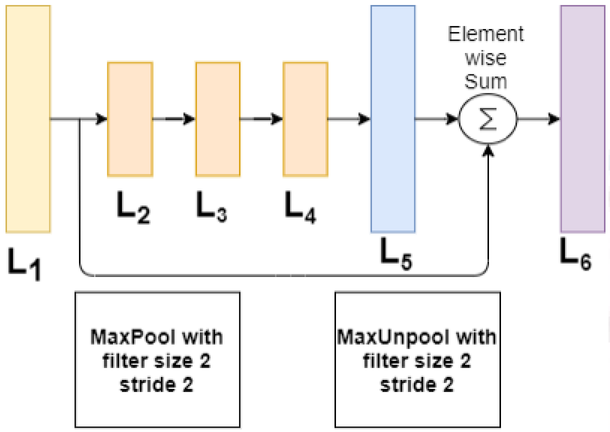
Course Outcome 3 (CO3):

1. Derive a mathematical expression to show L2 regularization as weight decay.
2. Explain how L2 regularization improves the performance of deep feed forward neural networks.
3. Explain how L1 regularization method leads to weight sparsity.

Course Outcome 4 (CO4):

1. Draw and explain the architecture of convolutional neural networks.
2. You are given a classification problem to classify the handwritten digits. Suggest a learning and/or inference machine with its architecture, an objective function, and an optimization routine, along with how input and output will be prepared for the classifier.
3. In a Deep CNN architecture the feature map L_1 was processed by the following operations as shown in the figure. First down sampled using max pool operation of size 2 and stride 2, and three convolution operations and finally max unpool operation and followed by an element wise sum. The feature map L_1 and L_4 are given below. Compute the matrix L6.

$$L_1 = \begin{matrix} 10 & 20 & 15 & 22 \\ 20 & 16 & 28 & 30 \\ 30 & 12 & 20 & 16 \\ 20 & 20 & 40 & 12 \end{matrix} \qquad L_4 = \begin{matrix} 10 & 20 \\ 20 & 30 \end{matrix}$$


Element wise Sum
Σ
L_1
L_2
L_3
L_4
L_5
L_6
MaxPool with filter size 2 stride 2
MaxUnpool with filter size 2 stride 2

4. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.

Course Outcome 5 (CO5):

1. Draw and explain the architecture of LSTM.
2. List the differences between LSTM and GRU

Model Question Paper

QP CODE: **PAGES:4**

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(HONORS), MONTH & YEAR

Course Code: CST 395

Course Name: Neural Networks and Deep Learning

Max.Marks:100 **Duration:3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. List and compare the types of machine learning algorithms

2. Suppose 10000 patients get tested for flu; out of them, 9000 are actually healthy and 1000 are actually sick. For the sick people, a test was positive for 620 and negative for 380. For healthy people, the same test was positive for 180 and negative for 8820. Construct a confusion matrix for the data and compute the

accuracy, precision and recall for the data

3. Illustrate the limitation of a single layer perceptron with an example

4. Specify the advantages of ReLU over sigmoid activation function.

5. Derive weight updating rule in gradient descent when the error function is a) mean squared error b) cross entropy

6. List any three methods to prevent overfitting in neural networks

7. What happens if the stride of the convolutional layer increases? What can be the maximum stride? Justify your answer.

8. Consider an activation volume of size 13×13×64 and a filter of size 3×3×64. Discuss whether it is possible to perform convolutions with strides 2, 3 and 5. Justify your answer in each case.

9. How does a recursive neural network work?

10. List down three differences between LSTM and RNN

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Prove that the decision boundary of binary logistic regression is linear **(9)**

(b) Given the following data, construct the ROC curve of the data. Compute the AUC. **(5)**

Threshold	**TP**	TN	FP	FN
1	0	25	0	29
2	7	25	0	22
3	18	24	1	11
4	26	20	5	3
5	29	11	14	0

6	29	0	25	0
7	29	0	25	0

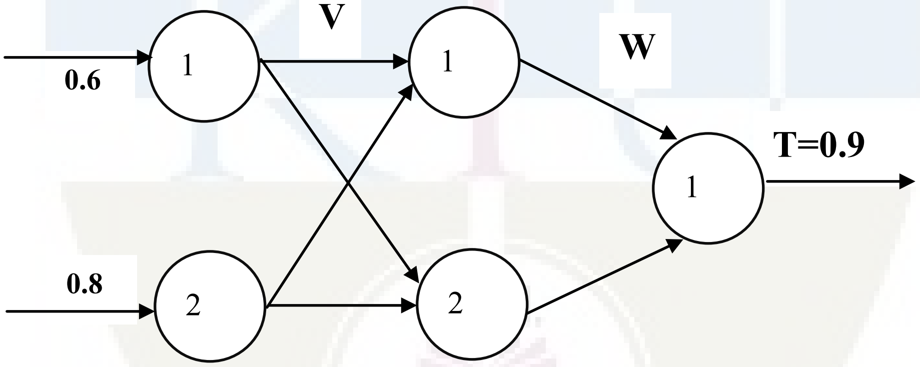
OR

12. (a) With an example classification problem, explain the following terms:
a) Hyper parameters b) Training set c) Validation sets d) Bias e) Variance **(8)**

(b) Determine the regression equation by finding the regression slope coefficient and the intercept value using the following data. **(6)**

x	55	60	65	70	80
y	52	54	56	58	62

13. (a) Update the parameters V_{11} in the given MLP using back propagation with learning rate as 0.5 and activation function as sigmoid. Initial weights are given as V_{11}= 0.2, V_{12}=0.1, V_{21}=0.1, V_{22}=0.3, V_{11}=0.2, W_{11}=0.5, W_{21}=0.2 **(10)**

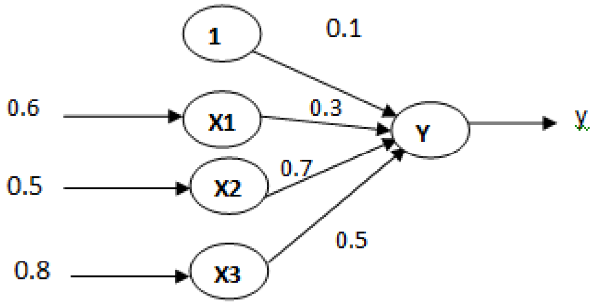


(b) Explain the importance of choosing the right step size in neural networks **(4)**

OR

14. (a) Explain in detail any four practical issues in neural network training **(8)**

(b) Calculate the output of the following neuron Y with the activation function as a) binary sigmoid b) tanh c)ReLU **(6)**



15. (a) Explain, what might happen in ADAGRAD, where momentum is expressed as $\Delta \square_{\square} = -\square\square_{\square}/\sqrt{(\sum_{\square=1}^{\square} \square_{\square}^{2})}$ where the denominator computes the L2 norm of all previous gradients on a per-dimension basis and □ is a global learning rate shared by all dimensions. **(6)**

(b) Differentiate gradient descent with and without momentum. Give equations for weight updation in GD with and without momentum. Illustrate plateaus, saddle points and slowly varying gradients. **(8)**

OR

16. (a) Suppose a supervised learning problem is given to model a deep feed forward neural network. Suggest solutions for the following a) small sized dataset for training b) dataset with both labelled and unlabeled data c) large data set but data from different distribution **(9)**

(b) Describe the effect in bias and variance when a neural network is modified with more number of hidden units followed with dropout regularization. **(5)**

17. (a) Draw and explain the architecture of Convolutional Neural Networks **(8)**

(b) Suppose that a CNN was trained to classify images into different categories. It performed well on a validation set that was taken from the same source as the training set but not on a testing set. What could be the problem with the training of such a CNN? How will you ascertain the problem? How can those problems be solved? **(6)**

OR

18. (a) Explain the following convolution functions a)tensors b) kernel flipping c) down sampling d) strides e) zero padding. **(10)**

(b) What is the motivation behind convolution neural networks? **(4)**

19. (a) Describe how an LSTM takes care of the vanishing gradient problem. Use some hypothetical numbers for input and output signals to explain the concept **(8)**

(b) Explain the architecture of Recurrent Neural Networks **(6)**

OR

20. (a) Explain LSTM based solution for anyone of the problems in the Natural Language Processing domain. **(8)**

(b) Discuss the architecture of GRU **(6)**

Teaching Plan

Module 1 : [Text book 1: Chapter 5, Textbook 2: Chapter 2](9 hours)		
1.1	Introduction, Learning algorithms - Supervised, Unsupervised, Reinforcement	1 hour
1.2	Overfitting, Underfitting, Hyperparameters	1 hour
1.3	Validation sets, Estimators -Bias and Variance. Challenges in machine learning.	1 hour
1.4	Simple Linear Regression	1 hour
1.5	Illustration of Linear Regression	1 hour
1.6	Logistic Regression	1 hour
1.7	Illustration of Logistic Regression	1 hour
1.8	Performance measures - Confusion matrix, Accuracy, Precision, Recall, Sensitivity, Specificity, ROC, AUC.	1 hour
1.9	Illustrative Examples for performance measures	1 hour
Module 2 : Text book 2, Chapter 1 (8 hours)		
2.1	Introduction to neural networks -Single layer perceptrons	1 hour
2.2	Multi Layer Perceptrons (MLPs), Representation Power of MLPs	1 hour
2.3	Activation functions - Sigmoid, Tanh, ReLU, Softmax. Risk minimization, Loss function	1 hour

2.4	Training MLPs with backpropagation	1 hour
2.5	Illustration of back propagation algorithm	1 hour
2.6	Practical issues in neural network training - The Problem of Overfitting, Vanishing and exploding gradient problems	1 hour
2.7	Difficulties in convergence, Local and spurious Optima, Computational Challenges.	1 hour
2.8	Applications of neural networks	1 hour
Module 3 : Text book 1: Chapter 7, 8, Text book 2, Chapter 3, 4 (10 hours)		
3.1	Introduction to deep learning, Deep feed forward network	1 hour
3.2	Training deep models - Introduction, setup and initialization issues	1 hour
3.3	Solving vanishing and exploding gradient problems	1 hour
3.4	Concepts of optimization, Gradient Descent (GD), GD with momentum.	1 hour
3.5	Nesterov accelerated GD, Stochastic GD.	1 hour
3.6	AdaGrad, RMSProp, Adam.	1 hour
3.7	Concepts of Regularization, L1 and L2 regularization.	1 hour
3.8	Early stopping, Dataset augmentation	1 hour
3.9	Parameter sharing and tying, Injecting noise at input, Ensemble methods	1 hour
3.10	Dropout, Parameter initialization.	1 hour
Module 4 : Text book 1, Chapter 9, Text book 2: Chapter 8 (8 hours)		
4.1	Convolutional Neural Networks, architecture	1 hour
4.2	Convolution and Pooling operation with example	1 hour
4.3	Convolution and Pooling as an infinitely strong prior	1 hour
4.4	Variants of convolution functions, structured outputs, data types	1 hour
4.5	Efficient convolution algorithms.	1 hour
4.6	Practical use cases for CNNs	1 hour
4.7	Case study - Building CNN with MNIST and AlexNet.	1 hour
4.8	Case study - Building CNN with MNIST and AlexNet	1 hour
Module 5 : Text book 1 :Chapter 10, 11, Text book 2:Chapter 7 (10 hours)		

5.1	Recurrent neural networks – Computational graphs, RNN design	1 hour
5.2	Encoder – decoder sequence to sequence architectures	1 hour
5.3	Deep recurrent networks- Architecture	1 hour
5.4	Recursive neural networks	1 hour
5.5	Modern RNNs - LSTM	1 hour
5.6	Modern RNNs - LSTM	1 hour
5.7	GRU	1 hour
5.8	Practical use cases for RNNs.	1 hour
5.9	Case study - Natural Language Processing.	1 hour
5.10	Case study - Natural Language Processing.	1 hour

CST 397	**PRINCIPLES OF MODEL CHECKING**	**Category**	**L**	**T**	**P**	**Credit**	**Year of Introduction**
		VAC	**3**	**1**	**0**	**4**	**2019**

Preamble:

This course covers the basic theory and algorithm for an automatic verification process namely, model checking. Model checking is a formal process for proving the correctness of a hardware/software which can be modelled as a finite-state transition system. This course introduces the topics - finite-state modelling of hardware/software, linear-time properties, classification of linear-time properties, Linear Temporal Logic (LTL), a formal language for property specification, LTL model checking algorithm and model checking case studies. Proving correctness of a hardware/software is essential in safety critical systems in domains such as avionics, health care and automotive.

Prerequisite: Nil

Course Outcomes: After the completion of the course, the student will be able to

CO#	**Course Outcomes**
CO1	Illustrate an application for model checking. **(Cognitive Knowledge Level: Understand)**
CO2	Describe finite-state modelling of hardware and software. **(Cognitive Knowledge Level: Understand)**
CO3	Identify the linear-time properties required to represent the requirements of a system. **(Cognitive Knowledge Level: Apply**)
CO4	Specify a given linear-time property in Linear Temporal Logic (LTL). (**Cognitive Knowledge Level: Apply**)
CO5	Perform LTL model checking with the tool SAL (Symbolic Analysis Laboratory). (**Cognitive Knowledge Level: Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO1 1	PO12
CO1	✓	✓	✓	✓	✓	✓						✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓
CO6	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Test 1 (Marks in percentage)**	**Test 2 (Marks in percentage)**	**End Semester Examination (Marks in percentage)**
Remember	30	30	30
Understand	40	40	40
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment Test	: **25 marks**
Continuous Assessment Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus. The second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module 1 (Introduction to Model Checking)

System Verification– Hardware and Software Verification, Model Checking, Characteristics of Model Checking. Transition Systems – Transition System, Direct Predecessors and Successors, Terminal State, Deterministic Transition System.
Executions - Execution Fragment, Maximal and Initial Execution Fragment, Execution, Reachable States. Modeling Hardware and Software Systems- Sequential Hardware Circuits, data Dependent Systems.

Module - 2 (Linear Time Properties)

Linear-Time (LT) Properties - Deadlock. Linear-Time Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Path. Traces - Trace and Trace Fragment, LT Properties - LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties. Safety Properties and Invariants - Invariants, Safety Properties, Trace Equivalence and Safety properties. Liveness Properties - Liveness Property, Safety vs. Liveness Properties. Fairness - Fairness, Unconditional, Weak and Strong Fairness, Fairness Strategies, Fairness and Safety. (Definition and examples only for all topics - no proof required).

Module - 3 (Regular Properties)

Regular Properties - Model Checking Regular Safety properties - Regular Safety property, Verifying Regular Safety Properties. Automata on Infinite Words - ω-Regular Languages and Properties, Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata (Definitions only). Model Checking ω-Regular Properties - Persistence Properties and Product, Nested Depth-First Search (Only algorithms required).

Module - 4 (Linear Time Logic)

Linear Temporal Logic (LTL) - Syntax, Semantics, Equivalence of LTL Formulae, Weak Until, Release and Positive Normal Form, Fairness, Safety and Liveness in LTL (Definitions only). Automata Based LTL Model Checking (Algorithms and examples only).

Module - 5 (Model Checking in SAL)

Introduction **-** Introduction to the tool Symbolic Analysis Laboratory (SAL). The Language of SAL - The expression language, The transition Language, The module language, SAL Contexts. SAL Examples - Mutual Exclusion, Peterson’s Protocol, Synchronous Bus Arbiter, Bounded Bakery protocol, Bakery Protocol, Simpson’s Protocol, Stack.

Text Books

1. Christel Baier and Joost-Pieter Katoen, Principles of Model Checking, The MIT Press. (Modules 1 - 4)
2. Leonardo de Moura, Sam Owre and N. Shankar, The SAL Language Manual, SRI International (http://sal.csl.sri.com/doc/language-report.pdf, Chapters 1, 3, 4, 5, 6, 7) (Module 5)

Reference Materials

1. SAL Examples (http://sal.csl.sri.com/examples.shtml) (Module 5)

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain how model checking can be effective in developing a nuclear power plant.

Course Outcome 2 (CO2):

1. Consider a message delivery system. The sender *s* is trying to send a series of messages to the receiver *r* in such a way that the $(i+1)^{st}$ message is sent only after the i^{th} message is delivered. There is a possibility of error in sending a message and in that case, *s* keeps on trying until it is able to send the message. Express this process as a transition system.

Course Outcome 3 (CO3):

1. Consider a shared memory segment *s* protected using a mutex lock variable *m*. Two processes p_1 and p_2 are trying to access *s*. Find the Linear Time properties of the system which will ensure safety, liveness and fairness.

Course Outcome 4 (CO4):

1. Express the Linear Time properties found in the above system using LTL.

Course Outcome 5 (CO5):

1. Model the above system using SAL and verify that the system avoids deadlock under all conditions.
2.

Model Question Paper

QP CODE: **PAGES: ___**

Reg No:_______________

Name:__________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(HONORS), MONTH & YEAR

Course Code: CST 397

Course Name : Principles of Model Checking

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

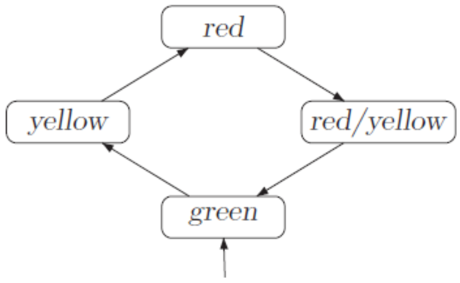
1. What is model checking? Give the schematic view of the model checking approach.

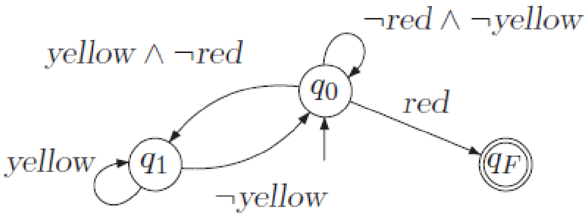
2. Give the transition system of a beverage vending machine.

3. What is an invariant in Linear Time (LT) properties? Give an example.

4. Give 3 Liveness properties in the Mutual Exclusion problem of processes.

5. Find the product automaton for the following Transition System and Non-Deterministic Finite Automaton (NFA).





6. Differentiate between Deterministic Buchi Automaton and Non-deterministic

Buchi Automaton. Give examples of each.

7. Express the following statements about traffic lights in Linear Temporal Logic (LTL).
   a. Once red, the light can not become green immediately.
   b. Once red, the light always becomes green eventually after being yellow for some time.

8. What is Positive Normal Form (PNF) in LTL? Give an example.

9. Write notes on Symbolic Analysis Laboratory (SAL).

10. What is a SAL context? Give an example.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain in detail the various phases of the model checking process. **(7)**

    (b) Explain the strengths and weaknesses of model checking. **(7)**

OR

12. (a) Explain the following terms in association with execution of a transition system. **(14)**
    a. Execution Fragment
    b. Maximal and Initial Execution Fragment
    c. Execution
    d. Reachable States

13. (a) With an example, explain the satisfaction relation for LT properties. **(7)**

    (b) What is trace equivalence in Transition Systems? Give an example to show that if two transition systems have the same trace equivalence, they satisfy the same LT properties. **(7)**

14. (a) Give the transition system for the fault tolerant variant of the dining philosophers problem. **(4)**

(b) With a suitable example, explain the algorithms to check whether a Transition System satisfies an invariant or not. **(10)**

15. (a) Give the algorithm for verifying Regular Safety Properties. Explain with an appropriate example. **(7)**

(b) With a suitable example, explain Regular Safety Properties. **(7)**

OR

16. (a) Explain ω -Regular Properties. **(4)**

(b) Illustrate how ω -Regular Properties are verified. **(10)**

17. (a) Explain the syntax of Linear Temporal Logic (LTL). **(7)**

(b) Explain the semantics of LTL. **(7)**

OR

18. (a) With an example, give the difference between *until* and *weak until* in LTL. **(4)**

(b) With a suitable example, explain automata based LTL model checking. **(10)**

19. (a) Explain Peterson's protocol. What are the LTL properties to be verified to ensure its correctness? **(8)**

(b) Write a SAL script for the verification of Peterson's protocol. **(6)**

OR

20. (a) Show the SAL model corresponding to Bakery protocol. **(8)**

(b) List any three Linear Time properties of this model and show their LTL specifications. **(6)**

Teaching plan

Module 1 (Introduction to Model Checking)		**8 Hours**
1.1	**System Verification** – Hard- and Software Verification, Model Checking, Characteristics of Model Checking	1 Hour
1.2	Transition Systems – Transition System, Direct Predecessors and Successors,	1 Hour
1.3	Terminal State, Deterministic Transition System,	1 Hour
1.4	**Executions** - Execution Fragment, Maximal and Initial Execution Fragment	1 Hour
1.5	Execution, Reachable States	1 Hour
1.6	Modeling Hardware and Software Systems - Sequential Hardware Circuits	1 Hours
1.7	Data Dependent Systems (Lecture 1)	1 Hour
1.8	Data Dependent Systems (Lecture 2)	1 Hour
Module 2 (Linear Time Properties)		**10 Hours**
2.1	Linear-Time (LT) Properties - Deadlock	1 Hour
2.2	Linear-Time Behavior - Paths and State Graph, Path Fragment	1 Hour
2.3	Maximal and Initial Path Fragment, Path	
2.4	Traces - Trace and Trace Fragment	1 Hour
2.5	LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties	1 Hour
2.6	Invariants	1 Hour
2.7	Safety Properties, Trace Equivalence and Safety properties	1 Hour
2.8	Liveness Property, Safety vs. Liveness Properties	1 Hour
2.9	Fairness, Unconditional, Weak and Strong Fairness	1 Hour
2.10	Fairness Strategies, Fairness and Safety	1 Hour
Module 3 (Regular Properties)		**8 Hours**
3.1	**Regular Properties** - Model Checking Regular Safety properties - Regular Safety property	1 Hour
3.2	Verifying Regular Safety Properties	1 Hour

3.3	Automata on Infinite Words - ω-Regular Languages and Properties	1 Hour
3.4	Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata	1 Hour
3.5	Model Checking ω-Regular Properties - Persistence Properties and Product - Lecture 1	1 Hour
3.6	Persistence Properties and Product - Lecture 2	1 Hour
3.7	Nested Depth-First Search (Lecture 1)	1 Hour
3.8	Nested Depth-First Search (Lecture 2)	1 Hour
Module 4 (Linear Time Logic)		**9 Hours**
4.1	Linear Temporal Logic – Linear Temporal Logic (LTL) - Syntax	1 Hour
4.2	Semantics - Lecture 1	1 Hour
4.3	Semantics - Lecture 2	1 Hour
4.4	Equivalence of LTL Formulae, Weak Until	1 Hour
4.5	Release and Positive Normal Form	1 Hour
4.6	Fairness, Safety and Liveness in LTL	1 Hour
4.7	Automata Based LTL Model Checking - Lecture 1	1 Hour
4.8	Automata Based LTL Model Checking - Lecture 2	1 Hour
4.9	Automata Based LTL Model Checking - Lecture 3	1 Hour
Module 5 (Model Checking in SAL)		**10 Hours**
5.1	Introduction - Introduction to the tool Symbolic Analysis Laboratory (SAL).	1 Hour
5.2	The Language of SAL - The expression language, The transition Language	1 Hour
5.3	The module language, SAL Contexts.	1 Hour
5.4	SAL Examples - Mutual Exclusion	1 Hour
5.5	Peterson’s Protocol	1 Hour
5.6	Synchronous Bus Arbiter	1 Hour
5.7	Bounded Bakery protocol,	1 Hour
5.8	Bakery Protocol	1 Hour

5.9	Simpson’s Protocol	1 Hour
5.10	Stack	1 Hour

SEMESTER VI

CST 302	COMPILER DESIGN	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble:

The purpose of this course is to create awareness among students about the phases of a compiler and the techniques for designing a compiler. This course covers the fundamental concepts of different phases of compilation such as lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization and code generation. Students can apply this knowledge in design and development of compilers.

Prerequisite: Sound knowledge in Data Structures, Formal Languages & Automata Theory.

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain the phases in compilation process(lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization and code generation) and model a lexical analyzer **(Cognitive Knowledge Level: Apply)**
CO2	Model language syntax using Context Free Grammar and develop parse tree representation using leftmost and rightmost derivations (**Cognitive Knowledge Level: Apply)**
CO3	Compare different types of parsers(Bottom-up and Top-down) and construct parser for a given grammar **(Cognitive Knowledge Level: Apply)**
CO4	Build Syntax Directed Translation for a context free grammar, compare various storage allocation strategies and classify intermediate representations **(Cognitive Knowledge Level: Apply)**
CO5	Illustrate code optimization and code generation techniques in compilation **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓	✓							✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	20	20	20
Understand	40	40	40
Apply	40	40	40
Analyze			

Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment - Test	: **25 marks**
Continuous Assessment - Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to compilers and lexical analysis)

Analysis of the source program - Analysis and synthesis phases, Phases of a compiler. Compiler writing tools. Bootstrapping. Lexical Analysis - Role of Lexical Analyser, Input Buffering, Specification of Tokens, Recognition of Tokens.

Module - 2 (Introduction to Syntax Analysis)

Role of the Syntax Analyser – Syntax error handling. Review of Context Free Grammars - Derivation and Parse Trees, Eliminating Ambiguity. Basic parsing approaches - Eliminating left recursion, left factoring. Top-Down Parsing - Recursive Descent parsing, Predictive Parsing, LL(1) Grammars.

Module - 3 (Bottom-Up Parsing)

Handle Pruning. Shift Reduce parsing. Operator precedence parsing (Concept only). LR parsing - Constructing SLR, LALR and canonical LR parsing tables.

Module - 4 (Syntax directed translation and Intermediate code generation)

Syntax directed translation - Syntax directed definitions, S-attributed definitions, L-attributed definitions, Bottom-up evaluation of S-attributed definitions. Run-Time Environments - Source Language issues, Storage organization, Storage-allocation strategies. Intermediate Code Generation - Intermediate languages, Graphical representations, Three-Address code, Quadruples, Triples.

Module 5 – (Code Optimization and Generation)

Code Optimization - Principal sources of optimization, Machine dependent and machine independent optimizations, Local and global optimizations. Code generation - Issues in the design of a code generator, Target Language, A simple code generator.

Text Books

1. Aho A.V., Ravi Sethi and D. Ullman. Compilers – Principles Techniques and Tools, Addison Wesley, 2006.

Reference Books

1. D.M.Dhamdhere, System Programming and Operating Systems, Tata McGraw Hill & Company, 1996.
2. Kenneth C. Louden, Compiler Construction – Principles and Practice, Cengage Learning Indian Edition, 2006.

3. Tremblay and Sorenson, The Theory and Practice of Compiler Writing, Tata McGraw Hill & Company,1984.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1) Explain the phases of a compiler with a neat diagram.

2) Define a token. Identify the tokens in the expression a := b + 10.

Course Outcome 2 (CO2):

1) Illustrate the process of eliminating ambiguity, left recursion and left factoring the grammar.

2) Is the following grammar ambiguous? If so eliminate ambiguity.

E→E + E | E*E |(E) | id

Course Outcome 3 (CO3):

1. What are the different parsing conflicts in the SLR parsing table?
2. Design a recursive descent parser for the grammar

   E→E + T | T

   T→T*F | F

   F→(E) | id
3. Construct canonical LR(0) collection of items for the grammar below.

   S → L = R

   S → R

   L → * R

   L → id

   R → L

   Also identify a shift reduce conflict in the LR(0) collection constructed above.

Course Outcome 4 (CO4):

1. Write the quadruple and triple representation of the following intermediate code

   R1 = C * D

   R2 = B + R1

   A = R2

   B[0] = A

2. Differentiate S-attributed Syntax Directed Translation(SDT) and L-attributed SDT. Write S - attributed SDT for a simple desktop calculator

Course Outcome 5 (CO5):

1. List out the examples of function preserving transformations.
2. What are the actions performed by a simple code generator for a typical three-address statement of the form x: = y op z.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION , MONTH & YEAR

Course Code: CST 302

Course Name: Compiler Design

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Specify the analysis and synthesis parts of compilation.
2. Define the terms token, lexemes and patterns with examples.
3. Is the grammar S --> S | (S) S / Ɛ ambiguous? Justify your answer.
4. What is left recursive grammar? Give an example. What are the steps in removing left recursion?
5. Compare different bottom-up parsing techniques.
6. What are the possible actions of a shift reduce parser.

7. Differentiate synthesized and inherited attributes with examples.

8. Translate a[i] = b * c – b * d, to quadruple.

9. What is the role of peephole optimization in the compilation process

10. What are the issues in the design of a code generator

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the different phases of a compiler with a running example. **(9)**

(b) List and explain any three compiler construction tools. **(5)**

OR

12. (a) What is a regular definition? Give the regular definition of an unsigned integer **(7)**

(b) Express the role of transition diagrams in recognition of tokens. **(7)**

13. (a) What is Recursive Descent parsing? List the challenges in designing such a parser? **(4)**

(b) Consider the following grammar **(10)**

E-→E or T | T

T→ T and F |F

F→not F | (E) | true | false

(i) Remove left recursion from the grammar.

(ii) Construct a predictive parsing table.

(iii) Justify the statement "The grammar is LL (1)".

OR

14. (a) What is Recursive Descent parsing? List the problems in designing such a parser **(4)**

(b) Design a recursive descent parser for the grammar S→cAd, A→ab/ b **(5)**

Find the FIRST and FOLLOW of the non-terminals S, A and B in the grammar **(5)**

S→aABe
A→Abc | b
B→d

15. (a) Construct the LR(0) set of items and their GOTO function for the grammar **(10)**
S→S S + | S S * | a

(b) Is the grammar SLR? Justify your answer **(4)**

OR

16. (a) Identify LR(1) items for the grammar **(7)**

S→ CC
C→ cC | d

(b) Construct LALR table for the above grammar **(7)**

17. (a) Design a Syntax Directed Translator(SDT) for the arithmetic expression (4 * 7 + 19) * 2 and draw an annotated parse tree for the same. **(8)**

(b) Consider the grammar with following translation rules and E as the start symbol **(6)**

E → E1 # T {E.value=E1.value x T.value ;}
| T{E.value=T.value ;}
T → T1 & F{ T.value=T1.value + F.value ;}
| F{T.value= F.value ; }
F → num { F.value=num. lvalue ;}

Compute E.value for the root of the parse tree for the expression
2#3 & 5# 6 &7

OR

18. (a) Write Syntax Directed Translator (SDT) and parse tree for infix to postfix translation of an expression. **(8)**

(b) Explain the storage allocation strategies. **(6)**

19. (a) Describe the principal sources of optimization **(7)**

(b) Illustrate the optimization of basic blocks with examples. **(7)**

OR

20. (a) Write the Code Generation Algorithm and explain the *getreg* function **(6)**

(b) Generate target code sequence for the following statement **(8)**

d := (a-b)+(a-c)+(a-c).

Teaching Plan

No	Contents	No. of Lecture Hours
Module - 1(Introduction to Compilers and lexical analyzer) (8 hours)		
1.1	Introduction to compilers, Analysis of the source program	1 hour
1.2	Phases of the compiler – Analysis Phases	1 hour
1.3	Phases of the Compiler - Synthesis Phases	1 hour
1.4	Symbol Table Manager and Error Handler	1 hour
1.5	Compiler writing tools, bootstrapping	1 hour
1.6	The role of Lexical Analyzer , Input Buffering	1 hour
1.7	Specification of Tokens	1 hour
1.8	Recognition of Tokens	1 hour

Module – 2 (Introduction to Syntax Analysis) (10 hours)		
2.1	Role of the Syntax Analyser, Syntax error handling	1 hour
2.2	Review of Context Free Grammars	1 hour
2.3	Parse Trees and Derivations	1 hour
2.4	Grammar transformations, Eliminating ambiguity	1 hour
2.5	Eliminating left recursion	1 hour
2.6	Left factoring the grammar	1 hour
2.7	Recursive Descent parsing	1 hour
2.8	First and Follow	1 hour
2.9	Predictive Parsing table constructor	1 hour
2.10	LL(1) Grammars	1 hour
Module - 3 (Bottom up parsing) (9 hours)		
3.1	Bottom-up parsing - Handle Pruning	1 hour
3.2	Shift Reduce parsing	1 hour
3.3	Operator precedence parsing (Concept only)	1 hour
3.4	LR parsing , SLR Grammar, items	1 hour
3.5	Augmented Grammar, Canonical collection of LR(0) items	1 hour
3.6	SLR Parser Table Construction	1 hour
3.7	Constructing Canonical LR Parsing Tables	1 hour
3.8	Constructing LALR Parsing Tables	1 hour
3.9	LALR parser	1 hour
Module - 4 (Syntax Directed Translation and Intermediate code Generation) (9 hours)		
4.1	Syntax directed definitions	1 hour
4.2	S- attributed definitions, L- attributed definitions	1 hour
4.3	Bottom- up evaluation of S- attributed definitions.	1 hour
4.4	Source Language issues	1 hour
4.5	Storage organization	1 hour

4.6	Storage- allocation strategies	1 hour
4.7	Intermediate languages , Graphical representations	1 hour
4.8	Three-Address code	1 hour
4.9	Quadruples, Triples	1 hour
Module - 5 (Code Optimization and Generation) (9 hours)		
5.1	Principal sources of optimization	1 hour
5.2	Machine dependent optimizations	1 hour
5.3	Machine independent optimizations	1 hour
5.4	Local optimizations	1 hour
5.5	Global optimizations	1 hour
5.6	Issues in the design of a code generator – Lecture 1	1 hour
5.7	Issues in the design of a code generator – Lecture 2	1 hour
5.8	Target Language	1 hour
5.9	Design of a simple code generator.	1 hour

CST 304	COMPUTER GRAPHICS AND IMAGE PROCESSING	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble:
The purpose of this course is to make awareness about strong theoretical relationships between computer graphics and image processing. This course helps the learner to understand three-dimensional environment representation in a computer, transformation of 2D/3D objects, basic mathematical techniques and algorithms used to build useful applications, imaging, and image processing techniques. The study of computer graphics and image processing develops the ability to create image processing frameworks for different domains and develops algorithms for emerging display technologies.

Prerequisite: A sound knowledge of Mathematics and a programming language.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Describe the working principles of graphics devices**(Cognitive Knowledge level: Understand)**
CO2	Illustrate line drawing, circle drawing and polygon filling algorithms(Cognitive Knowledge level: Apply)
CO3	Demonstrate geometric representations, transformations on 2D & 3D objects, clipping algorithms and projection algorithms(Cognitive Knowledge level: Apply)
CO4	Summarize visible surface detection methods(Cognitive Knowledge level: Understand)
CO5	Summarize the concepts of digital image representation, processing and demonstrate pixel relationships(Cognitive Knowledge level: Apply)
CO6	Solve image enhancement and segmentation problems using spatial domain techniques(Cognitive Knowledge level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓		✓									✓
CO5	✓	✓	✓	✓								
CO6	✓	✓	✓	✓		✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30

Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of SeriesTests1& 2)	**25 marks**
Continuous AssessmentAssignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one full question. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1(Basics of Computer graphics and Algorithms)

Basics of Computer Graphics and its applications. Video Display devices- Refresh Cathode Ray Tubes, Random Scan Displays and systems, Raster scan displays and systems. Line drawing algorithms- DDA, Bresenham's algorithm. Circle drawing algorithms- Midpoint Circle generation algorithm, Bresenham's algorithm.

Module - 2(Filled Area Primitives and transformations)

Filled Area Primitives- Scan line polygon filling, Boundary filling and flood filling. Two dimensional transformations-Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations.

Module - 3 (Clipping and Projections)

Window to viewport transformation. Cohen Sutherland Line clipping algorithm. Sutherland Hodgeman Polygon clipping algorithm. Three dimensional viewing pipeline. Projections- Parallel and Perspective projections. Visible surface detection algorithms- Depth buffer algorithm, Scan line algorithm.

Module - 4 (Fundamentals of Digital Image Processing)

Introduction to Image processing and applications. Image as 2D data. Image representation in Gray scale, Binary and Colour images. Fundamental steps in image processing. Components of image processing system. Coordinate conventions. Sampling and quantization. Spatial and Gray Level Resolution. Basic relationship between pixels– neighbourhood, adjacency, connectivity. Fundamentals of spatial domain-convolution operation.

Module - 5 (Image Enhancement in Spatial Domain and Image Segmentation)

Basic gray level transformation functions - Log transformations, Power-Law transformations, Contrast stretching. Histogram equalization. Basics of spatial filtering - Smoothing spatial filter- Linear and nonlinear filters, and Sharpening spatial filters-Gradient and Laplacian.

Fundamentals of Image Segmentation. Thresholding - Basics of Intensity thresholding and Global Thresholding. Region based Approach - Region Growing, Region Splitting and Merging. Edge Detection - Edge Operators- Sobel and Prewitt.

Text Book

1. Donald Hearn and M. Pauline Baker, Computer Graphics, PHI, 2e, 1996
2. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing. Pearson, 4e, 2017

References

1) William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics. McGraw Hill, 2001

2) Zhigang Xiang and Roy Plastock, Computer Graphics (Schaum's outline Series), McGraw Hill, 2019.
3) David F. Rogers , Procedural Elements for Computer Graphics, Tata McGraw Hill,2001.
4) M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, Thomson India Edition, 4e, 2017.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare the working principle of raster scan systems and random scan systems.
2. How much time is spent scanning across each row of pixels during screen refresh on a raster system with resolution of 1280*1024 and a refresh rate of 60 frames per second?

Course Outcome 2 (CO2):

1. Rasterize the line with end points(2,3) and (5,8) using Bresenham's line drawing algorithm.
2. Explain how the 4-connected area filling approach differs from 8- connected area filling in boundary filling algorithm

Course Outcome 3 (CO3):

1. Rotate a triangle ABC 45 degree counter clockwise about the pivot point (10,3), where the position vector of the coordinate ABC is given as A(4,1), B(5,2) and C(4,3).
2. Given a clipping window A(20,20), B(60,20), C(60,40) and D(20,40). Using Cohen Sutherland algorithm, find the visible portion of the line segment joining the points P(40,80) and Q(120,30)

Course Outcome 4 (CO4):

1. Explain scan line algorithm for detecting visible surfaces in an object.

Course Outcome 5 (CO5):

1. Give an image representation model and describe how the representation changes in grayscale, binary and colour images.
2. Consider an image segment shown below.

3 1 2 1 (q)

2 2 0 2

1 2 1 1

(p) 1 0 1 2

(a) Let V={0,1} and compute the length of the shortest 4-,8- and m- path between p and q. If a particular path does not exist between these two points , explain why?

(b) Repeat for V={1,2}.

3. The spatial resolution of an image is given by 128 X 128.What is its storage requirements if it is represented by 64 gray levels?

Course Outcome 6 (CO6):

1. A skilled medical technician is charged with the job of inspecting a certain class of monochrome images generated by electronic microscope. To facilitate the inspection, the technician uses image processing aids. However when he examines the images he finds the following problems.
   (a) Presence of bright isolated dots that are not of interest.
   (b) Lack of sharpness
   (c) Poor contrast

   Identify the sequence of preprocessing steps that the technician may use to overcome the above mentioned problems and explain it.

2. A 4x4, 4 bits/pixel original image is given by

$$\begin{pmatrix} 10 & 12 & 8 & 9 \\ 10 & 12 & 12 & 14 \\ 12 & 13 & 10 & 9 \\ 14 & 12 & 10 & 12 \end{pmatrix}$$

   (a) Apply histogram equalisation to the image by rounding the resulting image pixels to integers
   (b) Sketch the histogram of the original image and the histogram-equalised image.

3. You have Sobel operator and Laplacian operator for edge detection. Which operator will you select for edge detection in the case of noisy image? Explain.**(Assignment)**

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 304

Course Name: Computer Graphics and Image Processing

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Justify the approach of using integer arithmetic in Bresenham's line drawing algorithm.

2. Consider a raster system with a resolution of 1024*1024. What is the size of the raster needed to store 4 bits per pixel? How much storage is needed if 8 bits per pixel are to be stored?

3. Show that two successive reflections about either of the coordinate axes is equivalent to a single rotation about the coordinate origin.

4. Determine a sequence of basic transformations that are equivalent to the x-direction shearing matrix.

5. Find the window to viewport normalization transformation with window lower left corner at (1,1) and upper right corner at (2,6).

6. Find the orthographic projection of a unit cube onto the x=0, y=0 and z=0 plane.

7. Define Sampling and Quantization of an image.

8. Give any three applications of digital image processing.

9. A captured image appears very dark because of wrong lens aperture setting. Describe an enhancement technique which is appropriate to enhance such an image.

10. Suggest an approach of thresholding that should be used in case of uniform illumination. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Write Midpoint circle drawing algorithm and use it to plot a circle with radius=20 and center is (50,30). **(10)**

    (b) Draw the architecture of raster scan display systems and explain its working principle. **(4)**

OR

12. (a) Derive the initial decision parameter of Bresenham's line drawing algorithm and use the algorithm to rasterize a line with endpoints (2,2) and (10,10). **(10)**

    (b) Explain the working principle of color CRT monitors with suitable illustrations. **(4)**

13. (a) Compare boundary fill algorithm and flood fill algorithm. **(5)**

    (b) Reflect a triangle ABC about the line 3x-4y+8=0. The position vector of the coordinate ABC is given as A(4,1), B(5,2) and C(4,3). **(9)**

OR

14. (a) Explain the need of using vanishing points in projections. **(4)**

    (b) Explain Cohen-Sutherland line clipping algorithm. Use the algorithm to clip line P1(70, 20) and P2(100,10) against a window lower left hand corner (50,10) and upper right hand corner (80,40). **(10)**

15. (a) Describe Sutherland Hodegman polygon clipping algorithm and what are its **(7)**

limitations.

(b) Explain how visible surfaces can be detected using depth buffer algorithm. **(7)**

OR

16. (a) Describe Sutherland Hodegman polygon clipping algorithm and what are its limitations. **(7)**

(b) Explain how visible surfaces can be detected using depth buffer algorithm. **(7)**

17. (a) Explain the components of an image processing system with suitable diagram **(9)**

(b) Define Resolution of an image. Explain the spatial and gray level resolution of an image with an example. **(5)**

OR

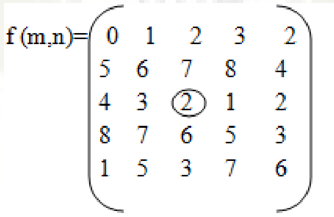
18. (a) Define 4-adjacency, 8 adjacency and m-adjacency. Consider the image segment shown. **(7)**

$$\begin{array}{cccccc} & 4 & 2 & 3 & 2 & (q) \\ & 3 & 3 & 1 & 3 & \\ & 2 & 3 & 2 & 2 & \\ (p) & 2 & 1 & 2 & 3 & \end{array}$$

Let V={1,2} and compute the length of the shortest 4- ,8- and m- path between p and q. If a particular path does not exist between these two points, explain why?

(b) Using any one application, explain the steps involved in image processing. **(7)**

19. (a) A 5x5 image patch is shown below. Compute the value of the marked pixel if it is smoothened by a 3x3 average filter and median filter. **(4)**



$$f(m,n)=\begin{pmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & \textcircled{2} & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{pmatrix}$$

(b) Define Image segmentation and describe in detail method of edge and region based segmentation technique. **(10)**

OR

20. (a) Distinguish between smoothing and sharpening filters in terms of **(10)**

    (i) Functionality
    (ii) Types
    (iii) Applications
    (iv) Mask Coefficients

    (b) Describe how an image is segmented using split and merge technique in association with the region adjacency graph. **(8)**

Teaching Plan

No	Contents	No of Lecture Hrs (45 hrs)
Module – 1 (Basics of Computer Graphics and Algorithms) (9 hrs)		
1.1	Basics of Computer Graphics and applications	1 hour
1.2	Refresh Cathode Ray Tubes	1 hour
1.3	Random Scan Displays and systems	1 hour
1.4	Raster scan displays and systems	1 hour
1.5	DDA Line drawing Algorithm	1 hour
1.6	Bresenham's line drawing algorithm	1 hour
1.7	Midpoint Circle generation algorithm	1 hour
1.8	Bresenham's Circle generation algorithm	1 hour
1.9	Illustration of line drawing and circle drawing algorithms	1 hour
Module - 2 (Filled Area Primitives and transformations) (9 hrs)		
2.1	Scan line polygon filling	1 hour
2.2	Boundary filling and flood filling	1 hour
2.3	Basic 2D transformations-Translation	1 hour

2.4	Basic 2D transformations- Rotation and Scaling	1 hour
2.5	Reflection and Shearing	1 hour
2.6	Composite transformations	1 hour
2.7	Matrix representations and homogeneous coordinates	1 hour
2.8	Basic 3D transformation-Translation and scaling	1 hour
2.9	Basic 3D transformation-Rotation	1 hour
Module - 3 (Clipping and Projections) (8 hrs)		
3.1	Window to viewport transformation	1 hour
3.2	Cohen Sutherland Line clipping algorithm	1 hour
3.3	Sutherland Hodgeman Polygon clipping algorithm	1 hour
3.4	Practice problems on Clipping algorithms	1 hour
3.5	Three dimensional viewing pipeline, Projections-Parallel projections	1 hour
3.6	Projections- Perspective projections	1 hour
3.7	Visible surface detection algorithms- Depth buffer algorithm	1 hour
3.8	Scan line visible surface detection algorithm	1 hour
Module - 4 (Fundamentals of Digital Image Processing) (8 hrs)		
4.1	Introduction to Image processing-Image as a 2D data, Image representation-Gray scale, Binary and Colour images.	1 hour
4.2	Fundamental steps in image processing and applications	1 hour
4.3	Components of image processing system	1 hour
4.4	Coordinate conventions, Sampling and quantization, Spatial and Gray Level Resolution	1 hour
4.5	Basic relationship between pixels – neighbourhood, adjacency, connectivity	1 hour
4.6	Illustration of basic relationship between pixels– neighbourhood,	1 hour

	adjacency, connectivity	
4.7	Fundamentals of spatial domain - Convolution operation	1 hour
4.8	Illustration of Convolution operation	1 hour
Module - 5 (Image Enhancement in spatial domain and Image Segmentation) (11 hrs)		
5.1	Basic gray level transformation functions- Log transformations.	1 hour
5.2	Power-Law transformations, Contrast stretching	1 hour
5.3	Histogram equalization	1 hour
5.4	Illustration of Histogram equalization	1 hour
5.5	Basics of spatial filtering, Smoothing spatial filter- Linear and nonlinear filters	1 hour
5.6	Sharpening spatial filtering-Gradient filter mask	1 hour
5.7	Sharpening spatial filtering-Laplacian filter mask	1 hour
5.8	Fundamentals of Image Segmentation, Basics of Intensity thresholding, Basic Global Thresholding	1 hour
5.9	Region Based Approach- Region Growing, Region Splitting and Merging	1 hour1
5.10	Basics of Edge Detection	1 hour
5.11	Sobel and Prewitt edge detection masks	1 hour

CST 306	ALGORITHM ANALYSIS AND DESIGN	Category	L	T	P	Credit	Year of Introduction
		PCC	3	1	0	4	2019

Preamble:

The course introduces students to the design of computer algorithms, as well as analysis of algorithms. Algorithm design and analysis provide the theoretical backbone of computer science and are a must in the daily work of the successful programmer. The goal of this course is to provide a solid background in the design and analysis of the major classes of algorithms. At the end of the course students will be able to develop their own versions for a given computational task and to compare and contrast their performance.

Prerequisite:

Strong Foundation in Mathematics, Programming in C, Data Structures and Graph Theory.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Analyze any given algorithm and express its time and space complexities in asymptotic notations. **(Cognitive Level: Apply)**
CO2	Derive recurrence equations and solve it using Iteration, Recurrence Tree, Substitution and Master’s Method to compute time complexity of algorithms. **(Cognitive Level: Apply)**
CO3	Illustrate Graph traversal algorithms & applications and Advanced Data structures like AVL trees and Disjoint set operations. **(Cognitive Level: Apply)**
CO4	Demonstrate Divide-and-conquer, Greedy Strategy, Dynamic programming, Branch-and Bound and Backtracking algorithm design techniques (**Cognitive Level: Apply)**
CO5	Classify a problem as computationally tractable or intractable, and discuss strategies to address intractability **(Cognitive Level: Understand)**
CO6	Identify the suitable design strategy to solve a given problem. **(Cognitive Level: Analyze)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓										√
CO6	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of SeriesTests1& 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Introduction to Algorithm Analysis)

Characteristics of Algorithms, Criteria for Analysing Algorithms, Time and Space Complexity - Best, Worst and Average Case Complexities, Asymptotic Notations - Big-Oh (O), Big- Omega (Ω), Big-Theta (Θ), Little-oh (o) and Little- Omega (ω) and their properties. Classifying functions by their asymptotic growth rate, Time and Space Complexity Calculation of simple algorithms.

Analysis of Recursive Algorithms: Recurrence Equations, Solving Recurrence Equations – Iteration Method, Recursion Tree Method, Substitution method and Master's Theorem (Proof not required).

Module–2 (Advanced Data Structures and Graph Algorithms)

Self Balancing Tree - AVL Trees (Insertion and deletion operations with all rotations in detail, algorithms not expected); Disjoint Sets- Disjoint set operations, Union and find algorithms.

DFS and BFS traversals - Analysis, Strongly Connected Components of a Directed graph, Topological Sorting.

Module–3 (Divide & Conquer and Greedy Strategy)

The Control Abstraction of Divide and Conquer- 2-way Merge sort, Strassen's Algorithm for Matrix Multiplication-Analysis. The Control Abstraction of Greedy Strategy- Fractional Knapsack Problem, Minimum Cost Spanning Tree Computation- Kruskal's Algorithms - Analysis, Single Source Shortest Path Algorithm - Dijkstra's Algorithm-Analysis.

Module-4 (Dynamic Programming, Back Tracking and Branch & Bound))

The Control Abstraction- The Optimality Principle- Matrix Chain Multiplication-Analysis, All Pairs Shortest Path Algorithm - Floyd-Warshall Algorithm-Analysis. The Control Abstraction of Back Tracking – The N Queen's Problem. Branch and Bound Algorithm for Travelling Salesman Problem.

Module-5 (Introduction to Complexity Theory)

Tractable and Intractable Problems, Complexity Classes – P, NP, NP- Hard and NP-Complete Classes- NP Completeness proof of Clique Problem and Vertex Cover Problem- Approximation algorithms- Bin Packing, Graph Coloring. Randomized Algorithms (Definitions of Monte Carlo and Las Vegas algorithms), Randomized version of Quick Sort algorithm with analysis.

Text Books

1. T.H.Cormen, C.E.Leiserson, R.L.Rivest, C. Stein, Introduction to Algorithms, 2nd Edition, Prentice-Hall India (2001)
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, 2nd Edition, Orient Longman Universities Press (2008)

3. Sara Baase and Allen Van Gelder ―Computer Algorithms, Introduction to Design and Analysis, 3rd Edition, Pearson Education (2009)

Reference Books

1. Jon Kleinberg, Eva Tardos, “Algorithm Design”, First Edition, Pearson (2005)
2. Robert Sedgewick, Kevin Wayne, “Algorithms”,4th Edition Pearson (2011)
3. GIlles Brassard, Paul Brately, “Fundamentals of Algorithmics”, Pearson (1996)
4. Steven S. Skiena, “The Algorithm Design Manual”, 2nd Edition, Springer(2008)

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$? Justify your answer.
2. What is the need of asymptotic analysis in calculating time complexity? What are the notations used for asymptotic analysis?
3. Calculate the time complexity for addition of two matrices.
4. Define time complexity and space complexity. Write an algorithm for adding n natural numbers and analyse the time and space requirements of the algorithm.

Course Outcome 2 (CO2):

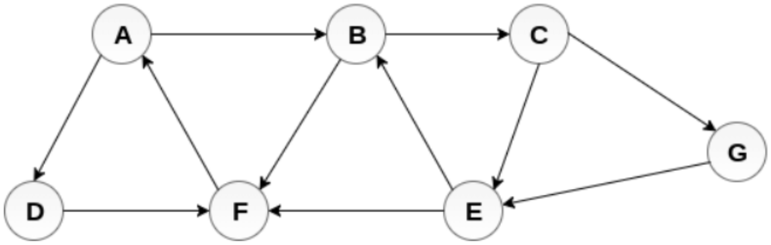
1. State Master’s theorem for solving recurrences.
2. Solve the recurrence T(n) = 3T(n-2), using iteration method
3. State the conditions in recurrences where Master Theorem is not applicable.
4. Solve the following recurrence equations using Master’s theorem.

   a) $T(n) = 8T(n/2) + 100\, n^2$

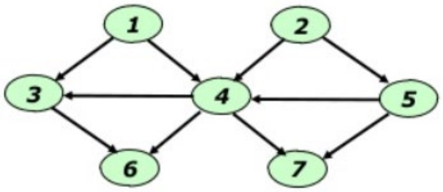
   b) $T(n) = 2T(n/2) + 10\, n$
5. Using Recursion Tree method, Solve T(n)= 2T(n/10)+ T(9n/10)+n. Assume constant time for small values of n.

Course Outcome 3 (CO3):

1. Explain the rotations performed for insertion in AVL tree with example.
2. Write down BFS algorithm and analyse the time complexity. Perform BFS traversal on the given graph starting from node A. If multiple node choices are available for next travel, choose the next node in alphabetical order.

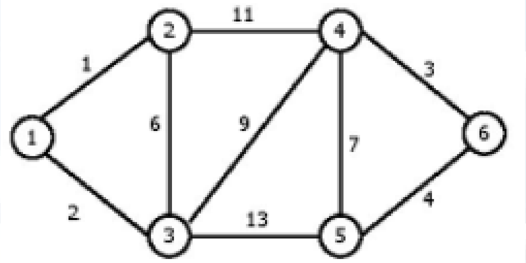


3. Find the minimum and maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0. (3)
4. Find any three topological orderings of the given graph.

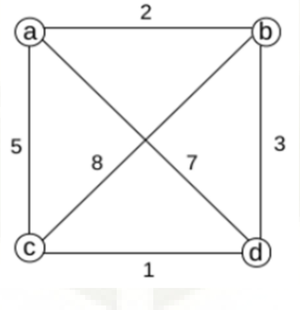


Course Outcome 4 (CO4):

1. Give the control abstraction for Divide and Conquer method.
2. Construct the minimum spanning tree for the given graph using Kruskal's algorithm. Analyse the complexity of the algorithm.



3. Compare Divide and Conquer and Dynamic programming methodologies
4. What is Principle of Optimality?
5. Define Travelling Salesman Problem (TSP). Apply branch and bound algorithm to solve TSP for the following graph, assuming the start city as 'a'. Draw the state space tree.



Course Outcome 5 (CO5):

1. Compare Tractable and Intractable Problems
2. With the help of suitable code sequence convince Vertex Cover Problem is an example of NP-Complete Problem

3. Explain Vertex Cover problem using an example. Suggest an algorithm for finding Vertex Cover of a graph.
4. Write short notes on approximation algorithms.
5. Compare Conventional quick sort algorithm and Randomized quicksort with the help of a suitable example?

Course Outcome 6 (CO6): (CO attainment through assignment only, not meant for examinations)

Choosing the best algorithm design strategy for a given problem after applying applicable design strategies – Sample Problems Given.

1. Finding the Smallest and Largest elements in an array of 'n' numbers
2. Fibonacci Sequence Generation.
3. Merge Sort
4. Travelling Sales Man Problem
5. 0/1 Knapsack Problem

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 306

Course Name: Algorithm Analysis and Design

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Define asymptotic notation? Arrange the following functions in increasing order of asymptotic growth rate.

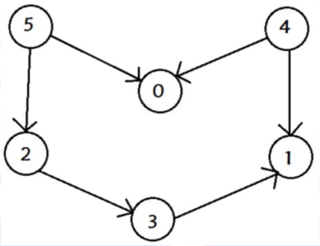
   n^3, 2^n, $\log n^3$, 2^{100}, $n^2 \log n$, n^n, $\log n$, $n^{0.3}$, $2^{\log n}$

2. State Master's Theorem. Find the solution to the following recurrence equations using Master's theorem.

   a) T (n) = 8T(n/2) + 100 n^2

   b) T (n) = 2T(n/2) + 10 n

3. Find any two topological ordering of the DAG given below.



4. Show the UNION operation using linked list representation of disjoint sets.

5. Write the control abstraction of greedy strategy to solve a problem.

6. Write an algorithm based on divide-and-conquer strategy to search an element in a given list. Assume that the elements of list are in sorted order.

7. List the sequence of steps to be followed in Dynamic Programming approach.

8. Illustrate how optimal substructure property could be maintained in Floyd-Warshall algorithm.

9. Differentiate between P and NP problems.

10. Specify the relevance of approximation algorithms.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Define Big O, Big Ω and Big Θ Notation and illustrate them graphically. **(7)**

    (b) Solve the following recurrence equation using recursion tree method **(7)**

    T(n) = T(n/3) + T(2n/3) + n , where n>1

    T(n) = 1, Otherwise

OR

12. (a) Explain the iteration method for solving recurrences and solve the following recurrence equation using iteration method. **(7)**

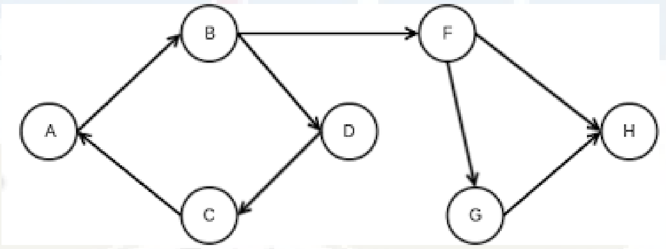
$$T(n) = 3T(n/3) + n; T(1) = 1$$

(b) Determine the time complexities of the following two functions fun1( ) and fun2( ). **(7)**

```
i)      int fun1(int n)
           {
              if (n <= 1) return n;
               return 2*fun1(n-1);
           }
ii)    int fun2 (int n)
            {
               if (n <= 1) return n;
               return fun2 (n-1) + fun2 (n-1)
            }
```

13. (a) Write DFS algorithm and analyse its time complexity. Illustrate the classification of edges in DFS traversal. **(7)**

(b) Find the strongly connected components of the digraph given below: **(7)**

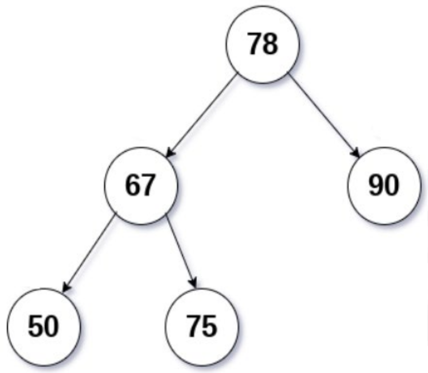


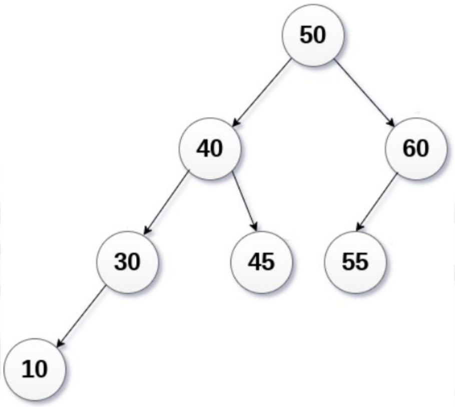
OR

14. (a) Illustrate the advantage of height balanced binary search trees over binary search trees? Explain various rotations in AVL trees with example. **(7)**

(b) Perform the following operations in the given AVL trees. **(7)**

i) Insert 70 ii) Delete 55





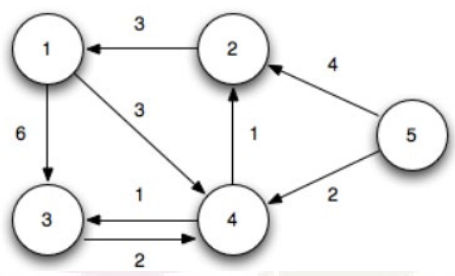
15. (a) State Fractional Knapsack Problem and write Greedy Algorithm for Fractional Knapsack Problem. **(7)**

(b) Find the optimal solution for the following Fractional Knapsack problem. Given the number of items(n) = 7, capacity of sack(m) = 15, W={2,3,5,7,1,4,1} and P = {10,5,15,7,6,18,3} **(7)**

OR

16. (a) Write and explain merge sort algorithm using divide and conquer strategy using the data {30, 19, 35, 3, 9, 46, 10}. Also analyse the time complexity. **(7)**

(b) Write the pseudo code for Dijkstra's algorithm. Compute the shortest distance from vertex 1 to all other vertices using Dijkstra's algorithm. **(7)**



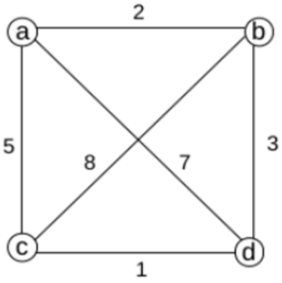
17. (a) Write Floyd-Warshall algorithm and analyse its complexity. **(5)**

(b) Write and explain the algorithm to find the optimal parenthesization of matrix chain product whose sequence of dimension is 4x10,10x3, 3x12,12x20. **(9)**

OR

18. (a) Explain the concept of Backtracking method using 4 Queens problem. **(7)**

(b) Define Travelling Salesman Problem (TSP). Apply branch and bound algorithm to solve TSP for the following graph, assuming the start city as 'a'. Draw the state space tree. **(7)**



19. (a) State bin packing problem? Explain the first fit decreasing strategy **(7)**

(b) Prove that the Clique problem is NP-Complete. **(7)**

OR

20. (a) Explain the need for randomized algorithms. Differentiate Las Vegas and Monte Carlo algorithms. **(6)**

(b) Explain randomized quicksort and analyse the expected running time of randomized quicksort with the help of a suitable example? **(9)**

Teaching Plan

No	Topic	No. of Hours (45 hrs)
Module -1 (Introduction to Algorithm Analysis) 9 hrs.		
1.1	Introduction to Algorithm Analysis: Characteristics of Algorithms.	1 hour
1.2	Criteria for Analysing Algorithms, Time and Space Complexity - Best, Worst and Average Case Complexities.	1 hour
1.3	Asymptotic Notations - Properties of Big-Oh (O), Big- Omega (Ω), Big-Theta (Θ), Little-Oh (o) and Little- Omega (ω).	1 hour
1.4	Illustration of Asymptotic Notations	1 hour

1.5	Classifying functions by their asymptotic growth rate	1 hour
1.6	Time and Space Complexity Calculation of algorithms/code segments.	1 hour
1.7	Analysis of Recursive Algorithms: Recurrence Equations, Solving Recurrence Equations – Iteration Method.	1 hour
1.8	Recursion Tree Method	1 hour
1.9	Substitution method and Master's Theorem and its Illustration.	1 hour
Module-2 (Advanced Data Structures and Graph Algorithms) 10 Hrs.		
2.1	Self Balancing Trees - Properties of AVL Trees, Rotations of AVL Trees	1 hour
2.2	AVL Trees Insertion and Illustration	1 hour
2.3	AVL Trees Deletion and Illustration	1 hour
2.4	Disjoint set operations.	1 hour
2.5	Union and find algorithms.	1 hour
2.6	Illustration of Union and find algorithms	1 hour
2.7	Graph Algorithms: BFS traversal, Analysis.	1 hour
2.8	DFS traversal, Analysis.	1 hour
2.9	Strongly connected components of a Directed graph.	1 hour
2.10	Topological Sorting.	1 hour
Module-3 (Divide & Conquer and Greedy Method) 8 Hrs		
3.1	Divide and Conquer: The Control Abstraction.	1 hour
3.2	2-way Merge Sort, Analysis.	1 hour
3.3	Strassen's Algorithm for Matrix Multiplication, Analysis	1 hour

3.4	Greedy Strategy: The Control Abstraction.	1 hour
3.5	Fractional Knapsack Problem.	1 hour
3.6	Minimum Cost Spanning Tree Computation- Kruskal's Algorithm, Analysis.	1 hour
3.7	Single Source Shortest Path Algorithm - Dijkstra's Algorithm	1 hour
3.8	Illustration of Dijkstra's Algorithm-Analysis.	1 hour
Module-4 (Dynamic Programming, Back Tracking and Branch and Bound) 8 Hrs.		
4.1	Dynamic Programming: The Control Abstraction, The Optimality Principle.	1 hour
4.2	Matrix Chain Multiplication-Analysis.	1 hour
4.3	Illustration of Matrix Chain Multiplication-Analysis.	1 hour
4.4	All Pairs Shortest Path Algorithm- Analysis and Illustration of Floyd-Warshall Algorithm.	1 hour
4.5	Back Tracking: The Control Abstraction .	1 hour
4.6	Back Tracking: The Control Abstraction – The N Queen's Problem.	1 hour
4.7	Branch and Bound:- Travelling salesman problem.	1 hour
4.8	Branch and Bound:- Travelling salesman problem.	1 hour
Module-5 (Introduction to Complexity Theory) 10 Hrs		
5.1	Introduction to Complexity Theory: Tractable and Intractable Problems.	1 hour
5.2	Complexity Classes – P, NP.	1 hour
5.3	NP- Hard and NP-Complete Problems.	1 hour
5.4	NP Completeness Proof of Clique Problem.	1 hour

5.5	NP Completeness Proof of Vertex Cover Problem.	1 hour
5.6	Approximation algorithms- Bin Packing Algorithm and Illustration.	1 hour
5.7	Graph Colouring Algorithm and Illustration.	1 hour
5.8	Randomized Algorithms (definitions of Monte Carlo and Las Vegas algorithms).	1 hour
5.9	Randomized Version of Quick Sort Algorithm with Analysis.	1 hour
5.10	Illustration of Randomized Version of Quick Sort Algorithm with Analysis.	1 hour

CST 308	COMPREHENSIVE COURSE WORK	Category	L	T	P	Credit	Year of Introduction
		PCC	1	0	0	1	2019

Preamble:

The objective of this Course work is to ensure the comprehensive knowledge of each student in the most fundamental core courses in the curriculum. Six core courses credited from Semesters 3, 4 and 5 are chosen for the detailed study in this course work. This course helps the learner to become competent in cracking GATE, placement tests and other competitive examinations

Prerequisite:

1. **Discrete Mathematical Structures**
2. **Data Structures**
3. **Operating Systems**
4. **Computer Organization And Architecture**
5. **Database Management Systems**
6. **Formal Languages And Automata Theory**

Course Outcomes: After the completion of the course the student will be able to

CO1	Comprehend the concepts of discrete mathematical structures (Cognitive Knowledge Level: **Understand)**
CO2 :	Comprehend the concepts and applications of data structures (Cognitive Knowledge Level: **Understand)**
CO3 :	Comprehend the concepts, functions and algorithms in Operating System (Cognitive Knowledge Level: **Understand)**)
CO4 :	Comprehend the organization and architecture of computer systems (Cognitive Knowledge Level: **Understand)**
CO5 :	Comprehend the fundamental principles of database design and manipulation (Cognitive Knowledge Level: **Understand)**
CO6 :	Comprehend the concepts in formal languages and automata theory Cognitive Knowledge Level: **Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓										✓
CO3	✓	✓										✓
CO4	✓	✓										✓
CO5	✓	✓										✓

Assessment Pattern

Bloom's Category	End Semester Examination
Remember	10
Understand	20
Apply	20
Analyse	
Evaluate	
Create	

Mark distribution

Total Marks	CIE	ESE	ESE Duration
50	0	50	1 hour

End Semester Examination Pattern: Objective Questions with multiple choice (Four). Question paper include fifty questions of one mark each covering the five identified courses.

Syllabus

Full Syllabus of all six selected Courses.

1. **Discrete Mathematical Structures**
2. **Data Structures**
3. **Operating Systems**
4. **Computer Organization And Architecture**
5. **Database Management Systems**
6. **Formal Languages And Automata Theory**

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	**DISCRETE MATHEMATICAL STRUCTURES (14 hours)**	
1.1	Mock Test on Module 1 and Module 2	1 hour
1.2	Mock Test on Module 3, Module 4 and Module 5	1 hour
2	**DATA STRUCTURES**	
2.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
2.2	Mock Test on Module 4 and Module 5	1 hour
3	**OPERATING SYSTEMS**	
3.1	Mock Test on Module 1 and Module 2	1 hour
3.2	Mock Test on Module 3, Module 4 and Module 5	1 hour
3.3	Feedback and Remedial	1 hour
4	**COMPUTER ORGANIZATION AND ARCHITECTURE**	
4.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
4.2	Mock Test on Module 4 and Module 5	1 hour
5	**DATABASE MANAGEMENT SYSTEMS**	

5.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
5.2	Mock Test on Module 4 and Module 5	1 hour
6	**FORMAL LANGUAGES AND AUTOMATA THEORY**	
6.1	Mock Test on Module 1, Module 2 and Module 3	1 hour
6.2	Mock Test on Module 4 and Module 5	1 hour
6.3	Feedback and Remedial	1 hour

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 10**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 308

Course Name: Comprehensive Course Work

Max. Marks: 50 **Duration: 1 Hour**

Objective type questions with multiple choices. Mark one correct answer for each question. Each Question Carries 1 Mark

1. What is the maximum possible number of relations from a set with 5 elements to another set with 4 elements?

   (A) 2^10 (B)2^16 (C)2^20 (D)2^25

2. The set {1,2,4,7,8,11,13,14} is a group under multiplication modulo 15. Find the inverse of element 13

   (A) 7 (B) 13 (C) 1 (D) 8

3. Consider the recurrence relation $a_1 = 2$, $a_n = 3n+a_{n-1}$ Then a_{72} is

(A) 7882 (B) 7883 (C) 7884 (D) 7885

4. Which among the following is a contradiction?
(A) $(p \wedge q) \vee \neg(p \vee q)$ (B) $(p \vee q) \wedge \neg(p \wedge q)$
(C) $(p \wedge q) \wedge \neg(p \vee q)$ (D) $(p \wedge q) \vee (p \wedge \neg q)$

5. The number of non-negative solutions to $x + y + z = 18$, with conditions $x \geq 3, y \geq 2, z \geq 1 is$
(A) 84 (B) 91 (C) 105 (D) 121

6. The solution of the recurrence relation $a_n = a_{n-1} + 2a_{n-2}$ with initial conditions $a_0 = 2, a_1 = 7$, is
(A) $3(2)^n - (-1)^n$ (B) $3(2)^n + (-1)^n$
(C) -$3(2)^n - (-1)^n$ (D) -$3(2)^n + (-1)^n$

7. Which among the following is not a subgroup of the set of Complex numbers under addition?
(A) R, the set of all Real numbers.
(B) Q^+, the set of positive rational numbers.
(C) Z, the set of all integers.
(D) The set iR of purely imaginary numbers including 0

8. Minimum number n of integers to be selected from $S = \{1,2,\ldots.,9\}$ to guarantee that the difference of two of the n integers is 5 is
(A) 3 (B) 4 (C) 6 (D) 9

9. Find the contrapositive the of statement "If it is a sunday, then I will wake up late"
(A) If I am not waking up late, then it is a suniday
(B) If I am not waking up late, then it is not a suniday
(C) If it is not a sunday, then I will not wake up late.
(D) It is not a sunday or I will wake up late

10. In the poset (Z^+, |) (where Z^+ is the set of all positive integers and | is the divides relation), which of the following are false?
I. 3 and 9 is comparable
II. 7 and 10 is comparable
III. The poset (Z+, |) is a total order
(A) I and III (B) II only (C) II and III (D) III only

11. Consider the following sequence of operations on an empty stack.
push(22); push(43); pop(); push(55); push(12); s=pop();

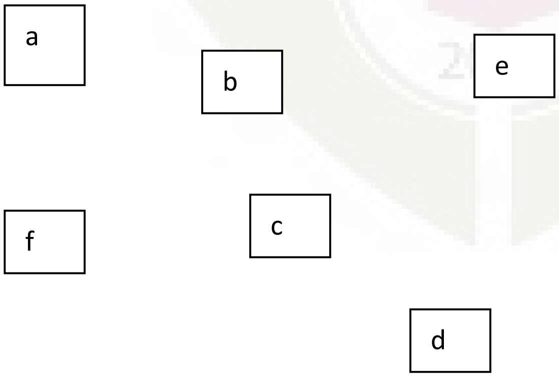
Consider the following sequence of operations on an empty queue.
enqueue(32);enqueue(27); dequeue(); enqueue(38); enqueue(12); q=dequeue();
The value of s+q is ___________
(A) 44 (B) 54 (C) 39 (D) 70

12. The following postfix expression with single digit operands is evaluated using a stack:
8 2 2 ^ / 4 3 * + 5 1 * -
Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are:
(A) 12,2 (B) 12,5 (C) 2,12 (D) 2,5

13. Construct a binary search tree by inserting 8, 6, 12, 3, 10, 9 one after another. To make the resulting tree as AVL tree which of the following is required?
(A) One right rotation only
(B) One left rotation followed by two right rotations
(C) One left rotation and one right rotation
(D) The resulting tree itself is AVL

14. In a complete 4-ary tree, every internal node has exactly 4 children or no child. The number of leaves in such a tree with 6 internal nodes is:
(A) 20 (B) 18 (C) 19 (D) 17

15. Consider the following graph with the following sequences
I. a b c f d e
II. a b e d f c
III. a b f c d e
IV. a f c b e d



Which are Depth First Traversals of the above graph?

(A) I, II and IV only (B) I and IV only
(C) II, III and IV only (D) I, III and IV only

16. Consider a hash table of size seven, with starting index zero, and a hash function (2x + 5) mod7. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 4, 9, 6 is inserted into the table using closed hashing? Note that '_' denotes an empty location in the table.
(A) 9, _, 1, 6, _, _, 4 (B) 1, _, 6, 9, _, _, 4
(C) 4, _, 9, 6, _, _, 1 (D) 1, _, 9, 6, _, _, 4

17. Consider the following C program where TreeNode represents a node in a binary tree

```
struct TreeNode{
struct TreeNode *leftChild;
struct TreeNode *rightChild;
int element;
};
int CountNodes(struct TreeNode *t)
{
if((t==NULL)||((t->leftChild==NULL) && (t->rightChild==NULL)))
        return 0;
else
{
        return 1+CountNodes(t->leftChild)+CountNodes(t->rightChild)
}
}
```

The value returned by CountNodes when a pointer to the root of a binary tree is passed as its argument is
(A) number of nodes
(B) number of leaf nodes
(C) number of non leaf nodes
(D) number of leaf nodes-number of non leaf nodes

18. How many distinct binary search trees can be created out of 6 distinct keys?
(A) 7 (B) 36 (C) 140 (D) 132

19. Suppose a disk has 400 cylinders, numbered from 0 to 399. At some time the disk arm is at cylinder 58, and there is a queue of disk access requests for cylinder 66, 349, 201, 110, 38, 84, 226, 70, 86. If Shortest-Seek Time First (SSTF) is being used for scheduling the disk access, the request for cylinder 86 is serviced after servicing ____________ number of

requests.
(A) 1 (B) 2 (C)3 (D)4

20. If frame size is 4KB then a paging system with page table entry of 2 bytes can address _______ bytes of physical memory.
(A) 2^12 (B) 2^16 (C) 2^18 (D) 2^28

21. Calculate the internal fragmentation if page size is 4KB and process size is 103KB.
(A) 3KB (B) 4KB (C) 1KB (D) 2KB

22. Which of the following scheduling policy is likely to improve interactiveness?
(A) FCFS (B) Round Robin
(C) Shortest Process Next (D) Priority Based Scgeduling

23. Consider the following program

```
          Semaphore X=1, Y=0
Void A ( )                                    Void B ( )
{                                             {
          While (1)                                     While (1)
          {                                             {
          P(X);                                         P(Y);
          Print'1';                                     P(X);
          V(Y);                                         Print'0';
          }                                             V(X);
}                                                       }
                                              }
```

The possible output of the program:
(A) Any number of 0's followed by any number of 1's.
(B) Any number of 1's followed by any number of 0's.
(C) 0 followed by deadlock
(D) 1 followed by deadlock

24. In a system using single processor, a new process arrives at the rate of 12 processes per minute and each such process requires 5 seconds of service time. What is the percentage of CPU utilization?
(A) 41.66 (B) 100.00 (C) 240.00 (D) 60.00

25. A system has two processes and three identical resources. Each process needs a maximum of two resources. This could cause
(A) Deadlock is possible (B) Deadlock is not possible

(C) Starvation may be present (D) Thrashing

26. Which of the following is true with regard to Round Robin scheduling technique?
(A) Responds poorly to short process with small time quantum.
(B) Works like SJF for larger time quantum
(C) Does not use a prior knowledge of burst times of processes.
(D) Ensure that the ready queue is always of the same size.

27. The size of the physical address space of a 32-bit processor is 2^W words. The capacity of cache memory is 2^N words. The size of each cache block is 2^K words. For a M-way set-associative cache memory, the length (in number of bits) of the tag field is
(A) $W - N + \log_2 M$ (B) $W - N - \log_2 M$
(C) $W - N - K - \log_2 M$ (D) $W - N - K + \log_2 M$

28. A 64-bit processor can support a maximum memory of 8 GB, where the memory is word-addressable (one word is of 64 bits). The size of the address bus of the processor is atleast ____ bits.
(A) 30 (B) 31 (C) 32 (D) None

29. The stage delays in a 4-stage pipeline are 900, 450, 400 and 350 picoseconds. The first stage (with delay 900 picoseconds) is replaced with a functionally equivalent design involving two stages with respective delays 600 and 550 picoseconds. The throughput increase of the pipeline is _______ percent.
(A) 38 (B) 30 (C) 58 (D) 50

30. Consider a direct mapped cache of size 256 Kilo words with block size 512 words. There are 6 bits in the tag. The number of bits in block (index) and word (offset) fields of physical address are is:
(A) block (index) field = 6 bits, word (offset) field = 9 bits
(B) block (index) field = 7 bits, word (offset) field = 8 bits
(C) block (index) field = 9 bits, word (offset) field = 9 bits
(D) block (index) field = 8 bits, word (offset) field = 8 bits

31. The memory unit of a computer has 1 Giga words of 64 bits each. The computer has instruction format, with 4 fields: an opcode field; a mode field to specify one of 12 addressing modes; a register address field to specify one of 48 registers; and a memory address field. If an instruction is 64 bits long, how large is the opcode field?
(A) 34 bits (B) 24 bits (C) 20 bits (D) 14 bits

32. A computer has 64-bit instructions and 28-bit address. Suppose there are 252 two-address instructions. How many 1-address instructions can be formulated?

(A) 2^24 (B) 2^26 (C) 2^28 (D) 2^30

33. Determine the number of clock cycles required to process 200 tasks in a six-segment pipeline.(Assume there were no stalls),each segment takes 1 cycle.
(A) 1200 cycles (B) 206 cycles (C) 207 cycles (D) 205 cycles

34. Match the following Lists:

P.DMA	1.Priority Interrupt
Q. Processor status Word	2.I/O Transfer
R. Daisy chaining	3.CPU
S. Handshaking	4.Asynchronous Data Transfer

(A) P-1, Q-3, R-4, S-2 (B) P-2, Q-3, R-1, S-4
(C) P-2, Q-1, R-3, S-4 (D) P-4, Q-3, R-1, S-2

35. Let E1, E2 and E3 be three entities in an E/R diagram with simple single-valued attributes. R1 and R2 are two relationships between E1 and E2, where R1 is one-to-many, R2 is many-to-many. R3 is another relationship between E2 and E3 which is many-to-many. R1, R2 and R3 do not have any attributes of their own. What is the minimum number of tables required to represent this situation in the relational model?
(A) 3 (B) 4 (C) 5 (D) 6

36. Identify the minimal key for relational scheme R(U, V, W, X, Y, Z) with functional dependencies F = {U → V, V → W, W → X, VX → Z}
(A) UV (B) UW (C) UX (D) UY

37. It is given that: “Every student need to register one course and each course registered by many students”, what is the cardinality of the relation say “Register” from the “Student” entity to the “Course” entity in the ER diagram to implement the given requirement.
(A) M:1 relationship (B) M:N relationship
(C) 1:1 relationship (D) option (B) or(C)

38. Consider the relation branch(branch_name, assets, branch_city)
SELECT DISTINCT T.branch_name FROM branch T, branch S WHERE T.assets > L.assets AND S.branch_city = "TVM" .
Finds the names of
(A) All branches that have greater assets than all branches located in TVM.
(B) All branches that have greater assets than some branch located in TVM.
(C) The branch that has the greatest asset in TVM.
(D) Any branch that has greater asset than any branch located in TVM.

39. Consider the following relation instance, where “A” is primary Key.

A1	**A2**	**A3**	**A4**
1	1	1	Null
5	2	5	1
9	5	13	5
13	13	9	15

Which one of the following can be a foreign key that refers to the same relation?
(A) A2 (B) A3 (C) A4 (D) ALL

40. A relation R(ABC) is having the tuples(1,2,1),(1,2,2),(1,3,1) and (2,3,2). Which of the following functional dependencies holds well?
(A) A → BC (B) AC → B (C) AB → C (D) BC → A

41. Consider a relation R with attributes A, B, C, D and E and functional dependencies A → BC, BC → E, E →DA. What is the highest normal form that the relation satisfies?
(A) BCNF (B) 3 NF (C) 2 NF (D) 1 NF

42. For the given schedule S, find out the conflict equivalent schedule.
S : r1(x); r2(Z) ; r3(X); r1(Z); r2(Y); r3(Y);W1(X); W2(Z); W3(Y); W2(Y)
(A) T1→T2→T3 (B) T2->T1->T3
(C) T3→T1→T2 (D) Not conflict serializable

43. Which of the following strings is in the language defined by the grammar:
S → aX
X → aX | bX | b
(A) aaaba (B) babab (C) aaaaa (D) ababb

44. Consider the regular expression (x+y)*xyx(x+y)* where Σ = (x,y). If L is the language represented by this regular expression, then what will be the minimum number of states in a DFA recognizing L ?
(A) 2 (B) 3 (C) 4 (D) 5

45. Which of the following cannot handle the same set of languages?
(A) Deterministic Finite Automata and Non-Deterministic Finite Automata
(B) Deterministic Push Down Automata and Non-Deterministic Push Down Automata
(C) All of these
(D) None of these

46. Consider L be a context-free language and M be a non-context-free language. Which among the following is TRUE?

(I) L will definitely pass the pumping lemma test for CFLs.
(II) M will definitely pass the pumping lemma test for CFLs.
(III) L will not definitely pass the pumping lemma test for CFLs.
(IV) M will not definitely pass the pumping lemma test for CFLs.
(V) L may or maynot pass the pumping lemma test for CFLs.
(VI) M may or maynot pass the pumping lemma test for CFLs.
(A) I, II (B) II, V (C) I, VI (D) IV, V

47. Which of the following problem(s) is/are decidable?
(I) Whether a CFG is empty or not.
(II) Whether a CFG generates all possible strings.
(III) Whether the language generated by a Turing Machine is regular.
(IV) Whether the language generated by DFA and NFA are same.
(A) I and II (B) II and III (C) II and IV (D) I and IV

48. Which of the following is/are TRUE?
(I) Regular languages are closed under complementation.
(II) Recursive languages are closed under complementation.
(III) Context free languages are closed under complementation.
(IV) Context free languages are not closed under complementation.
(A) I, II and III (B) I, II and IV (C) II and III (D) III only

49. Which of the following regular expressions defined over the alphabet $\Sigma = \{0,1\}$ defines the language of all strings of length l where l is a multiple of 3?
(A) (0 + 1 + 00 + 11 + 000 +111)* (B) (000 + 111)*
(C) ((0 + 1)(0 + 1)(0 + 1))* (D) ((000 + 01 + 1)(111 + 10 + 0))*

50. Determine the minimum number of states of a DFA that recognizes the language over the alphabet {a,b} consisting of all the strings that contain at least three a's and at least four b's.

(A) 6 (B) 12 (C) 15 (D) 20

ANSWER KEY:-

QNo	Ans. Key	QNo	Ans. Key	QNo	Ans. Key	QNo	Ans. Key	QNo	Ans. Key
1	(C)	11	(C)	21	(C)	31	(B)	41	(A)

2	(A)	12	(A)	22	(B)	32	(D)	42	(D)
3	(B)	13	(A)	23	(D)	33	(D)	43	(D)
4	(C)	14	(C)	24	(B)	34	(B)	44	(C)
5	(B)	15	(A)	25	(B)	35	(C)	45	(B)
6	(A)	16	(D)	26	(C)	36	(D)	46	(C)
7	(B)	17	(C)	27	(A)	37	(A)	47	(D)
8	(C)	18	(D)	28	(A)	38	(B)	48	(B)
9	(B)	19	(C)	29	(D)	39	(B)	49	(C)
10	(C)	20	(D)	30	(C)	40	(D)	50	(D)

CSL332	NETWORKING LAB	CATEGORY	L	T	P	Credit	Year of Introduction
		PCC	0	0	3	2	2

Preamble:

The course enables the learners to get hands-on experience in network programming using Linux System calls and network monitoring tools. It covers implementation of network protocols and algorithms, configuration of network services and familiarization of network simulators. This helps the learners to develop, implement protocols and evaluate its performance for real world networks.

Prerequisite: Sound knowledge in Programming in C, Data Structures and Computer Networks

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Use network related commands and configuration files in Linux Operating System. **(Cognitive Knowledge Level: Understand).**
CO2	Develop network application programs and protocols. **(Cognitive Knowledge Level: Apply)**
CO3	Analyze network traffic using network monitoring tools. **(Cognitive Knowledge Level: Apply)**
CO4	Design and setup a network and configure different network protocols. **(Cognitive Knowledge Level: Apply)**
CO5	Develop simulation of fundamental network concepts using a network simulator. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓					✓		✓		✓
CO2	✓	✓	✓	✓				✓		✓		✓
CO3	✓	✓	✓	✓	✓			✓		✓		✓

CO4	✓	✓	✓	✓	✓	✓		✓		✓		✓
CO5	✓	✓	✓	✓	✓			✓		✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and teamwork
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Test (Internal Exam) Marks in percentage**	**End Semester Examination Marks in percentage**
Remember	**20**	**20**
Understand	**20**	**20**
Apply	**60**	**60**
Analyze		
Evaluate		
Create		

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**75**	**75**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	: **15 marks**
Continuous Evaluation in Lab	: **30 marks**
Continuous Assessment Test	: **15 marks**
Viva voce	: **15 marks**

Internal Examination Pattern:

The Internal examination shall be conducted for 100 marks, which will be converted to out of 15, while calculating Internal Evaluation marks. The marks will be distributed as, Algorithm - 30 marks, Program - 20 marks, Output - 20 marks and Viva - 30 marks.

End Semester Examination Pattern:

The End Semester Examination will be conducted for a total of 75 marks and shall be distributed as, Algorithm - 30 marks, Program - 20 marks, Output - 20 marks and Viva- 30 marks.

Operating System to Use in Lab	: Linux
Compiler/Software to Use in Lab	: gcc, NS2
Programming Language to Use in Lab	: Ansi C

Fair Lab Record:

All the students attending the Networking Lab should have a Fair Record. Every experiment conducted in the lab should be noted in the fair record. For every experiment, in the fair record, the right hand page should contain experiment heading, experiment number, date of experiment, aim of the experiment, procedure/algorithm followed, other such details of the experiment and final result. The left hand page should contain a print out of the respective code with sample input and corresponding output obtained. All the experiments noted in the fair record should be verified by the faculty regularly. The fair record, properly certified by the faculty, should be produced during the time of End Semester Examination for the verification by the examiners.

Syllabus

***Mandatory**

(Note: At least one program from each topic in the syllabus should be completed in the Lab)

1. Getting started with the basics of network configuration files and networking commands in Linux.*
2. To familiarize and understand the use and functioning of system calls used for network programming in Linux.*
3. Implement client-server communication using socket programming and TCP as transport layer protocol*
4. Implement client-server communication using socket programming and UDP as transport layer protocol*
5. Simulate sliding window flow control protocols.* (Stop and Wait, Go back N, Selective Repeat ARQ protocols)
6. Implement and simulate algorithm for Distance Vector Routing protocol or Link State Routing protocol.*
7. Implement Simple Mail Transfer Protocol.
8. Implement File Transfer Protocol.*
9. Implement congestion control using a leaky bucket algorithm.*
10. Understanding the Wireshark tool.*
11. Design and configure a network with multiple subnets with wired and wireless LANs using required network devices. Configure commonly used services in the network.*
12. Study of NS2 simulator*

Networking Lab-Practice Questions

1. a) View the configuration, including addresses of your computers network interfaces.

   b) Test the network connectivity between your computer and several other computers.

   c) View the active TCP connections in the computer after visiting a website.

   d) Find the hardware/MAC address of another computer in the network using ARP.
2. Write the system calls used for creating sockets and transferring data between two nodes.
3. a) Implement a multi-user chat server using TCP as transport layer protocol.

   b) Implement a simple web proxy server that accepts HTTP requests and forwarding to remote servers and returning data to the client using TCP

4. Implement a Concurrent Time Server application using UDP to execute the program at a remote server. Client sends a time request to the server, server sends its system time back to the client. Client displays the result.

5. a) Implement Stop-and-Wait ARQ flow control protocol.

   b) Implement Go-Back--N ARQ flow control protocol.

   c) Implement Selective Repeat ARQ flow control protocol.

6. Implement Distance Vector Routing algorithm or Link State Routing algorithm..

7. Implement Simple Mail Transfer Protocol.

8. Develop a concurrent file server which will provide the file requested by a client if it exists. If not, the server sends appropriate message to the client. Server should also send its process ID (PID) to clients for display along with the file or the message.

9. Implement leaky bucket algorithm for congestion control.

10. a) Using Wireshark, Capture packets transferred while browsing a selected website. Investigate the protocols used in each packet, the values of the header fields and the size of the packet.

    b) Using Wireshark, observe three way handshaking connection establishment, three way handshaking connection termination and Data transfer in client server communication using TCP.

    c) Explore at least the following features of Wireshark: filters, Flow graphs (TCP), statistics, and protocol hierarchies.

11. Design and configure a network (wired and wireless LANs) with multiple subnets using required network devices. Configure at least three of the following services in the network- TELNET, SSH, FTP server, Web server, File server, DHCP server and DNS server.

12. a) The network consists of TCP source node (n0) and destination node (n1) over an area size of 500m x 500m. Node (n0) uses Agent/TCP/Reno as the sending TCP agent and FTP traffic source. Node (n1) is the receiver of FTP transfers, and it uses Agent/TCP sink as its TCP-agent for the connection establishment. Run the simulation for 150 seconds and show the TCP window size in two static nodes scenario with any dynamic routing protocol. Run the script and analyze the output graph for the given scenario.

    b) Simulate the transmission of ping messages over a star network topology consisting of ‘n’ nodes and find the number of packets dropped due to congestion using NS2simulator.

    c) Simulate Link State Protocol or Distance Vector Routing protocol in NS2.

Reference Books:

1. W. Richard Stevens, Bill Fenner, Andy Rudoff, UNIX Network Programming: Volume 1, The Sockets Networking API, 3rd Edition, Pearson, 2015
2. Lisa Bock, Learn Wireshark: Confidently navigate the Wireshark interface and solve real-world networking problems, Packt Publishing, 2019
3. Teerawat Issariyakul, Ekram Hossain, Introduction to Network Simulator NS2,2nd Edition, Springer,2019

CSD 334	MINI PROJECT	Category	L	T	P	Credit	Year of Introduction
		PCC	0	0	3	2	2019

Preamble:

The objective of this course is to apply the fundamental concepts of Software Engineering principles for the effective development of an application/research project. This course helps the learners to practice the different steps to be followed in the software development process such as literature review and problem identification, preparation of Software Requirement Specification &Software Design Document (SDD), testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

Prerequisite:

A sound knowledge in any programming language and fundamental concepts of Software Engineering.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Identify technically and economically feasible problems **(Cognitive Knowledge Level: Apply)**
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes **(Cognitive Knowledge Level: Apply)**
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (**Cognitive Knowledge Level: Apply)**
CO4	Prepare technical report and deliver presentation **(Cognitive Knowledge Level: Apply)**
CO5	Apply engineering and management principles to achieve the goal of the project **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	75	75	

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Project Guide	**15 marks**
Project Report	**10 marks**

Evaluation by the Committee (will be evaluating the level of completion
and demonstration of functionality/specifications, presentation,
oral examination, work knowledge and involvement) : **40 marks**

Student Groups with 3 or 4 members should identify a topic of interest in consultation with a Faculty/Advisor. Review the literature and gather information pertaining to the chosen topic. State the objectives and develop a methodology to achieve the objectives. Carryout the design/fabrication or develop codes/programs to achieve the objectives by strictly following steps specified in the teaching plan. Innovative design concepts, performance, scalability, reliability considerations, aesthetics/ergonomic, user experience and security aspects taken care of in the project shall be given due weight.

The progress of the mini project is evaluated based on a minimum of two reviews. The review committee may be constituted by the Head of the Department comprising HoD or a senior faculty member, Mini Project coordinator and project guide. The internal evaluation shall be made based on the progress/outcome of the project, reports and a viva-voce examination, conducted internally by a 3-member committee. A project report is required at the end of the semester. The project has to be demonstrated for its full design specifications.

End Semester Examination Pattern:

The marks will be distributed as

Presentation	: **30 marks**
Demo	: **20 marks**
Viva	: **25 marks.**
Total	: **75 marks**.

TEACHING PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc. and get department approval. Register Online course/ Collect study materials.
3. Create Software Requirements Specification (SRS Document)
4. Create Software Design Document (SDD). This may include designs like,
    a. System Architecture Design
    b. Application Architecture Design
    c. GUI Design (Mockups)
    d. API Design
    e. Database Design
    f. Technology Stack
5. Create Test Plan, Test Scenarios and Test Cases (Test Case Document) & Traceability Matrix
6. Create a Project Plan (with Modules, Tasks, Resources, Time schedule) [May use any project management tool or excel for this] – Choose and follow agile or waterfall models.
7. Development
    a. Set coding standards
    b. Environment Setup
    c. Source Code Control Setup (Like Subversion(SVN), Git)
    d. Development
    e. Unit Testing
    f. Integration Testing
    g. Testing /Quality Assurance(QA)
        i. Functional Testing
        ii. Load Testing
        iii. Report Bugs
    h. Resolve Bugs & Retest

8. Deployment (of software from local development environment to a production environment)
9. Test Run & Get Results
10. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman18, Bold; Heading 2 – Times New Roman16, Bold; Heading 3 – Times New Roman14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.
- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figuretitle under the figure and table title above the table.

- **Suggestive order of documentation:**

  i. Top Cover
  ii. Title page
  iii. Certification page
  iv. Acknowledgement
  v. Abstract
  vi. Table of Contents
  vii. List of Figures and Tables
  viii. Chapters
  ix. Appendices, if any
  x. References/Bibliography

SEMESTER VI

PROGRAM ELECTIVE I

CST 312	FOUNDATIONS OF MACHINE LEARNING	Category	L	T	P	Credit	Year of Introduction
		PEC	2	1	0	3	2019

Preamble:

This course enables the learners to understand the mathematical foundations of Machine Learning concepts. This course covers Linear Algebra, Probability and Distributions. Concepts in this course help the learners to identify the inherent assumptions & limitations of the current methodologies and develop new Machine Learning solutions.

Prerequisite: A sound background in higher secondary school Mathematics.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Illustrate operations and applications of linear equations, matrix algebra, vector spaces, eigen values & eigenvectors **(Cognitive Knowledge Level: Apply)**
CO 2	Illustrate the concepts of orthogonality & diagonalization. **(Cognitive Knowledge Level: Apply)**
CO 3	Solve computational problems using probability and random variables. **(Cognitive Knowledge Level: Apply)**
CO 4	Identify an appropriate probability distribution for a given discrete or continuous random variable and use its properties. (**Cognitive Knowledge Level: Apply)**
CO 5	Illustrate moment generating function, law of large numbers and central limit theorems **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	✓	✓	✓	✓								✓
CO 2	✓	✓	✓	✓								✓
CO 3	✓	✓	✓	✓								✓
CO 4	✓	✓	✓	✓								✓

CO 5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination**
	1	**2**	
Remember	**30%**	**30%**	**30%**
Understand	**30%**	**30%**	**30%**
Apply	**40%**	**40%**	**40%**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	**: 10 marks**
Continuous Assessment Tests	**: 25 marks**
Continuous Assessment Assignment	**: 15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer anyone. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module 1 (LINEAR ALGEBRA)

Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence, Basis and Rank, Linear Mappings.

Module 2 (LINEAR ALGEBRA)

Norms - Inner Products, Lengths and Distances, Angles and Orthogonality. Orthonormal Basis, Orthogonal Complement, Orthogonal Projections. Matrix Decompositions - Eigenvalues and Eigenvectors, Eigen decomposition and Diagonalization.

Module 3 (PROBABILITY AND DISTRIBUTIONS)

Probability Space - Sample Spaces, Probability Measures, Computing Probabilities, Conditional Probability, Baye's Rule, Independence. Random Variables - Discrete Random Variables (Bernoulli Random Variables, Binomial Distribution, Geometric and Poisson Distribution, Continuous Random Variables (Exponential Density, Gamma Density, Normal Distribution, Beta Density)

Module 4 (RANDOM VARIABLES)

Functions of a Random Variable. Joint Distributions - Independent Random Variables, Conditional Distributions, Functions of Jointly Distributed Random Variables.

Expected Values - Expected Value of a Random Variable, Expectations of Functions of Random Variables, Expectations of Linear Combinations of Random Variables, Variance and Standard Deviation, Covariance and Correlation, Conditional Expectation

Module 5 (LIMIT THEOREMS)

Moment-Generating Function. Limit Theorems(Proof not expected) - Law of Large Numbers, Convergence in Distribution and the Central Limit Theorem. Distributions derived from the Normal Distribution - Chi-square, t, and F Distributions, Sample Mean and the Sample Variance.

Text book:

1. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Mathematics for Machine Learning, Cambridge University Press (freely available at https://mml – book.github.io)
2. John A. Rice, Mathematical Statistics and Data Analysis, University of California, Berkeley, Third edition, published by Cengage.

Reference books:

1. Gilbert Strang, Linear Algebra and Its Applications, 4th Edition,
2. Axler, Sheldon, Linear Algebra Done Right, 2015 Springer
3. Stephen Boyd and Lieven Vandenberghe, Introduction to Applied Linear Algebra, 2018 published by Cambridge University Press

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Find the set ***S*** of all solutions in ***x*** of the following inhomogeneous linear systems ***Ax = b***, where ***A*** and ***b*** are defined as follows:

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

2. Determine the inverses of the following matrix if possible

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

3. Are the following sets of vectors linearl independent?

$$\boldsymbol{x}_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

4. A set of ***n*** linearly independent vectors in $\boldsymbol{R^n}$ forms a basis. Does the set of vectors **(2, 4,−3) (0, 1, 1)** , **(0, 1,−1**) form a basis for $\boldsymbol{R^3}$? Explain your reasons.

Course Outcome 2 (CO2):

1. Determine which of the following sets are orthogonal sets.

$$\left\{\begin{bmatrix}3\\1\\1\end{bmatrix},\begin{bmatrix}-1\\2\\1\end{bmatrix},\begin{bmatrix}-1/2\\-2\\7/2\end{bmatrix}\right\} \qquad \left\{\begin{bmatrix}1\\-1\\1\end{bmatrix},\begin{bmatrix}2\\1\\-1\end{bmatrix},\begin{bmatrix}3\\0\\-3\end{bmatrix}\right\} \qquad \left\{\begin{bmatrix}3\\-2\\1\\3\end{bmatrix},\begin{bmatrix}-1\\3\\-3\\4\end{bmatrix},\begin{bmatrix}3\\8\\7\\0\end{bmatrix}\right\}$$

2. Find the characteristic equation, eigenvalues, and eigenspaces corresponding to each eigenvalue of the following matrix.

$$\begin{bmatrix}2 & 0 & 4\\0 & 3 & 0\\0 & 1 & 2\end{bmatrix}$$

3. Diagonalize the following matrix, if possible

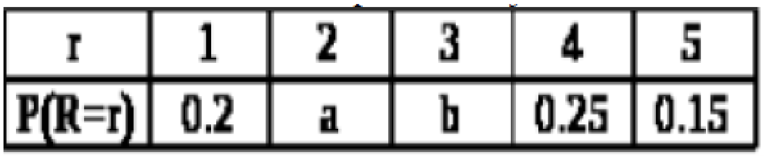
$$\begin{bmatrix}3 & 0 & 0 & 0\\0 & 2 & 0 & 0\\0 & 0 & 2 & 0\\1 & 0 & 0 & 3\end{bmatrix}$$

Course Outcome 2 (CO3):

1. Let ***J*** and ***T*** be independent events, where ***P(J)=0.4*** and ***P(T)=0.7***.
   - i. Find ***P(J∩T)***
   - ii. Find ***P(J∪T)***
   - iii. Find ***P(J∩T')***

2. Let ***A*** and ***B*** be events such that ***P(A)=0.45*** , ***P(B)=0.35*** and ***P(A∪B)=0.5***. Find ***P(A/B)***.

3. A random variable **R** has the probability distribution as shown in the following table:



r	1	2	3	4	5
P(R=r)	0.2	a	b	0.25	0.15

i. Given that ***E(R)=2.85***, find ***a*** and ***b***.
ii. Find ***P(R>2)***.

4. A biased coin (with probability of obtaining a head equal to ***p > 0***) is tossed repeatedly and independently until the first head is observed. Compute the probability that the first head appears at an even numbered toss.

5. Two players A and B are competing at a quiz game involving a series of questions. On any individual question, the probabilities that A and B give the correct answer are ***p*** and ***q*** respectively, for all questions, with outcomes for different questions being independent. The game finishes when a player wins by answering a question correctly. Compute the probability that A wins if
   i. A answers the first question,
   ii. B answers the first question.

6. A coin for which ***P(heads) = p*** is tossed until two successive tails are obtained. Find the probability that the experiment is completed on the ***n***th toss.

Course Outcome- 3 (CO4):

1. An urn contains **p** black balls, **q** white balls, and **r** red balls; and **n** balls are chosen without replacement.
   a. Find the joint distribution of the numbers of black, white, and red balls in the sample.
   b. Find the joint distribution of the numbers of black and white balls in the sample.
   c. Find the marginal distribution of the number of white balls in the sample.
2. Suppose that two components have independent exponentially distributed lifetimes, T_1 and T_2, with parameters α and β, respectively. Find (a) $P(T_1 > T_2)$ and (b) $P(T_1 > 2\, T_2)$.
3. Let Z_1 and Z_2 be independent random variables each having the standard normal distribution. Define the random variables X and Y by $X = Z_1 + 3Z_2$ and $Y = Z_1 + Z_2$. Argue that the joint distribution of (X, Y) is a bivariate normal distribution. What are the parameters of this distribution?

4. Given a continuous random variable ***x***, with cumulative distribution function $F_x(x)$, show that the random variable $y = F_x(x)$ is uniformly distributed.
5. You roll a fair dice twice. Let the random variable ***X*** be the product of the outcomes of the two rolls. What is the probability mass function of ***X***? What are the expected values and the standard deviation of ***X***?
6. Let ***X*** be a continuous random variable with the density function $f(x) = 2x, 0 \leq x \leq 1$

   a. Find $E(X)$.

   b. Find $E(X^2)$ and $Var(X)$.

Course Outcome 5 (CO5):

1. Find the moment-generating function of a Bernoulli random variable, and use it to find the mean, variance, and third moment.
2. Use moment-generating functions to show that if **X** and **Y** are independent, then $Var(aX + bY) = a^2Var(X) + b^2Var(Y)$.
3. Suppose that you bet Rs 5 on each of a sequence of 50 independent fair games. Use the central limit theorem to approximate the probability that you will lose more than Rs 75.
4. Suppose that the number of insurance claims, ***N***, filed in a year is Poisson distributed with $E(N) = 10{,}000$. Use the normal approximation to the Poisson to approximate $P(N > 10{,}200)$.

Model Question paper

QP Code : **Total Pages: 4**

Reg No.:________________ Name:____________________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION (ELECTIVE), MONTH and YEAR

Course Code: CST 312

Course Name: FOUNDATIONS OF MACHINE LEARNING

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks.

1 Show that with the usual operation of scalar multiplication but with addition on reals given by $x \# \ y = 2(x + y)$ is not a vector space.

2 Are the following vectors linearly independent? Justify your answer.

$$x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

3 Find the eigenvalues of the following matrix in terms of k. Can you find an eigenvector corresponding to each of the eigenvalues?

$$\begin{bmatrix} 1 & k \\ 2 & 1 \end{bmatrix}$$

4 Find a unit vector in $\mathbf{R}^2$ that is orthogonal to **(-1 , 2)**.

5 The first three digits of a telephone number are 452. If all the sequences of the remaining four digits are equally likely, what is the probability that a randomly selected telephone number contains seven distinct digits?

6 Show that if two events $\boldsymbol{A}$ and $\boldsymbol{B}$ are independent, then $\boldsymbol{A}$ and $\boldsymbol{B'}$ are independent.

7 Prove that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent if and only if $f_{X|Y}(x|y) = f_X(x)$ for all $\boldsymbol{x}$ and $\boldsymbol{y}$.

8 If $\boldsymbol{X}$ is a discrete uniform random variable, i.e., $P(X = k) = 1/n$ for $k = 1, 2, \dots , n$, find $E(X)$ and $Var(X)$.

9 Compare the Poisson cdf and the normal approximation for (a) $\lambda = \mathbf{10}$, (b) $\lambda = \mathbf{20}$, and (c) $\lambda = \mathbf{40}$.

10 State law of large numbers.

10 x 3 = 30

PART B

Answer any one Question from each module. Each question carries 14 Marks

11 a) Find all solutions to the system of linear equations **(8)**

$$\begin{aligned} -4x + 5z &= -2 \\ -3x - 3y + 5z &= 3 \\ -x + 2y + 2z &= -1 \end{aligned}$$

b) Consider the transformation $\boldsymbol{T(x, y) = (x + y, x + 2y, 2x + 3y)}$. Obtain **ker T** and use this to calculate the nullity. Also find the transformation matrix for **T**. **(6)**

OR

12 a) Consider the following linear mapping **(8)**

$$\Phi : \mathbb{R}^3 \to \mathbb{R}^4$$

$$\Phi\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = \begin{bmatrix} 3x_1 + 2x_2 + x_3 \\ x_1 + x_2 + x_3 \\ x_1 - 3x_2 \\ 2x_1 + 3x_2 + x_3 \end{bmatrix}$$

i. Find the transformation matrix T.

ii. Determine rank(T).

iii. Compute the kernel and image of the mapping and find their dimension

b) Prove that all vectors orthogonal to $\mathbf{[2, -3, 1]^T}$ forms a subspace $\boldsymbol{W}$ of $\boldsymbol{R^3}$. What is ***dim (W)*** and why? **(6)**

13 a) Find an orthonormal basis of $\boldsymbol{R^3}$ consisting of eigenvectors for the following matrix **(8)**

$$\begin{bmatrix} 1 & 0 & -2 \\ 0 & 5 & 0 \\ -2 & 0 & 4 \end{bmatrix}$$

b) Find a 3 × 3 orthogonal matrix $\boldsymbol{S}$ and a 3 × 3 diagonal matrix $\boldsymbol{D}$ such that $\boldsymbol{A = SDS^T}$ **(6)**

OR

14 a) Find an orthogonal basis for the subspace of $\boldsymbol{R^4}$ spanned by $\boldsymbol{\{ w_1 = (1, 1, 3, 2), w_2 = (1, -2, 0, -1), w_3 = (0, 2, 1, 2) \}}$. **(8)**

b) Find the characteristic equation, eigenvalues, and eigenspaces corresponding tc each eigenvalue of the following matrix **(6)**

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

15 a) Three players play 10 independent rounds of a game, and each player has probability 1/3 of winning each round. Find the joint distribution of the numbers of games won by each of the three players. **(7)**

b) An experiment consists of throwing a fair coin four times. Find the probability mass function and the cumulative distribution function of the following random variables: **(7)**

i. the number of heads before the first tail

ii. the number of heads following the first tail

iii. the number of heads minus the number of tails

iv. the number of tails times the number of heads.

OR

16 a) A factory runs three shifts. On a given day, 1% of the items produced by the first shift are defective, 2% of the second shift's items are defective, and 5% of the third shift's items are defective. If the shifts all have the same productivity, what percentage of the items produced in a day are defective? If an item is defective, what is the probability that it was produced by the third shift? **(8)**

b) Show that if A and B are two independent events, then $P(A \cup B) = P(A) + P(B) - P(A)P(B)$ **(6)**

17 a) Find the joint density of $X + Y$ and X/Y, where X and Y are independent exponential random variables with parameter λ. Show that $X + Y$ and X/Y are independent. **(8)**

b) Let X be a discrete random variable that takes on values 0, 1, 2 with probabilities 1/2 , 3/8 , 1/8 , respectively. **(6)**

i. Find **E(X)** and **Var(X)**.

ii. Let $Y = X^2$. Find the probability mass function of Y and use it to find **E(Y)**.

18 a) A random square has a side length that is a uniform [0, 1] random variable. Find the expected area of the square. **(7)**

b) Let X be a continuous random variable with probability density function on $0 <= x <= 1$ defined by $f(x) = 3x^2$. Find the pdf of $Y = X^2$. **(7)**

19 a) Using the fact that the mean of the chi-squared distribution is $(n-1)$, prove that $E(S^2) = \sigma^2$. **(7)**

b) i. Random samples of size 36 are taken from an infinite population whose mean is 80 and standard deviation is 18. Find the mean and standard error of the **(7)** sampling distribution.

ii. Why is the Central Limit Theorem so important to statistical analysis?

OR

20 a) A six-sided die is rolled 100 times. Using the normal approximation, find the probability that the face showing a six turns up between 15 and 20 times. Find the probability that the sum of the face values of the 100 trials is less than 300. **(8)**

b) Determine an interval **(a, b)** such that ***P[a ≤ t ≤ b] = 0.80***, and that 10% of the area is on each side of ***a*** and ***b***, assuming that the sample is of size 21. **(6)**

Teaching Plan		
No	**Topic**	**No. of Lectures (35)**
1	**Module-1 (LINEAR ALGEBRA) TB-1(Ch 2,3,4) (6 hours)**	
1.1	Systems of Linear Equations – Matrices, Solving Systems of Linear Equations.	1 hour
1.2	Vector Spaces, sub space	1 hour
1.3	Linear Independence,	1 hour
1.4	Basis and Rank	1 hour
1.5.	Linear Mappings- Kernel, Range	1 hour
1.6.	Linear Mappings- Rank, Nullity	
2	**Module-2 (LINEAR ALGEBRA) (6 hours)**	
2.1.	Norms, Inner Products, Lengths and Distances, Angles and Orthogonality,	1 hour
2.2	Orthonormal Basis, Orthogonal Complement,	1 hour
2.3	Orthogonal Projections	1 hour
2.4.	Eigenvalues and Eigenvectors	1 hour
2.5.	Eigen decomposition	1 hour
2.6.	Eigen Diagonalization	1 hour
3.	**Module-3 (PROBABILITY AND DISTRIBUTIONS) TB-2(Ch 1,2) (9 hours)**	

3.1	Sample Spaces, Probability Measures, Computing Probabilities	1 hour
3.2	Conditional Probability,	1 hour
3.3	Baye’s Rule	1 hour
3.4	Independence of events	1 hour
3.5	Discrete Random Variables -Bernoulli Random Variables, Binomial Distribution	1 hour
3.6	Discrete Random Variables -Geometric Distribution	1 hour
3.7	Discrete Random Variables -Poisson Distribution	1 hour
3.8	Continuous Random Variables - Exponential Density, Gamma Density,	1 hour
3.9	Continuous Random Variables - Normal Distribution, Beta Density	1 hour
4.	**Module-4 (RANDOM VARIABLES) TB-2 (Ch 3, 4, 5, 6) (9 hours)**	
4.1	Functions of a Random Variable	1 hour
4.2	Joint Distributions - Independent Random Variables	1 hour
4.3	Conditional Distributions	1 hour
4.4	Functions of Jointly Distributed Random Variables	1 hour
4.5	Expected Value of a Random Variable,	1 hour
4.6	Expectations of Functions of Random Variables,	1 hour
4.7	Expectations of Linear Combinations of Random Variables	1 hour
4.6	Variance and Standard Deviation	1 hour
4.9	Covariance and Correlation	1 hour

5	**Module-5 (LIMIT THEOREMS) (6 hours)**	
5.1	Conditional Expectation,	1 hour
5.2	Moment-Generating Function	1 hour
5.3	Limit Theorems(Proof not expected) - Law of Large Numbers,	1 hour
5.4	Convergence in Distribution and the Central Limit Theorem.	1 hour
5.5	Distributions derived from the Normal Distribution - Chi-square and, and F Distributions,	1 hour
5.6	Distributions derived from the Normal Distribution - Sample Mean and the Sample Variance.	1 hour

CST 322	DATA ANALYTICS	Category	L	T	P	Credits	Year of Introduction
		PEC	2	1	0	3	2019

Preamble:

This course helps the learner to understand the basic concepts of data analytics. This course covers mathematics for data analytics, predictive and descriptive analytics of data, Big data and its applications, techniques for managing big data and data analysis & visualization using R programming tool. It enables the learners to perform data analysis on a real world scenario using appropriate tools.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Illustrate the mathematical concepts for data analytics **(Cognitive Knowledge Level: Apply)**
CO2	Explain the basic concepts of data analytics **(Cognitive Knowledge Level: Understand)**
CO3	Illustrate various predictive and descriptive analytics algorithms (**Cognitive Knowledge Level: Apply)**
CO4	Describe the key concepts and applications of Big Data Analytics **(Cognitive Knowledge Level: Understand)**
CO5	Demonstrate the usage of Map Reduce paradigm for Big Data Analytics **(Cognitive Knowledge Level: Apply)**
CO6	Use R programming tool to perform data analysis and visualization **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓	✓							✓
CO6	✓	✓	✓	✓	✓							✓

Abstract POs Defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of Series Tests 1& 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which students should answer any one. Each question can have a maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Mathematics for Data Analytics)

Descriptive statistics - Measures of central tendency and dispersion, Association of two variables - Discrete variables, Ordinal and Continuous variable, Probability calculus - probability distributions, Inductive statistics - Point estimation, Interval estimation, Hypothesis Testing - Basic definitions, t-test

Module - 2 (Introduction to Data Analytics)

Introduction to Data Analysis - Analytics, Analytics Process Model, Analytical Model Requirements. Data Analytics Life Cycle overview. Basics of data collection, sampling, preprocessing and dimensionality reduction

Module - 3 (Predictive and Descriptive Analytics)

Supervised Learning - Classification, Naive Bayes, KNN, Linear Regression. Unsupervised Learning - Clustering, Hierarchical algorithms – Agglomerative algorithm, Partitional algorithms - K- Means. Association Rule Mining - Apriori algorithm

Module - 4 (Big Data Analytics)

Big Data Overview – State of the practice in analytics, Example Applications - Credit Risk Modeling, Business Process Analytics.Big Data Analytics using Map Reduce and Apache Hadoop, Developing and Executing a HadoopMapReduce Program.

Module - 5 (R programming for Data Analysis)

Overview of modern data analytic tools.Data Analysis Using R - Introduction to R - R Graphical User Interfaces, Data Import and Export, Attribute and Data Types, Descriptive Statistics, Exploratory Data Analysis - Visualization Before Analysis, Dirty Data, Visualizing a Single Variable, Examining Multiple Variables, Data Exploration Versus Presentation, Statistical Methods for Evaluation

Text Book

1. Bart Baesens," Analytics in a Big Data World: The Essential Guide to Data Science and its Business Intelligence and Analytic Trends", John Wiley & Sons, 2013.
2. David Dietrich, "EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", John Wiley & Sons, 2015.
3. Jaiwei Han, MichelineKamber, "Data Mining Concepts and Techniques", Elsevier, 2006.
4. Christian Heumann and Michael Schomaker, "Introduction to Statistics and DataAnalysis", Springer, 2016

References

1. Margaret H. Dunham, Data Mining: Introductory and Advanced Topics. Pearson, 2012.
2. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the measures of central tendency.
2. Drive the mean and variance of normal distribution.
3. Collect sample data associated with a real world scenario, and identify central tendency and dispersion measures. Explain your inferences.

Course Outcome 2 (CO2):

1. Explain the life cycle of Data Analytics.
2. Discuss in detail the relevance of data sampling.

Course Outcome 3 (CO3):

1. The following table shows the midterm and final exam marks obtained for students in a database course.

X (Midterm exam)	Y (Final exam)
72	84
50	63

81	77
74	78
94	90
86	75
59	49
83	79
65	77
33	52
88	74
81	90

a) Use the method of least squares to find an equation for the prediction of a student's final exam marks based on the student's midterm grade in the course.

b) Predict the final exam marks of a student who received an 86 on the midterm exam.

2. Perform knn classification on the following dataset and predict the class for the data point X (P1 = 3, P2 =7), assuming the value of k as 3.

P1	**P2**	**Class**
7	7	False
7	4	False
3	4	True
1	4	True

Course Outcome 4 (CO4):

1. List down the characteristics of Big Data.
2. Illustrate process discovery task in business analytics using the scenario of insurance claim handling process. Draw the annotated process map.

Course Outcome 5 (CO5):

1. Explain how fault tolerance is achieved in HDFS.
2. Write down the pseudocode for Map and Reduce functions to solve any one data analytic problem.

Course Outcome 6 (CO6):

1. Illustrate any three R functions used in data analytics.
2. Explain the different categories of attributes and data types in R.

Model Question Paper

QP CODE:

Reg No:______________

Name :______________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 322

Course Name: Data Analytics

Max.Marks :100 **Duration: 3 Hrs**

PART A

(Answer all Questions. Each question carries 3 Marks)

1. Outline the errors that arise in hypothesis testing.

2. The number of members of a millionaires' club were as follows:

Year	2011	2012	2013	2014	2015	2016
Members	23	24	27	25	30	28

(a)What is the average growth rate of the membership?
(b)Based on the results of (a), how many members would one expect in 2018?

3. List and explain any two methods for dealing with missing values in a dataset.

4. Consider the following data (in increasing order) for the attribute age: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70. Sketch an example for stratified sampling using samples of size 5 and the strata “youth,” “middle-aged,” and “senior.”

5. Why is k nearest neighbor classifier called a lazy learner?

6. Find the absolute support, relative support and confidence of the rule (bread => jam) in the following set of transactions

   T1 {bread, butter}, T2{bread, jam, milk}

   T3{Milk, curd}, T4{bread, jam}

7. Explain the 3 Vs of Big Data.

8. Discuss the application of big data analytics in credit risk modeling.

9. Why is Exploratory Data Analysis important in business application ?

10. Explain how box plots be used for data summarization.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Illustrate the Maximum Likelihood Estimation of Bernoulli distribution. **(8)**

(b) A hiking enthusiast has a new app for his smartphone which summarizes his hikes by using a GPS device. Let us look at the distance hiked (in km) and maximum altitude (in m) for the last 10 hikes: **(6)**

Distance	12.5	29.9	14.8	18.7	7.6	16.2	16.5	27.4	12.1	17.5
Altitude	342	1245	502	555	398	670	796	912	238	466

Calculate the arithmetic mean and median for both distance and altitude.

OR

12. (a) Explain the steps in conducting a hypothesis test. **(8)**

(b) A total of 150 customers of a petrol station are asked about their satisfaction with their car and motorbike insurance. The results are summarized below: Determine and interpret Pearson's χ2 statistic and Cramer's V. **(6)**

	Satisfied	Unsatisfied	Total
Car	33	25	58
Car (Diesel engine)	29	31	60
Motor bike	12	20	32
Total	74	76	150

13. (a) Explain the data analytical process model. **(8)**

(b) Discuss the methods for handling noisy data. Consider the following sorted data for price (in dollars) 4, 8, 15, 21, 21, 24, 25, 28, 34. **(6)**
Illustrate smoothing by bin means and bin boundaries

OR

14. (a) a) What is the need for sampling in data analytics? Discuss the different sampling techniques. **(8)**

(b) Use these methods to *normalize* the following group of data: **(6)**

200, 300, 400, 600, 1000

(i) min-max normalization by setting *min* = 0 and *max* = 1

(ii) z-score normalization

(iii) normalization by decimal scaling .

15. (a) A database has five transactions. Let min_sup be 60% and min_conf be 80%.
.

TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

(a) Find all frequent itemsets using Apriori algorithm **(10)**

(b) Generate strong association rules from any one 3 itemset. **(4)**

OR

16. (a) Explain agglomerative hierarchical clustering with an example. **(8)**

(b) Suppose that the data mining task is to cluster points (with (x, y) representing location) into three clusters, where the points areA1(2,10), A2 (2,5), A3 (8,4), B1 (5,8), B2 (7,5), B3 (6,4), C1(1,2), C2 (4,9). The distance function is Euclidean distance. Suppose initially we assign A1, B1, and C1as the center of each cluster, respectively. Use the k-means algorithm to show only **(6)**

(a) The three cluster centers after the first round of execution.

(b) The final three clusters.

17. (a) Illustrate the working of a Map Reduce program with example. **(8)**

(b) Explain the data analytic architecture with a diagram. **(6)**

OR

18. (a) Discuss the architecture of HDFS and its features. **(8)**

(b) Illustrate the use of big data analytics in credit risk modeling. **(6)**

19. (a) List and explain the R functions used in descriptive statistics. **(8)**

(b) Explain hypothesis testing using ANOVA. **(6)**

OR

20. (a) Discuss the data visualization for multiple variables in R **(8)**

(b) Describe the R functions used for cleaning dirty data. **(6)**

(5 x 14 = 70)

Teaching Plan

No	Contents	No of Lecture Hrs
Module – 1(Mathematics for Data Analytics) (7 hrs)		
1.1	Descriptive statistics - Measures of central tendency	1
1.2	Measures of dispersion	1
1.3	Association of two variables - Discrete Variables	1
1.4	Association of two variables - Ordinal and Continuous variable	1
1.5	Probability calculus - Probability distributions	1
1.6	Inductive statistics - Point estimation, Interval estimation	1
1.7	Hypothesis Testing - Basic definitions, t-test	1
Module – 2 (Introduction to Data Analytics) (6 hrs)		
2.1	Introduction to Data Analysis –Analytics, Analytics process model	1
2.3	Analytical model requirements	1
2.4	Data Analytics Life Cycle overview	1
2.5	Basics of data collection	1
2.6	Basics of sampling and preprocessing	1
2.7	Dimensionality reduction	1
Module - 3 (Predictive and Descriptive Analytics) (8 hrs)		
3.1	Supervised Learning, Naive Bayes classification	1
3.2	KNN algorithm	1

3.3	Linear Regression	1
3.4	Unsupervised Learning- Clustering	1
3.5	Hierarchical algorithms Agglomerative algorithm	1
3.6	Partitional algorithms -K- Means	1
3.7	Association Rule Mining	1
3.8	Apriori algorithm	1
Module - 4 (Big Data Analytics) (7 hrs)		
4.1	Big Data Overview – State of the practice in analytics.	1
4.2	Example Applications - Credit Risk Modeling	1
4.3	Business Process Analytics.	1
4.4	Big Data Analytics using Map Reduce and Apache Hadoop	1
4.5	Big Data Analytics using Map Reduce and Apache Hadoop	1
4.6	Developing and Executing a Hadoop MapReduce Program	1
4.7	Developing and Executing a Hadoop MapReduce Program	1
Module - 5 (R programming for Data Analysis) (8 hrs)		
5.1	Overview of modern data analytic tools, Introduction to R, R Graphical User Interfaces	1
5.2	Data Import and Export, Attribute and Data Types	1

5.3	Descriptive Statistics	1
5.4	Exploratory Data Analysis, Visualization Before Analysis	1
5.5	Dirty Data, Visualizing a Single Variable	1
5.6	Examining Multiple Variable	1
5.7	Data Exploration Versus Presentation	1
5.8	Statistical Methods for Evaluation	1

CST 332	FOUNDATIONS OF SECURITY IN COMPUTING	Category	L	T	P	Credit	Year Of Introduction
		PEC	2	1	0	3	2019

Preamble: The purpose of this course is to create awareness among learners about the fundamentals of security and number theory. This course covers Integer & Modular Arithmetic, Primes & Congruences, Discrete Logarithms & Elliptic Curve Arithmetic and an overview of computer security. The concepts covered in this course enable the learners in effective use of cryptographic algorithms and to identify the security threats in computing.

Prerequisite: A sound knowledge in Mathematics, Discrete Computational Structures, Operating Systems and Database Systems.

Course Outcomes: After the completion of the course, the student will be able to

CO1	Illustrate the operations and properties of algebraic structures, integer arithmetic and modular arithmetic. **(Cognitive Knowledge Level: Understand)**
CO2	Use the concepts of prime numbers and factorization for ensuring security in computing systems **(Cognitive Knowledge Level: Apply)**
CO3	Illustrate the concepts of Linear Congruence, Primitive Roots, Discrete Logarithms and Elliptic Curve Arithmetic **(Cognitive Knowledge Level: Apply)**
CO4	Summarize the threats and attacks related to computer and program security **(Cognitive Knowledge Level: Understand)**
CO5	Outline the key aspects of operating system and database security **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓			✓		✓				✓
CO5	✓	✓	✓			✓		✓				✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Test 1 (%)**	**Test 2 (%)**	**End Semester Examination (%)**
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyse			

Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment Test	: **25 marks**
Continuous Assessment Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Modular Arithmetic)

Integer arithmetic - Integer division, Divisibility, Greatest Common Divisor (GCD), Euclid's algorithm for GCD, Extended Euclid’s algorithm, Linear Diophantine Equations. Modular arithmetic - Operations, Properties. Algebraic structures - Groups, Rings, Fields, Finite fields, GF(p), GF (2^n).

Module-2 (Prime Numbers and Factorization)

Prime numbers - Prime numbers and prime-power factorization, Fermat and Mersenne primes, Fermat's theorem, Applications, Euler's theorem, Euler's totient function, Applications. Primality testing – Deterministic algorithms and Probabilistic algorithms. Factorization - Fermat's factorization, Pollard p-1 method.

Module-3 (Linear Congruence, Primitive Roots and Elliptic Curve Arithmetic)

Linear congruence - Simultaneous linear congruence, Chinese Remainder Theorem (CRT). Congruence with a prime - Power modulus, Arithmetic modulo p, Pseudoprimes and Carmichael numbers, Solving congruence modulo prime powers. Primitive roots - Existence of primitive roots for primes, Discrete logarithms. Elliptic curve arithmetic – Prime curves, Binary curves, Addition of two points, Multiplication of a point by a constant.

Module-4 (Computer and Program Security)

Introduction to computer security – Threats, Vulnerabilities, Controls. Browser attack types, Web attacks targeting users, Email attack types. Introduction to program security - Non-malicious programming oversights, Malware.

Module-5 (Operating System and Database Security)

Operating system security – Security in operating system, Security in design of operating system. Database security – Security requirements of databases, Reliability and integrity, Database disclosure.

Text Books

1. Behrouz A Forouzan, Cryptography and Network Security, 3/e, Tata McGraw-Hill.
2. Charles P Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Security in Computing, 5/e, Prentice Hall.
3. G.A. Jones & J.M. Jones, Elementary Number Theory, Springer UTM, 2007

References

1. William Stallings, Cryptography and Network Security Principles and Practices, 4/e, Pearson Ed.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Find the n- bit word that is represented by the polynomial x^2 +1 in GF(2^5).
2. Solve the linear Diophantine equation 21x + 14y=35.

Course Outcome 2 (CO2):

1. Prove that a Carmichael number cannot be the product of two distinct primes.
2. Use the Pollard p-1 method to find a factor of 57247159 with the bound B=8.

Course Outcome 3 (CO3):

1. Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.
2. In the elliptic curve E(1,2) over the field GF(11), find the equation of the curve and all the points on the curve.

Course Outcome 4 (CO4):

1. List three controls that could be applied to detect or prevent off-by-one errors.
2. How does fake email messages act as spam?

Course Outcome 5 (CO5):

1. Discuss the importance of auditability and access control in database security.
2. Explain the various factors which can make data sensitive.

Model Question Paper

QP CODE: **PAGES: ___**

Reg No:_______________
Name:__________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 332
Course Name : FOUNDATIONS OF SECURITY IN COMPUTING

Max Marks: 100 **Duration: 3 Hours**

PART A

(Answer All Questions. Each question carries 3 marks)

1. List the four properties of divisibility with examples.

2. Find gcd (401,700) using Euclid's algorithm.

3. Use Fermat's Little theorem to show that 91 is not a prime.

4. If m is relatively prime to n , show that $\Phi(mn) = \Phi(m)\ \Phi(n)$.

5. Solve the congruence relation 103x ≡ 57 (mod 211).

6. Find a solution for the congruence 3x ≡ 5 mod 7^3

7. What are the problems created by an off-by-one error?

8. How does a clickjacking attack succeed?

9. Explain the significance of correctness and completeness in the design of operating systems.

10. How does the two-phase update technique help the database manager in handling failures?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) For the group G = < Z_6*, x>, prove that it is an Abelian group. Also show the result of 5 x 1 and 1 ÷ 5. **(6)**

(b) Find a particular and the general solution to the following linear Diophantine equations. **(8)**
i) 19 x+13y = 20 ii) 40 x + 16 y =88

OR

12. (a) Describe the properties of modular arithmetic and modulo operator. **(6)**

(b) Using Extended Euclidean algorithm, find the multiplicative inverse of (i) 131 in Z_{180} and (ii) 23 in Z_{100}. **(8)**

13. (a) State and prove Fermat's theorem. **(6)**

(b) Explain Fermat's factorization method and use it to factor 809009. **(8)**

OR

14. (a) Define Euler's totient function. Prove that, Ø(pq)=(p-1)(q-1) where p and q are prime numbers. **(7)**

(b) Define Fermat primes. Show that any two distinct Fermat numbers are relatively prime. **(7)**

15. (a) Using Chinese Remainder Theorem, solve the system of congruence, x ≡2(mod 3), x ≡3(mod 5), x ≡2(mod 7). **(7)**

(b) Define Carmichael number and show that a Carmichael number must be the product of at least three distinct primes. **(7)**

OR

16. (a) For the group G = <Z_{19*}, x>, find the primitive roots in the group. **(6)**

(b) Consider the elliptic curve $y^2 = x^3 + x + 1$ defined over Z_{23}. If P = (3, 10) and Q = (9,7) are two points on the elliptic curve, find 2P and P + Q. **(8)**

17. (a) Distinguish the terms vulnerability, threat and control. **(4)**

(b) With the help of suitable examples, explain the security problems created by incomplete mediation and time-of-check to time-of use. **(10)**

OR

18. (a) Differentiate between man-in-the-browser attack and page-in-the-middle attack. **(4)**

(b) Explain the four aspects of malicious code infection. **(10)**

19. (a) List any six computer security related functions addressed by operating systems. **(6)**

(b) How does a kernelized design support in enforcing security mechanisms? **(8)**

OR

20. (a) Explain any four security requirements of databases. **(4)**

(b) How can database disclosure be prevented? With the help of suitable examples, explain any six types of disclosure. **(10)**

Teaching Plan

No	Contents	No.of Lecture Hrs
Module-1 (Modular Arithmetic) (6 hrs)		
1.1	Integer arithmetic, Integer division, Divisibility, Greatest Common Divisor (GCD)	1
1.2	Euclid's algorithm for GCD, Extended Euclid’s algorithm	1
1.3	Linear Diophantine Equations	1
1.4	Modular arithmetic operations, Properties of modular arithmetic	1
1.5	Groups, Rings and Fields	1
1.6	Finite fields – GF(p), GF(2^n)	1
Module-2 (Prime Numbers and Factorization) (7 hrs)		
2.1	Prime numbers and prime-power factorization	1
2.2	Fermat and Mersenne primes	1
2.3	Fermat’s theorem, Applications – Exponentiation, Multiplicative inverse	1
2.4	Euler’s theorem, Euler’s totient function, Applications	1
2.5	Primality testing – Deterministic algorithms – Divisibility algorithm	1

2.6	Primality testing – Probabilistic algorithms-Fermat test, Square root test, Miller - Rabin test	1
2.7	Factorization - Fermat’s factorization, Pollard p-1 method	1
Module-3 (Linear Congruence, Primitive Roots and Elliptic Curve Arithmetic) (7 hrs)		
3.1	Linear congruence, Simultaneous linear congruence	1
3.2	Chinese Remainder Theorem (CRT)	1
3.3	Congruence with a Prime-Power Modulus, Arithmetic modulo p	1
3.4	Pseudo-primes and Carmichael numbers	1
3.5	Solving congruence modulo prime powers	1
3.6	Primitive roots, Existence of primitive roots for primes, Discrete logarithms	1
3.7	Elliptic curve arithmetic – Prime curves, Binary curves, Addition of two points, Multiplication of a point by a constant	1
Module-4 (Computer and Program Security) (7 hrs) (Text book2: Chapters 1, 3, 4)		
4.1	Threats, Vulnerabilities, Controls	1
4.2	Browser attack types	1
4.3	Web attacks targeting users	1
4.4	Email attack types	1
4.5	Non-malicious programming oversights (Lecture 1)	1
4.6	Non-malicious programming oversights (Lecture 2)	1
4.7	Malware – Four aspects of infection	1
Module-5 (Operating System and Database Security) (8 hrs)(Text book2: Chapters 5, 7)		
5.1	Security in operating system (Lecture 1)	1
5.2	Security in operating system (Lecture 2)	1
5.3	Security in design of operating system (Lecture 1)	1

5.4	Security in design of operating system (Lecture 2)	1
5.5	Security requirements of databases	1
5.6	Reliability & integrity	1
5.7	Database disclosure (Lecture 1)	1
5.8	Database disclosure (Lecture 2)	1

CST 342	AUTOMATED VERIFICATION	Category	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course is intended to impart the basic theory and algorithm for an automatic verification process namely model checking. This course covers finite-state modelling of hardware/software, linear-time properties, classification of linear-time properties, Linear Temporal Logic (LTL) - a formal language for property specification, LTL model checking algorithm and model checking case studies. This course enables the learners to prove correctness of a hardware/software used in safety critical systems in domains such as avionics, health care and automotive.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO1	Illustrate an application for model checking. **(Cognitive Knowledge Level: Understand)**
CO2	Describe finite-state modelling for hardware and software. **(Cognitive Knowledge Level: Understand)**
CO3	Identify linear-time properties required to represent the requirements of a system. **(Cognitive Knowledge Level: Apply)**
CO4	Specify a given linear-time property in Linear Temporal Logic (LTL). **(Cognitive Knowledge Level: Apply)**
CO5	Perform LTL model checking using the tool Symbolic Analysis Laboratory (SAL). **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓	✓							✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks (Out 15, 10 marks shall be given for a model checking project to be implemented in SAL.)

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to Model Checking)

System Verification – Hardware and Software Verification, Model Checking, Characteristics of Model Checking.

Transition Systems – Transition System, Direct Predecessors and Successors, Terminal State, Deterministic Transition System.

Executions - Execution Fragment, Maximal and Initial Execution Fragment, Execution, Reachable States.

Module - 2 (Linear Time Properties)

Linear-Time (LT) Properties - Deadlock. Linear-Time Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Path. Traces - Trace and Trace Fragment, LT Properties - LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties. Safety Properties and Invariants - Invariants, Safety Properties, Trace Equivalence and Safety properties. Liveness Properties - Liveness Property, Safety vs. Liveness Properties. Fairness - Fairness, Unconditional, Weak and Strong Fairness, Fairness Strategies, Fairness and Safety. (Definition and examples only for all topics - no proof required).

Module - 3 (Regular Properties)

Regular Properties - Model Checking Regular Safety properties - Regular Safety property, Verifying Regular Safety Properties. Automata on Infinite Words - ω-Regular Languages and Properties, Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata (Definitions only). Model Checking ω-Regular Properties - Persistence Properties and Product, Nested Depth-First Search (Only algorithms required).

Module - 4 (Linear Time Logic)

Linear Temporal Logic (LTL) - Syntax, Semantics, Equivalence of LTL Formulae, Weak Until, Release and Positive Normal Form, Fairness, Safety and Liveness in LTL (Definitions only). Automata Based LTL Model Checking (Algorithms and examples only).

Module - 5 (Model Checking in SAL)

Introduction **-** Introduction to the tool Symbolic Analysis Laboratory (SAL).

The Language of SAL - The expression language, The transition Language, The module language, SAL Contexts.

SAL Examples - Mutual Exclusion, Peterson's Protocol, Synchronous Bus Arbiter, Bounded Bakery protocol, Bakery Protocol, Traffic Signalling System.

Text Books

1. Christel Baier and Joost-Pieter Katoen, Principles of Model Checking, The MIT Press. (Modules 1 - 4)
2. Leonardo de Moura, Sam Owre and N. Shankar, The SAL Language Manual, SRI International (http://sal.csl.sri.com/doc/language-report.pdf, Chapters 1, 3, 4, 5, 6, 7) (Module 5)

Reference Materials

1. SAL Examples (http://sal.csl.sri.com/examples.shtml) (Module 5)

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Illustrate how model checking can make a system design reliable, based on a required set of properties/constraints.

Course Outcome 2 (CO2):

1. Consider a message delivery system. The sender *s* is trying to send a series of messages to the receiver *r* in such a way that the *(i+1)*st message is sent only after the *i*th message is delivered. There is a possibility of error in sending a message and in that case, *s* keeps on

trying until it is able to send the message. Show a finite state transition system modeling this system.

Course Outcome 3 (CO3):

1. Consider a shared memory segment *s* protected using a mutex lock variable *m*. Two processes p_1 and p_2 are trying to access *s*. List the Linear Time properties of the system which will ensure safety, liveness and fairness.

Course Outcome 4 (CO4):

1. Show the LTL specifications of the safety, liveness and fairness properties listed for the assessment question given in CO3.

Course Outcome 5 (CO5):

1. Model the system mentioned in the question given in CO3 in SAL and verify that the system is correct with respect to the LTL properties shown as the answer for CO4.

Model Question paper

QP CODE: **PAGES: 3**

Reg No:______________ **Name :______________**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST342

Course Name: Automated Verification

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all questions. Each question carries 3 marks.

1. Define model checking. Show the schematic diagram of the model checking approach.

2. Show a transition system modeling a coffee/Tea vending machine.

3. Define invariant as a Linear Time (LT) property. Give an example

4. List any three Linear Time properties in the Mutual Exclusion problem of processes.

5. Illustrate the construction of a product automaton from two automata.

6. Differentiate between Deterministic Buchi Automaton and Non-deterministic Buchi Automaton.

7. Specify the following statements about traffic lights in Linear Temporal Logic (LTL).
   a. Once red, the light can not become green immediately.
   b. Once red, the light always becomes green eventually after being yellow for some time.

8. What is Positive Normal Form (PNF) in LTL? Give an example.

9. List any three applications of the tool Symbolic Analysis Laboratory (SAL).

10. What is a SAL context? Give an example.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain in detail the various phases of the model checking process. **(8)**

(b) Explain the strengths and weaknesses of model checking. **(6)**

OR

12. (a) Define and illustrate the following terms of a transition system. **(14)**
   a. Execution Fragment
   b. Maximal and Initial Execution Fragment
   c. Execution
   d. Reachable States

13. (a) With an example, explain the satisfaction relation for LT properties. (7)

(b) What is trace equivalence in Transition Systems? Give an example to show that if two transition systems satisfy the trace equivalence property, then they satisfy the same set of LT properties. (7)

OR

14. (a) Give the transition system for the fault tolerant variant of the dining philosophers problem. (4)

(b) With a suitable example, explain the algorithms to check whether a Transition System satisfies an invariant or not. (10)

15. (a) Explain Regular Safety Properties with a suitable example. (7)

(b) Illustrate an algorithm for verifying Regular Safety Properties. (7)

OR

16. (a) Explain ω-Regular Properties. (4)

(b) Illustrate how ω-Regular Properties are verified. (10)

17. (a) Explain the syntax of Linear Temporal Logic (LTL). (7)

(b) Explain the semantics of LTL. (7)

OR

18. (a) With an example, give the difference between until and weak until in LTL. (4)

(b) With a suitable example, explain automata based LTL model checking. (10)

19. (a) Explain Peterson's protocol. What are the LTL properties to be verified to ensure its correctness? (8)

(b) Write a SAL script for the verification of Peterson's protocol. (6)

OR

20. (a) Show the SAL model corresponding to Bakery protocol. **(8)**

(b) List any three Linear Time properties of this model and show their LTL **(6)**

Teaching Plan

Module 1 (Introduction to Model Checking)		**4 Hours**
1.1	**System Verification** – Hardware and Software Verification, Model Checking, Model Checking	1 Hour
1.2	**Transition Systems** – Transition System, Direct Predecessors and Successors, Terminal State, Deterministic Transition System	1 Hour
1.3	**Executions** - Execution Fragment, Maximal and Initial Execution Fragment	1 Hour
1.4	Execution, Reachable States	1 Hour
Module 2 (Linear Time Properties)		**8 Hours**
2.1	Linear-Time (LT) Properties - Deadlock	1 Hour
2.2	Linear-Time Behavior - Paths and State Graph, Path Fragment, Maximal and Initial Path Fragment, Path	1 Hour
2.3	Traces - Trace and Trace Fragment	1 Hour
2.4	LT Property, Satisfaction Relation for LT Properties, Trace Equivalence and LT Properties	1 Hour
2.5	Invariants	1 Hour
2.6	Safety Properties, Trace Equivalence and Safety properties	1 Hour
2.7	Liveness Property, Safety vs. Liveness Properties	1 Hour
2.8	Fairness, Unconditional, Weak and Strong Fairness, Fairness Strategies, Fairness and Safety	1 Hour
Module 3 (Regular Properties)		**9 Hours**
3.1	**Regular Properties** - Model Checking Regular Safety properties - Regular Safety property	1 Hour
3.2	Verifying Regular Safety Properties	1 Hour
3.3	Automata on Infinite Words - ω -Regular Languages and Properties	2 Hour

3.4	Nondeterministic Buchi Automata (NBA), Deterministic Buchi Automata (DBA), Generalised Buchi Automata	1 Hour
3.5	Model Checking ω-Regular Properties - Persistence Properties and Product - Lecture 1	1 Hour
3.6	Persistence Properties and Product - Lecture 2	1 Hour
3.7	Nested Depth-First Search (Lecture 1)	1 Hour
3.8	Nested Depth-First Search (Lecture 2)	1 Hour
Module 4 (Linear Time Logic)		**7 Hours**
4.1	**Linear Temporal Logic –** Linear Temporal Logic (LTL) - Syntax	1 Hour
4.2	Semantics - Lecture 1	1 Hour
4.3	Equivalence of LTL Formulae, Weak Until	1 Hour
4.4	Release and Positive Normal Form	1 Hour
4.5	Fairness, Safety and Liveness in LTL	1 Hour
4.6	Automata Based LTL Model Checking (Lecture 1)	1 Hour
4.7	Automata Based LTL Model Checking (Lecture 2)	1 Hour
Module 5 (Model Checking in SAL)		**7 Hours**
5.1	Introduction - Introduction to the tool Symbolic Analysis Laboratory (SAL).	1 Hour
5.2	The Language of SAL - The expression language, The transition Language	1 Hour
5.3	The module language, SAL Contexts.	1 Hour
5.4	SAL Examples - Mutual Exclusion	1 Hour
5.5	Peterson’s Protocol, Synchronous Bus Arbiter	1 Hour
5.6	Bounded Bakery protocol, Bakery Protocol	1 Hour
5.7	Traffic Signalling System	1 Hour

CST 362	PROGRAMMING IN PYTHON	Category	L	T	P	Credit	Year of Introduction
		PEC	2	1	0	3	2019

Preamble: The objective of the course is to equip the learners to develop multi-module software solutions for real world computational problems using Python. It encompasses the Python programming environment, syntax, data representations, intermediate level features, GUI programming, Object Oriented Programming and data processing. This course lays the foundation to develop modular software solutions including complex interactive applications, network applications, and data-driven intelligent applications.

Prerequisite: Basic knowledge in Computational Problem Solving, A course in any programming language.

Course Outcomes: After the completion of the course the student will be able to

CO1	Write, test and debug Python programs **(Cognitive Knowledge level: Apply)**
CO2	Illustrate uses of conditional (if, if-else and if-elif-else) and iterative (while and for) statements in Python programs. **(Cognitive Knowledge level: Apply)**
CO3	Develop programs by utilizing the Python programming constructs such as Lists, Tuples, Sets and Dictionaries. **(Cognitive Knowledge level: Apply)**
CO4	Develop graphical user interface for solutions using Python libraries. (**Cognitive Knowledge level: Apply)**
CO5	Implement Object Oriented programs with exception handling. **(Cognitive Knowledge level: Apply)**
CO6	Write programs in Python to process data stored in files by utilizing Numpy, Matplotlib, and Pandas. **(Cognitive Knowledge level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✔	✔	✔		✔							✔
CO2	✔	✔	✔									✔
CO3	✔	✔	✔	✔	✔							✔
CO4	✔	✔	✔	✔	✔							✔
CO5	✔	✔	✔	✔	✔							✔
CO6	✔	✔	✔	✔	✔	✔						✔

Abstract POs defined by National Board of Accreditation

#PO	Broad PO	#PO	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (*Marks in percentage*)	Test 2 (*Marks in percentage*)	End Semester Examination Marks
Remember	20	20	20

Understand	35	35	35
Apply	45	45	45
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 sub-divisions and carries 14 marks.

SYLLABUS

Module -1 (Programming Environment and Python Basics) (6 hours)

Getting started with Python programming – Interactive shell, IDLE, iPython Notebooks, Detecting and correcting syntax errors, How Python works. The software development process – A case study. Basic coding skills – strings, assignment, and comments, Numeric data types and character sets, Expressions, Using inbuilt functions and modules. Control statements – Iteration with for/while loop, Formatting text for output, A case study, Selection structure (if-else, switch-case), Conditional iteration with while, A case study, Testing control statements, Lazy evaluation.

Module -2 (Building Python Programs) (8 hours)

Strings and text files – Accessing characters, substrings, Data encryption, Strings and number system, String methods, Text files, A case study on text analysis. Design with Functions – Functions as Abstraction Mechanisms, Problem solving with top-down design, Design with recursive functions, Managing a program's namespace, Higher-Order Functions. Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension. Work with tuples. Sets. Work with dates and times, A case study with lists. Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup. Case Study – Data Structure Selection.

Module -3 (Graphics) (7 hours)

Graphics – Terminal-based programs, Simple Graphics using Turtle, Operations, 2D Shapes, Colors and RGB Systems, A case study. Image Processing – Basic image processing with inbuilt functions. Graphical User Interfaces – Event-driven programming, Coding simple GUI-based programs : Windows, Labels, Displaying images, Input text entry, Popup dialog boxes, Command buttons, A case study.

Module -4 (Object Oriented Programming) (7 hours)

Design with classes - Objects and Classes, Methods, Instance variables, Constructor, Accessor and Mutator, Data-Modeling Examples, Structuring classes with inheritance and polymorphism. Abstract classes, Interfaces, Exceptions - Handle a single exception, handle multiple exceptions.

Module -5 (Data Processing) (9 hours)

The os and sys modules, NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Random numbers. Plotting and visualization. Matplotlib - Basic plot, Ticks, Labels, and Legends. Working with CSV files. – Pandas - Reading, Manipulating, and Processing Data. Introduction to Micro services using Flask.

Text Books:

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017
3. Flask: Building Python web services, Jack Stouffer, Shalabh Aggarwal, Gareth Dwyer, PACKT Publishing Limited, 2018

Reference Books:

1. Zed A Shaw, Learn Python 3 The Hard Way, Addison-Wesley, 2017
2. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
3. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016
4. Charles Severance. Python for Informatics: Exploring Information,

Sample Course Level Assessment Questions

Course Outcome1(CO1):

1. What is type conversion? How is it done in Python?
2. Write a note on the Python editors.

Course Outcome 2(CO2):

1. Write a Python program which takes a positive integer **n** as input and finds the sum of cubes all positive even numbers less than or equal to the number.
2. What is printed when the below code is executed?

```
mysum = 0
   for i in range(5, 11, 2):
               mysum += i
               if mysum == 5:
                           break
                     mysum += 1
   print(mysum)
```

What would be the output if 'break' is replaced with 'continue' in the above code fragment?

Course Outcome 3(CO3):

1. Given is a list of of words, *wordlist*, and a string, *name.* Write a Python function which takes *wordlist* and *name* as input and returns a tuple. The first element of the output tuple is the number of words in the *wordlist* which have *name* as a substring in it. The second element of

the tuple is a list showing the index at which the *name* occurs in each of the words of the *wordlist* and a 0 if it doesn't occur.

2. What is the value of L after you run the code below?

```
L = ["life", "answer", 42, 0]
for thing in L:
      if thing == 0:
        L[thing] = "universe"
      elif thing == 42:
          L[1] = "everything"
```

Course Outcome 4(CO4):

1. A bouncy program is defined as follows – The program computes and displays the total distance traveled by a ball, given three inputs—the initial height from which it is dropped, its bounciness index, and the number of bounces. Given the inputs write a GUI-based program to compute the total distance traveled.
2. Write a Python program to find the quadrant of a point, say (x,y).

Course Outcome 5(CO5):

1. Write a Python program to implement the addition, subtraction, and multiplication of complex numbers using classes. Use constructors to create objects. The input to the program consist of real and imaginary parts of the complex numbers.
2. Explain inheritance in Python using suitable examples.

Course Outcome 6(CO6):

1. Given a file "auto.csv" of automobile data with the fields *index, company, body-style, wheel-base, length, engine-type, num-of-cylinders, horsepower, average-mileage,* and *price*, write python code to
    1. Clean and Update the CSV file
    2. Print total cars of all companies
    3. Find the average mileage of all companies
    4. Find the highest priced car of all companies.
2. Given two matrices A and B, write a program to find the product of A and B^T.

Model Question Paper

QP CODE: PAGES:

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 362

Course name : PROGRAMMING IN PYTHON

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Write a Python program to reverse a number and also find the sum of digits of the number. Prompt the user for input.
2. Explain the concept of scope and lifetime of variables in Python programming language, with a suitable example.
3. Illustrate format specifiers and escape sequences with examples.
4. Compare tuples, lists, and dictionaries with examples.
5. Describe the following dictionary methods with an example.
   i. *get( )* ii. *Keys( )* iii. *pop( )* iv. *update( )* v. *values( )* vi. *items( )*
6. Differentiate the terminal-based and GUI-based programming in Python.
7. What is polymorphism? Give an example in the context of OOP in Python.
8. How is exception handling accomplished in Python programs?
9. Explain the **os** and **os.path** modules in Python with examples. Also, discuss the *walk( )* and *getcwd( )* methods of the **os** module.
10. What are the important characteristics of CSV file format.

PART-B

(Answer any one full question from each module)

11. (a) Write a Python code to check whether a given year is a leap year or not [An year is a leap year if it's divisible by 4 but not divisible by 100 except for those divisible by 400]**.** (6)

(b) What are the possible errors in a Python program. Write a Python program to print the value of $\boldsymbol{2^{2^n}+n+5}$ for ***n*** provided by the user. (8)

OR

12. (a) Write a Python program to find the value for *sin(x)* up to *n* terms using the series (6)

$$\mathbf{sin}(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\mathbf{1!}} - \frac{\boldsymbol{x^3}}{\mathbf{3!}} + \frac{\boldsymbol{x^5}}{\mathbf{5!}} - \frac{\boldsymbol{x^7}}{\mathbf{7!}} + \cdots \quad \text{where } \boldsymbol{x} \text{ is in degrees}$$

(b) Write a Python code to determine whether the given string is a Palindrome or not using slicing. Do not use any string function. (8)

13. (a) Write a Python code to create a function called *list_of_frequency* that takes a string and prints the letters in non-increasing order of the frequency of their occurrences. Use dictionaries. (5)

(b) Write a Python program to read a list of numbers and sort the list in a non-decreasing order without using any built in functions. Separate function should be written to sort the list wherein the name of the list is passed as the parameter. (9)

OR

14. (a) Illustrate the following Set methods with an example. (6)

i. *intersection( )* ii. *Union( )* iii. *Issubset( )* iv. *Difference( )* v. *update( )* vi. *discard( )*

(b) Write a Python program to check the validity of a password given by the user. (8)

The Password should satisfy the following criteria:

1. Contains at least one letter between **a** and **z**
2. Contains at least one number between **0** and **9**
3. Contains at least one letter between **A** and **Z**
4. Contains at least one special character from **$, #, @**
5. Minimum length of password: **6**

15. (a) Write a program to draw a hexagon using turtle. (5)

(b) Write a note on the image processing function in Python. (9)

OR

16. (a) Describe the features of event driven programming. (4)

(b) Write a GUI-based program that allows the user to convert temperature values between degrees Fahrenheit and degrees Celsius. The interface should have labeled entry fields for these two values. These components should be arranged in a grid where the labels occupy the first row and the corresponding fields occupy the second row. At start-up, the Fahrenheit field should contain 32.0, and the Celsius field should contain 0.0. The third row in the window contains two command buttons, labeled >>>> and <<<<. When the user presses the first button, the program should use the data in the Fahrenheit field to compute the Celsius value, which should then be output to the Celsius field. The second button should perform the inverse function. (10)

17. (a) How can a class be instantiated in Python? Write a Python program to express the instances as return values to define a class RECTANGLE with parameters *height, width, corner_x,* and *corner_y* and member functions to find center, area, and perimeter of an instance. (10)

(b) Explain inheritance in Python. Give examples for each type of inheritance. (4)

OR

18. (a) Write a Python class named ***Circle*** constructed by a radius and two methods which will compute the area and the perimeter of a given circle (6)

(b) Write Python program to create a class called as **Complex** and implement *__add__( )* method to add two complex numbers. Display the result by overloading the + Operator. (8)

19. (a) Write a Python program to add two matrices and also find the transpose of the resultant matrix. (8)

(b) Given a file "auto.csv" of automobile data with the fields *index, company, body-style, wheel-base, length, engine-type, num-of-cylinders, horsepower, average-mileage,* and *price***,** write Python codes using Pandas to (6)

1) Clean and Update the CSV file
2) Print total cars of all companies

3) Find the average mileage of all companies
4) Find the highest priced car of all companies.

OR

20. (a) Write Python program to write the data given below to a CSV file. (5)

SN	Name	Country	Contribution	Year
1	Linus Torvalds	Finland	Linux Kernel	1991
2	Tim Berners-Lee	England	World Wide Web	1990
3	Guido van Rossum	Netherlands	Python	1991

(b) Given the sales information of a company as CSV file with the following fields *month_number, facecream, facewash, toothpaste, bathingsoap, shampoo, moisturizer, total_units, total_profit*. Write Python codes to visualize the data as follows (9)

1) Toothpaste sales data of each month and show it using a scatter plot
2) Face cream and face wash product sales data and show it using the bar chart

Calculate total sale data for last year for each product and show it using a Pie chart.

(14X5=70)

Teaching Plan

Module 1: Programming Environment and Python Basics		**(6 hours)**
1.1	Getting started with Python programming – Interactive shell, IDLE, iPython Notebooks, Detecting and correcting syntax errors, How Python works.	**1 hour**
1.2	The software development process – A case study.	**1 hour**
1.3	Basic coding skills – strings, assignment, and comments, Numeric data types and character sets	**1 hour**
1.4	Expressions, Using inbuilt functions and modules.	**1 hour**
1.5	Control statements – Definite Iteration with for loop, Formatting text for output, Selection structure (if-else, switch-case), Conditional iteration with	**1 hour**

	while loop, A case study	
1.6	Testing the control statements, Lazy evaluation.	**1 hour**
Module 2: Building Python Programs		**(8 hours)**
2.1	Strings – Accessing characters, substrings, Data encryption, Strings and number system, String methods,	**1 hour**
2.2	Text files, A case study on text analysis.	**1 hour**
2.3	Design with Functions – Functions as Abstraction Mechanisms, Problem solving with top-down design,	**1 hour**
2.4	Design with recursive functions, Managing a program's namespace, Higher-Order Functions.	**1 hour**
2.5	Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension.	**1 hour**
2.6	Work with tuples. Sets. Work with dates and times, A case study with lists.	**1 hour**
2.7	Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup.	**1 hour**
2.8	Case Study - Data Structure Selection.	**1 hour**
Module 3: Graphics		**(7 hours)**
3.1	Graphics – Simple Graphics using Turtle, Operations, 2D Shapes,	**1 hour**
3.2	Colors and RGB Systems, A case study.	**1 hour**
3.3	Image Processing – Basic image processing with inbuilt functions.	**1 hour**
3.4	Graphical User Interfaces – Event-driven programming	**1 hour**
3.5	Coding simple GUI-based programs : Windows, Labels, Displaying images,	**1 hour**
3.6	Coding simple GUI-based programs : Input text entry, Popup dialog boxes, Command buttons	**1 hour**
3.7	A case study - GUI	**1 hour**

Module 4: Object Oriented Programming		**(7 hours)**
4.1	Design with classes : Objects and Classes, Methods, Instance Variables	**1 hour**
4.2	Constructor, Accessors, and Mutators	**1 hour**
4.3	Structuring classes with Inheritance	**1 hour**
4.4	Polymorphism	**1 hour**
4.5	Abstract Classes	**1 hour**
4.6	Interfaces	**1 hour**
4.7	Exceptions : Handle a single exception, handle multiple exceptions	**1 hour**
Module 5: Data Processing		**(9 hours)**
5.1	The *os* and *sys* modules, NumPy : Basics, Creating arrays, Arithmetic, Slicing	**1 hour**
5.2	Matrix Operations, Random numbers.	**1 hour**
5.3	Matplotlib : Basic plot, Ticks, Labels, and Legends	**1 hour**
5.4	Working with CSV files	**1 hour**
5.5	Pandas : Reading, Manipulating	**1 hour**
5.6	Pandas : Processing Data and Visualize.	**1 hour**
5.7	Introduction to Microservices using Flask	**1 hour**
5.8	Introduction to Microservices using Flask	**1 hour**
5.9	Introduction to Microservices using Flask	**1 hour**

CST 372	DATA AND COMPUTER COMMUNICATION	Category	L	T	P	Credits	Year of Introduction
		PEC	2	1	0	3	2019

Preamble:

The purpose of this course is to prepare learners to understand the communication entities and the associated issues in data transmission. This course covers fundamental concepts of data transmission in digital and analog form, transmission media, concepts of encoding, multiplexing, spread spectrum and switching methods. This course helps the learner to gain insight into the important aspects of data communication and computer networking systems and enables to apply in practical applications.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Identify the characteristics of signals for analog and digital transmissions (**Cognitive knowledge: Apply**)
CO2	Identify the issues in data transmission (**Cognitive knowledge: Apply**)
CO3	Select transmission media based on characteristics and propagation modes (**Cognitive knowledge: Apply**)
CO4	Choose appropriate signal encoding techniques for a given scenario (**Cognitive knowledge: Apply**)
CO5	Illustrate multiplexing and spread spectrum technologies (**Cognitive knowledge: Apply**)
CO6	Use error detection, correction and switching techniques in data communication (**Cognitive knowledge: Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓		✓									✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓
CO6	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Test : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Data Transmission Basics)

Communication model - Simplex, Half duplex, Full duplex transmission. Periodic analog signals - Sine wave, Amplitude, Phase, Wavelength, Time and frequency domain, Bandwidth. Analog & digital data and signals. Transmission impairments - Attenuation, Delay distortion, Noise. Data rate limits - Noiseless channel, Nyquist bandwidth, Noisy channel, Shannon's capacity formula.

Module-2 (Transmission Media)

Guided transmission media - Twisted pair, Coaxial cable, Optical fiber. Unguided media - Radio waves, Terrestrial microwave, Satellite microwave, Infrared. Wireless propagation - Ground wave propagation, Sky wave propagation, Line-of-Sight (LoS) propagation.

Module-3 (Digital Transmission and Analog Transmission)

Digital data to digital signal – Non-Return-to-Zero (NRZ), Return-to-Zero (RZ), Multilevel binary, Biphase. Analog data to digital signal - Sampling theorem, Pulse Code Modulation (PCM), Delta Modulation (DM). Digital data to analog signal - Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), Phase Shift Keying (PSK). Analog data to analog signal - Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM).

Module-4 (Multiplexing and Spread Spectrum)

Multiplexing - Frequency Division Multiplexing (FDM), Wavelength Division Multiplexing (WDM), Time Division Multiplexing (TDM), Characteristics, Synchronous TDM, Statistical TDM. Spread spectrum techniques - Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), Code Division Multiplexing, Code Division Multiple Access (CDMA).

Module-5 (Error Detection, Correction and Switching)

Digital data communication techniques - Asynchronous transmission, Synchronous transmission. Detecting and correcting errors - Types of errors, Parity check, Checksum, Cyclic Redundancy Check (CRC), Forward Error Correction (FEC), Hamming distance, Hamming code. Basic principles of switching - Circuit switching, Packet switching, Message switching.

Text Books

1. Forouzan B. A., Data Communications and Networking, 5/e, McGraw Hill, 2013.

2. William Stallings, Data and Computer Communication 9/e, Pearson Education, Inc.

References

1. Schiller J., Mobile Communications, 2/e, Pearson Education, 2009.
2. Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. If the spectrum of a channel is between 3 MHz and 4 MHz and SNR_{dB} = 24 dB, calculate the Shannon capacity.
2. Assume that a periodic signal is composed of five sine waves with frequencies 200, 400, 600, 800 and 1000 Hz. Determine the bandwidth. Draw the spectrum assuming all components have a maximum amplitude of 5 V.

Course Outcome 2 (CO2):

1. Given a receiver with an effective noise temperature of 294 K and a bandwidth of 10 MHz. Find the thermal noise level at the receiver side in dBW.
2. The loss in a cable is usually defined in decibels per kilometer (dB/km). If the signal at the beginning of a cable with -0.3 db/km has a power of 2 mW, determine the power of the signal at 5 km.

Course Outcome 3 (CO3):

1. Explain the reflective property of a parabolic antenna.
2. Two separate frequencies are used for uplink and downlink transmission in satellite communication. Give reason.

Course Outcome 4 (CO4):

1. Encode the data sequence 101011100 using Multilevel binary and Biphase schemes.
2. Encode the data bits 00101101110001 using 2B1Q encoding scheme. Assume negative original level.

Course Outcome 5 (CO5):

1. The frequency spectrum of input signals will move to high frequency bands by the FDM process. Justify.
2. Four channels are multiplexed using TDM. If each channel sends 100 bytes/sec and we multiplex one byte per channel, determine the frame size, duration of a frame, frame rate and bit rate of link.

Course Outcome 6 (CO6):

1. Using the divisor polynomial $x^4 + x + 1$, determine the Cyclic Redundancy Check (CRC) for the dataword 10110100. Also, perform the checking at the receiver side.
2. How many redundancy bits are required to generate the Hamming code for a 7-bit data? Assuming even parity, generate the Hamming code for the 7-bit dataword 1001101. If the fifth bit from the left of the received codeword is changed to 0, can

this be detected? Give reasons for your answer.

Model Question Paper

QP CODE: **PAGES: 3**

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 372

Course Name : Data and Computer Communication

Max Marks: 100 **Duration: 3 Hours**

PART A
(Answer All Questions. Each question carries 3 marks)

1. Define bandwidth. Find the lowest frequency, if a periodic signal has a bandwidth of 20 Hz and the highest frequency is 60 Hz. Draw the spectrum if the signal contains all frequencies of the same amplitude.

2. Assume that a TV picture is to be transmitted over a channel with 4.5 MHz bandwidth and a 35 dB Signal-to-Noise-Ratio. Find the capacity of the channel.

3. How does twisting affect the performance in a twisted pair cable?

4. Which wireless propagation method is suitable for satellite communication? Justify your answer.

5. Explain the two main distortions that can occur in a delta modulated waveform. How can it be avoided?

6. Illustrate the equivalent square wave pattern of the bit string 01001101 using Non-Return-to-Zero (NRZ) - Level and NRZ-Invert encoding schemes.

7. Apply Direct Sequence Spread Spectrum to the data 101 using the Barker sequence 10110111000. Show the encoding and decoding steps.

8. Compare synchronous and statistical time division multiplexing.

9. Find the minimum hamming distance for the following cases:

   a) Detection of two errors

b) Correction of two errors

c) Detection of three errors

10. Find the parity bit for simple even parity check for the following.

a) 1001010

b) 0001100

c) 1000000

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) With the help of suitable figures, distinguish between time and frequency domain representations. **(4)**

(b) Describe the different types of transmission impairments. **(10)**

OR

12. (a) Calculate the bandwidth, if a periodic signal is decomposed into 4 sine waves with frequencies 50 Hz, 100 Hz, 150 Hz and 200 Hz. Draw the spectrum, assuming all components having an amplitude in the range 6-12 V and all are multiples of two in the increasing order. **(6)**

(b) Distinguish between Nyquist bandwidth and Shannon capacity. Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with (i) Two signal levels (ii) Four signal levels. Determine the maximum bit rate in both cases. **(8)**

13. (a) For a parabolic reflective antenna operating at 12 GHz with a diameter of 2 m, calculate the effective area and the antenna gain. **(6)**

(b) List any four advantages and disadvantages of twisted pair, coaxial cable and fiber optic cable. **(8)**

OR

14. (a) Compare the features of terrestrial microwave and satellite microwave. **(6)**

(b) With the help of suitable diagrams, differentiate Multi-mode and Single-mode optical fibres. How are the rays propagated in Step-index and Graded-index Multi-mode fibres? **(8)**

15. (a) Distinguish between data rate and signal rate. **(4)**

(b) How is polar encoding done? Encode the pattern 010011001110 using the two Biphase schemes. **(10)**

OR

16. (a) Show the equivalent analog sine wave pattern of the bit string 010011010 using Amplitude Shift Keying, Frequency Shift Keying and Phase Shift Keying. **(4)**

(b) State Sampling theorem. Explain Pulse Code Modulation with suitable figures. **(10)**

17. (a) Four channels are multiplexed using Time Division Multiplexing. If each channel sends 100 bytes/sec and we multiplex one byte per channel, determine the frame size, duration of a frame, frame rate and bit rate of the link. **(6)**

(b) Explain the working of Frequency Hopping Spread Spectrum with an example. **(8)**

OR

18. (a) Explain any three techniques by which the disparity in input data rate is handled by Time Division Multiplexing. Give examples. **(4)**

(b) Suppose Alice and Bob are communicating using Code Division Multiple Access. Alice uses the code [+1 +1] and Bob uses the code [+1 -1]. Alice sends a data bit 0 and Bob sends a data bit 1. Show the data in the channel and how they can detect what the other person has sent. **(10)**

19. (a) Explain two-dimensional parity check with examples. **(4)**

(b) Describe the need for a switch in a communication system. What are the different phases in circuit switching? **(10)**

OR

20. (a) Explain the virtual circuit approach of packet switching with a suitable example. **(6)**

(b) Find the Hamming code for the data word 1011001. Assume odd parity. **(8)**

Teaching Plan

No	Contents	No.of Lecture Hrs (35 hrs)
Module-1 (Data Transmission Basics) (6 hrs)		
1.1	Introduction, Communication model, Simplex, Half duplex, Full duplex transmission, Periodic analog signals, Sine wave, Amplitude, Phase, Wavelength	1
1.2	Time and frequency domain, Bandwidth	1
1.3	Analog & digital data and signals	1
1.4	Transmission impairments, Attenuation, Delay distortion, Noise	1
1.5	Data rate limits, Noiseless channel, Nyquist bandwidth	1
1.6	Noisy channel, Shannon's capacity formula	1
Module-2 (Transmission Media) (6 hrs)		
2.1	Guided transmission media, Twisted pair, Coaxial cable	1
2.2	Optical fiber	1
2.3	Unguided media, Radio waves	1
2.4	Terrestrial microwave, Satellite microwave	1
2.5	Infrared	1
2.6	Wireless Propagation, Ground wave, Sky wave, Line-of-Sight (LoS) propagation	1
Module-3 (Digital Transmission and Analog Transmission) (8 hrs)		
3.1	Digital data to digital signal, Non-Return-to-Zero (NRZ), Return-to-Zero (RZ)	1
3.2	Multilevel binary and Biphase	1
3.3	Analog data to digital signal, Sampling theorem, Pulse Code Modulation (PCM)	1

3.4	Delta Modulation (DM)	1
3.5	Digital data to analog signal, Amplitude Shift Keying (ASK)	1
3.6	Frequency Shift Keying (FSK), Phase Shift Keying (PSK)	1
3.7	Analog data to analog signal, Amplitude Modulation (AM)	1
3.8	Frequency Modulation (FM), Phase Modulation (PM)	1
Module-4 (Multiplexing and Spread Spectrum) (7 hrs)		
4.1	Multiplexing, Frequency Division Multiplexing (FDM), Wavelength Division Multiplexing (WDM)	1
4.2	Time Division Multiplexing (TDM), Characteristics, Synchronous TDM, Statistical TDM	1
4.3	Spread spectrum techniques, Direct Sequence Spread Spectrum (DSSS)	1
4.4	Frequency Hopping Spread Spectrum (FHSS)	1
4.5	Code Division Multiplexing	1
4.6	Code Division Multiple Access (CDMA) (Lecture 1)	1
4.7	CDMA (Lecture 2)	1
Module-5 (Error Detection, Correction and Switching) (8 hrs)		
5.1	Digital data communication techniques, Asynchronous & Synchronous transmission	1
5.2	Detecting and correcting errors, Types of errors, Parity check, Checksum	1
5.3	Cyclic Redundancy Check (CRC)	1
5.4	Forward Error Correction (FEC), Hamming distance	1
5.5	Hamming code	1
5.6	Basic principles of switching, Circuit switching	1
5.7	Packet switching	1
5.8	Message switching	1

SEMESTER VI

MINOR

CST 382	INTRODUCTION TO SOFTWARE TESTING	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This is a course in theoretical computer science that includes test cases for white-box, black-box, and grey-box approaches. This course describes the various techniques for test case design used to test software artifacts, including requirements, design, and code. The course includes different techniques for test case design based on graphs, programming language syntaxes and inputs. The course also covers symbolic execution using PEX tool.

Course Outcomes: After the completion of the course the student will be able to:-

CO1	List a range of different software testing techniques and be able to apply specific unit testing method to the projects using Junit**. (Cognitive Knowledge Level: Understand)**
CO2	Explain mutation testing method for a given piece of code to identify hidden defects that can't be detected using other testing methods. **(Cognitive Knowledge Level: Understand)**
CO3	Explain graph coverage criteria in terms of control flow graph and data flow graph for a given program. **(Cognitive Knowledge Level: Understand**)
CO4	Demonstrate the importance of black-box approaches in terms of domain and functional testing. **(Cognitive Knowledge Level: Understand)**
CO5	Illustrate the use of PEX tool with symbolic execution. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO 9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓						✓		✓

CO3	✓	✓	✓	✓						✓		✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓						✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test 1 (Marks)	**Test 2 (Marks)**	
Remember	**30**	**30**	**30**
Understand	**40**	**40**	**40**
Apply	**30**	**30**	**30**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment - Test	: **25 marks**
Continuous Assessment - Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to Software Testing)

Some Popular Errors – Ariane 5, Therac 25, Intel Pentium Bug. What is Software testing? Why should it be tested? Software Quality, Role of Testing. Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking. Software Testing Terminologies - Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria. Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing, Functional testing, Stress testing, Performance testing, Usability testing and Regression testing. Testing Methods - Black Box testing, White Box testing, Grey Box testing.

Module - 2 (Unit Testing)

Concept of Unit testing. Static Unit testing. Dynamic Unit testing - Control Flow testing, Data Flow testing, Domain testing, Functional Program testing. Mutation testing - Mutation and Mutants, Mutation operators, Mutation score. Junit - Framework for Unit testing. Case Study - Mutation testing using Junit and Muclipse.

Module - 3 (Unit Testing - White Box Approaches)

Overview of Graph Coverage Criteria. Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage, Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage. Data Flow Criteria - du paths, du pairs. Subsumption Relationships among Graph Coverage Criteria. Graph Coverage for Source Code - Control flow graphs for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program – Statistics. Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph, Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root. Case Study - Graph Based testing using JUnit Framework.

Module - 4 (Unit Testing - Black Box Approaches)

Domain Testing / Input Space Partitioning - Partitions of a set. Input domain modelling - Interface-based approach, Functionality-based approach. Identifying values. Multiple partitions of the inputdomain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage. TriTyp example. Functional Testing - Functional Testing Concepts of Howden. Functional testing - Important Steps. Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing. Case Study - Black Box testing approaches using JUnit.

Module - 5 (Grey Box Testing Approaches)

Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages. Techniques of Grey Box Testing - Matrix Testing, Regression Testing, Orthogonal Array Testing or OAT, Pattern Testing. An Introduction to PEX - Parameterized Unit Testing, The Testing Problem. Symbolic Execution – Example, Symbolic execution tree. PEX application Case Study – PEX.

Text Books

1. Paul Ammann and JeffOffutt ,Introduction to Software Testing.
2. KshirasagarNaik and PriyadarshiTripathy, Software Testing And Quality Assurance: Theory And Practice.

Reference Materials

1. https://www.csc.ncsu.edu/academics/undergrad/honors/thesis/muclipsebinder.pdf - Muclipse tutorial.
2. King, James C, "Symbolic Execution and Program Testing", Association for Computing Machinery, July 1976.
3.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Explain the following types of testing methods with examples.

(i) Balck-box testing.

(ii) White-box testing.

(iii) Grey-box testing.

Course Outcome 2 (CO2): Define 12 mutants for the following method *power*() using effective mutation operators. Try to use each mutation operator at least once. Approximately, how many mutants do you think there would be, if all mutants for *power*() were created?

```
public static int power (int left, int right)
{
//************************************
// Raises Left to the power of Right
// precondition : Right >= 0
// postcondition: Returns Left**Right
//************************************
          intrslt;
         rslt = Left;
```

```
        if (Right == 0)
        {
                rslt = 1;
        }
        else
        {
                for (int i = 2; i <= Right; i++)
                rslt = rslt * Left;
        }
        return (rslt);
        }
```

Course Outcome 3 (CO3): Draw the control flow graph and data flow graph of given piece of code.

```
public static double ReturnAverage(int value[],int AS, int MIN, int MAX){
/*
Function: ReturnAverageComputes the averageof all those numbers in the input array in
the positive range [MIN, MAX]. The maximumsize of the array is AS. But, the array size
could be smaller than AS in which case the endof input is represented by -999.
*/
int i, ti, tv, sum;
doubleav;
i = 0; ti = 0; tv = 0; sum = 0;
while (ti< AS && value[i] != -999) {
ti++;
if (value[i] >= MIN && value[i] <= MAX) {
tv++;
sum = sum + value[i];
}
i++;
}
if (tv> 0)
av = (double)sum/tv;
else
av = (double) -999;
return (av);
}
```

Course Outcome 4 (CO4): Explain the following with examples.

1. Input domain modelling.
2. All Combinations Coverage (ACoC)
3. Each Choice Coverage (ECC)
4. Pair-wise Coverage
5. T-wise Coverage
6. Base Choice Coverage
7. Multiple Base Choices Coverage.

Course Outcome 5 (CO5): Draw the symbolic execution tree for the following program code and explain the symbolic execution of testme (α1, α2).

```
int twice (int v) {
 return 2 ∗ v;
 }
 void testme (int x, int y ) {
 z = twice ( y);
 if ( z == x ){
 if ( x > y + 10)
ERROR;
 }
}
int main() {
 x = sym input();
 y = sym input();
testme ( x , y);
 return(0);
}
```

Model Question Paper

QP CODE: **PAGES: 4**

Reg No:______________ **Name :______________**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 382

Course Name: Introduction to Software Testing

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Explain the differences between Validation and Verification.

2. Explain the differences between Fault, Error, and Bug?

3. Define Ground string, Mutation score, and Mutants.

4. What are the functions of Test driver and Test stubs in dynamic unit testing?

5. Define Node coverage, Edge coverage and Prime path coverage in a control flow graph.

6. What are du paths and du pairs in a data flow graph?

7. Explain the two approaches in input domain modelling.

8. Explain the difference between Equivalence Class Partitioning and Boundary Value Analysis.

9. Briefly explain three techniques of Grey box testing.

10. Explain the concept of symbolic execution with the help of a toy example.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the following types of testing
(i) Black Box testing (ii) White Box testing (iii) Grey Box testing **(14)**

(iv) Unit testing (v) Integration testing (vi) System testing (vii) Acceptance testing

OR

12. (a) Explain the following coverage criterias based on the code fragment given below. (i) Functional coverage (ii) Statement coverage (iii)Conditional coverage (iv)Branch coverage **(8)**

```
int foo (int x, int y){
    int z = 0;
    if ((x > 0) && (y > 0)){
    z = x;}
    return z;
    }
```

(b) Write positive and negative test cases for an ATM Machine? **(6)**

13. (a) Explain Dynamic unit test environment with a neat figure. **(8)**

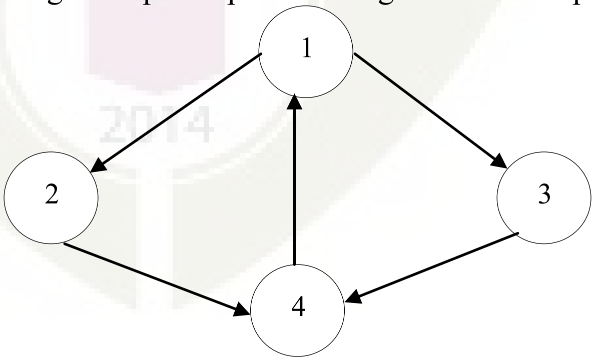
(b) Explain the major difference between control flow testing and data flow testing**.** **(6)**

OR

14. Explain seven types of mutation operators with neat examples. **(14)**

15. (a) Explain touring, side trips and detours with a neat example. **(7)**

(b) Explain simple path coverage and prime path coverage with the help of CFG given below. **(7)**



OR

16. (a) Draw CFG fragment for

(i) Simple *if* (ii) Simple *while* loop (iii) Simple *for* loop **(7)**

(b) Explain the following concepts with examples. **(7)**

(i)Call graph (ii) Inheritance graph (iii) Coupling du-pairs

17. (a) What are the four important steps in functional testing? **(7)**

(b) Briefly explain input domain modelling approaches. **(7)**

OR

18. (a) Consider the triangle classification program with a specification: **(6)**

The program reads floating values from the standard input. The three values *A*, *B*, and *C* are interpreted as representing the lengths of the sides of triangle. The program then prints a message to the standard output that states whether the triangle, if it can be formed, is scalene, isosceles, equilateral, or right angled. Determine the following for the above program:

(i) For the boundary condition $A + B > C$ case (scalene triangle), identify test cases to verify the boundary.

(ii) For the boundary condition $A = C$ case (isosceles triangle), identify test cases to verify the boundary.

(iii) For the boundary condition $A = B = C$ case (equilateral triangle), identify test cases to verify the boundary.

(b) Develop a decision table to generate test cases for this specification. **(8)**

19. (a) Explain the importance of grey box testing, its advantages and disadvantages. **(9)**

(b) Explain the concept of symbolic execution tree. **(5)**

OR

20. (a) Consider the code fragment given below: - **(7)**

```
1. POWER: PROCEDURE(X, Y);
2. Z ← 1;
3. J ← 1;
4. LAB:   IF Y ≥ J THEN
```

```
5.  DO; Z← Z * X;
6.  J ← J + 1;
7.  GO TO LAB;  END;
8.  RETURN (Z) ;
9.  END;
```

a) Explain Symbolic execution of POWER (αl, α2).

(b) Explain Execution tree for POWER (αl, α2) in the above code fragment. **(7)**

TEACHING PLAN

Index	Topics	No. of Hours (45)
	Module 1 (Introduction to Software Testing) 9 Hours	
1.1	Some Popular Errors– Ariane 5, Therac 25, Intel Pentium Bug.	1 Hour
1.2	What is Software testing? Why should it be tested? Software Quality, Role of Testing.	1 Hour
1.3	Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking.	1 Hour
1.4	Software Testing Terminologies- Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria.	1 Hour
1.5	Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing	1 Hour
1.6	Functional testing, Stress testing	1 Hour
1.7	Performance testing, Usability testing and Regression testing.	1 Hour
1.8	Testing Methods - Black Box testing	1 Hour
1.9	Grey Box testing.	1 Hour
Module 2 (Unit testing) 8 Hours		

2.1	Concept of Unit testing.	1 Hour
2.2	Static Unit testing.	1 Hour
2.3	Dynamic Unit testing - Control Flow testing, Data Flow testing	1 Hour
2.4	Domain testing, Functional Program testing.	
2.5	Mutation testing - Mutation and Mutants, Mutation operators, Mutation score.	1 Hour
2.6	Junit - Framework for Unit testing.	1 Hour
2.7	Case Study - Mutation testing using Junit	1 Hour
2.8	Case Study - Mutation testing using Muclipse	1 Hour
Module 3 (Unit Testing:- White Box Approaches) 10 Hours		
3.1	Overview of Graph Coverage Criteria	1 Hour
3.2	Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage	1 Hour
3.3	Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage.	1 Hour
3.4	Data Flow Criteria - du paths, du pairs	1 Hour
3.5	Subsumption Relationships among Graph Coverage Criteria.	1 Hour
3.6	Graph Coverage for Source Code - Control Flow Graphs (CFG) for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program - Statistics	1 Hour
3.7	Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph,	1 Hour

3.8	Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root	1 Hour
3.9	Case Study - Graph Based testing using JUnit Framework. (Lecture 1)	1 Hour
3.10	Case Study - Graph Based testing using JUnit Framework. (Lecture 2)	1 Hour
Module 4 (Unit Testing:- Black Box Approaches) 9 Hours		
4.1	Domain Testing / Input Space Partitioning - Partitions of a set.	1 Hour
4.2	Input domain modelling - Interface-based approach, Functionality-based approach.	1 Hour
4.3	Identifying values.	1 Hour
4.4	Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage.	1 Hour
4.5	TriTyp example.	1 Hour
4.6	Functional Testing - Functional Testing Concepts of Howden. Important Steps.	1 Hour
4.7	Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis	1 Hour
4.8	Decision Tables, Random Testing.	1 Hour
4.9	Case Study - Black Box testing approaches using JUnit.	1 Hour
Module 5 (Grey Box Testing Approaches) 9 Hours		
5.1	Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages.	1 Hour
5.2	Techniques of Grey Box Testing - Matrix Testing, Regression Testing,	1 Hour

5.3	Orthogonal Array Testing or OAT, Pattern Testing.	1 Hour
5.4	An Introduction to Pex - Parameterized Unit Testing, The Testing Problem.	1 Hour
5.5	Symbolic Execution – Example, Symbolic execution tree.	1 Hour
5.6	PEX application.	1 hour
5.7	Case Study – PEX (Lecture 1)	1 Hour
5.8	Case Study – PEX (Lecture 2)	1 Hour
5.9	Case Study – PEX (Lecture 3)	1 Hour

CST 384	CONCEPTS IN DEEP LEARNING	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course aims to introduce the learner to an overview of the concepts and algorithms involved in deep learning. Deep learning is a subfield of machine learning, a subfield of artificial intelligence. Basic concepts and application areas of machine learning, deep networks, convolutional neural network and recurrent neural network are covered here. This is a foundational program that will help students understand the capabilities, challenges, and consequences of deep learning and prepare them to participate in the development of leading-edge AI technology. They will be able to gain the knowledge needed to take a definitive step in the world of AI.

Prerequisite: Sound knowledge in Basics of linear algebra and probability theory.

CO1	Demonstrate basic concepts in machine learning.**(Cognitive Knowledge Level: Understand)**
CO2	Illustrate the validation process of machine learning models using hyper-parameters and validation sets. **(Cognitive Knowledge Level: Understand)**
CO3	Demonstrate the concept of the feed forward neural network and its training process. **(Cognitive Knowledge Level: Apply)**
CO4	Build CNN and Recurrent Neural Network (RNN) models for different use cases. **(Cognitive Knowledge Level: Apply)**
CO5	Use different neural network/deep learning models for practical applications. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	**30**	**30**	**30**
Understand	**40**	**40**	**40**
Apply	**30**	**30**	**30**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

INTRODUCTION TO DEEP LEARNING

(General Instructions: Instructors are to introduce students to any one software platform and demonstrate the working of the algorithms in the syllabus using suitable use cases and public datasets to give a better understanding of the concepts discussed. Tutorial hour may be used for this purpose)

Module-1 (Introduction)

Key components - Data, models, objective functions, optimization algorithms, Learning algorithm. Supervised learning- regression, classification, tagging, web search, page ranking, recommender systems, sequence learning, Unsupervised learning, Reinforcement learning, Historical Trends in Deep Learning. Other Concepts - overfitting, underfitting, hyperparameters and validation sets, estimators, bias and variance.

Module- 2 (Optimization and Neural Networks)

Neural Networks –Perceptron, Gradient Descent solution for Perceptron, Multilayer perceptron, activation functions, architecture design, chain rule, back propagation, gradient based learning. Introduction to optimization– Gradient based optimization, linear least squares. Stochastic gradient descent, Building ML algorithms and challenges.

Module -3 (Convolutional Neural Network)

Convolutional Neural Networks – convolution operation, motivation, pooling, Structure of CNN, Convolution and Pooling as an infinitely strong prior, variants of convolution functions, structured outputs, data types, efficient convolution algorithms. Practical challenges of common deep learning architectures- early stopping, parameter sharing, dropout. Case study: AlexNet, VGG, ResNet.

Module- 4 (Recurrent Neural Network)

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs.

Module-5 (Application Areas)

Applications – computer vision, speech recognition, natural language processing, common word embedding: continuous Bag-of-Words, Word2Vec, global vectors for word representation (GloVe). Research Areas – autoencoders, representation learning, boltzmann machines, deep belief networks.

Text Book

1. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, MIT Press 2015 ed.
2. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learning, August 2019.
3. Neural Networks and Deep Learning, Aggarwal, Charu C., c Springer International Publishing AG, part of Springer Nature 2018

Reference Books

1. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks by Russell Reed, Robert J MarksII, A Bradford Book,2014
2. Practical Convolutional Neural Networks by MohitSewak, Md. Rezaul Karim, PradeepPujari,Packt Publishing 2018
3. Hands-On Deep Learning Algorithms with Python by SudharsanRavichandran,Packt Publishing 2019
4. Deep Learning with Python by Francois Chollet,Manning Publications Co.,2018

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

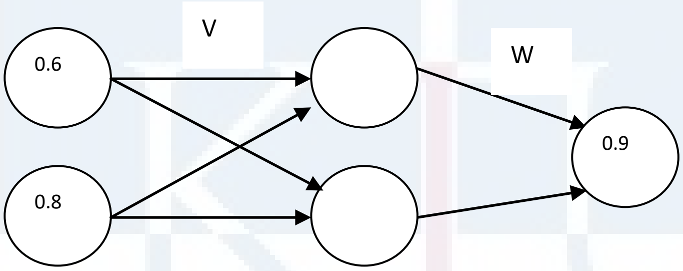
1. Compare regression and classification.
2. Define supervised learning? Distinguish between regression and classification.
3. Discuss the different learning approaches used in machine learning.

Course Outcome 2 (CO2):

1. What are hyperparameters? Why are they needed?
2. What issues are to be considered while selecting a model for applying machine learning in a given problem?

Course Outcome 3 (CO3):

1. Update the parameters V_{11} in the given MLP using back propagation with learning rate as 0.5 and activation function as sigmoid. Initial weights are given as V_{11}= 0.2, V_{12}=0.1, V_{21}=0.1, V_{22}=0.3, V_{11}=0.2, W_{11}=0.5, W_{21}=0.2



2. Draw the architecture of a multi-layer perceptron.
3. Derive update rules for parameters in the multi-layer neural network through the gradient descent.

Course Outcome 4 (CO4):

1. Give two benefits of using convolutional layers instead of fully connected ones for visual tasks.
2. Suppose that a CNN was trained to classify images into different categories. It performed well on a validation set that was taken from the same source as the training set but not on a testing set. What could be the problem with the training of such a CNN? How will you ascertain the problem? How can those problems be solved?
3. Explain how the cell state is updated in the LSTM model from Ct-1 to Ct
4. Show the steps involved in an LSTM to predict stock prices.

Course Outcome 5 (CO5):

1. Explain how the cell state is updated in the LSTM model from Ct-1 to Ct
2. Show the steps involved in an LSTM to predict stock prices.
3. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.

Course Outcome 6 (CO6):

1. Development a deep learning solution for problems in the domain i) natural language processing or ii Computer vision (Assignment
2. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.

Model Question Paper

QP CODE: PAGES:4

Reg No:_______________

Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 384

Course Name: CONCEPTS IN DEEP LEARNING

Max. Marks:100 **Duration: 3 Hours**

PART A
Answer all Questions. Each question carries 3 Marks

1. Distinguish between supervised learning and Reinforcement learning. Illustrate with an example.

2. Differentiate classification and regression.

3. Compare overfitting and underfitting. How it can affect model generalization.

4. Why does a single perceptron cannot simulate simple XOR function? Explain how this limitation is overcome?

5. Illustrate the strengths and weaknesses of convolutional neural networks.

6. Illustrate convolution and pooling operation with an example

7. How many parameters are there in AlexNet? Why the dataset size (1.2 million) is important for the success of AlexNet?

8. Explain your understanding of unfolding a recursive or recurrent computation into a computational graph.

9. Illustrate the use of deep learning concepts in Speech Recognition.

10. What is an autoencoder? Give one application of an autoencoder

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." What is your understanding of the terms task, performance and experience. Explain with two example **(10)**

(b) "How does bias and variance trade-off affect machine learning algorithms? **(4)**

OR

12. (a) Illustrate the concepts of Web search, Page Ranking, Recommender systems with suitable examples. **(10)**

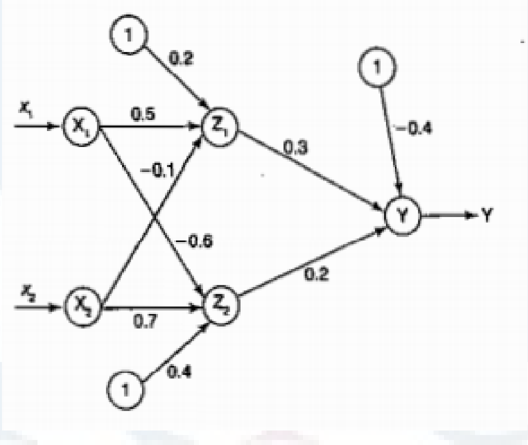
(b) List and discuss the different hyper parameters used in fine tuning the **(4)** traditional machine learning models

13. (a) How multilayer neural networks learn and encode higher level features from input features. **(7)**

(b) Explain gradient decent and delta rule? Why stochastic approximation to gradient descent is needed? **(7)**

OR

14. (a) Find the new weights for the network using backpropogation algorithm, the network is given with a input pattern[-1,1] and target output as +1, Use learning rate of alpha=0.3 and bipolar sigmoid function. **(7)**



(b) Write an algorithm for backpropgation which uses stochastic gradient descent method. Comment on the effect of adding momentum to the network. **(7)**

15. (a) Input to CNN architecture is a color image of size 112x112x3. The first convolution layer comprises of 64 kernels of size 5x5 applied with a stride of 2 and padding 0. What will be the number of parameters? **(5)**

(b) Let X=[-1, 0, 3, 5] W=[.3 ,.5 .2,.1] be the the input of ith layer of a neural network and to apply softmax function. What should be the output of it? **(4)**

(c) Draw and explain the architecture of convolutional network **(5)**

OR

16. (a) Explain the concept behind i) Early stopping ii) dropout iii) weight decay **(9)**

(b) How backpropagation is used to learn higher-order features in a convolutional Network? **(5)**

17. (a) Explain the working of RNN and discuss how backpropagation through time is used in recurrent networks. **(8)**

(b) Describe the working of a long short term memory in RNNs. **(6)**

OR

18. (a) What is the vanishing gradient problem and exploding gradient problem? **(8)**

(b) Why do RNNs have a tendency to suffer from exploding/vanishing gradient? How to overcome this challenge? **(6)**

19. (a) Explain any two word embedding techniques **(8)**

(b) Explain the merits and demerits of using Auto encoders in Computer Vision. **(6)**

OR

20. (a) Illustrate the use of representation learning in object classification. **(7)**

(b) Compare Boltzmann Machine with Deep Belief Network. **(7)**

Teaching Plan

CONCEPTS IN DEEP LEARNING (45 Hours)		
Module 1 : Introduction (9 hours)		
1.1	Key components - Data, models, objective functions, optimization algorithms. (TB2: Section 1.1-1.2)	1 hour

1.2	Learning algorithm (TB1: Section 5.1), Supervised learning- regression, classification (TB2: Section 1.3.1)	1 hour
1.3	tagging, web search, page ranking (TB2: Section 1.3.1)	1 hour
1.4	Recommender systems, Sequence learning, Unsupervised learning, Reinforcement learning(TB2: Section 1.3.2-1.3.4)	1 hour
1.5	Historical Trends in Deep Learning (TB1: Section 1.2).	1 hour
1.6	Concepts: over-fitting, under-fitting, hyperparameters and validation sets. (TB1: Section 5.2-5.3)	1 hour
1.7	Concepts: Estimators, bias and variance. (TB1: Section 5.4)	1 hour
1.8	Demonstrate the concepts of supervised learning algorithms using a suitable platform.	1 hour
1.9	Demonstrate the concepts of unsupervised using a suitable platform.	1 hour
Module 2 : Optimization and Neural Networks (9 hours)		
2.1	Perceptron, Stochastic Gradient descent, Gradient descent solution for perceptron (TB3: Section 1.1 - 1.2.1)	1 hour
2.2	Multilayer perceptron (TB3: Section 1.2.2), (TB1: Section 6.1,6.3)	1 hour
2.3	Activation functions- Sigmoid, tanh, Softmax, ReLU, leaky ReLU (TB3: Section 1.2.1.3 - 1.2.1.5)	1 hour
2.4	Architecture design (TB1: Section 6.4, TB3: Section 1.6)	1 hour
2.5	Chain rule, back propagation (TB3: Section 1.3)	1 hour

2.6	Gradient based learning (TB1: Section 6.2)	1 hour
2.7	Gradient based optimization (TB1: Section 4.3)	1 hour
2.8	Linear least squares using a suitable platform. (TB1: Section 4.5)	1 hour
2.9	Building ML Algorithms and Challenges (TB3: 1.4, TB1: 5.10-5.11)	1 hour
Module 3 :Convolution Neural Network (10 hours)		
3.1	Convolution operation, Motivation, pooling (TB1:Section 9.1-9.3)	1 hour
3.2	Structure of CNN (TB3: Section 8.2)	1 hour
3.3	Convolution and Pooling as an infinitely strong prior (TB1: Section 9.4)	1 hour
3.4	Variants of convolution functions – multilayer convolutional network, tensors, kernel flipping, downsampling, strides and zero padding. (TB1: Section 9.5)	1 hour
3.5	Variants of convolution functions - unshared convolutions, tiled convolution, training different networks. (TB1: Section 9.5)	1 hour
3.6	Structured outputs, data types (TB1: Section 9.6-9.7)	1 hour
3.7	Efficient convolution algorithms. (TB1: Section 9.8,9.10)	1 hour
3.8	Practical challenges of common deep learning architectures- early Stopping (TB3: 4.6)	1 hour
3.9	Practical challenges of common deep learning architectures- parameter sharing, drop-out (TB3: Section 4.9, 4.5.4)	1 hour
3.10	Case Study: AlexNet,VGG, ResNet. (TB3: Section 8.4.1-8.4.3,8.4.5)	1 hour

Module 4 :Recurrent Neural Network (8 hours)		
4.1	Computational graphs (TB1: Section 10.1)	1 hour
4.2	RNN (TB1: Section 10.2-10.3)	1 hour
4.3	Encoder – decoder sequence to sequence architectures. (TB1: Section 10.4)	1 hour
4.4	Deep recurrent networks (TB1: Section 10.5)	1 hour
4.5	Recursive neural networks , Modern RNNs, (TB1: Section 10.6, 10.10)	1 hour
4.6	LSTM and GRU (TB1: Section 10.10, TB3: Section 7.5-7.6)	1 hour
4.7	Practical use cases for RNNs. (TB1: Section 11.1-11.4)	1 hour
4.8	Demonstrate the concepts of RNN using a suitable platform.	1 hour
Module 5 : Applications and Research (9 hours)		
5.1	Computer vision. (TB1: Section 12.2)	1 hour
5.2	Speech recognition. (TB1: Section 12.3)	1 hour
5.3	Natural language processing. (TB1: Section 12.4)	1 hour
5.4	Common Word Embedding -: Continuous Bag-of-Words, Word2Vec (TB3: Section 2.6)	1 hour
5.5	Common Word Embedding -: Global Vectors for Word Representation(GloVe) (TB3: Section 2.9.1- Pennigton 2014)	1 hour
5.6	Brief introduction on current research areas- Autoencoders, Representation learning. (TB3: Section 4.10)	1 hour

5.7	Brief introduction on current research areas- representation learning. (TB3: Section 9.3)	1 hour
5.8	Brief introduction on current research areas- Boltzmann Machines, Deep belief networks. (TB1: Section 20.1, TB3 Section 6.3)	1 hour
5.9	Brief introduction on current research areas- Deep belief networks. (TB1: Section 20.3)	1 hour

CST 386	WIRELESS NETWORKS AND IoT APPLICATIONS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course equips the learners with fundamental wireless technologies for the Internet of Things(IoT) and the IoT ecosystem. It covers the underlying concepts in wireless networks, communication mechanisms, protocols, hardware, software, and the cloud platforms for IoT. The students will be able to design smart IoT applications for real world problems..

Prerequisite: Sound knowledge in Data Communication, Computer Networks and Programming in C

Course Outcomes: After the completion of the course the students will be able to

CO1	Recognize wireless technologies required for IoT ecosystem **(Cognitive Knowledge Level : Understand)**
CO2	Perceive the concept of IoT and M2M architecture, IoT examples, and Data Management in IoT **(Cognitive Knowledge Level :Apply)**
CO3	Outline the hardware components used in IoT including Sensors, Actuators and development boards **(Cognitive Knowledge Level : understand)**
CO4	Explain the software components of IoT **(Cognitive Knowledge Level :Understand)**
CO5	Demonstrate the protocols used in IoT and build IoT Programs **(Cognitive Knowledge Level : Apply)**
CO6	Build IoT-based smart real-time applications such as Smart Healthcare, Smart Agriculture, Smart Environment and Smart Home **(Cognitive Knowledge Level : Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓									✓

CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓	✓	✓	✓							✓
CO6	✓	✓	✓	✓	✓	✓						

Abstract POs Defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and teamwork
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Blooms Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test 1 (Percentage)	**Test 2 (Percentage)**	
Remember	**30**	**30**	**30**
Understand	**50**	**40**	**40**
Apply	**20**	**30**	**30**

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 Hours**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus, and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have a maximum 2 subdivisions and carries 14 marks.

Syllabus

Module- 1 (Introduction to IoT and wireless technologies required for IoT)

Internet of Things, Role of Things and the Internet, Wireless IoT. Wireless Networks - Network Topologies, Types of Networks. Role of Wireless Standards in IoT. Protocol Stack - OSI Model, TCP/IP Model, IEEE 802 Reference Model, Protocols for Wireless IoT. Bluetooth - Transceiver, Frequency Channels, Typical Range, Access and Spread Spectrum, Modulation and Data Rate, Error Correction and Detection, Network Topology. ITU G.9959, Zwave, IEEE 802.15.4, Zigbee Specification, Thread, WiFi, 6LowPAN, IPv6, LoRaWAN.

Module- 2 (IoT architecture, Data and Device management)

Internet of Things - IoT Architectural View, Technology Behind IoT - Server End Technology, Sources of Internet of Things, M2M Communication. IoT Application Areas. IoT Examples. IoT Data Management - Device Management Gateways. Design Principles for Web Connectivity - Web Communication Protocols for Connected Devices, Web Connectivity for Connected Devices using Gateways. Internet Connectivity Principles – Internet Connectivity, Internet based communication, IP addressing in the IoT.

Module- 3 (Data Acquiring and Enabling Technologies)

Data Acquiring and Storage for IoT Sevices- Organization of Data, Big data, Acquiring Methods, Management Techniques, Analytics, Storage Technologies. Cloud Computing for Data storage - IoT Cloud based Services using Xively, Nimbits, and Other Platforms. Sensor Technologies for IoT Devices - Sensor Technology, Participatory Sensing, Industrial IoT and Automotive IoT, Actuators for Various Devices, Sensor Data Communication Protocols, Wireless Sensor network Technology

Module-4 (Prototyping the Embedded Devices for IoT)

Embedded Computing Basics, Embedded Hardware Unit. Embedded Platforms for Prototyping - Arduino, Intel Galileo, Intel Edison, Raspberry Pi, BeagleBone, mBed. Prototyping and Designing the Software for IoT Applications- Introduction, Prototyping Embedded DeviceSoftware- Programming using Arduino, Programming for an Arduino Controlled Traffic Control Lights at a Road Junction, Basic Arduino Programs to Blink LED, Find the Distance using Ultrasonic Sensor, Estimate Room Temperature, Measuring Soil Moisture Level

Module 5 (Business Models and Case Studies)

Business Models and Processes using IoT. Value Creation in the Internet of Things. Cloud PaaS- Xively, Nimbits, IBM Bluemix, CISCO IoT, AWS IoT, TCS Connected AWS Platform, Case studies- Smart Home, Smart Environment, Smart healthcare, Smart agriculture

Text Books

1. Daniel Chew, "Wireless Internet of Things -A Guide to the lower layers", IEEE Standards and Association, IEEE Press, Wiley
2. Rajkamal, "Internet of Things : Architecture and Design Principles", McGraw Hill (India) Private Limited.

References

1. ArshadeepBahga, Vijay Madisetti, "Internet of Things: A hands-on approach", University Press, 2015 (First edition)
2. Dieter Uckelmann, Mark Harrison, Michahelles Florian (Ed.), Architecting the internet of things, Springer, 2011
3. Dr. Ovidiu Vermesan, Dr. Peter Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers, 2013
4. Simon Monk, "Programming Arduino: Getting Started with Sketches", McGraw Hill Publications

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare Bluetooth and Bluetooth LE power classes
2. Demonstrate Zigbee Specification Protocol Stack

Course Outcome 2 (CO2):

1. What are the major components of IOT system? Briefly explain each
2. Correlate M2M architectural Levels with IOT architectural Levels

Course Outcome 3 (CO3):

1. Describe the use of GPIO pins ?
2. What are actuators ? Mention the roles of actuators in IoT systems

Course Outcome 4(CO4):

1. Identify the role of HBase in Hadoop File System
2. Differentiate Edge computing and Distributed computing
3. Illustrate open protocols, tools and frameworks generally used in M2M

Course Outcome 5(CO5):

1. What do you mean by Arduino sketches?
2. Write an Arduino program to blink LED

Course Outcome 6(CO6):

1. How IoT technology helps TELEMEDICINE in India?
2. How soil moisture can be detected in Smart Agriculture?

Model Question Paper

QP CODE: PAGES :2

Reg No: _______________

Name: _________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 386

Course Name: WIRELESS NETWORKS AND IoT APPLICATIONS

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Illustrate Role of *things* and *internet* in IoT

2. What is Bluetooth? Explain the range and frequency channels of Bluetooth?

3. List any three the features of Constrained Application Protocol (COAP).

4. Compare Raspberry Pi and BeagleBoard boards.

5. Identify the role of HBase in Hadoop File System.

6. Differentiate Edge computing and Distributed computing.

7. Give an example of Raspberry Pi applications for Industrial IoT.

8. What are the on-board functional units in Intel Galileo?

9. Interpret the concept of value creation in IoT.

10. Explain the use of PaaS in IoT Smart applications with any three examples.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Compare various Network topologies used in Wireless Networks. **(8)**

(b) Describe the following wireless technologies on i) *Zigbee* ii)*WiFi* iii) *Thread.* **(6)**

OR

12. (a) Explain protocol stacks used in wireless networks for IoT applications. **(8)**

(b) Illustrate the Architectural design of LoRaWAN. **(6)**

13. (a) Define M2M. Explain M2M architecture. Correlate M2M architectural levels with IoT architectural levels. **(8)**

(b) Compare SOAP and REST protocols. **(6)**

OR

14. (a) Summarize different Online Transactions and Processing techniques. **(8)**

(b) Identify the functions of Device-Management Gateway . **(6)**

15. (a) Define actuators ? Describe the roles of actuators in IoT systems. **(8)**

(b) Explain the usage contexts of analog sensors and digital sensors. **(6)**

OR

16. (a) How data collection, storage & computing services done using Nimbits? **(10)**

(b) List any four features of Xively. **(4)**

17. (a) What do you mean by Arduino sketches? **(4)**

(b) Write an Arduino program to blink LED **(10)**

OR

18. (a) Demonstrate an example of Raspberry Pi applications for Industrial IoT. **(10)**

(b) Compare the features of Arduino-R3 and Arduino Yun boards. **(4)**

19. (a) Explain various tasks of a smart irrigation monitoring service. **(8)**

(b) Demonstrate the tasks of Soil-Moisture monitoring service. **(6)**

OR

20. (a) a) Mr. Kiran Mathew has been a chronic diabetic patient for the past few years. He was under regular check up at the hospital every two weeks. All of a sudden the pandemic like COVID-19 arises in the country and the government issues a lockdown for a period of two months. Illustrate how Mr. Kiran can be monitored by the health care worker using intelligent healthcare techniques. **(10)**

(b) Mention any four sensors used in smart healthcare **(4)**

TEACHING PLAN

No	Contents	No of Lecture Hrs(45)
Module – 1 (Introduction to IoT and wireless technologies required for IoT) (8 hrs) (TB-1, Chapter 1...)		
1.1	Internet Of Things, Role of things and internet ,Wireless IoT	1
1.2	Wireless Networks- Network Topologies-Types of Networks,Role of	1

	Wireless standards in IoT	
1.3	Protocol Stack-OSI Model- TCP/IP Model-IEEE 802 reference model	1
1.4	Protocols for Wireless IoT-Bluetooth-Transceiver, Frequency Channels-Typical Range, Access and Spread Spectrum, Modulation and Data Rate	1
1.5	Error Correction and Detection-Network Topology.	1
1.6	ITU G.9959, Zwave, IEEE 802.15.4, Zigbee Specification	1
1.7	Thread, Wifi, 6LowPAN, IPv6	1
1.8	LoRaWAN	1
Module- 2 (IOT architecture, Data and Device management) (9hrs)		
2.1	Internet of Things -IoT Architectural view	1
2.2	Technology Behind IOT-Server End Technology,Sources of Internet of Things	1
2.3	M2M Communication.	1
2.4	IoT Application Areas. IOT Examples.	1
2.5	IoT Data Management, Device Management Gateways.	1
2.6	Design Principles for Web Connectivity	1
2.7	Web communication protocols for connected devices,	1
2.8	Web connectivity for connected devices using Gateways.	1
2.9	Internet connectivity Principles – Internet Connectivity, Internet based communication, IP addressing in the IoT.	1
Module- 3 (Data Acquiring and Enabling Technologies (8 hrs)		
3.1	Data acquiring and storage for IoT devices- Organization of Data, Big data	1
3.2	Acquiring methods, management techniques, Analytics, Storage technologies.	1
3.3	Cloud computing for Data storage-IoT Cloud based services using Xively,	1

	Nimbits, and other platforms.	
3.4	Cloud computing-Nimbits	1
3.5	Sensor Technologies for IoT Devices-Sensor Technology, Participatory sensing	1
3.6	Industrial IoT and Automotive IoT	1
3.7	Actuators for various devices, Sensor data communication protocols	1
3.8	Wireless Sensor network Technology	1
Module 4(Prototyping the Embedded Devices for IoT)(9hrs)		
4.1	Introduction, Embedded Computing Basics, Embedded Hardware Unit.	1
4.2	Embedded Platforms for Prototyping-Arduino, Intel Galileo	1
4.3	Intel Edison, Raspberry Pi, BeagleBone, mBed	1
4.4	Prototyping and designing the software for IoT applications-Introduction, Prototyping embedded device software	1
4.5	Prototyping and designing the software for IoT applications-Introduction, Prototyping embedded device software	1
4.6	Programming concepts in Arduino	1
4.7	Programming for an arduino controlled traffic control lights at a road junction	1
4.8	Basic Arduino programs to blink LED, Find the distance using ultrasonic sensor	1
4.9	Estimate room temperature, Measuring soil moisture level	1
Module 5 (higher level protocols and case studies)(9 hrs)		
5.1	Business Models and Processes using IOT, Value creation in the Internet of Things.	1

5.2	Xively, Nimbits, IBM Bluemix	1
5.3	CISCO IoT, AWS IoT, TCS Connected AWS Platform	1
5.4	Case Study- Smart Environment	1
5.5	Case Study- Smart Environment	1
5.6	Case study Smart Home	1
5.7	Case study Smart Home	1
5.8	Case study Smart healthcare (Lecture I)	1
5.9	Case study Smart healthcare (Lecture II)	1
5.10	Case study -Smart agriculture (Lecture I)	1
5.11	Case study -Smart agriculture (Lecture II)	1

SEMESTER VI

HONOURS

CST 394	NETWORK SECURITY	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

The purpose of this course is to create a better understanding of the network security concepts. This course covers network security standards, email security services, web security mechanisms, firewalls and wireless security mechanisms. This course helps the learner to gain insight into the key aspects of secure network communication and enables to apply in real-life scenarios.

Prerequisite: A sound background in Number Theory and Cryptographic Algorithms.

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Identify the key aspects of security, intrusion detection systems and digital signature schemes **(Cognitive Knowledge Level: Apply)**
CO2	Explain the security standards used in network communication **(Cognitive Knowledge Level:Understand)**
CO3	Identify the mechanisms in email security services **(Cognitive Knowledge Level: Apply)**
CO4	Summarize the protocols used to provide web security **(Cognitive Knowledge Level: Understand)**
CO5	Explain the fundamental concepts of wireless network security and firewalls **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓		✓						✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Network Security Basics)

Introduction to network security - Security requirements, Challenges of security, Network security model. Malicious programs – Worms, Viruses, Trojans, Spyware, Adware. Intrusion Detection Systems (IDS) - Uses, Techniques. Digital signatures - ElGamal, Schnorr, Digital Signature Standard (DSS).

Module – 2 (Network Security Standards)

Kerberos v4 – Configuration, Authentication, Encryption, Message formats. Kerberos v5 – Cryptographic algorithms, Message formats. Public Key Infrastructure (PKI) – Trust models, Revocation. Real-time communication security – Perfect Forward Secrecy (PFS), Denial-of-Service protection, Endpoint identifier hiding, Live partner reassurance. Internet Protocol Security (IPSec) - Authentication Header (AH), Encapsulating Security Payload (ESP), Internet Key Exchange (IKE) phases.

Module – 3 (Email Security)

Introduction to email security - Security services for email, Establishing keys, Privacy, Authentication, Message integrity, Non-repudiation. Privacy Enhanced Mail (PEM) – Encryption, Source authentication and integrity protection, Message formats. Secure/Multipurpose Internet Mail Extensions (S/MIME) – Messages, Differences from PEM. Pretty Good Privacy (PGP) - Encoding, Certificate and key revocation, Anomalies, Object formats.

Module – 4 (Web Security)

Introduction to web security - Web security considerations, Threats. Secure Sockets Layer (SSL) – Architecture, Protocols, Transport Layer Security (TLS) – Differences from SSL. Hypertext Transfer Protocol Secure (HTTPS) – Connection initiation, Closure. Secure Shell (SSH) – Transport layer protocol, User authentication protocol, Connection protocol.

Module – 5 (Wireless Network Security and Firewalls)

IEEE 802.11 Wireless LAN - Network components, Architectural model, Services. IEEE 802.11i wireless LAN security - Services, Phases of operation. Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2, Wireless Application Protocol (WAP) – Services, Protocol architecture. Firewalls – Need for firewalls, Packet filters, Circuit-level firewalls, Application layer firewalls.

Text Books

1. C. Kaufman, R. Perlman and M. Speciner, “Network Security: Private Communication in a Public World”, 2/e, PHI.
2. William Stallings, “Cryptography and Network Security Principles and Practice”, 5/e, Pearson

Education Asia.

References

1. Behrouz A. Forouzan, Debdeep Mukhopadhyay, “Cryptography and Network Security”, 3/e, Tata McGraw Hill.
2. Tyler Wrightson, “Wireless Network Security A Beginner’s Guide”, 2012, Tata McGraw Hill.
3. William Stallings, “Network Security Essentials: Applications and Standards”, 4/e, Prentice Hall.
4. Schiller J., Mobile Communications, 2/e, Pearson Education.
5. Roberta Bragg et. al., “Network Security: The Complete Reference”, Tata McGraw Hill.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Using the Schnorr digital signature scheme, let q = 83, p = 997 and d = 23. Find values for e_1 and e_2.
2. The Digital Signature Algorithm (DSA) specifies that if the signature generation process results in a value of zero, a new value of k should be generated and the signature should be recalculated. Give reason.

Course Outcome 2 (CO2):

1. In Kerberos v4, the authenticator field is not of security benefit when asking the Key Distribution Center (KDC) for a ticket for Bob, but useful when logging in as Bob. Give reasons for your answer.
2. How does the stateless cookie protocol provide clogging protection?

Course Outcome 3 (CO3):

1. If Alice is sending an ENCRYPTED message, she first signs the message digest with her private key and then encrypts the message digest with the pre-message secret key. Why this last encryption was considered necessary for encrypted messages and not for MIC-CLEAR or MIC-ONLY?
2. Which security services are considered desirable in the following cases? (i) Sending a purchase order (ii) Sending a ransom note. (iii) Sending a mission description to security officials.
3. Explain the security mechanism used in Gmail communication.

Course Outcome 4 (CO4):

1. Is it possible in SSL for the receiver to reorder SSL record blocks that arrive out of order? If so, how it can be done? If not, why?
2. Describe any five web security threats, their consequences and countermeasures.

Course Outcome 5 (CO5):

1. Explain the security areas addressed by IEEE 802.11i.
2. Describe the advantages and disadvantages of application layer firewalls.

Model Question Paper

QP CODE:
Reg. No:______________
Name:________________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH. DEGREE (HONORS) EXAMINATION, MONTH &YEAR
Course Code: CST 394

Course Name: Network Security

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Distinguish between signature-based and anomaly-based intrusion detection techniques.

2. A trusted third party is considered as a main component in a network security model. Why?

3. How is endpoint identifier hiding achieved in real-time communication?

4. Show how encryption is used to provide privacy and integrity in Kerberos v5.

5. End-to-end privacy is essential for e-mail security. How is this achieved?

6. List the four steps for preparing an EnvelopedData MIME entity.

7. Show the operation of a Secure Sockets Layer (SSL) Record protocol.

8. For Secure Shell (SSH) packets, what is the advantage of not including the MAC in the scope of packet encryption?

9. List the three security services provided by IEEE 802.11i.

10. Define the terms Access Point, Basic Service Set, Extended Service Set.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Using the ElGamal scheme, let p = 881 and d = 700, find values for e1 and e2. Choose r = 17. Find the value of S1 and S2 if M = 400. **(8)**

    (b) Explain the requirements and challenges of network security. **(6)**

OR

12. (a) In ElGamal, Schnorr and DSS, what happens if an attacker can find the value of random secret key used by the signer? Also, what happens if a user uses the same value of random secret key to sign two messages? Explain your answer for each scheme separately. **(8)**

    (b) Explain the network security model with the help of a neat diagram. **(6)**

13. (a) Alice wishes to log into Bob's workstation remotely. List the steps involved in this communication if Kerberos v4 is used. **(7)**

    (b) How does Diffie-Hellman technique provide perfect forward secrecy using signature keys? **(7)**

OR

14. (a) Explain the algorithm for Message Authentication Code (MAC) calculation and verification in Kerberos v5 rsa-md5-des. **(8)**

    (b) Compare the aggressive mode and main mode of Phase 1 Internet Key Exchange (IKE). **(6)**

15. (a) Describe the different methods by which authentication of source is performed in email communication. **(7)**

    (b) Explain the Signed data and Clear-signed data functions provided by S/MIME. **(7)**

OR

16. (a) Explain the advantages of Pretty Good Privacy (PGP) over Privacy Enhanced Mail (PEM). **(7)**

(b) Define non-repudiation. Describe the different ways by which it is implemented in email communication. **(7)**

17. (a) Describe the significance of pseudo-random function of Transport Layer Security. **(7)**

(b) Explain the four different phases of Secure Sockets Layer (SSL) HandshakeProtocol. **(7)**

OR

18. (a) Describe how connection initiation and connection closure is done in Hyper Text Transfer Protocol Secure (HTTPS). **(7)**

(b) Illustrate the sequence of events in Secure Shell (SSH) transport layer protocol packet exchanges. **(7)**

19. (a) Explain the Discovery phase and Authentication phase of IEEE 802.11i operation. **(7)**

(b) Why are firewalls needed? Compare the features of packet filters and circuit level firewalls. **(7)**

OR

20. (a) Explain the two authentication methods used in Wired Equivalent Privacy (WEP). **(7)**

(b) Describe the three transaction classes provided by Wireless Transaction Protocol. **(7)**

Teaching Plan

No	Contents	No of Lecture Hrs
Module - 1 (Network Security Basics) (7 hrs)		
1.1	Security requirements, Challenges of security	1
1.2	Network security model	1
1.3	Worms, Viruses, Trojans, Spyware, Adware	1
1.4	Intrusion Detection Systems (IDS) uses, Techniques	1
1.5	ElGamal digital signature	1
1.6	Schnorr digital signature	1
1.7	Digital Signature Standard (DSS)	1
Module - 2 (Network Security Standards) (12 hrs)		
2.1	Kerberos v4 configuration, Authentication	1
2.2	Kerberos v4 encryption	1
2.3	Kerberos v4 message formats	1
2.4	Kerberos v5 cryptographic algorithms – rsa-md5-des, des-mac, des-mac-k	1
2.5	Kerberos v5 cryptographic algorithms - rsa-md4-des, rsa-md4-des-k, Encryption for privacy and integrity	1
2.6	Kerberos v5 message formats	1
2.7	Public Key Infrastructure (PKI) trust models	1
2.8	PKI revocation	1
2.9	Perfect Forward Secrecy (PFS), Denial-of-Service protection	1
2.10	Endpoint identifier hiding, Live partner reassurance	1
2.11	Internet Protocol Security (IPSec) Authentication Header (AH), Encapsulating Security Payload (ESP)	1

2.12	Internet Key Exchange (IKE) phases	1
Module - 3 (Email Security) (9 hrs)		
3.1	Security services for email, Establishing keys, Privacy	1
3.2	Authentication, Message integrity, Non-repudiation	1
3.3	Privacy Enhanced Mail (PEM) encryption, Source authentication	1
3.4	PEM integrity protection, Message formats (Lecture 1)	1
3.5	PEM message formats (Lecture 2)	1
3.6	Secure/Multipurpose Internet Mail Extensions (S/MIME) – Messages, Differences from PEM	1
3.7	Pretty Good Privacy (PGP) encoding, Certificate and key revocation, Anomalies	1
3.8	PGP Object formats (Lecture 1)	1
3.9	PGP Object formats (Lecture 2)	1
Module – 4 (Web Security)(9 hrs)		
4.1	Web security considerations, Threats, Secure Sockets Layer (SSL) architecture	1
4.2	SSL protocols (Lecture 1)	1
4.3	SSL protocols (Lecture 2)	1
4.4	Transport Layer Security (TLS) differences from SSL (Lecture 1)	1
4.5	TLS differences from SSL (Lecture 2)	1
4.6	Hypertext Transfer Protocol Secure (HTTPS) connection initiation, Closure	1
4.7	Secure Shell (SSH) transport layer protocol	1
4.8	SSH user authentication protocol	1
4.9	SSH connection protocol	1

Module - 5 (Wireless Security and Firewalls) (8 hrs)		
5.1	IEEE 802.11 Wireless LAN network components, Architectural model, Services	1
5.2	IEEE 802.11i wireless LAN security services, Phases of operation (Lecture 1)	1
5.3	IEEE 802.11i phases of operation (Lecture 2)	1
5.4	Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2	1
5.5	Wireless Application Protocol (WAP) services, Protocol architecture (Lecture 1)	1
5.6	WAP protocol architecture (Lecture 2)	1
5.7	Need for firewalls, Packet filters	1
5.8	Circuit-level firewalls, Application layer firewalls	1

CST 396	ADVANCED TOPICS IN MACHINE LEARNING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course enables the learners to understand the advanced concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning and the naive Bayes algorithm, basic clustering algorithms, auto encoders, sampling methods and PAC learning. This course helps the students to provide machine learning based solutions to real world problems.

Prerequisite: Basic understanding of probability theory, linear algebra, multivariate calculus and multivariate probability theory.

CO1	Illustrate the concepts of regression and classification techniques **(Cognitive Knowledge Level: Apply)**
CO2	Demonstrate various unsupervised learning techniques **(Cognitive Knowledge Level: Apply)**
CO3	Choose suitable model parameters for different machine learning techniques and to evaluate a model performance **(Cognitive Knowledge Level: Apply)**
CO4	Explain the framework of PAC learning, basic concepts of VC dimension and non-uniform learnability **(Cognitive Knowledge Level: Understand)**
CO5	Construct Bayesian models for data and apply computational techniques to draw inferences **(Cognitive Knowledge Level: Apply)**
CO6	Illustrate the concepts of sampling algorithms, auto encoder, generative adversarial networks **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO 2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO1 2
CO1	✓	✓	✓	✓	✓	✓						✓
CO2	✓	✓	✓	✓	✓	✓						✓
CO3	✓	✓	✓	✓	✓	✓						✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓	✓							✓
CO6	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test1 (Percentage)	**Test2 (Percentage)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module -1 (Supervised Learning)

Overview of machine learning - supervised, semi-supervised, unsupervised learning, reinforcement learning Regression algorithms: least squares linear regression, gradient descent, closed form, normal equations, regularization techniques (LASSO, RIDGE), polynomial regression. Discriminative Methods - Logistic Regression, Decision Tree Learning. Generative Methods - Naive Bayes Classifier, Gaussian Discriminant Analysis (GDA).

Module -2 (Unsupervised Learning)

Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, K-medoids clustering, Gaussian mixture models: Expectation Maximization (EM) algorithm for Gaussian mixture model.

Module -3 (Practical aspects in machine learning)

Classification Performance measures - Precision, Recall, Accuracy, F-Measure, ROC, AUC, generalisation and overfitting, cross-validation, bias-variance tradeoff, error estimation, parameter and model selection. Ensemble Methods - Bagging, Boosting, Adaboost, Random Forests.

Module -4 (Statistical Learning Theory)

Models of learnability- learning in the limit, probably approximately correct (PAC) learning. Sample complexity- quantifying the number of examples needed to PAC learn, Computational complexity of training, Sample complexity for finite hypothesis spaces, PAC results for learning conjunctions, Sample complexity for infinite hypothesis spaces, Vapnik-Chervonenkis(VC) dimension.

Module -5 (Advanced Machine Learning Topics)

Graphical models - Bayesian belief networks, Markov random fields(MRFs), Inference on chains and factor graphs, inference on clique trees. Monte Carlo methods – Basic sampling algorithms, rejection sampling, importance sampling, Markov chain Monte Carlo(MCMC), Gibbs sampling. Variational methods. Auto Encoder, Variational AutoEncoder, Generative Adversarial Networks

Textbook

1. Christopher M. Bishop. Pattern recognition and machine learning. Springer 2006.
2. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
3. Mohammed J. Zaki and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, First South Asia edition, 2016.
4. Ian Goodfellow, Yoshua Bengio and Aaron Courville. Deep Learning. MIT Press 2016.
5. Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. Foundations of Machine Learning. Second edition. MIT Press 2018.
6. Tom Mitchell. Machine Learning. McGraw Hill 1997.
7. Richard O. Duda, Peter E . Hart, David G. Stork. Pattern classification, Second Edition. Wiley.
8. Jiawei Han, Micheline Kamber, Jian Pei. Data Mining Concepts and Techniques, Third Edition. Morgan Kaufmann.
9. David Foster. Generative Deep Learning - Teaching Machines to Paint, Write, Compose, and Play. O'Reilly Media, Inc., June 2019.

Reference Books

1. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. MIT Press 2012
2. Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. MIT Press 2005.

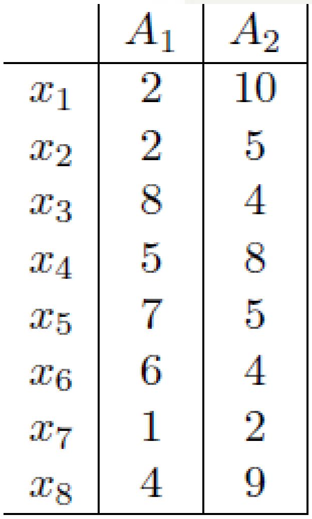
Sample Course Level Assessment Questions

Course Outcome1 (CO1):

1. Consider a naive Bayes classifier with 3 boolean input variables, $\mathbf{X_1}$, $\mathbf{X_2}$ and $\mathbf{X_3}$, and one boolean output, **Y**. How many parameters must be estimated to train such a naive Bayes classifier? How many parameters would have to be estimated to learn the above classifier if we do not make the naive Bayes conditional independence assumption?
2. Describe the ID3 algorithm. Is the order of attributes identical in all branches of the decision tree?
3. Explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.
4. Suppose that you are asked to perform linear regression to learn the function that outputs y, given the D-dimensional input x. You are given N independent data points, and that all the D attributes are linearly independent. Assuming that D is around 100, would you prefer the closed form solution or gradient descent to estimate the regressor?
5. Suppose you have a three class problem where class label $y \in 0, 1, 2$ and each training example X has 3 binary attributes $X_1, X_2, X_3 \in \mathbf{0, 1}$. How many parameters (probability distribution) do you need to know to classify an example using the Naive Bayes classifier?

Course Outcome 2(CO2):

1. Which similarity measure could be used to compare feature vectors of two images? Justify your answer.
2. Illustrate the strength and weakness of k-means algorithm.
3. Suppose you want to cluster the eight points shown below using **k**-means



	A_1	A_2
x_1	2	10
x_2	2	5
x_3	8	4
x_4	5	8
x_5	7	5
x_6	6	4
x_7	1	2
x_8	4	9

Assume that **k = 3** and that initially the points are assigned to clusters as follows:

C_1 = {x_1, x_2, x_3}, C_2 = {x_4, x_5, x_6}, C_3 = {x_7, x_8}. Apply the **k**-means algorithm until convergence, using the Manhattan distance.

4. Cluster the following eight points representing locations into three clusters: ***A_1(2, 10), A_2(2, 5), A_3(8, 4), A_4(5, 8), A_5(7, 5), A_6(6, 4), A_7(1, 2), A_8(4, 9)***.

   Initial cluster centers are: ***A_1(2, 10), A_4(5, 8)*** and ***A_7(1, 2)***.

   The distance function between two points ***a = (x_1, y_1)*** and ***b = (x_2, y_2)*** is defined as ***D(a, b)*** = $|x_2 - x_1| + |y_2 - y_1|$

   Use **k**-Means Algorithm to find the three cluster centers after the second iteration.

Course Outcome 3(CO3):

1. What is ensemble learning? Can ensemble learning using linear classifiers learn classification of linearly non-separable sets?
2. Describe boosting. What is the relation between boosting and ensemble learning?
3. Classifier A attains 100% accuracy on the training set and 70% accuracy on the test set. Classifier B attains 70% accuracy on the training set and 75% accuracy on the test set. Which one is a better classifier. Justify your answer.
4. What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why?
5. Suppose there are three classifiers A,B and C. The (FPR, TPR) measures of the three classifiers are as follows – A (0, 1), B (1, 1) , C (1,0.5). Which can be considered as a perfect classifier? Justify your answer.

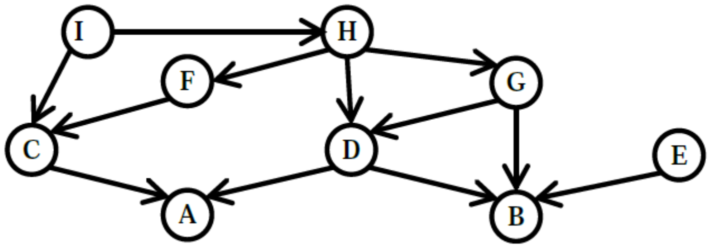
Course Outcome 4(CO4): .

1. A monotone conjunction is a conjunction of the variables such that no variable is negated. Show that the concept class of monotone conjunction is probably approximately correct (PAC)-learnable.
2. Consider a Boolean classification problem with ***n*** binary variables and a hypothesis space ***H***, where each hypothesis is a decision tree of depth 2, using only two variables. How many training examples, ***m*** suffice to assure that with probability at least 0.99, any consistent learner using ***H*** will output a hypothesis with true error at most 0.05
3. Show that the concept class C containing the set of all boolean functions on n variable is not PAC-learnable.

4. What is the Vapnik-Chervonenkis(VC)-dimension of a circle centered at the origin.
5. A hypothesis space that has a high VC dimension is good, bad, or neither? Explain in terms of both (a) richness or expressive power of the hypotheses, and (b) sample complexity.

Course Outcome 5(CO5):

1. Write down the factored conditional probability expression that corresponds to the graphical Bayesian Network shown below.



2. How do we learn the conditional probability tables(CPT) in Bayesian networks if information about some variables is missing? How are these variables called?

Course Outcome 6 (CO6):

1. Derive an algorithm using the inverse transform method to generate a random sample from the exponential distribution.
2. Explain the pros and cons of importance sampling versus rejection sampling.
3. Sketch the core idea of the Monte Carlo method. What is a sample? What is a direct sampling method? Why can't it be used directly to do any inference? What is rejection sampling? What is its major disadvantage?
4. Generative Adversarial Networks(GANs) include a generator and a discriminator. Sketch a basic GAN using those elements, a source of real images, and a source of randomness.
5. The word "adversarial" in the acronym for GANs suggests a two-player game. What are the two players, and what are their respective goals?

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 5**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION (HONORS), MONTH & YEAR

Course Code: CST 396

Course Name: Advanced Topics in Machine Learning

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1.	Suppose you have a dataset with m = 1000000 examples and n = 200000 features for each example. You want to use multivariate linear regression to fit the parameters to our data. Should you prefer gradient descent or the normal equation? Justify your answer.	
2.	Define Information gain? How is that different from Gain ratio? Give the advantage of using Gain ratio measure?	
3.	What is cluster analysis? Identify two applications where cluster analysis can be applied to multimedia data?	
4.	Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8): (i) Compute the Euclidean distance between the two objects. (ii) Compute the Manhattan distance between the two objects.	
5.	Suppose there are three classifiers A,B and C. The (FPR, TPR) measures of the three classifiers are as follows – A (0, 1), B (1, 1) , C (1,0.5). Which can be considered as a perfect classifier? Justify your answer.	
6.	How Bias-Variance Tradeoff affects machine learning algorithms?	
7.	For a particular learning task, if the requirement of error parameter ε changes from 0.1 to 0.01. How many more samples will be required for probably approximately correct(PAC) learning?	

8. Suppose we have a hypothesis set that labels all points inside an interval ***[a, b]*** as class 1. Find its Vapnik-Chervonenkis(VC)- dimension?

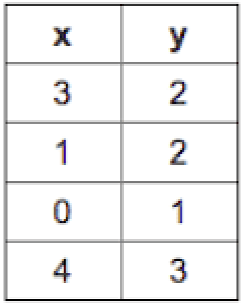
9. Given a density function ***f(x)***, the rejection sampling is a method that can generate data points from the density function ***f***. List the three steps to generate a random sample from f using rejection sampling.

10. How does the variational auto-encoder(VAE) architecture allow it to generate new data points, compared to auto-encoder, which cannot generate new data points? **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Consider the hypothesis for the linear regression h_θ (x) = θ_0 + θ_1x, and the cost function J(θ_0, θ_1) = 1/2m Σ $_{i=1}$ to m (h_θ $(x^{(i)})$ – $y^{(i)})^2$ where m is the number of training examples. Given the following set of training examples. **(5)**



x	y
3	2
1	2
0	1
4	3

Answer the following questions :

1) Find the value of h_θ (2) if θ_0= 0 and θ_1 = 1.5

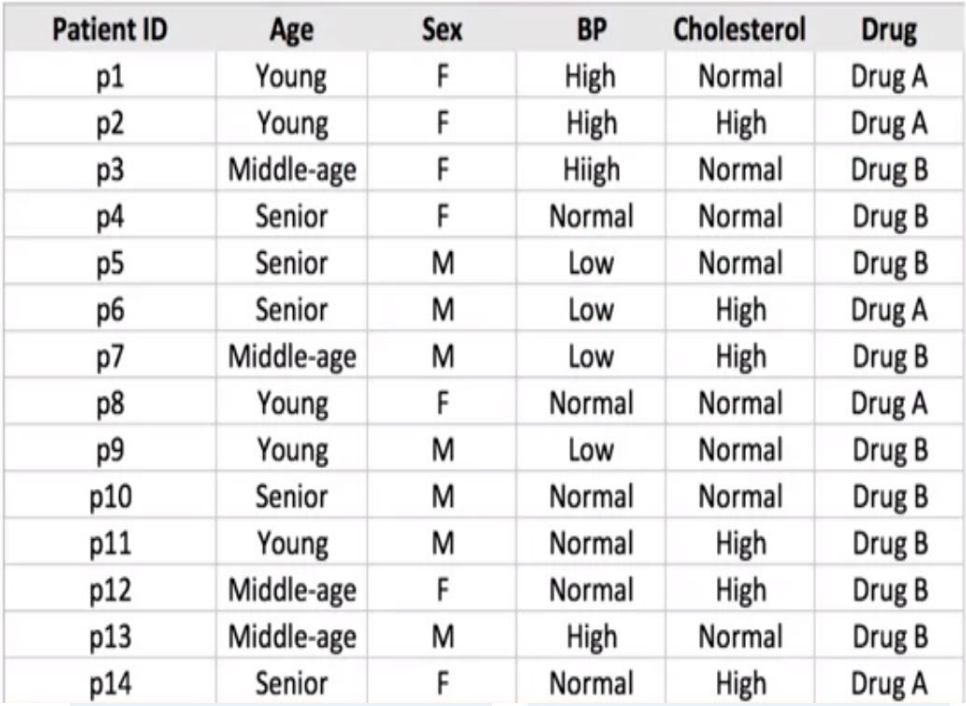
2) Find the value of J(0,1)

3) Suppose the value of J(θ_0, θ_1) = 0. What can be inferred from this.

(b) Write a gradient descent algorithm for multivariate regression? Compare the gradient and analytical solution to the multivariate regression? **(9)**

OR

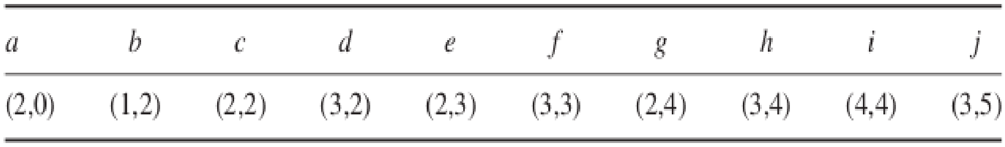
12. (a) Consider the collection of training samples (S) in the Figure given below. Drug is the target attribute which describes the Drug suggested for each patient. Find the value of the following . i) Gain(S, Sex) ii) Gain (S, Cholesterol) **(9)**



Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

(b) Explain how LASSO regression helps to reduce the overfitting problem? **(5)**

13. (a) Suppose that we have the following data: **(9)**



a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*
(2,0)	(1,2)	(2,2)	(3,2)	(2,3)	(3,3)	(2,4)	(3,4)	(4,4)	(3,5)

Identify the cluster by applying the k-means algorithm, with k = 2. Try using initial cluster centers as far apart as possible.

(b) Describe EM algorithm for Gaussian mixtures. **(5)**

OR

14. (a) Illustrate the strength and weakness of k-means in comparison with the k-medoids algorithm. **(4)**

(b) Suppose that we have the following data . Use single linkage Agglomerative clustering to identify the clusters. Draw the Dendogram. Use Euclidean distance measure **(10)**

	X	Y
P1	0.4	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

15. (a) Define Precision, Recall, Accuracy and F-measure? **(4)**

(b) What does it mean for a classifier to have a high precision but low recall? **(3)**

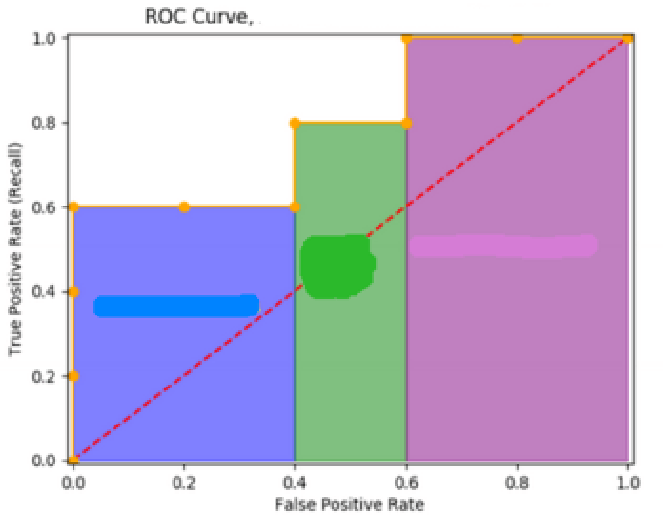
(c) Fill in the missing values in the accompanying three class confusion matrix. Given that model accuracy is 72% and classification error for class 2 is 20%. Find also the precision and recall for class1 **(7)**

		Predicted		
		Class 1	**Class 2**	**Class 3**
Actual	**Class 1**	14	2	5
	Class 2	?(X)	40	2
	Class 3	1	?(Y)	18

OR

16. (a) What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why? **(4)**

(b) Given the following ROC Curve? Find the AUC? **(3)**



(c) How does random forest classifier work? Why is a random forest better than a decision tree? **(7)**

17. (a) Show that the concept class Cn of the conjunction of boolean literals up to n variables is probably approximately correct(PAC)-learnable. **(8)**

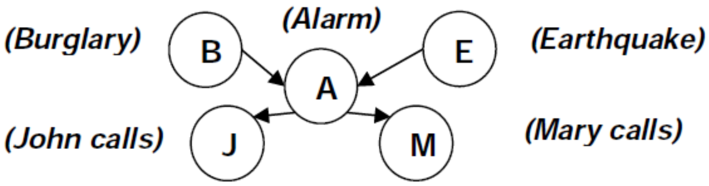
(b) Explain the concept of Vapnik-Chervonenkis (VC) dimension using shattering. How the number of training examples required to train the model is related to the VC dimension and what is its relation with training and test errors. **(6)**

OR

18. (a) Consider a Boolean classification problem with ***n*** binary variables and a hypothesis space ***H***, where each hypothesis is a decision tree of depth 1. How many training examples, ***m*** suffice to assure that with probability at least 0.99, any consistent learner using ***H*** will output a hypothesis with true error at most 0.05? **(7)**

(b) Prove that VC(H) ≤ log2 |H|, where H is a hypothesis space. (|H| denotes the **(7)** cardinality of the hypothesis space)

19. (a) Shown below is the Bayesian network corresponding to the Burglar Alarm problem, P(J | A) P(M | A) P(A | B, E) P(B) P(E). The probability tables show the probability that variable is True, e.g., P(M) means P(M = t). Find P(J = t ∧ M = f ∧ A = f ∧ B = f ∧ E = t). **(7)**



P(E)
.002

A	P(M)
t	.70
f	.01

B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

P(B)
.001

A	P(J)
t	.90
f	.05

(b) Derive an algorithm using the inverse transform method to generate a random sample from the distribution with density $f_X(x) = 3\,x^2,\ 0 < x < 1$. **(7)**

OR

20. (a) Draw the Bayesian Network that corresponds to this conditional probability: ***P(A | B,C,E) P(B | D,E) P(C | F,H) P(D | G) P(E| G,H) P(F | H) P(G) P(H)*** **(6)**

(b) What is effective sample size (ESS)? Why is a large ESS necessary but not sufficient for good MCMC mixing? **(3)**

(c) Describe the overall Gibbs sampling algorithm briefly **(5)**

Teaching Plan

Module 1 : (Supervised Learning)(10 hours)		
1.1	Supervised, semi-supervised, unsupervised learning, reinforcement learning (TB 2: Ch 1)	1 hour
1.2	Least squares linear regression (TB 2: Section 2.6)	1 hour
1.3	Gradient descent, closed form, normal equations (TB 2: Section 5.8)	1 hour
1.4	Regularization techniques (LASSO, RIDGE) (TB 4: Section 7.1)	1 hour
1.5	Polynomial regression (TB 2: Section 2.6)	1 hour
1.6	Logistic Regression (TB 6: Section 3.3)	1 hour
1.7	Decision Tree Learning (ID3) (TB 8: Section 8.2)	1 hour
1.8	Decision Tree Learning (C4.5) (TB 8: Section 8.2)	1 hour
1.9	Naive Bayes Classifier (TB 8: Section 8.3)	1 hour
1.10	Gaussian Discriminant Analysis (GDA) (TB 7: Section 5.2,5.3)	1 hour
Module 2 : (Unsupervised Learning)(8 hours)		
2.1	Similarity measures (TB 8: Section 2.4)	1 hour
2.2	Hierarchical Agglomerative Clustering (TB 3: Chapter 14)	1 hour
2.3	Hierarchical Agglomerative Clustering (TB 3: Chapter 14)	
2.4	K-means partitional clustering (TB 3: Chapter 13)	1 hour
2.5	K-medoids partitional clustering	
2.6	Gaussian mixture models (TB 3: Chapter 13)	1 hour
2.7	Expectation Maximization (EM) algorithm for Gaussian mixture model Lecture-1 (TB 3: Chapter 13)	1 hour
2.8	Expectation Maximization (EM) algorithm for Gaussian mixture model Lecture-2 (TB 3: Chapter 13)	1 hour
Module 3 : (Practical aspects in machine learning) (6 hours)		

3.1	Precision, Recall, Accuracy, F-Measure, ROC, AUC (TB8.5/TB 3: Chapter 22.1)	1 hour
3.2	Generalisation and overfitting, cross-validation (TB 2: Section 2.7,4.8)	1 hour
3.3	Bias-variance tradeoff (TB 2: Chapter 22.3)	1 hour
3.4	Error estimation, parameter and model selection (TB 3: Chapter 8.5)	1 hour
3.5	Bagging, Boosting (TB 8: Chapter 8.6)	1 hour
3.6	Adaboost, Random Forests (TB 8: Chapter 8.6)	1 hour
Module 4 : (Statistical Learning Theory) (TB 5 – Chapter 2, 3.3)(7 hours)		
4.1	Learning in the limit, probably approximately correct (PAC) learning	1 hour
4.2	Quantifying the number of examples needed to PAC learn	1 hour
4.3	Computational complexity of training	1 hour
4.4	Sample complexity for finite hypothesis spaces	1 hour
4.5	PAC results for learning conjunctions	1 hour
4.6	Sample complexity for infinite hypothesis spaces	1 hour
4.7	Vapnik-Chervonenkis(VC) dimension	1 hour
Module 5 : (Advanced Machine Learning Topics) (13 hours)		
5.1	Bayesian belief networks (TB 1 – Chapter 8)	1 hour
5.2	Markov random fields (TB 1 – Chapter 8)	1 hour
5.3	Inference on chains and factor graphs (TB 1 – Chapter 8)	1 hour
5.4	Inference on clique trees (TB 1 – Chapter 8)	1 hour
5.5	Basic sampling algorithms (TB 1 – Chapter 11)	1 hour
5.6	Rejection sampling (TB 1 – Chapter 11)	1 hour
5.7	Importance sampling (TB 1 – Chapter 11)	1 hour
5.8	Markov chain Monte Carlo(MCMC) (TB 1 – Chapter 11)	1 hour
5.9	Gibbs sampling (TB 1 – Chapter 11)	1 hour

5.10	Variational method (TB 1 – Chapter 10)	1 hour
5.11	Auto Encoder (TB 4 – Chapter 14)	1 hour
5.12	Variational AutoEncoder (TB 9 – Chapter 3)	1 hour
5.13	Generative Adversarial Networks (TB 9 – Chapter 4)	1 hour

CST 398	THEORY OF COMPUTABILITY AND COMPLEXITY	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This is a theoretical course in computer science to enable the learners to know the fundamentals of computability and complexity theories. It covers the notions of computability/decidability, the process of reduction to prove decidability/undecidability and the classification of problems into class P, class NP and class NP Complete based on the time complexity of solving the problems. This course helps the learner to identify whether a real life problem is decidable/undecidable and also to classify a decidable problem into tractable or intractable, based on the time complexity class it belongs.

Prerequisite: Sound knowledge in Data Structures and Formal Languages and Automata Theory.

Mapping of course outcomes with program outcomes

CO1	Illustrate relative computing powers of Finite State Automata, Push Down Automata, Linear Bounded Automata and Turing Machines.**(Cognitive Knowledge Level: Apply)**
CO2	Prove that a given language is undecidable/not semi-decidable by using the reduction process.**(Cognitive Knowledge Level: Apply)**
CO3	Describe the time complexity of a given problem as a function of the number of steps required by a Turing machine to solve it. **(Cognitive Knowledge Level: Understand)**
CO4	Utilize polynomial time reduction to prove that a given problem is NP Complete. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓

CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests1&2)	**25 marks**
Continuous AssessmentAssignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to Formal Language Theory and Regular Languages)

Finite State Automata, Push Down Automata, Linear Bounded Automata, Turing Machines, Recursive Languages, Recursively Enumerable Languages, Universal Turing Machine, Enumeration Machine, Two Counter Machine.

Module– 2 (Undecidability)

Halting Problem, Language representation of a problem, Reduction - applications, Rice’s First and Second Theorem with proof.

Module - 3 (Overview of Complexity Classes)

Measuring time complexity, Asymptotic notations - Big O and small-o, Analysing algorithms, Complexity relationship among models. Complexity classes- Class P, example problems in class P, Class NP, Polynomial time verification, example problems in class NP.

Module- 4 (NP Completeness)

Satisfiability problem, Polynomial time reducibility, Overview of Graphs, NP Complete Problems, Cook-Levin theorem (SAT is NP Complete).

Module- 5 (More NP Complete Problems)

CLIQUE, Vertex Cover and Hamiltonian path with proof of correctness of NP Completeness.

Text Books

1. Dexter C. Kozen, Automata and Computability, Springer (1999)
2. Michael Sipser, Introduction to the Theory of Computation, Second Edition

Reference Books

1. Douglas B. West, Introduction to Graph Theory, Second Edition

Course Level Assessment Questions

Course Outcome1 (CO1):

Identify the class of the following languages in Chomsky Hierarchy:

1. Design a Finite State Automaton for the language $L = \{axb | x \in \{a, b\}^*\}$
2. Design a Push Down Automaton for the language $L = \{a^n b^n | n \geq 0\}$
3. Design a Linear Bounded Automaton for the language $L = \{a^n b^n c^n | n \geq 0\}$
4. Design a Turing Machine for the language $L = \{ww | w \in \{a, b\}^*\}$

Course Outcome 2(CO2):

Without using Rice's Theorem prove that neither the set FIN (refer Text Book 1) nor its complement is recursively enumerable.

Course Outcome 3(CO3):

Show that the language $L = \{a^n b^n | n \geq 0\}$ can be decided by a deterministic Turing Machine in quadratic time.

Course Outcome 4(CO4): .

Using polynomial time reduction, prove that SUBSET-SUM (refer Text Book 2) problem is NP Complete.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH. DEGREE EXAMINATION(HONORS), MONTH & YEAR

Course Code: CST 398

Course Name: Theory of Computability and Complexity

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Design a Deterministic Finite state Automaton (DFA) for the language:$L = \{x \in \{0,1\}^* | x does\ not\ contain\ consecutive\ zeros\}$.

2. Design a Pushdown Automaton (PDA) for the language $L = \{a^m b^n | m \geq 0 and n > m\}$ (no explanation is required, just list the transitions in the PDA).

3. List any *six* undecidable problems.

4. Illustrate how a problem can be represented as a language.

5. Describe the term time complexity class.

6. Define the term polynomial time verification. Describe its usage.

7. Define the term polynomial time reduction. Describe its usage.

8. Define vertex cover. Illustrate with the help of a graph.

9. Illustrate CLIQUE problem with an example.

10. State Hamiltonian path problem. Show an example.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Why the family of languages recognized by Turing machines is called Recursively Enumerable? Explain the working of an enumeration machine. **(8)**

(b) Illustrate the functioning of a Universal Turing Machine. **(6)**

OR

12. (a) Illustrate the functioning of a two counter machine. **(4)**

(b) Prove that Turing Machines and Two Counter Machines are equivalent in power. **(10)**

13. (a) Prove using Diagonalisation that halting problem is undecidable. **(8)**

(b) Prove Using reduction that state entry problem of Turing machines is undecidable. **(6)**

OR

14. (a) State and prove Rice's first theorem. **(8)**

(b) Prove Using reduction that whether a Turing Machine accepts empty string (or null string) is undecidable. **(6)**

15. (a) Show that the language $L = \{a^n b^n | n \geq 0\}$ can be decided by a deterministic Turing Machine in $O(n \cdot logn)$ time. **(7)**

(b) Let $t(n)$ be a function, where $n \in \mathbb{N}$ and $t(n) \geq n$. Then, prove that every $t(n)$ time nondeterministic single-tape Turing machine has an equivalent $2^{O(t(n))}$ time single-tape deterministic Turing machine. **(7)**

OR

16. (a) Prove that every context free language is a member of class P. **(8)**

(b) When is a problem said to be in class NP? **(6)**
Prove that Hamiltonian path problem of a directed graph is in class NP.

17. (a) Define Independent set in a graph. Prove that a graph G of n vertices with an independent set of size k contains a vertex cover of size $n - k$. **(8)**

(b) Define the complexity class NP Complete. Explain the significance of an NP Complete problem. **(6)**

OR

18. (a) Define the complement of a graph. Prove that the complement of a graph G of n vertices with a CLIQUE of size k contains an independent set of size k. **(7)**

(b) What is satisfiability problem. Prove that satisfiability problem is in class NP. **(7)**

19. (a) Illustrate Hamiltonian path in a Graph. **(4)**

(b) Prove that Hamiltonian path problem is in the class NP Complete. **(10)**

OR

20. (a) Prove that Vertex Cover problem is in the class NP Complete. **(8)**

(b) Why is it useful to identify that a problem is in the class NP Complete? **(6)**

Teaching Plan

No	Contents	No. of Lecture Hours (45 hrs)
Module-1(Overview of Automata Theory) (10 hours)		
1.1	Finite State Automata	1 hour
1.2	Push Down Automata	1 hour
1.3	Linear Bounded Automata	1 hour
1.4	Turing Machines	1 hour
1.5	Recursive Languages	1 hour
1.6	Recursively Enumerable Languages	1 hour
1.7	Universal Turing Machine	1 hour
1.8	Enumeration Machine	1 hour
1.9	Two Counter Machines	1 hour
1.10	Proof that two Counter Machines and Turing machines are equivalent	1 hour
Module-2 (Undecidability) (10 hours)		
2.1	Halting problem of Turing machine	1 hour
2.2	Proof of undecidability of Halting Problem	1 hour
2.3	Language representation of a problem	1 hour
2.4	Reduction	1 hour
2.5	Applications of reduction - Lecture 1	1 hour

2.6	Applications of reduction - Lecture 2	1 hour
2.7	Rice's First Theorem	1 hour
2.8	Proof of Rice's First Theorem	1 hour
2.9	Rice's Second Theorem	1 hour
2.10	Proof of Rice's Second Theorem	1 hour
Module-3 (Overview of Complexity Classes) (10 hours)		
3.1	Measuring time complexity, Asymptotic notations - Big O and small-o	1 hour
3.2	Analysing algorithms - time complexity class	1 hour
3.3	Complexity relationship among models - Single tape Turing Machine	1 hour
3.4	Multi-tape Turing Machine, Nondeterministic Turing Machine	1 hour
3.5	Class P	1 hour
3.6	Example problems in Class P	1 hour
3.7	Class NP	1 hour
3.8	Polynomial time verification	1 hour
3.9	Example problems in Class NP - Lecture 1	1 hour
3.10	Example problems in Class NP - Lecture 2	1 hour
Module-4 (NP Completeness) (9 hours)		
4.1	Satisfiability problem	1 hour
4.2	Polynomial time reducibility	1 hour
4.3	Overview of Graphs, CLIQUE, INDEPENDENT SET	1 hour
4.4	Vertex Cover	1 hour
4.5	Reducing 3SAT problem to CLIQUE - Lecture 1	1 hour
4.6	Reducing 3SAT problem to CLIQUE - Lecture 2	1 hour
4.7	NP Complete Problems	1 hour
4.8	Cook-Levin theorem, Proof - Lecture 1	1 hour
4.9	Proof - Lecture 2	1 hour
Module-5 (More NP Complete Problems) (6 hours)		

5.1	CLIQUE	1 hour
5.2	Vertex Cover - Lecture 1	1 hour
5.3	Vertex Cover - Lecture 2	1 hour
5.4	Hamiltonian path - Lecture 1	1 hour
5.5	Hamiltonian path - Lecture 2	1 hour
5.6	Hamiltonian path - Lecture 3	1 hour

SEMESTER VII

CST401	**ARTIFICIAL INTELLIGENCE**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**
		PCC	**2**	**1**	**0**	**3**

Preamble: The course aims to introduce the fundamental principles of intelligent systems to students. This involves ideas about the characteristics of intelligent systems, knowledge representation schemes, logic and inference mechanisms. The course helps the learner to understand the design of self learning systems along with some of their typical applications in the emerging scenario where the business world is being transformed by the progress made in machine learning.

Prerequisite : NIL

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Explain the fundamental concepts of intelligent systems and their architecture. (**Cognitive Knowledge Level: Understanding)**
CO2	Illustrate uninformed and informed search techniques for problem solving in intelligent systems. **(Cognitive Knowledge Level: Understanding)**
CO3	Solve Constraint Satisfaction Problems using search techniques. (**Cognitive Knowledge Level: Apply**)
CO4	Represent AI domain knowledge using logic systems and use inference techniques for reasoning in intelligent systems. (**Cognitive Knowledge Level: Apply**)
CO5	Illustrate different types of learning techniques used in intelligent systems **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											
CO2	✓	✓										✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓			✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	60	30	40
Apply	20	40	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of SeriesTests1& 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Introduction)
Introduction – What is Artificial Intelligence(AI) ? The Foundations of AI, History of AI, Applications of AI. Intelligent Agents – Agents and Environments, Good behavior: The concept of rationality, nature of Environments, Structure of Agents.

Module – 2 (Problem Solving)
Solving Problems by searching-Problem solving Agents, Example problems, Searching for solutions, Uninformed search strategies, Informed search strategies, Heuristic functions.

Module - 3 (Search in Complex environments)
Adversarial search - Games, Optimal decisions in games, The Minimax algorithm, Alpha-Beta pruning. Constraint Satisfaction Problems – Defining CSP, Constraint Propagation- inference in CSPs, Backtracking search for CSPs, Structure of CSP problems.

Module - 4 (Knowledge Representation and Reasoning)
Logical Agents – Knowledge based agents, Logic, Propositional Logic, Propositional Theorem proving, Agents based on Propositional Logic. First Order Predicate Logic – Syntax and Semantics of First Order Logic, Using First Order Logic, Knowledge representation in First Order Logic. Inference in First Order Logic – Propositional Vs First Order inference, Unification and Lifting, Forward chaining, Backward chaining, Resolution.

Module - 5 (Machine Learning)

Learning from Examples – Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and choosing the best hypothesis, Regression and classification with Linear models.

Text Book

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3rd Edition. Prentice Hall.

References

1. Nilsson N.J., Artificial Intelligence - A New Synthesis, Harcourt Asia Pvt. Ltd.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain about the basic types of agent programs in intelligent systems.
2. For the following activities, give a PEAS description of the task environment and characterize it in terms of the task environment properties.

   a) Playing soccer.

   b) Bidding on an item at an auction.

Course Outcome 2 (CO2):

1. Differentiate between uninformed and informed search strategies in intelligent systems.
2. Illustrate the working of Minimax search procedure.

Course Outcome 3 (CO3):

1. Solve the following crypt arithmetic problem by hand, using the strategy of backtracking with forward checking and the MRV & least-constraining-value heuristics.

$$\begin{array}{r} T\ W\ O \\ +\ T\ W\ O \\ \hline F\ O\ U\ R \end{array}$$

Course Outcome 4 (CO4):

1. Prove, or find a counter example to, the following assertion:

   If α |= γ or β |= γ (or both) then (α ∧ β) |= γ
2. For each pair of atomic sentences, find the most general unifier if it exists:

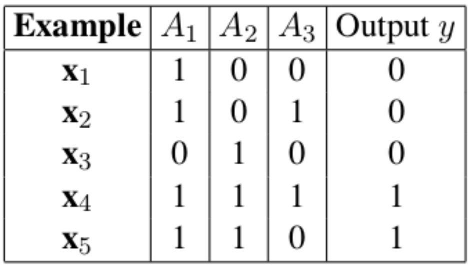
   a) P (A, B, B), P (x, y, z).

   b) Q(y, G(A, B)), Q(G(x, x), y).

Course Outcome 5 (CO5):

1. Consider the following data set comprised of three binary input attributes (A1 , A2 , and

A3) and one binary output.



Example	A_1	A_2	A_3	Output y
$\mathbf{x}_1$	1	0	0	0
$\mathbf{x}_2$	1	0	1	0
$\mathbf{x}_3$	0	1	0	0
$\mathbf{x}_4$	1	1	1	1
$\mathbf{x}_5$	1	1	0	1

Use the DECISION-TREE-LEARNING algorithm to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

2. What is multivariate linear regression? Explain.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST401

Course Name: Artificial Intelligence

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1 What is a rational agent? Explain.

2 Describe any two ways to represent states and the transitions between them in agent programs.

3 Differentiate between informed search and uninformed search.

4 Define heuristic function? Give two examples.

5 What are the components of a Constraint Satisfaction Problem? Illustrate with an example.

6 Formulate the following problem as a CSP. Class scheduling: There is a fixed number of professors and classrooms, a list of classes to be offered, and a list of possible time slots for classes. Each professor has a set of classes that he or she can teach.

7 What is a knowledge based agent? How does it work?

8. Represent the following assertion in propositional logic:
"A person who is radical (R) is electable (E) if he/she is conservative (C), but otherwise is not electable."

9 Describe the various forms of learning?

10 State and explain Ockham's razor principle **(10x3=30**]

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the structure Goal-based agents and Utility-based agents with the help of diagrams. **(8)**

(b) For the following activities, give a PEAS description of the task environment and characterize it in terms of the task environment properties. **(6)**

a) Playing soccer

b) Bidding on an item at an auction.

OR

12. (a) Explain the structure Simple reflex agents and Model-based reflex agents with the help of diagrams. **(8)**

(b) Discuss about any five applications of AI. **(6)**

13. (a) Explain Best First Search algorithm. How does it implement heuristic search? **(6)**

(b) Describe any four uninformed search strategies. **(8)**

OR

14. (a) Write and explain A* search algorithm. **(6)**

(b) Explain the components of a well defined AI problem? Write the standard formulation of 8-puzzle problem. **(8)**

15. (a) (a) Solve the following crypt arithmetic problem by hand, using the strategy of backtracking with forward checking and the MRV and least-constraining-value heuristics. **(8)**

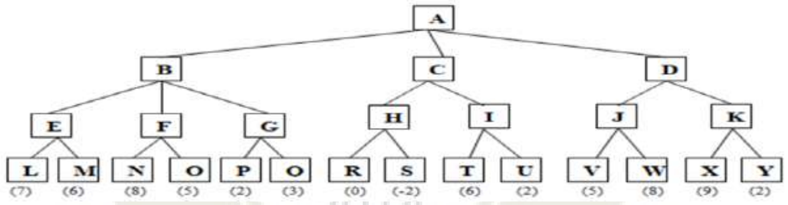
$$\begin{array}{r} T\ W\ O \\ +\ T\ W\ O \\ \hline F\ O\ U\ R \end{array}$$

(b) What is local consistency in CSP constraint propagation? Explain different types local consistencies. **(6)**

OR

16. (a) Illustrate the use of alpha-beta pruning in games. **(6)**

(b) Consider the following game tree in which static evaluation score are all from the players point of view: static evaluation score range is (+10 to -10) **(8)**



Suppose the first player is the maximizing player. What move should be chosen? Justify your answer.

17. (a) Convert the following sentences into first order logic: **(6)**

Everyone who loves all animals is loved by someone.

Anyone who kills an animal is loved by no one.

Jack loves all animals.

Either Jack or Curiosity killed the cat, who is named Tuna.

Did Curiosity kill the cat?

(b) Give a resolution proof to answer the question "Did Curiosity kill the cat? " **(8)**

OR

18. (a) Prove or find a counter example to the following assertion in propositional logic: **(6)**

If α |= (β ∧γ) then α |= β and α |= γ.

(b) For each pair of atomic sentences, give the most general unifier if it exists: **(8)**

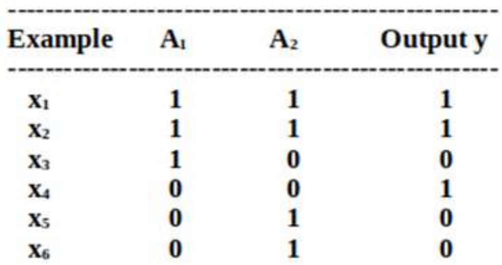
Older (Father (y), y), Older (Father (x), John).

19. (a) How is best hypothesis selected from alternatives? **(8)**

(b) Explain Univariate Linear Regression. **(6)**

OR

20. (a) Consider the following data set comprised of two binary input attributes (A1 and A2) and one binary output. **(8)**



Example	A_1	A_2	Output y
x_1	1	1	1
x_2	1	1	1
x_3	1	0	0
x_4	0	0	1
x_5	0	1	0
x_6	0	1	0

Use the DECISION-TREE-LEARNING algorithm to learn a decision tree for these data. Show the computations made to determine the attribute to split at each node.

(b) Explain Linear classification with logistic regression **(6)**

Teaching Plan

No	Contents	No of Lecture Hrs (36)
Module – 1 (Introduction) (7 hrs)		
1.1	Introduction, What is Artificial Intelligence(AI)?	1
1.2	The foundations of AI, The history of AI	1
1.3	Applications of AI	1
1.4	Intelligent Agents – Agents and Environments	1
1.5	Good behavior: The concept of rationality	1
1.6	The nature of Environments	1
1.7	The structure of Agents	1
Module - 2 (Problem Solving by searching) (7 hrs)		
2.1	Solving Problems by searching-Problem solving Agents	1
2.2	Illustration of the problem solving process by agents	1
2.3	Searching for solutions	1
2.4	Uninformed search strategies:BFS, Uniform-cost search, DFS, Depth-limited search, Iterative deepening depth-first search	1
2.5	Informed search strategies: Best First search	1
2.6	Informed search strategies: A* Search	1
2.7	Heuristic functions	1
Module - 3 (Problem Solving in complex environments) (7 hrs)		
3.1	Adversarial search - Games	1
3.2	Optimal decisions in games, The Minimax algorithm	1
3.3	Alpha-Beta pruning	1
3.4	Constraint Satisfaction Problems – Defining CSP	1
3.5	Constraint Propagation- inference in CSPs	1
3.6	Backtracking search for CSPs	1
3.7	The structure of problems	1

Module - 4 (Knowledge Representation and Reasoning) (9 hrs)		
4.1	Logical Agents – Knowledge based agents and logic	1
4.2	Propositional Logic	1
4.3	Propositional Theorem proving	1
4.4	Agents based on Propositional Logic	1
4.5	First Order Predicate Logic – Syntax and Semantics of First Order Logic	1
4.6	Using First Order Logic, Knowledge representation in First Order Logic	1
4.7	Inference in First Order Logic – Propositional Vs First Order inference, Unification and Lifting	1
4.8	Forward chaining, Backward chaining	1
4.9	Resolution	1
Module - 5 (Machine Learning)(6 hrs)		
5.1	Learning from Examples – Forms of Learning	1
5.2	Supervised Learning	1
5.3	Learning Decision Trees	1
5.4	Generaliztion and overfitting	1
5.5	Evaluating and choosing the best hypothesis	1
5.6	Regression and classification with Linear models.	1

CSL411	COMPILER LAB	CATEGORY	L	T	P	CREDIT
		PCC	0	0	3	2

Preamble: This course aims to offer students hands-on experience on compiler design concepts. Students will be able to familiarize with tools such as LEX and YACC and automate different phases of a compiler. This course helps the learners to enhance the capability to design and implement a compiler.

Prerequisite: A sound knowledge in C programming, Data Structures, Formal languages and Automata Theory and Compiler design.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Implement lexical analyzer using the tool LEX. **(Cognitive Knowledge Level: Apply)**
CO 2	Implement Syntax analyzer using the tool YACC. **(Cognitive Knowledge Level: Apply)**
CO 3	Design NFA and DFA for a problem and write programs to perform operations on it. **(Cognitive Knowledge Level: Apply)**
CO 4	Design and Implement Top-Down parsers. **(Cognitive Knowledge Level: Apply)**
CO 5	Design and Implement Bottom-Up parsers. **(Cognitive Knowledge Level: Apply)**
CO 6	Implement intermediate code for expressions. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	✓	✓	✓	✓	✓			✓		✓		✓
CO 2	✓	✓	✓	✓	✓			✓		✓		✓
CO 3	✓	✓	✓	✓				✓		✓		✓
CO 4	✓	✓	✓	✓				✓		✓		✓
CO 5	✓	✓	✓	✓				✓		✓		✓
CO 6	✓	✓	✓	✓				✓		✓		✓

Assessment Pattern

Bloom's Category	Continuous Assessment Test %	End Semester Examination %
Remember	20	20
Understand	20	20
Apply	60	60
Analyze		
Evaluate		
Create		

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	75	75	3 hours

Continuous Internal Evaluation Pattern:

Attendance	:	15 marks
Continuous Evaluation in Lab	:	30 marks
Continuous Assessment Test	:	15 marks
Viva-voce	:	15 marks

Internal Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks which will be converted out of 15 while calculating Internal Evaluation marks.

End Semester Examination Pattern: The marks will be distributed as Algorithm 30 marks, Program 20 marks, Output 20 marks and Viva 30 marks. Total 100 marks will be converted out of 75 for End Semester Examination.

Operating System to Use in Lab	**:** Linux
Compiler/Software to Use in Lab	**:** gcc, lex, yacc
Programming Language to Use in Lab	**:** Ansi C

Fair Lab Record:

All Students attending the Compiler Lab should have a Fair Record. The fair record should be produced in the University Lab Examination. Every experiment conducted in the lab should be noted in the fair record. For every experiment in the fair record the right hand page should contain Experiment Heading, Experiment Number, Date of Experiment, Aim of Experiment, Details of Experiment including algorithm and Result of Experiment. The left hand page should contain a print out of the code used for the experiment and sample output obtained for a set of input.

SYLLABUS

1. Implementation of lexical analyzer using the tool LEX.
2. Implementation of Syntax analyzer using the tool YACC.
3. Application problems using NFA and DFA.
4. Implement Top-Down Parser.
5. Implement Bottom-up parser.
6. Simulation of code optimization Techniques.
7. Implement Intermediate code generation for simple expressions.
8. Implement the back end of the compiler.

PRACTICE QUESTIONS

List of Exercises/Experiments:

1. Design and implement a lexical analyzer using C language to recognize all valid tokens in the input program. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments.
2. Implement a Lexical Analyzer for a given program using Lex Tool.
3. Write a lex program to display the number of lines, words and characters in an input text.
4. Write a LEX Program to convert the substring *abc* to *ABC* from the given input string.
5. Write a lex program to find out total number of vowels and consonants from the given input sting.
6. Generate a YACC specification to recognize a valid arithmetic expression that uses operators +, – , *,/ and parenthesis.

7. Generate a YACC specification to recognize a valid identifier which starts with a letter followed by any number of letters or digits.
8. Implementation of Calculator using LEX and YACC
9. Convert the BNF rules into YACC form and write code to generate abstract syntax tree.
10. Write a program to find ε – closure of all states of any given NFA with ε transition.
11. Write a program to convert NFA with ε transition to NFA without ε transition.
12. Write a program to convert NFA to DFA.
13. Write a program to minimize any given DFA.
14. Write a program to find First and Follow of any given grammar.
15. Design and implement a recursive descent parser for a given grammar.
16. Construct a Shift Reduce Parser for a given language.
17. Write a program to perform constant propagation.
18. Implement Intermediate code generation for simple expressions.
19. Implement the back end of the compiler which takes the three address code and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, jump etc.

CSQ413	SEMINAR	CATEGORY	L	T	P	CREDIT
		PWS	0	0	3	2

Preamble: The course ‘Seminar’ is intended to enable a B.Tech graduate to read, understand, present and prepare report about an academic document. The learner shall search in the literature including peer reviewed journals, conference, books, project reports etc., and identify an appropriate paper/thesis/report in her/his area of interest, in consultation with her/his seminar guide. This course can help the learner to experience how a presentation can be made about a selected academic document and also empower her/him to prepare a technical report.

Course Objectives:

- To do literature survey in a selected area ofstudy.
- To understand an academic document from the literate and to give a presentation about it.
- To prepare a technical report.

Course Outcomes [COs] : After successful completion of the course, the students will be able to:

CO1	Identify academic documents from the literature which are related to her/his areas of interest (Cognitive knowledge level: **Apply**).
CO2	Read and apprehend an academic document from the literature which is related to her/ his areas of interest (Cognitive knowledge level: **Analyze**).
CO3	Prepare a presentation about an academic document (Cognitive knowledge level: **Create**).
CO4	Give a presentation about an academic document (Cognitive knowledge level: **Apply**).
CO5	Prepare a technical report (Cognitive knowledge level: **Create**).

Mapping of course outcomes with program outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	1		2	1					3
CO2	3	3	2	3		2	1					3
CO3	3	2			3			1		2		3
CO4	3				2			1		3		3
CO5	3	3	3	3	2	2		2		3		3

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

General Guidelines

- The Department shall form an Internal Evaluation Committee (IEC) for the seminar with academic coordinator for that program as the Chairperson/Chairman and seminar coordinator & seminar guide as members. During the seminar presentation of a student, all members of IEC shall be present.

- Formation of IEC and guide allotment shall be completed within a week after the University examination (or last working day) of the previous semester.

- Guide shall provide required input to their students regarding the selection of topic/ paper.

- Choosing a seminar topic: The topic for a UG seminar should be current and broad based rather than a very specific research work. It's advisable to choose a topic for the Seminar to be closely linked to the final year project area. Every member of the project team could choose or be assigned Seminar topics that covers various aspects linked to the Project area.

- A topic/paper relevant to the discipline shall be selected by the student during the semester break.

- Topic/Paper shall be finalized in the first week of the semester and shall be submitted to the IEC.

- The IEC shall approve the selected topic/paper by the second week of the semester.

- Accurate references from genuine peer reviewed published material to be given in the report and to be verified.

Evaluation pattern

Total marks: 100, only CIE, minimum required to pass 50

Seminar Guide: 20 marks (Background Knowledge – 10 (The guide shall give deserving marks for a candidate based on the candidate's background knowledge about the topic selected), Relevance of the paper/topic selected – 10).

Seminar Coordinator: 20 marks (Seminar Diary – 10 (Each student shall maintain a seminar diary and the guide shall monitor the progress of the seminar work on a weekly basis and shall approve the entries in the seminar diary during the weekly meeting with the student), Attendance – 10).

Presentation: 40 marks to be awarded by the IEC (Clarity of presentation – 10, Interactions – 10 (to be based on the candidate's ability to answer questions during the interactive session of her/his presentation), Overall participation – 10 (to be given based on her/his involvement during interactive sessions of presentations by other students), Quality of the slides – 10).

Report: 20 marks to be awarded by the IEC (check for technical content, overall quality, templates followed, adequacy of references etc.).

CSD415	PROJECT PHASE I	CATEGORY	L	T	P	CREDIT
		PWS	0	0	6	2

Preamble: The course ‘Project Work’ is mainly intended to evoke the innovation and invention skills in a student. The course will provide an opportunity to synthesize and apply the knowledge and analytical skills learned, to be developed as a prototype or simulation. The project extends to 2 semesters and will be evaluated in the 7th and 8th semester separately, based on the achieved objectives. One third of the project credits shall be completed in 7th semester and two third in 8th semester. It is recommended that the projects may be finalized in the thrust areas of the respective engineering stream or as interdisciplinary projects. Importance should be given to address societal problems and developing indigenous technologies.

Course Objectives

- To apply engineering knowledge in practical problem solving.
- To foster innovation in design of products, processes or systems.
- To develop creative thinking in finding viable solutions to engineering problems.

Course Outcomes [COs] : After successful completion of the course, the students will be able to:

CO1	Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: **Apply**).
CO2	Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: **Apply**).
CO3	Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: **Apply**).
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: **Apply**).
CO5	Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: **Analyze**).
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: **Apply**).

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	1	2	2	2	1	1	1	1	2
CO2	2	2	2		1	3	3	1	1		1	1
CO3									3	2	2	1
CO4					2			3	2	2	3	2
CO5	2	3	3	1	2							1
CO6					2			2	2	3	1	1

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

PROJECT PHASE I

Phase 1 Target

- Literature study/survey of published literature on the assigned topic
- Formulation of objectives
- Formulation of hypothesis/ design/ methodology
- Formulation of work plan and task allocation.
- Block level design documentation
- Seeking project funds from various agencies
- Preliminary Analysis/Modeling/Simulation/Experiment/Design/Feasibility study
- Preparation of Phase 1 report

Evaluation Guidelines & Rubrics

Total: 100 marks (Minimum required to pass: 50 marks).

- Project progress evaluation by guide: 30 Marks.
- Interim evaluation by the Evaluation Committee: 20 Marks.
- Final Evaluation by the Evaluation Committee: 30 Marks.
- Project Phase - I Report (By Evaluation Committee): 20 Marks.

(The evaluation committee comprises HoD or a senior faculty member, Project coordinator and project supervisor).

Evaluation by the Guide

The guide/supervisor shall monitor the progress being carried out by the project groups on a regular basis. In case it is found that progress is unsatisfactory it shall be reported to the Department Evaluation Committee for necessary action. The presence of each student in the group and their involvement in all stages of execution of the project shall be ensured by the guide. Project evaluation by the guide: 30 Marks. This mark shall be awarded to the students in his/her group by considering the following aspects:

Topic Selection: innovativeness, social relevance etc. (2)

Problem definition: Identification of the social, environmental and ethical issues of the project problem. (2)

Purpose and need of the project: Detailed and extensive explanation of the purpose and need of the project. (3)

Project Objectives: All objectives of the proposed work are well defined; Steps to be followed to solve the defined problem are clearly specified. (2)

Project Scheduling & Distribution of Work among Team members: Detailed and extensive Scheduling with timelines provided for each phase of project. Work breakdown structure well defined. (3)

Literature survey: Outstanding investigation in all aspects. (4)

Student's Diary/ Daily Log: The main purpose of writing daily diary is to cultivate the habit of documenting and to encourage the students to search for details. It develops the students' thought process and reasoning abilities. The students should record in the daily/weekly activity diary the day to day account of the observations, impressions, information gathered and suggestions given, if any. It should contain the sketches & drawings related to the observations made by the students. The daily/weekly activity diary shall be signed after every day/week by the guide. (7)

Individual Contribution: The contribution of each student at various stages. (7)

EVALUATION RUBRICS for PROJECT Phase I: Interim Evaluation

No.	Parameters	Marks	Poor	Fair	Very Good	Outstanding
1-a	Topic identification, selection, formulation of objectives and/or literature survey. (Group assessment) [CO1]	10	The team has failed to come with a relevant topic in time. Needed full assistance to find a topic from the guide. They do not respond to suggestions from the evaluation committee and/or the guide. No literature review was conducted. The team tried to gather easy information without verifying the authenticity. No objectives formed yet.	The team has identified a topic. The originally selected topic lacks substance and needs to be revised. There were suggestions given to improve the relevance and quality of the project topic. Only a few relevant references were consulted/ studied and there is no clear evidence to show the team's understanding on the same. Some objectives identified, but not clear enough.	Good evidence of the group thinking and brainstorming on what they are going to build. The results of the brainstorming are documented and the selection of topic is relevant. The review of related references was good, but there is scope of improvement. Objectives formed with good clarity, however some objectives are not realistic enough.	The group has brainstormed in an excellent manner on what they were going to build. The topic selected is highly relevant, real world problem and is potentially innovative. The group shows extreme interest in the topic and has conducted extensive literature survey in connection with the topic. The team has come up with clear objectives which are feasible.
			(0 – 3 Marks)	(4 – 6 Marks)	(7 - 9 Marks)	(10 Marks)
1-b	Project Planning, Scheduling and Resource/ Tasks Identification and allocation. (Group assessment) [CO4]	10	No evidence of planning or scheduling of the project. The students did not plan what they were going to build or plan on what materials / resources to use in the project. The students do not have any idea on the budget required. The team has not yet decided on who does what. No project journal kept.	Some evidence of a primary plan. There were some ideas on the materials /resources required, but not really thought out. The students have some idea on the finances required, but they have not formalized a budget plan. Schedules were not prepared. The project journal has no details. Some evidence on task allocation among the team members.	Good evidence of planning done. Materials were listed and thought out, but the plan wasn't quite complete. Schedules were prepared, but not detailed, and needs improvement. Project journal is presented but it is not complete in all respect / detailed. There is better task allocation and individual members understand about their tasks. There is room for improvement.	Excellent evidence of enterprising and extensive project planning. Gantt charts were used to depict detailed project scheduling. A project management/version control tool is used to track the project, which shows familiarity with modern tools. All materials / resources were identified and listed and anticipation of procuring time is done. Detailed budgeting is done. All tasks were identified and incorporated in the schedule. A well-kept project journal shows evidence for all the above, in addition to the interaction with the project guide. Each member knows well about their individual tasks.
			(0 – 3 Marks)	(4 – 6 Marks)	(7 - 9 Marks)	(10 Marks)

Phase 1 Interim Evaluation Total Marks: 20

EVALUATION RUBRICS for PROJECT Phase I: Final Evaluation

Sl. No.	Parameters	Marks	Poor	Fair	Very Good	Outstanding
1-c	Formulation of Design and/or Methodology and Progress. (Group assessment) [CO1]	5	None of the team members show any evidence of knowledge about the design and the methodology adopted till now/ to be adopted in the later stages. The team has not progressed from the previous stage of evaluation.	The students have some knowledge on the design procedure to be adopted, and the methodologies. However, the team has not made much progress in the design, and yet to catch up with the project plan.	The students are comfortable with design methods adopted, and they have made some progress as per the plan. The methodologies are understood to a large extent.	Shows clear evidence of having a well- defined design methodology and adherence to it. Excellent knowledge in design procedure and its adaptation. Adherence to project plan is commendable.
			(0 – 1 Marks)	(2 – 3 Marks)	(4 Marks)	(5 Marks)
1-d	Individual and Teamwork Leadership (Individual assessment) [CO3]	10	The student does not show any interest in the project activities, and is a passive member.	The student show some interest and participates in some of the activities. However, the activities are mostly easy and superficial in nature.	The student shows very good interest in project, and takes up tasks and attempts to complete them. Shows excellent responsibility and team skills. Supports the other members well.	The student takes a leadership position and supports the other team members and leads the project. Shows clear evidence of leadership.
			(0 – 3 Marks)	(4 – 6 Marks)	(7 - 9 Marks)	(10 Marks)
1-e	Preliminary Analysis/ Modeling / Simulation/ Experiment / Design/ Feasibility study [CO1]	10	The team has not done any preliminary work with respect to the analysis/modeling/ simulation/experiment/design/feasibility study/ algorithm development.	The team has started doing some preliminary work with respect to the project. The students however are not prepared enough for the work and they need to improve a lot.	There is some evidence to show that the team has done good amount of preliminary investigation and design/ analysis/ modeling etc. They can improve further.	Strong evidence for excellent progress in the project. The team has completed the required preliminary work already and are poised to finish the phase I in an excellent manner. They have shown results to prove their progress.
			(0 – 3 Marks)	(4 – 6 Marks)	(7 - 9 Marks)	(10 Marks)

1-f	Documentation and presentation. (Individual & group assessment). [CO6]	5	The team did not document the work at all. The project journal/diary is not presented. The presentation was shallow in content and dull in appearance. The individual student has no idea on the presentation of his/her part.	Some documentation is done, but not extensive. Interaction with the guide is minimal. Presentation include some points of interest, but overall quality needs to be improved. Individual performance to be improved.	Most of the project details were documented well enough. There is scope for improvement. The presentation is satisfactory. Individual performance is good.	The project stages are extensively documented in the report. Professional documentation tools like LaTeX were used to document the progress of the project along with the project journal. The documentation structure is well-planned and can easily grow into the project report. The presentation is done professionally and with great clarity. The individual's performance is excellent.
			(0 – 1 Marks)	(2 – 3 Marks)	(4 Marks)	(5 Marks)
Total		**30**	**Phase - I Final Evaluation Marks: 30**			

EVALUATION RUBRICS for PROJECT Phase I: Report Evaluation

Sl. No.	Parameters	Marks	Poor	Fair	Very Good	Outstanding
1-g	Report [CO6]	20	The prepared report is shallow and not as per standard format. It does not follow proper organization. Contains mostly Unacknowledged content. Lack of effort in preparation is evident.	Project report follows the standard format to some extent. However, its organization is not very good. Language needs to be improved. All references are not cited properly in the report.	Project report shows evidence of systematic documentation. Report is following the standard format and there are only a few issues. Organization of the report is good. Most of references are cited properly.	The report is exceptionally good. Neatly organized. All references cited properly. Diagrams/Figures, Tables and equations are properly numbered, and listed and clearly shown. Language is excellent and follows standard styles.
			(0 - 7 Marks)	(8 - 12 Marks)	(13 - 19 Marks)	(20 Marks)
				Phase - I Project Report Marks: 20		

SEMESTER VII

PROGRAM ELECTIVE II

CST413	MACHINE LEARNING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course enables the learners to understand the advanced concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning and the Naive Bayes algorithm, basic clustering algorithms and classifier performance measures. This course helps the students to provide machine learning based solutions to real world problems.

Prerequisite: Basic understanding of probability theory, linear algebra and Python Programming

Course Outcomes: After the completion of the course the student will be able to

CO1	Illustrate Machine Learning concepts and basic parameter estimation methods. (Cognitive Knowledge Level: **Apply)**
CO2	Demonstrate supervised learning concepts (regression, linear classification). **(**Cognitive Knowledge Level: **Apply**)
CO3	Illustrate the concepts of Multilayer neural network and Support Vector Machine **(**Cognitive Knowledge Level: **Apply**)
CO4	Describe unsupervised learning concepts and dimensionality reduction techniques. (Cognitive Knowledge Level: **Apply**)
CO5	Solve real life problems using appropriate machine learning models and evaluate the performance measures (Cognitive Knowledge Level: **Apply**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓	✓	✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Overview of machine learning)

Machine learning paradigms-supervised, semi-supervised, unsupervised, reinforcement learning. Basics of parameter estimation **-** maximum likelihood estimation(MLE) and maximum a posteriori estimation(MAP). Introduction to Bayesian formulation.

Module-2 (Supervised Learning)

Regression - Linear regression with one variable, Linear regression with multiple variables, solution using gradient descent algorithm and matrix method, basic idea of overfitting in regression. Linear Methods for Classification- Logistic regression, Naive Bayes, Decision tree algorithm ID3.

Module-3 (Neural Networks (NN) and Support Vector Machines (SVM))

Perceptron, Neural Network - Multilayer feed forward network, Activation functions (Sigmoid, ReLU, Tanh), Backpropagation algorithm.

SVM - Introduction, Maximum Margin Classification, Mathematics behind Maximum Margin Classification, Maximum Margin linear separators, soft margin SVM classifier, non-linear SVM, Kernels for learning non-linear functions, polynomial kernel, Radial Basis Function(RBF).

Module-4 (Unsupervised Learning)

Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, Expectation maximization (EM) for soft clustering. Dimensionality reduction – Principal Component Analysis.

Module-5 (Classification Assessment)

Classification Performance measures - Precision, Recall, Accuracy, F-Measure, Receiver Operating Characteristic Curve(ROC), Area Under Curve(AUC. Bootstrapping, Cross Validation, Ensemble methods, Bias-Variance decomposition. Case Study: Develop a classifier for face detection.

Text Book

1. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
2. Mohammed J. Zaki and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, First South Asia edition, 2016.
3. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 2016
4. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.

Reference Books

1. Christopher Bishop. Neural Networks for Pattern Recognition,Oxford University Press, 1995.
2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, MIT Press 2012.
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements Of Statistical Learning, Second edition Springer 2007.
4. P. Langley, Elements of Machine Learning, Morgan Kaufmann, 1995.
5. Richert and Coelho, Building Machine Learning Systems with Python.
6. Davy Cielen, Arno DB Meysman and Mohamed Ali.Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Dreamtech Press 2016.

Course Level Assessment Questions

Course Outcome1 (CO1):

1. A coin is tossed 100 times and lands heads 62 times. What is the maximum likelihood estimate for ***θ*,** the probability of heads.
2. Suppose data ***x_1, ..., x_n*** are independent and identically distributed drawn from an exponential distribution ***exp(λ)***. Find the maximum likelihood for λ.
3. Suppose ***x_1, ..., x_n*** are independent and identically distributed(iid) samples from a distribution with density

$$f_X(x \mid \theta) = \begin{cases} \dfrac{\theta x^{\theta-1}}{3^{\theta}}, & 0 \le x \le 3 \\ 0, & \text{otherwise} \end{cases}$$

   Find the maximum likelihood estimate(MLE) for ***θ***.
4. Find the maximum likelihood estimator (MLE) and maximum a posteriori (MAP) estimator for the mean of a univariate normal distribution. Assume that we have ***N*** samples, ***x_1,..., x_N*** independently drawn from a normal distribution with known variance ***σ^2*** and unknown mean ***μ*** and the prior distribution for the mean is itself a normal distribution with mean ***ν*** and variance ***β^2***. What happens to the MLE and MAP estimators as the number of samples goes to infinity.

Course Outcome 2(CO2):

1. Suppose that you are asked to perform linear regression to learn the function that outputs ***y***, given the ***D***-dimensional input ***x***. You are given ***N*** independent data points, and that all the ***D*** attributes are linearly independent. Assuming that ***D*** is around 100, would you prefer the closed form solution or gradient descent to estimate the regressor?
2. Suppose you have a three class problem where class label ***y*** **∈ 0, 1, 2** and each training example ***X*** has 3 binary attributes ***X_1, X_2, X_3*** **∈ 0, 1**. How many parameters (probability distribution) do you need to know to classify an example using the Naive Bayes classifier?

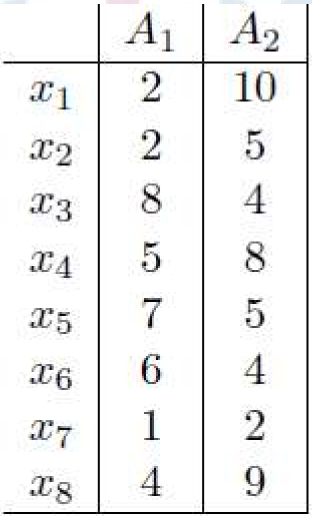
Course Outcome 3(CO3):

1. What are support vectors and list any three properties of the support vector classifier solution?
2. Why do you use kernels to model a projection from attributes into a feature space, instead of simply projecting the dataset directly?

3. Describe how Support Vector Machines can be extended to make use of kernels. Illustrate with reference to the Gaussian kernel ***K(x, y) = e***$^{-z}$, where $z = (x-y)^2$.
4. Briefly explain one way in which using tanh instead of logistic activations makes optimization easier.
5. ReLU activation functions are most used in neural networks instead of the tanh activation function. Draw both activation functions and give a) an advantage of the ReLU function compared to the tanh function. b) a disadvantage of the ReLU function compared to the tanh function.

Course Outcome 4(CO4): .

1. Which similarity measure could be used to compare feature vectors of two images? Justify your answer.
2. Illustrate the strength and weakness of k-means algorithm.
3. Suppose you want to cluster the eight points shown below using **k**-means



	A_1	A_2
x_1	2	10
x_2	2	5
x_3	8	4
x_4	5	8
x_5	7	5
x_6	6	4
x_7	1	2
x_8	4	9

Assume that **k = 3** and that initially the points are assigned to clusters as follows:

$C_1 = \{x_1, x_2, x_3\}$, $C_2 = \{x_4, x_5, x_6\}$, $C_3 = \{x_7, x_8\}$. Apply the **k**-means algorithm until convergence, using the Manhattan distance.

4. Cluster the following eight points representing locations into three clusters: $A_1(2, 10)$, $A_2(2, 5)$, $A_3(8, 4)$, $A_4(5, 8)$, $A_5(7, 5)$, $A_6(6, 4)$, $A_7(1, 2)$, $A_8(4, 9)$.

Initial cluster centers are: $A_1(2, 10)$, $A_4(5, 8)$ and $A_7(1, 2)$.

The distance function between two points $a = (x_1, y_1)$ and $b = (x_2, y_2)$ is defined as $D(a, b) = |x_2 - x_1| + |y_2 - y_1|$

Use **k**-Means Algorithm to find the three cluster centers after the second iteration.

Course Outcome 5(CO5):

1. What is ensemble learning? Can ensemble learning using linear classifiers learn classification of linearly non-separable sets?
2. Describe boosting. What is the relation between boosting and ensemble learning?
3. Classifier A attains 100% accuracy on the training set and 70% accuracy on the test set. Classifier B attains 70% accuracy on the training set and 75% accuracy on the test set. Which one is a better classifier. Justify your answer.
4. What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why?
5. Suppose there are three classifiers A,B and C. The (FPR, TPR) measures of the three classifiers are as follows – A (0, 1), B (1, 1) , C (1,0.5). Which can be considered as a perfect classifier? Justify your answer.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST413

Course Name: Machine Learning

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Define supervised learning? Name special cases of supervised learning depending on whether the inputs/outputs are categorical, or continuous.

2. Differentiate between Maximum Likelihood estimation (MLE) and Maximum a Posteriori (MAP) estimation?

3. What is overfitting and why is it a problem?

4. Specify the basic principle of gradient descent algorithm.

5. Suppose that you have a linear support vector machine(SVM) binary classifier. Consider a point that is currently classified correctly, and is far away from the decision boundary. If you remove the point from the training set, and re-train the classifier, will the decision boundary change or stay the same? Justify your answer.

6. Mention the primary motivation for using the kernel trick in machine learning algorithms?

7. Expectation maximization (EM) is designed to find a maximum likelihood setting of the parameters of a model when some of the data is missing. Does the algorithm converge? If so, do you obtain a locally or globally optimal set of parameters?

8. Illustrate the strength and weakness of k-means algorithm.

9. Classifier A attains 100% accuracy on the training set and 70% accuracy on the test set. Classifier B attains 70% accuracy on the training set and 75% accuracy on the test set. Which one is a better classifier. Justify your answer.

10. How does bias and variance trade-off affect machine learning algorithms?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Suppose that X is a discrete random variable with the following probability mass function: where $0 \leq \theta \leq 1$ is a parameter. The following 10 independent observations **(7)**

X	0	1	2	3
$P(X)$	$2\theta/3$	$\theta/3$	$2(1-\theta)/3$	$(1-\theta)/3$

were taken from such a distribution: (3, 0, 2, 1, 3, 2, 1, 0, 2, 1). What is the maximum likelihood estimate of θ.

(b) Suppose you have a three class problem where class label $\boldsymbol{y} \in \mathbf{0, 1, 2}$ and each training example $\boldsymbol{X}$ has 3 binary attributes $\boldsymbol{X_1, X_2, X_3} \in \mathbf{0, 1}$. How many parameters (probability distribution) do you need to know **(7)** to classify an example using the Naive Bayes classifier?

OR

12. (a) Consider the geometric distribution, which has p.m.f $P(X = k) = (1 - \theta)^{k-1}\theta$. Assume that n i.i.d data are drawn from that distribution. **(7)**

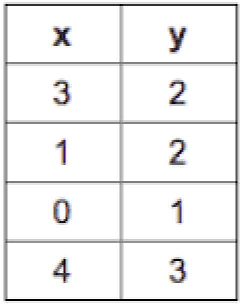
i. Write an expression for the log-likelihood of the data as a function of the parameter θ.

ii. Find the maximum likelihood estimate for θ?

ii. Let θ has a beta prior distribution. What is the posterior distribution of θ?

(b) Find the maximum a posteriori (MAP) estimator for the mean of a univariate normal distribution. Assume that we have ***N*** samples, $\boldsymbol{x_1, \ldots, x_N}$ independently drawn from a normal distribution with known variance $\boldsymbol{\sigma^2}$ and unknown mean $\boldsymbol{\mu}$ and the prior distribution for the mean is itself a normal distribution with mean $\boldsymbol{v}$ and variance $\boldsymbol{\beta^2}$. **(7)**

13. (a) Consider the hypothesis for the linear regression $h_\theta(x) = \theta_0 + \theta_1 x$, and the cost function $J(\theta_{0,}\ \theta_1) = 1/2m\ \Sigma_1^m(h_\theta(x^{(i)}) - y^{(i)})^2$where m is the number of training examples. Given the following set of training examples. **(7)**



x	y
3	2
1	2
0	1
4	3

Answer the following questions :

1) Find the value of h_θ (2) if $\theta_0 = 0$ and $\theta_1 = 1.5$

2) Find the value of J(0,1)

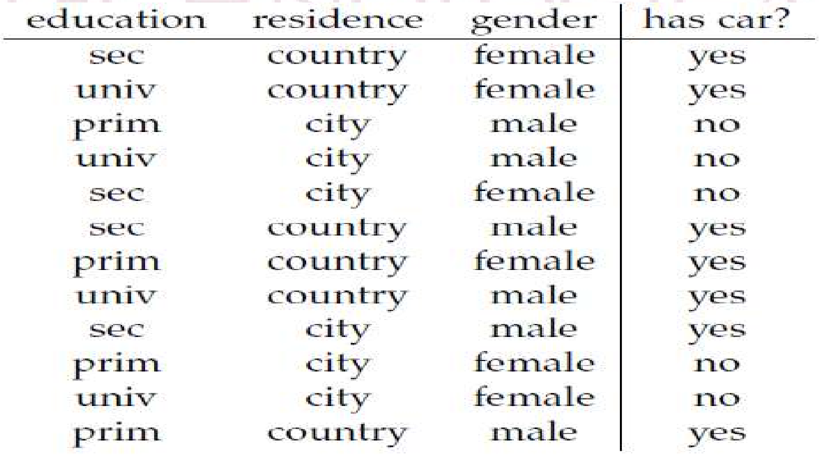
3) Suppose the value of $J(\theta_0, \theta_1) = 0$. What can be inferred from this.

(b) Assume we have a classification problem involving 3 classes: professors, students, and staff members. There are 750 students, 150 staff members and 100 professors. All professors have blond hair, 50 staff members have blond hair, and 250 students have blond hair. Compute the information gain of the test ”hair color = blond” that returns true or false. **(3)**

(c) Explain the significance of regularization. How do Ridge differs from Lasso regularization? **(4)**

OR

14. (a) The following dataset can be used to train a classifier that determines whether a given person is likely to own a car or not. There are three features: education level (primary, secondary, or university); residence (city or country); gender (female, male). **(7)**



education	residence	gender	has car?
sec	country	female	yes
univ	country	female	yes
prim	city	male	no
univ	city	male	no
sec	city	female	no
sec	country	male	yes
prim	country	female	yes
univ	country	male	yes
sec	city	male	yes
prim	city	female	no
univ	city	female	no
prim	country	male	yes

Use ID3 Algorithm and find the best attribute at the root level of the tree

(b) Consider a linear regression problem y = w1x + w0, with a training set having m examples $(x_1, y_1), \ldots (x_m, y_m)$. Suppose that we wish to minimize the mean 5th degree error (loss function) given by $1/m\ \Sigma_1^m (y_i - w_1 x_i - w_0)^5$. **(7)**

1. Calculate the gradient with respect to the parameter w_1.

2. Write down pseudo-code for on-line gradient descent on w_1.

3. Give one reason in favor of on-line gradient descent compared to batch-gradient descent, and one reason in favor of batch over on-line.

15. (a) Consider a support vector machine whose input space is 2-D, and the inner products are computed by means of the kernel ***K(x, y) = (x.y + 1)***^{2}***-1***, where ***x.y*** denotes the ordinary inner product. Show that the mapping to feature space that is implicitly defined by this kernel is the mapping to 5-D given by **(8)**

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2}\, x_1 \\ \sqrt{2}\, x_2 \end{bmatrix}.$$

(b) Consider a neuron with four inputs, and weight of edge connecting the inputs are 1, 2, 3 and 4. Let the bias of the node is zero and inputs are 2, 3, 1, 4. If the activation function is linear f(x)=2x, compute the output of the neuron. **(3)**

(c) Compare ReLU with Sigmoid function **(3)**

OR

16. (a) State the mathematical formulation to express Soft Margin as a constraint optimization problem. **(10)**

(b) What is the basic idea of back propagation algorithm **(4)**

17. (a) Suppose that we have the following data (one variable). Use single linkage Agglomerative clustering to identify the clusters. **(8)**

Data: (2, 5, 9, 15, 16, 18, 25, 33, 33, 45).

(b) Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8): **(6)**

(i) Compute the Euclidean distance between the two objects.

(ii) Compute the Manhattan distance between the two objects.

(iii) Compute the Minkowski distance between the two objects, using p = 3

OR

18. (a) Suppose that we have the following data: **(8)**

(2, 0), (1, 2), (2, 2), (3, 2), (2, 3), (3, 3), (2, 4), (3, 4), (4, 4), (3, 5)

Identify the cluster by applying the k-means algorithm, with k = 2. Try using initial cluster centers as far apart as possible

(b) Describe EM algorithm for Gaussian Mixtures **(8)**

19. (a) Suppose the dataset had 9700 cancer-free images from 10000 images from cancer patients. Find precision, recall and accuracy ? Is it a good classifier? Justify. **(7)**

Actual Class\Predicted class	cancer = yes	cancer = no	Total
cancer = yes	**90**	**210**	300
cancer = no	**140**	**9560**	9700
Total	230	9770	10000

(b) What is Principal Component Analysis (PCA)? Which eigen value indicates the direction of largest variance? **(7)**

OR

20. (a) Assume you have a model with a high bias and a low variance. What are the characteristics of such a model? **(6)**

(b) What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why? **(8)**

Teaching Plan

No	Contents	No. of Lecture Hours (37 hrs)
Module -1 (Overview of machine learning) (7 hours)		
1.1	Supervised, semi-supervised, unsupervised learning, reinforcement learning (Text Book (TB) 1: Chapter 1)	1 hour
1.2	Maximum likelihood estimation(MLE) (TB 1: Section 4.2)	1 hour
1.3	Maximum likelihood estimation (MLE)- example (TB 1: Section 4.2)	1 hour
1.4	Maximum a posteriori estimation(MAP) (TB 4: Section 6.2)	1 hour
1.5	Maximum a posteriori estimation(MAP)-example (TB 4: Section 6.2)	1 hour
1.6	Bayesian formulation (TB 1: Section 14.1, 14.2)	1 hour
1.7	Bayesian formulation -example (TB 1: Section 14.1, 14.2)	1 hour
Module-2 (Supervised Learning) (7 hours)		
2.1	Linear regression with one variable (TB 1: Section 2.6)	1 hour
2.2	Multiple variables, Solution using gradient descent algorithm and matrix method (No derivation required) (TB 1: Section 5.8)	1 hour
2.3	Overfitting in regression, Lasso and Ridge regularization	1 hour
2.4	Logistic regression	1 hour
2.5	Naive Bayes (TB 2: Section 18.2)	1 hour
2.6	Decision trees (TB 2: Chapter 19)	1 hour
2.7	Decision trees- ID3 algorithm (TB 2: Chapter 19)	1 hour
Module-3 (Neural Networks and Support Vector Machines) (9 hours)		
3.1	Perceptron, Perceptron Learning	1 hour
3.2	Multilayer Feed forward Network, Activation Functions (Sigmoid, ReLU, Tanh)	1 hour
3.3	Back Propagation Algorithm	1 hour
3.4	Illustrative Example for Back Propagation	1 hour
3.5	Introduction, Maximum Margin Hyperplane,	1 hour
3.6	Mathematics behind Maximum Margin Classification	1 hour
3.7	Formulation of maximum margin hyperplane and solution	1 hour

3.8	Soft margin SVM, Solution of Soft margin SVM	1 hour
3.9	Non-linear SVM , Kernels for learning non-linear functions, Examples - Linear, RBF, Polynomial	1 hour
Module-4 (Unsupervised Learning) (7 hours)		
4.1	Similarity measures- Minkowski distance measures(Manhattan, Euclidean), Cosine Similarity	1 hour
4.2	Clustering - Hierarchical Clustering (TB 2: Chapter 14)	1 hour
4.3	K-means partitional clustering (TB 2: Chapter 13)	1 hour
4.4	Expectation maximization (EM) for soft clustering (TB 2: Chapter 13)	1 hour
4.5	Expectation maximization (EM) for soft clustering (TB 2: Chapter 13)	1 hour
4.6	Dimensionality reduction – Principal Component Analysis (TB 1: Section 6.3)	1 hour
4.7	Dimensionality reduction – Principal Component Analysis (TB 1: Section 6.3)	1 hour
Module-5 (Classification Assessment) (7 hours)		
5.1	Performance measures - Precision, Recall, Accuracy, F-Measure, ROC, AUC. (TB 2: Chapter 22.1)	1 hour
5.2	Boot strapping, Cross validation	1 hour
5.3	Ensemble methods- bagging, boosting	1 hour
5.4	Bias-Variance decomposition (TB 2: Chapter 22.3)	1 hour
5.5	Bias-Variance decomposition (TB 2: Chapter 22.3)	1 hour
5.6	Face detection (TB 3: Chapter 5 Section Application: A Face Detection Pipeline)	1 hour
5.7	Face detection (TB 3: Chapter 5 Section Application: A Face Detection Pipeline)	1 hour

CST423	CLOUD COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learners to understand cloud computing concepts. This course includes basic understanding of virtualization, fundamentals of cloud security, cloud computing based programming techniques and different industry popular cloud computing platforms. This course enables the student to suggest cloud based solutions to real world problems.

Prerequisite: Basic understanding of computer networks and operating systems.

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain the various cloud computing models and services. **(Cognitive Knowledge Level: Understand)**
CO2	Demonstrate the significance of implementing virtualization techniques. **(Cognitive Knowledge Level: Understand**)
CO3	Explain different cloud enabling technologies and compare private cloud platforms **(Cognitive Knowledge Level: Understand**)
CO4	Apply appropriate cloud programming methods to solve big data problems. (Cognitive Knowledge Level: **Apply**)
CO5	Describe the need for security mechanisms in cloud (**Cognitive Knowledge Level: Understand**)
CO6	Compare the different popular cloud computing platforms **(Cognitive Knowledge Level: Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											✓
CO2	✓	✓	✓									✓
CO3	✓											✓
CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓										✓
CO6	✓				✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus

Module 1: Fundamental Cloud Computing (7 Hours)

Traditional computing- Limitations. Overview of Computing Paradigms-Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing. NIST reference Model-Basic terminology and concepts. Cloud characteristics, benefits and challenges, Roles and Boundaries. Cloud delivery (service) models-Infrastructure-as-a-Service (IaaS), Platform-as-a-Service(PaaS),Software-as-a-Service (SaaS), XaaS (Anything-as-a-service)-Cloud deployment models- Public cloud, Community cloud, Private cloud, Hybrid cloud.

Module 2: Virtualization (7 Hours)

Introduction to virtualization-Virtualizing physical computing resources, Virtual Machines (Machine virtualization), non-virtualized v/s virtualized machine environments. Types of VMs-process VM v/s system VM, Emulation, interpretation and binary translation. Hardware-level virtualization- Hypervisors/VMM. Types of Hypervisors. Full Virtualization, Para-Virtualization, Hardware-assisted virtualization, OS level virtualization. Basics of Network Virtualization, Storage Virtualization and Desktop Virtualization, Pros and cons of virtualization. Case Study- Xen: Para-virtualization, VMware: full virtualization.

Module 3: Cloud-Enabling Technologies, Private cloud platforms and programming (7 Hours)

Broadband networks and internet architecture- Internet Service Providers (ISPs), Data center technology, Web technology, Multitenant technology, Service technology. Resource provisioning techniques-static and dynamic provisioning.

Open-source software platforms for private cloud-OpenStack, CloudStack, Basics of Eucalyptus, Open Nebula, Nimbus.

Cloud Programming- Parallel Computing and Programming Paradigms. Map Reduce – Hadoop Library from Apache, HDFS, Pig Latin High Level Languages, Apache Spark.

Module 4: Fundamental Cloud Security (7 Hours)

Basic terms and concepts in security- Threat agents, Cloud security threats/risks, Trust. Operating system security-Virtual machine security- Security of virtualization- Security Risks Posed by Shared Images, Security Risks Posed by Management OS. Infrastructure security-Network Level Security, Host Level Security, Application level security, Security of the Physical Systems. Identity & Access Management- Access Control.

Module 5: Popular Cloud Platforms (9 Hours)

Amazon Web Services(AWS):- AWS ecosystem- Computing services, Amazon machine images, Elastic Compute Cloud (EC2), Advanced compute services. Storage services-Simple Storage System (Amazon S3), Elastic Block Store (Amazon EBS), Database Services, Amazon CDN Services and Communication services.

Google Cloud Platform:- IaaS Offerings: Compute Engine (GCE), Cloud Storage, PaaS Offerings: Google App Engine (GAE), Storage services, Application services, Compute services, Database Services, SaaS Offerings: Gmail, Docs, Google Drive.

Microsoft Azure: Azure Platform Architecture, Hyper-V, Azure Virtual Machine, Compute services, Storage services.

Text Books

1. Thomas, E., Zaigham M., Ricardo P "Cloud Computing Concepts, Technology & Architecture.", (2013 Edition). Prentice Hall.
2. Buyya, R., Vecchiola, C., & Selvi, S. T. “Mastering cloud computing: foundations and applications programming”, (2017 Edition), Morgan Kaufmann.
3. Bhowmik, S., “Cloud computing”, (2017 Edition). Cambridge University Press.

References

1. Marinescu, D. C., "Cloud computing: theory and practice.", (2017 Edition). Morgan Kaufmann.
2. Buyya, R., Broberg, J., & Goscinski, A. M., "Cloud computing: Principles and paradigms" (2011 Edition). John Wiley & Sons.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. "*A hybrid cloud is a combination of two or more other cloud deployment models*". Justify the statement with an example.
2. What are the main characteristics of a Platform-as-a-Service solution?
3. How does cloud computing help to reduce the time to market for applications and to cut down capital expenses?
4. Differentiate public and private clouds in terms of flexibility.

Course Outcome 2 (CO2):

1. Define virtualization. What is the role of VMM in virtualization?
2. Explain various implementation levels of Virtualization.
3. State the differences between a traditional computer and a virtual machine.

Course Outcome 3 (CO3):

1. Differentiate between on-premise and cloud-based internetworking.
2. What are the benefits of Data Center Technologies?
3. What are the characteristics of Multi-tenant technology?
4. How can virtualization be implemented at the hardware level?

Course Outcome 4 (CO4):

1. Write a Hadoop MapReduce program that counts the number of occurrences of each character in a file.
2. Write a Hadoop MapReduce program to find the maximum temperature in the weather dataset.

Course Outcome 5 (CO5):

1. Why is it harder to establish security in the cloud?
2. Explain in detail about the security issues one should discuss with a cloud-computing vendor.
3. List and Explain major cloud security challenges.

Course Outcome 6 (CO6):

1. Explain the cloud based databases.
2. With a neat diagram, write about Google App Engine for PaaS applications.
3. Differentiate between amazon SimpleDB and Amazon RDS.
4. *"Storage services in the cloud are offered in two different forms as IaaS and as SaaS"*. Explain.

Model Question Paper

QP Code: **Total Pages : 3**

Reg No:_________________ **Name :____________________**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST423
Course Name: Cloud Computing

Duration: 3 Hrs **Max. Marks :100**

PART A

Answer all Questions. Each question carries 3 Marks (10 x 3 = 30 Marks)

1. Is the IT outsourcing model of traditional computing similar to cloud computing? Justify.

2. Why is grid computing considered as the predecessor of cloud computing? Explain.

3. What is virtualization and what are its benefits?

4. Explain why a hypervisor is also called a virtual machine monitor?

5. Differentiate between multi-tenancy and virtualization.

6. "*The field of service technology is a keystone foundation of cloud computing*". Explain.

7. Discuss any two identity management techniques used in cloud computing.

8. Differentiate between mandatory access control (MAC) and discretionary Access Control (DAC).

9. Differentiate between Amazon S3 and Amazon EBS.

10. Explain the database service offered by google cloud.

(10 x3 =30)

PART B

Answer any one Question from each Module. Each question carries 14 Marks

11. **(a)** Discuss the cloud computing reference model. **(8)**

(b) Which are the basic components of an IaaS-based solution for cloud computing? Also provide some examples of IaaS implementations. **(6)**

OR

12. **(a)** List down the characteristics and challenges of cloud computing. **(6)**

(b) Classify the various types of clouds. **(8)**

13. **(a)** List and discuss various types of virtualization. **(8)**

(b) Differentiate between full virtualization and paravirtualization. **(6)**

OR

14. **(a)** What is Xen? Discuss its elements for virtualization. **(8)**

(b) Explain the design requirements for Virtual Machine Monitor (VMM). **(6)**

15. **(a)** Explain the broadband networks and internet architecture. **(8)**

(b) List and explain the technologies and components of data centers. **(6)**

OR

16. **(a)** What are the major functions of the MapReduce framework? Explain the logical data flow of MapReduce function using a suitable example . **(8)**

(b) Write a Hadoop MapReduce program that counts the number of occurrences of each word in a file. **(6)**

17. **(a)** Explain common threats and vulnerabilities in cloud-based environments with suitable examples. **(8)**

(b) Discuss the security risks posed by shared images with suitable examples. **(6)**

OR

18. **(a)** Explain the operating system security in cloud computing. **(8)**

(b) What do you mean by threat agents?. Explain different types of threat agents. **(6)**

19. **(a)** Describe Amazon EC2 and its basic features. **(8)**

(b) Illustrate the architecture of Amazon S3. **(6)**

OR

20. **(a)** Describe the core components of Google AppEngine. **(8)**

(b) Explain the architecture of Windows Azure. **(6)**

Teaching Plan

No	**Contents**	**No. of Lecture Hours (37 hrs)**
Module 1 (Fundamental Cloud Computing) (6 hours)		
1.1	Traditional computing: Limitations.	1
1.2	Overview of Computing Paradigms: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing.	1
1.3	NIST reference Model, Basic terminology and concepts.	1
1.4	Cloud characteristics and benefits, challenges. Roles and Boundaries.	1
1.5	Cloud delivery (service) models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), XaaS (Anything-as-a-service).	1
1.6	Cloud deployment models: Public cloud, Community cloud, Private cloud, Hybrid cloud.	1

Module 2(Virtualization)(7 Hours)		
2.1	Introduction to virtualization, Virtualizing physical computing resources Virtual Machines (Machine virtualization):- non-virtualized v/s virtualized machine environments.	1
2.2	Types of VMs: process VM v/s system VM, Emulation, interpretation and binary translation.	1
2.3	Hardware-level virtualization: Hypervisors/VMM, Types of Hypervisors.	1
2.4	Full Virtualization, Para-Virtualization, Hardware-assisted virtualization, OS level virtualization.	1
2.5	Basics of Network Virtualization, Storage Virtualization and Desktop Virtualization, Pros and cons of virtualization.	1
2.6	Case Study: Xen: Para-virtualization.	1
2.7	Case Study: VMware: full virtualization.	1
Module 3 (Cloud-Enabling Technologies, Private cloud platforms and programming) (9 Hours)		
3.1	Broadband networks and internet architecture: Internet Service Providers (ISPs), Data center technology, Web technology, Multitenant technology, Service technology.	1
3.2	Resource provisioning techniques: static and dynamic provisioning.	1
3.3	Open-source software platforms for private cloud: OpenStack, CloudStack.	1
3.4	Basics of Eucalyptus, Open-Nebula, Nimbus.	1
3.5	Cloud Programming: Parallel Computing and Programming Paradigms.	1
3.6	Map Reduce.	1
3.7	Hadoop Library from Apache, HDFS.	1
3.8	Pig Latin High Level Languages	1
3.9	Apache Spark.	1

Module 4 (Fundamental Cloud Security) (7 Hours)		
4.1	Basic terms and concepts in security, Threat agents.	1
4.2	Cloud security threats/risks, Trust.	1
4.3	Operating system security, Virtual machine security.	1
4.4	Security of virtualization.	1
4.5	Security Risks posed by Shared Images, Security Risks posed by Management OS.	1
4.6	Infrastructure security: - Network Level Security, Host Level Security, Application level security, Security of the Physical Systems.	1
4.7	Identity & Access Management, Access Control.	1
Module 5 (Popular Cloud Platforms) (8 Hours)		
5.1	Amazon Web Services(AWS):- AWS ecosystem, Computing services: Amazon machine images, Elastic Compute Cloud (EC2).	1
5.2	Advanced computing services, Storage services: Simple Storage System (Amazon S3), Elastic Block Store (Amazon EBS).	1
5.3	Database Services, Amazon CDN Services and Communication services.	1
5.4	Google Cloud Platform:- IaaS Offerings: Compute Engine (GCE), Cloud Storage.	1
5.5	PaaS Offerings: Google App Engine (GAE), Storage services, Application services, Compute services.	1
5.6	Database Services, SaaS Offerings: Gmail, Docs, Google Drive.	1
5.7	Microsoft Azure: Azure Platform Architecture, Hyper-V, Azure Virtual Machine.	1
5.8	Azure Compute services, Storage services.	1

CST433	SECURITY IN COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learners to explore various algorithms to offer confidentiality, integrity, authentication &non-repudiation services and different attacks on system security with their countermeasures. It covers classical encryption techniques, symmetric and public key crypto-system, key distribution techniques, authentication functions, intruders, malicious software, and DDoS attacks. The concepts covered in this course enable the learners in effective use of cryptographic algorithms and appropriate countermeasures for securing real life applications.

Prerequisite: A fundamental knowledge in mathematical foundations of security.

Course Outcomes: After the completion of the course, the student will be able to

CO1	Identify the security services provided against different types of security attacks. **(Cognitive Knowledge Level: Understand)**
CO2	Illustrate classical encryption techniques for information hiding. **(Cognitive Knowledge Level: Apply)**
CO3	Illustrate symmetric/asymmetric key cryptosystems for secure communication. **(Cognitive Knowledge Level: Apply)**
CO4	Explain message integrity and authentication methods in a secure communication scenario. **(Cognitive Knowledge Level: Understand)**
CO5	Interpret public/secret key distribution techniques for secure communication. **(Cognitive Knowledge Level: Understand)**
CO6	Identify the effects of intruders, malicious software and distributed denial of service attacks on system security. **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓		✓							✓
CO4	✓	✓	✓			✓						✓
CO5	✓	✓	✓									✓
CO6	✓	✓	✓			✓		✓				✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (%)	Test 2 (%)	End Semester Examination (%)
Remember	30	30	30
Understand	40	40	40

Apply	30	30	30
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment Test	: **25 marks**
Continuous Assessment Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Basics of Security and Traditional Cryptosystems)
OSI security architecture – Security attacks, Services, Mechanisms. Cryptography vs Cryptanalysis. Classical encryption techniques – Symmetric cipher model. Substitution ciphers – Monoalphabetic vs Polyalphabetic ciphers, Caesar cipher, Affine cipher, Playfair cipher, Vigenere cipher, Hill cipher. Transposition ciphers – Keyless, Keyed, Double transposition.

Module-2 (Modern Symmetric Key Cryptosystems)
Symmetric key ciphers – Block vs Stream ciphers, Block cipher components, Product ciphers, Feistel and Non-Feistel ciphers. Data Encryption Standard (DES) – Structure, Key generation, Design criteria, Weaknesses, Double DES, Triple DES. Advanced Encryption Standard (AES) – Structure, Key expansion. Block cipher modes of operation – Electronic Codebook Mode (ECB), Cipher Block Chaining Mode (CBC), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), Counter Mode (CTR). Stream ciphers – Structure, RC4.

Module-3 (Public Key Cryptosystems)
Introduction to public key cryptosystems – Principles, Applications, Requirements, Conventional vs Public key cryptosystems. RSA cryptosystem – Algorithm, Security, Attacks. ElGamal cryptosystem – Algorithm. Diffie-Hellman key exchange – Algorithm, Man-in-the-middle attack. Elliptic Curve Cryptography (ECC) – ElGamal ECC, Key exchange using ECC.

Module-4 (Message Integrity and Authentication)
Hash functions – Security requirements, Secure Hash Algorithm (SHA-512). Message Authentication Code (MAC) – Requirements, Uses, Hash-based MAC (HMAC), Cipher-based MAC (CMAC). Digital signatures – Attacks, Forgeries, Requirements, Direct vs Arbitrated digital signatures, RSA digital signature, ElGamal digital signature, Digital Signature Standard (DSS).

Module-5 (Key Distribution and System Security)
Key management – Distribution of secret keys using symmetric and asymmetric encryption, Distribution of public keys. System security – Intruders, Intrusion detection techniques, Password management. Malicious software – Viruses, Related threats, Countermeasures. Distributed Denial of Service (DDoS) attacks – Types, Countermeasures.

Text Books
1. William Stallings, Cryptography and Network Security Principles and Practice, 4/e, Pearson Ed.
2. Behrouz A Forouzan, Cryptography and Network Security, 3/e, Tata McGraw-Hill.

References

1. Charles P Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Security in Computing, 5/e, Prentice Hall.
2. G.A. Jones & J.M. Jones, Elementary Number Theory, Springer UTM, 2007.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Define the type of security attack in the following case: A student breaks into a teacher's office to obtain a copy of the next day's exam question paper.
2. Which security mechanism is provided in the following case: A bank requires the customer's signature for a withdrawal.

Course Outcome 2 (CO2):

1. Alice wishes to send the message "COME BACK EARLY" to Bob, using Playfair cipher. The key to be used is "SAFFRON". Show the process of encryption.
2. Using Affine cipher, encrypt "HOT" and decrypt "JDG". Key is (7, 3).
3. Implement the Vigenere cipher method in a suitable programming language. (Assignment)

Course Outcome 3 (CO3):

1. If the DES key with parity bit is 0123 ABCD 2562 1456, find the first round key.
2. In RSA, given p=19, q=23, public key(e)=3, find n, ϕ(n) and private key(d).
3. Implement any two symmetric/asymmetric encryption techniques in a suitable programming language. (Assignment)

Course Outcome 4 (CO4):

1. Describe the steps involved in generating a Hash-based MAC.
2. Using ElGamal scheme, generate the signatures for the message M=400 with p=881, d=700 and r=17.
3. A company wishes to implement a secure authentication mechanism for communication. As a system security admin suggest any two ways of implementing such a mechanism. (Assignment)

Course Outcome 5 (CO5):

1. List any two ways in which secret keys can be distributed to two communicating parties.
2. Explain the significance of a public-key authority in the distribution of public keys.

Course Outcome 6 (CO6):

1. What are false positives and negatives in the context of Intrusion Detection Systems? How can we reduce these two?
2. Distinguish between a direct DDoS attack and a reflector DDoS attack.
3. Bob works as a network administrator in ABC & Co. On a day of his absence, he shared his admin password with one of his colleagues, John, to manage a network issue. Later John started misusing this privilege by launching DoS attacks in the network. Describe the ethical issues in this scenario and how can this be avoided? (Assignment)

Model Question Paper

QP CODE: **PAGES: ___**

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST433
Course Name: SECURITY IN COMPUTING

Max Marks: 100 **Duration: 3 Hours**

PART A
(Answer All Questions. Each question carries 3 marks)

1. Differentiate between passive attack and active attack.

2. Use an Affine cipher to encrypt the message “SECURITY” with the key pair(7,2) in modulus 26.

3. Compare stream cipher and Block cipher with example.

4. Differentiate between diffusion and confusion.

5. Define the elliptic curve logarithm problem.

6. Consider an ElGamal scheme with a common prime q = 71 and a primitive root α = 7. If B has a public key $Y_B = 3$ and A chose the random number k = 2, what is the ciphertext of the message M = 30?

7. Give the requirements of MAC function.

8. Specify the different types of forgery in digital signature.

9. List three different classes of intruders.

10. Mention the phases of operation of a virus.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Illustrate the two approaches to attack a conventional encryption scheme. **(4)**

(b) A Hill cipher is setup with the key matrix $\begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}$. **(10)**

Encrypt the text “COMPUTER”. Show the calculations for the corresponding decryption of the ciphertext to recover the original text back.

OR

12. (a) Encrypt the text “this is an exercise and complete it” using transposition cipher with the key (3,2,1,4,5). Show decryption of the ciphertext to recover the original text back. **(6)**

(b) Encrypt the message “the house is being sold tonight”using the following ciphers. Ignore the space between words. **(8)**

i) Vigenere cipher with key = “largest”.

ii) Autokey system of Vigenere cipher with key =“largest”.

13. (a) How is round key generated in DES? **(4)**

(b) Illustrate AES encryption in detail. **(10)**

OR

14. (a) Explain the construction of S-box in AES. **(5)**

(b) Summarize the primitive operations in RC4 algorithm. **(9)**

15. (a) Compare the Cipher Block Chaining Mode (CBC) and Cipher Feedback Mode (CFB) of block ciphers. **(6)**

(b) Explain RSA cryptosystem. In an RSA cryptosystem a participant A uses two prime numbers p=13 and q=17 to generate public key and private key. The public key of A is 35. Find the private key of A. **(8)**

OR

16. (a) Illustrate ElGamal cryptosystem. **(6)**

(b) Consider a Diffie–Hellman scheme with a common prime q=11 and a primitive root α=2. **(8)**

i) Show that 2 is a primitive root of 11.

ii) If User A has public key Y_A= 9, what is A's private key X_A?

iii) If User A has public key Y_B= 3, what is the shared secret key K, shared with A?

17. (a) Describe different types of arbitrated digital signature techniques. **(6)**

(b) Explain Cipher – Based Message Authentication Code. **(8)**

OR

18. (a) Explain the attacks on digital signature. **(5)**

(b) Describe the working of SHA-512 with diagrams. **(9)**

19. (a) Explain four techniques used to avoid guessable passwords. **(6)**

(b) Describe the different techniques for public key distribution. **(8)**

OR

20. (a) Explain different types of Simple DDoS attack and its countermeasures. **(6)**

(b) Differentiate between statistical anomaly detection and rule-based intrusion detection. **(8)**

Teaching Plan

No	Contents	No.of Lecture Hours (35Hrs)
	Module-1 (Basics of Security and Traditional Cryptosystems) (6 hrs)	
1.1	OSI security architecture – Security attacks, Services, Mechanisms	1
1.2	Cryptography vs Cryptanalysis. Classical encryption techniques – Symmetric cipher model	1
1.3	Substitution ciphers – Monoalphabetic vs Polyalphabetic ciphers, Caesar cipher, Affine cipher	1
1.4	Playfair cipher, Vigenere cipher	1
1.5	Hill cipher	1
1.6	Transposition ciphers – Keyless, Keyed, Double transposition	1
	Module-2 (Modern Symmetric Key Cryptosystems) (9hrs)	
2.1	Symmetric key ciphers – Block vs Stream ciphers, Block cipher components, Product ciphers, Feistel and Non-Feistel ciphers	1
2.2	Data Encryption Standard (DES) – Structure, Key generation	1
2.3	Design criteria, Weaknesses	1
2.4	Double DES, Triple DES	1
2.5	Advanced Encryption Standard (AES) – Overall Structure	1
2.6	Stages of encryption/decryption	1
2.7	Key expansion	1
2.8	Block cipher modes of operation – Electronic Codebook Mode (ECB), Cipher Block Chaining Mode (CBC), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), Counter Mode (CTR).	1
2.9	Stream ciphers – Structure, RC4	1
	Module-3 (Public Key Cryptosystems)(7hrs)	
3.1	Public key cryptosystems – Principles, Applications, Requirements, Conventional vs Public key cryptosystems	1

3.2	RSA cryptosystem – Algorithm	1
3.3	RSA Security, Attacks	1
3.4	ElGamal cryptosystem – Algorithm	1
3.5	Diffie-Hellman key exchange – Algorithm, Man-in-the-middle attack	1
3.6	Elliptic Curve Cryptography (ECC) – ElGamal ECC	1
3.7	Key exchange using ECC	1
Module-4 (Message Integrity and Authentication) (6 hrs)		
4.1	Hash functions – Security requirements, Secure Hash Algorithm (SHA-512)	1
4.2	Message Authentication Code (MAC) – Requirements, Uses	1
4.3	Hash-based MAC (HMAC), Cipher-based MAC (CMAC)	1
4.4	Digital signatures – Attacks, Forgeries, Requirements, Direct Vs Arbitrated digital signatures	1
4.5	RSA digital signature, ElGamal digital signature	1
4.6	Digital Signature Standard (DSS)	1
Module-5 (Key Distribution and System Security) (7hrs)		
5.1	Key management – Distribution of secret keys using symmetric and asymmetric encryption	1
5.2	Distribution of public keys	1
5.3	System security – Intruders, Intrusion detection techniques	1
5.4	Password management	1
5.5	Malicious software – Viruses, Related threats	1
5.6	Virus countermeasures	1
5.7	Distributed Denial of Service (DDoS) attacks – Types, Countermeasures	1

CST443	MODEL BASED SOFTWARE DEVELOPMENT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: The objective of the course is to familiarize learners about the concepts and advantages of using model based software development. This course covers the methodologies in developing the model of a software, perform analysis on the model and automatic generation of code from the model. The OSATE framework and its plugins using the Architecture Analysis and Design Language(AADL) language is used in the course to demonstrate the end-to-end concept of MBSD which helps the learners to get a hands on experience.

Prerequisite: Software Engineering

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain the relevance of model based software development in the software development process. **(Cognitive Knowledge level: Understand**)
CO2	Explain Model Driven Architecture with Computation Independent Model (CIM), Platform Independent Model(PIM), Platform Specific Model (PSM). (**Cognitive Knowledge level: Apply**)
CO3	Illustrate software modeling with Architecture Analysis and Design Language (AADL). (**Cognitive Knowledge level: Apply**)
CO4	Explain error annex using error modelling concepts and illustrate error modelling in AADL. (**Cognitive Knowledge level: Understand**)
CO5	Illustrate the process of code generation from an AADL model. (**Cognitive Knowledge level: Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination Marks
Remember	30	30	30
Understand	50	50	50
Apply	20	20	20
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to Model Based Software Development)

Software faults, Introduction to Model checking, Introduction to Automated Testing, Model Based Software Development (MBSD) – Need, MBSD Approach, Learning MBSD from the perspective of Architecture Analysis and Design Language (AADL).

Module - 2 (More on MBSD)

MBSD based software development – Requirements, Analysis, Design and Implementation. Model-Driven Architecture **-** Definitions and Assumptions, Overview of MBSD methodology, The modeling levels-Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM). Introduction to AADL, Basic Comparison of AADL with other modeling languages - Comparison with UML.

Module -3 (Modeling using AADL)

Modeling: Developing a Simple Model - Define the components - Explain with example (powerboat autopilot system), Develop a top-level model - Use example Powerboat Autopilot (PBA) system.

AADL: Components - Software, Hardware, Composite, Runtime semantics, Language syntax, AADL declarations, AADL classifiers, AADL system models and specifications

Case Study: Powerboat Autopilot System.

Module - 4 (Model Analysis)

Safety Analysis -Fault tree analysis, Minimal cutsets. Error Modeling in AADL-Error Model Libraries and Subclause Annotations, Error Types and Common Type Ontology, Error Sources and Their Impact, Component Error Behavior, Compositional Abstraction of Error Behavior, Use of Properties in Architecture Fault Models, Error modeling example.

Module - 5 (Code Generation)

Need for code generation, Categorization, Code Generation Techniques, Code Generation in AADL Model – Ocarina.

Text Books

1. Marco, Brambilla, Jordi Cabot, Manuel Wimmer, Model-Driven Software Engineering in Practice, 2/e, Synthesis Lectures on Software Engineering, 2017.
2. Christel Baier and Joost-Pieter Katoen, Principles of model checking, The MIT Press.
3. Thomas Stahl and Markus Volter, Model-Driven Software Development, Wiley, 2006.
4. David P. Gluch, Peter H. Feiler, Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language, Adison-Wesley, 2015.

References:

1. Automated software testing : http://www2.latech.edu
2. Peter H. Feiler, David P. Gluch, John J. Hudak.The Architecture Analysis & Design Language(AADL): An Introduction.
3. de Niz, Dionisio, Diagrams and Languages for Model-Based Software Engineering of EmbeddedSystems: UML and AADL
4. FAA System Safety Handbook, Chapter 8: Safety Analysis/Hazard Analysis Tasks
5. Enno Ruijters, Marielle Stoelinga, Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools.
6. Larson, Brian &Hatcliff, John & Fowler, Kim &Delange, Julien. (2013). Illustrating the AADL error modeling annex (v.2) using a simple safety-critical medical device. ACM SIGAda Ada Letters. 33. 65-84. 10.1145/2527269.2527271.
7. Delange, Julien&Feiler, Peter &Hudak, John &Gluch, Dave. (2016). Architecture Fault Modeling and Analysis with the Error Model Annex, Version 2. 10.13140/RG.2.1.4224.7927.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Justify the need of model based software development?
2. Explain the advantages of model based software development?

Course Outcome 2 (CO2):

1. Explain infrastructure of model driven architecture.

2. Describe about MDA modeling levels.

Course Outcome 3 (CO3):

1. Illustrate the basic components of an AADL Model.

2. Assume we have a system to regulate the fuel valve of a boiler by monitoring the steam flow and steam pressure. Identify the basic components of this system and design its AADL model.

Course Outcome 4 (CO4):

1. Suppose we have an isolette system which ensures the temperature is within a specified temperature range with following components:

   i) temperature sensor - detects air temperature.

   iii) heat source - supply hot air to maintain temperature.

   iv) operator interface - specify target temperature range(lower desired temperature, upper desired temperature.)

   iv) thermostat - takes as input an air temperature value from a temperature sensor and controls a heat source to produce an air temperature within a target range.

   Model the error flows, error propagations, component error behaviour and error properties for the value error in the isolette system.

Course Outcome 5 (CO5):

1. Illustrate code generation from an AADL model.

Model Question Paper

QP CODE:

Reg No: _______________

Name: __________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST443

Course Name: Model Based Software Development

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. List any three advantages of automated software testing.

2. Specify the steps and their purpose in the model checking process.

3. Compare Analysis And Design Language (AADL) with Unified modeling language (UML).

4. Describe the design phase in the model based software development process.

5. Represent interface component with an out data port and an out event port in AADL.

a) textual b)graphical

6. Give the textual top level model of a powerboat autopilot system in AADL.

7. What is an error type? Mention any two pre-declared timing and value errors in AADL.

8. Define : (i) Fault Tree Analysis (ii) Minimal cutsets

9. Explain templates and filtering code generation technique.

10. How does automated code generation help to deal with faults in a software system?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain model based software development approach. **(12)**

(b) Why is model based software development important? **(2)**

OR

12. (a) What are software faults? Mention any three software faults and its consequences. **(5)**

(b) Explain two approaches for ensuring software reliability? **(9)**

(i) Model Checking

(ii) Automated Testing

13. (a) Illustrate model based software development process. **(8)**

(b) Explain infrastructure of model driven architecture. **(6)**

OR

14. (a) What is AADL? Compare AADL and UML. **(6)**

(b) Explain in detail about MDA modeling levels. **(8)**

15. (a) Illustrate the components of an AADL model. **(12)**

(b) What is the AADL language syntax? **(2)**

OR

16. (a) Explain the following:

i) AADL classifiers

ii) AADL declarations **(2)**

(2)

(b) Design an AADL model which controls the speed of a vehicle. Also describe the basic components of the designed model. **(10)**

17. (a) Illustrate how value error can be modelled using AADL in the isolette system. **(10)**

(b) With a diagram explain error propagation, termination and transformation in AADL models. **(4)**

OR

18. (a) Illustrate error state machines in AADL using proper textual representations. **(8)**

(b) Suppose we have a train door controller system with following components **(6)**

i) door_controller - ensures safe opening of the door.

ii) train_controller - sends train speed and transit status to the door_controller.

iii) alarm - triggered when an emergency occurs in other components.

Model the error flows, error propagations, component error behaviour and error properties for the value error in the component door_controller.

19. (a) Explain templates and meta model type code generation? **(4)**

(b) Illustrate how the code can be generated from an AADL model. **(10)**

OR

20. (a) Describe any four code generation techniques. **(10)**

(b) Explain the advantages of automatic code generation. **(4)**

Teaching Plan

Sl No	**Contents**	**Number of Lecture Hours (35)**
	Module 1 (Introduction) (7 Hours)	
1.1	Software faults	1
1.2	Introduction to Model Checking	1
1.3	Introduction to Automated Testing (Lecture 1)	1
1.4	Introduction to Automated Testing (Lecture 2)	1
1.5	Need for MBSD, MBSD Approach	1
1.6	Architecture centric model driven software development	1
1.7	AADL and architecture-centric model-based software systems	1
	Module 2 (Model Based Software Development) (7 Hours)	
2.1	Model based software development process	1
2.2	Overview of MBSD methodology	1
2.3	Model Driven Architecture	1
2.4	MDA Definitions and Assumptions	1
2.5	The modeling levels	1
2.6	Introduction to AADL	1
2.7	Comparison of AADL with other modeling languages	1
	Module 3 (Modeling using AADL) (7 Hours)	
3.1	Modeling in detail: AADL components	1
3.2	Modeling in detail: Developing a simple model	1

3.3	Modeling in detail: Define top level model with an example	1
3.4	AADL in detail: Explain AADL components, Language syntax	1
3.5	AADL declarations and classifiers	1
3.6	AADL system models and specifications	1
3.7	Case study: Power boat auto pilot system	1
	Module 4 (Model Analysis)(7 Hours)	
4.1	Introduction to safety analysis	1
4.2	Fault tree analysis, minimal cutsets	1
4.3	Error modeling with AADL - Error Model Libraries and Subclause Annotations	1
4.4	Error modeling with AADL - Error Types and Common Type Ontology,	1
4.5	Error modeling with AADL - Error Sources and Their Impact, Component Error Behavior	1
4.6	Error modelling with AADL - Compositional Abstraction of Error Behavior, Use of Properties in Architecture Fault Models	1
4.7	Illustrate isolette error model	1
	Module 5 (Code Generation) (7 Hours)	
5.1	Code generation and its advantages	1
5.2	Categorization	1
5.3	Code generation techniques - Templates + filtering, Template + metamodel, Frame processors	1
5.4	Code generation techniques - API-based generators, In-line generation, Code attributes	1
5.5	Code generation techniques - Code weaving Commonalities and Differences Between the Different Code generation Approaches	1
5.6	Code generation in AADL - Ocarina	1
5.7	Illustration of code generation using AADL model	1

CST463	WEB PROGRAMMING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learners to understand the web programming concepts. It includes the essential frontend and backend technologies needed for the development of web applications. The learners will have an opportunity to gain necessary web development skills such as HTML, CSS, JavaScript, PHP, MySQL integration, JSON and Laravel framework.

Prerequisite: Knowledge of Programming is required.

Course Outcomes: After the completion of the course the student will be able to

CO1	Use HyperText Markup Language (HTML) for authoring web pages and understand the fundamentals of WWW. **(Cognitive Knowledge Level: Understand)**
CO2	Construct and visually format responsive, interactive web pages using CSS and JavaScript (JS) **(Cognitive Knowledge Level: Apply)**
CO3	Construct websites using advanced sever side programming tool PHP **(Cognitive Knowledge Level: Apply)**
CO4	Develop dynamic web applications using PHP and perform MySQL database operations. **(Cognitive Knowledge Level: Apply)**
CO5	Explain the importance of object exchange formats using JSON and the MVC based web application development frameworks (Laravel) **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓				✓							✓
CO2	✓	✓	✓		✓							✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓							✓

CO5	✓	✓			✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**20**	**20**	**20**
Understand	**40**	**40**	**40**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks.

First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus

Module – 1 (WWW, HTML)

Introduction to the Internet & WWW: Evolution of Internet & World Wide Web- Web Basics, URI's & URL-MIME.

Introduction to HTML5: Structuring & editing an HTML5 document, Fundamentals of HTML - Headings-Hyper Links- Images - Special Characters & Horizontal Rules-Lists- Tables -Forms - Internal Linking- Meta Elements-HTML5 Form input types -Input and Data List Elements and autocomplete attribute- Page Structure Elements -Multimedia-HTML5 Audio & video elements..

Module -2 (CSS, JavaScript)

Introduction to Stylesheets : Introduction to CSS-Basic syntax and structure-Inline Styles, Embedded Style Sheets, Conflict Resolution, Linking External Style Sheets-Exploring CSS Selectors-Properties, values, Positioning Elements: Absolute Positioning, Relative Positioning -

Backgrounds-List Styles-Element Dimensions- Table Layouts-Box Model and Text Flow-div and span -Basics of Responsive CSS, Media port & Media Queries.

Introduction to JavaScript : Introduction to Scripting- Programming fundamentals of JavaScript -Obtaining User Input with prompt Dialogs-Arithmetic-Decision Making -Control Statements -Functions -Arrays -Objects -Document Object Model (DOM) -Form processing

Module- 3 (PHP Basics)

PHP Language Structure: Introduction- **B**uilding blocks of PHP-Variables, Data Types -simple PHP program-Converting between Data Types- Operators and Expressions -Flow Control functions - Control statements- Working with Functions- Initialising and Manipulating Arrays--Objects- String Comparisons-String processing with Regular Expression

Module -4 (PHP- MySQL, JSON)

Advanced PHP: Form processing and Business Logic-Cookies- Sessions & MySQL Integration-Connecting to MySQL with PHP- Performing CREATE, DELETE, INSERT, SELECT and UPDATE operations on MySQL table -Working with MySQL data-Reading from Database-Dynamic Content.

Module- 5 (JSON, Laravel)

JSON Data Interchange Format: Syntax, Data Types, Object, JSON Schema, Manipulating JSON data with PHP

Web Development Frameworks: Laravel Overview**-**Features of Laravel-Setting up a Laravel Development Environment-Application structure of Laravel-Routing -Middleware-Controllers-Route Model Binding-Views-Redirections-Request and Responses.

Text Books

1 Paul J. Deitel, Harvey M. Deitel, Abbey Deitel, Internet & World Wide Web How to Program 5th Edition **[Module 1,2,3,4]**

2. Lindsay Bassett, Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON 1st Edition, O'Reilly **[Module 5]**

3. Julie C. Meloni, Pearson -PHP, MySQL & JavaScript All in One, Sams Teach Yourself,5th Ed [**Module 4]**

4. Matt Stauffer," LARAVEL up and Running, A framework for building modern PHP apps"1st Edition, O'REILLY **[Module 5]**

Reference Books

1. Robert W Sebesta, Programming the World Wide Web, 7/e, Pearson Education Inc,8th Edition
2. Larry Ullman, Pearson- PHP 6 and MySQL 5 for Dynamic Web Sites: Visual QuickPro Guide
3. Eric van der Vlist, Danny Ayers, Erik Bruchez, Joe Fawcett, Alessandro Vernet",Wrox-Professional Web 2.0 Programming, Wiley-India edition
4. Web Technologies Black Book 2018(As per Mumbai University Syllabus) HTML, CSS3, JavaScript, iQuery, AJAX,PHP,XML,MVC and Laravel DT Editorial Services (ISBN: 9789386052490)

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Construct a valid HTML document for your personal Profile registration page for a Job Site www.123Jobs.com. Add relevant HTML elements in a table, to accept a minimum of 10 different fields which includes your name, address, phone, email address, your picture, your college; your branch, fields for your personal history (Minimum 3 fields), favourite theory and practical subjects (Checkbox), Username, Password(password)

2. What is MIME? Give the expansion of MIME. List four examples for MIME types. State the reason why MIME type specification is necessary in a request-response transaction between a browser and server.

3. What is codec? Recognize the role of controls attribute in <video> & <audio> tag in HTML. Use the COVID vaccination promotional video ‘MySafety.mp4’ in a web page with suitable HTML code, ‘autoplay’ option enabled and displayed in a standard dimension 750 X500.

Course Outcome 2 (CO2):

1. Organize a sample web page for the event ‘Raagam2021’ at your campus and use embedded Style sheets to apply a minimum 5 styles. State the Style Specification format of embedded style sheets.

2. Write CSS style rules to implement the following in a web page:

   a. to display the content of hyperlinks with yellow background color and in italics

   b. to display the contents of unordered lists in bold and in Arial font

   c. to display a background image titled “birds.jpg” with no tiling.

3. Write the code for an HTML document with embedded JavaScript scripts, which initially displays a paragraph with text "Welcome" and a button titled "Click". When the button is clicked, the message "Hello from JavaScript" in bold should replace the paragraph text

Course Outcome 3 (CO3):

1. Write a PHP program to store the name and roll no of 10 students in an Associative Array and Use foreach loop to process the array and Perform asort, rsort and ksort in the array. Illustrate with suitable output data
2. Design an HTML page which enters a given number, write a PHP program to display a message indicating, whether the number is odd or even, when clicking on the submit button.
3. Write a PHP program to compute the sum of the positive integers up to 100 using do while.

Course Outcome 4 (CO4):

1. Write a PHP form handling program to verify the user authentication credentials of a web page using MySQL connection and store the userid value as a Session variable if the userid is valid.
2. Create a valid HTML document for yourself, including your name, address, and email address. Also add your college; your major and the course. Perform form handling in PHP and process the output using POST method.
3. Write an embedded PHP script which displays the factorial of all numbers from 1 to 10 in a table in the web page. The factorial should be calculated and returned from a function. The table headings should be "Number" and "Factorial"

Course Outcome 5 (CO5):

1. What is Route Model Binding in Laravel? Which types of route model binding are supported in Laravel?
2. Explain how laravel performs route handling using routes calling controller methods?
3. List the data types used in JSON? Explain the use of parse () and stringify() functions in JSON with examples.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST463

Course Name: Web Programming

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Define WWW. List any two examples of web server & web browser. Differentiate between URL and a domain?

2. Write the syntax of the URL? Rewrite the default URL of your university website by adding a subdomain named ‘Research’ and a web page named ‘FAQ.html’. Also link this URL through the logo of ‘kturesearch.png’ placed in a web page. The FAQ page should be opened in a new window.

3. Illustrate the implementation of a JavaScript function greeting () using external .js file, to display a welcome message, when you click on a Button in an HTML page.

4. What are different ways of adjusting spacing in a text with suitable example.

5. Discuss the various CSS style sheet levels with suitable examples. How are conflicts resolved when multiple style rules apply to a single web page element?

6. Describe how input from an HTML form is retrieved in a PHP program, with an example

7. Write a PHP program to check whether a number is prime number or not.

8. Discuss the various steps for establishing PHP-MySQL connection with a MySQL database ?

9. Describe the schema of a document implemented in JSON with suitable examples

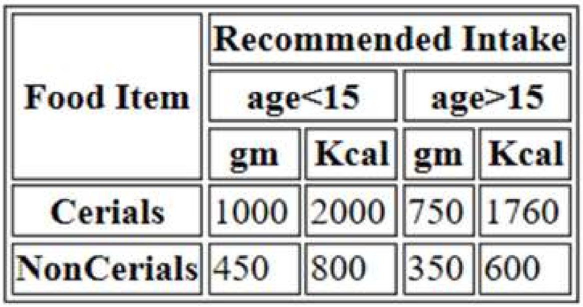
10. Explain the role of Resource controllers in Laravel.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Design a webpage that displays the following table. **(6)**



Food Item	Recommended Intake			
	age<15		age>15	
	gm	Kcal	gm	Kcal
Cerials	1000	2000	750	1760
NonCerials	450	800	350	600

(b) What is the difference between radio buttons and checkboxes when implemented using HTML? Write HTML code to implement a form which has the following elements: **(8)**

i. A textbox which can accept a maximum of 25 characters

ii. Three radio buttons with valid Label, Names and values

iii. Three check boxes buttons with valid Label, Names and values

iv. A selection list containing four items, two which are always visible

v. A submit button clicking on which will prompt the browser to send the form data to the server “http://www..mysite.com/reg.php” using “POST” method and reset button to clear its contents. You can use any text of your choice to label the form elements.

OR

12. (a) Write the equivalent HTML code to implement the following in a web page: (i) An image titled “birds.jpg” with a height of 100 pixels and width of 200 pixels. If the image cannot be accessed, a message “No image available” should be displayed (ii) A hyperlink to the URL “www.mysite.com/birds.jpg”. The hyperlink should have the label “Click Here”. **(6)**

(b) Create a static HTML document for your portfolio, which includes the following contents: your name, address, Mobile Number and email address. Also add the details about your college, university, your major and the batch **(8)** of study. Include a picture of yourself and at least one other image (friend/pet/role model) to the document with a short description about that. Add three paragraphs about your personal history, with links to your social media profile. Also create an ordered list for describing your Skill Set & an unordered list showing your Strengths & Weaknesses.

13. (a) Illustrate the usage of JavaScript DOM in event handling and explain any three methods with example. **(8)**

(b) Write CSS and the corresponding HTML code for the following: **(6)**

i. Set the background color for the hover and active link states to "green"

ii. Set the list style for unordered lists to "square".

iii. Set "Flower.png" as the background image of the page and set 3% margin for the pages

iv. Set dashed border for left and right and double border for top & bottom of a table with 2 rows.

OR

14. (a) List the order of precedence of style levels. Organize a sample web page for providing ‘KTU BTech Honours Regulation 19’ for KTU and use embedded Style sheet to apply minimum 5 styles for list, tables and pages. **(6)**

(b) Illustrate the different ways of Array declaration in JavaScript. Describe the function of the following JavaScript Array object methods with examples. **(8)**
(i) join (ii) slice

15. (a) Explain any six string handling functions used in PHP with example. **(6)**

(b) How does a PHP array differ from an array in C? List the different ways to create an array in PHP with an example. Explain any 4 functions that deals with PHP array. **(8)**

OR

16. (a) During the process of fetching a web page from a web server to a client browser, at what point does an embedded PHP script get executed. What are the two modes that the PHP processor operates in? Explain **(6)**

(b) Why is PHP considered to be dynamically typed? Distinguish between **(8)** implode and explode function in PHP with suitable examples.

17. (a) Write equivalent PHP statements corresponding to the following: **(8)**

i. Declare an associative array named "ages" to store the key-value pairs ("Alice", 30), ("Bob", 30), ("Harry", 35), ("Mary", 32).

ii. Modify the value associated with the key "Mary" to 28.

iii. Sort the array according to values maintaining the key-value relationships and print the sorted key-value pairs.

iv. The entry identified by the key "Bob"

(b) What are the uses of cookies in web pages? Describe syntax for setting cookies in PHP. How can you access and delete the cookie using setcookie() function? **(6)**

OR

18. (a) Write a PHP form handling program to perform the user registration of any website with a minimum of 5 different fields and insert the data into a MySQL table after establishing necessary connections with the DB, **(8)**

(b) Design the HTML page which enters a given number and embed the PHP code to display a message indicating, whether the number is odd or even, when clicking on the 'CHECK NUMBER' button. **(6)**

19. (a) With a neat diagram, explain about Laravel MVC Framework. **(6)**

(b) Discuss in detail about Laravel's Routing mechanisms. **(8)**

OR

20. (a) Enumerate the data types in JSON. Illustrate the document definition of a 'Student document 'using JSON Schema. **(8)**

(b) Discuss the following in Laravel Views **(6)**

i. Creating & Rendering Views

ii. Passing Data to Views

iii. Sharing Data with All Views

Teaching Plan

No	Contents	No of Lecture Hrs (35 hrs)
	Module 1 (7 hours)	
	Introduction to Internet and WWW	
1.1	Evolution of Internet &World Wide Web- Web Basics URI's & URL -MIME [Book 1 - Chapter 1]	1
	Introduction to HTML5	
1.2	Structuring & editing an HTML5 document- Fundamentals of HTML, Headings-Images [Book 1 - Chapter 2]	1
1.3	Hyper Links, Internal Linking- Lists [Book 1 - Chapter 2]	1
1.4	Special Characters & Horizontal Rules- meta Elements- div and span [Book 1 - Chapter 2]	1
1.5	Tables- Forms [Book 1 - Chapter 2]	1
1.6	HTML5 Form input types, input and data list Elements and autocomplete attributes-Page Structure Elements [Book 1 - Chapter 3]	1
1.7	Multimedia-HTML5 Audio & video elements [Book 1 - Chapter 9]	1
	Module 2 (10 hours)	
	Introduction to Cascading Style Sheets(CSS)	
2.1	Introduction to CSS3-Basic syntax and structure-Inline Styles [Book 1 - Chapter 4]	1
2.2	Embedded Style Sheets-Linking External Style Sheets [Book 1 - Chapter 4]	1
2.3	Exploring CSS Selectors-Properties-values [Book 1 - Chapter 4]	1
2.4	Positioning Elements: Absolute Positioning- Relative Positioning -Backgrounds-List Styles- Table Layouts [Book 1 - Chapter 4]	1

2.5	Box Model and Text Flow, Basics of Responsive CSS-Media port & Media Queries [Book 1 - Chapter 4]	1
Introduction to JavaScript		
2.6	Introduction to Scripting- Programming fundamentals of JavaScript -Obtaining User Input with prompt Dialogs [Book 1 - Chapter 6]	1
2.7	Arithmetic-Decision Making [Book 1 - Chapter 6]	1
2.8	Control Statements [Book 1 - Chapter 7]- Functions [Book 1 - Chapter 9]	1
2.9	Arrays [Book 1 - Chapter 10] - Objects [Book 1 - Chapter 11]	1
2.10	Document Object Model (DOM)- Form processing [Book 1 - Chapter 12,13]	1
Module 3 (6 hours)		
Introduction to PHP		
3.1	Building blocks of PHP-Variables, Data Types simple PHP program [Book 3- Chapters 4]	1
3.2	Converting between Data Types, Operators and Expressions -Flow Control functions [Book 1- Chapters 19]	1
3.3	Control Statements -Working with Functions [Book 3- Chapters 6]	1
3.4	Initialising and Manipulating Arrays- Objects [Book 1- Chapters 19]	1
3.5	Working with Strings-String processing with Regular expression, Pattern Matching [Book 1- Chapters 19]	1
3.6	Form processing and Business Logic [Book 1- Chapters 19]	1
Module 4 (6 hours)		
PHP -MYSQL		
4.1	Cookies- Sessions [Book 1- Chapters 19]	1
4.2	PHP& MySQL Integration-Connecting to MySQL with PHP . [Book 4- Chapters 18]	1

4.3	Working with MySQL data [Book 4- Chapters 18]	1
4.4	Performing CREATE, DELETE, INSERT operations on MySQL table from PHP Program. [Book 4- Chapters 16]	1
4.5	Performing SELECT and UPDATE operations on MySQL table from PHP Program. [Book 4- Chapters 16]	1
4.6	Building Dynamic Content in PHP application [Book1- Chapter19]	1
Module 5 (6 hours)		
	JSON	
5.1	JSON Data Interchange Format -Syntax, Data Types, Object [Book 2 - Chapters 1-2]	1
5.2	JSON Schema, Manipulating JSON data with PHP [Book 2 - Chapter 3,4]	1
	LARAVEL	
5.3	Laravel Overview**-** Design Pattern- Laravel Features [Book 4- Chapters 1] Setting up a Laravel Development Environment-Application structure of Laravel [Book 4- Chapters 2]	1
5.4	**Laravel Basics** Routing -middleware - Controllers [Book 4- Chapters 3]	1
5.5	Route Model Binding-Views-Redirections [Book 4- Chapters 3]	1
5.6	**Blade Templating-**echoing data, control structures [Book 4- Chapters 4]	1

CST473	NATURAL LANGUAGE PROCESSING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course enables the learners to understand the concepts of Natural Language Processing. The course covers basic pre-processing steps, language models, text classification using machine learning algorithms, information and relation extraction methods, Information Retrieval, Question Answer Systems and Machine Translation models. This course enables the students to apply techniques and methods to solve challenging real-world problems in NLP.

Prerequisite: Nil.

Course Outcomes: After the completion of the course the student will be able to

CO1	Summarize basic concepts and learning methods for NLP **(Cognitive Knowledge Level: Understand)**
CO2	Demonstrate the relevance of pre-processing methods on text data**(Cognitive Knowledge Level: Apply)**
CO3	Compare different language modelling techniques**(Cognitive Knowledge Level: Apply)**
CO4	Make use of NLP techniques in Text Classification and Information Retrieval**(Cognitive Knowledge Level: Apply)**
CO5	Explain Information Extraction, Relation Detection, QA Systems and Machine Translation**(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											✓
CO2	✓	✓	✓		✓							✓
CO3	✓	✓	✓		✓							✓
CO4	✓	✓	✓									✓
CO5	✓											✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Introduction to NLP)

NLP Tasks and Applications, Language-Building Blocks, Challenges of NLP, Machine Learning for NLP – Naïve Bayes Classifier, Logistic Regression, Support Vector Machines, Approaches to NLP-- Heuristics-Based NLP, Machine Learning-based NLP.

Module – 2 (Pre-processing and Representation Models)

NLP System Pipeline--Steps--Data Acquisition, Text Extraction and Clean-up, Pre-processing, Feature Engineering, Modelling, Evaluation, Post-Modelling Phases

Text Representation--Vector Space Models--Basic Vectorization Approaches--One-Hot Encoding, Bag of Words, Bag of N-Grams TF-IDF; Distributed Representations-- Word Embeddings, Doc2Vec.

Module - 3 (Classification and Information Extraction)

Text Classification--Text classification applications – Pipeline for building text classification systems, Naïve Bayes for Sentiment Classification – Naïve Bayes Classifier Training – Optimizing for Sentiment Analysis, Logistic Regression, Support Vector Machine for Text Classification

Information Extraction(IE)—IE Applications – The General Pipeline for IE - Named Entity Recognition(NER), Ambiguity in Named Entity Recognition – NER as Sequence Labeling – Evaluation of NER.

Module - 4 (Relation Detection and Information Retrieval)

Relation Detection and Classification – Supervised Learning Approaches to Relation Analysis – Lightly Supervised Approaches to Relation Analysis – Evaluation of Relation Analysis systems

Information Retrieval – Term weighting and document scoring – Inverted Index – Evaluation of Information Retrieval Systems.

Module - 5 (QA Systems and Machine Translation)

Question-Answering Systems – Factoid Question Answering – Question Processing – Passage Retrieval – Answer Processing – Evaluation of Factoid Answers

Machine Translation – Why Machine Translation is Hard – Classical Machine Translation – Direct Translation – Transfer – Statistical Machine Translation- The Phrase based Translation model – Alignment in MT – Training Alignment Models – Symmetrizing Alignments for Phrase-based MT – Decoding for Phrase-based Statistical MT

Text Books

1. Daniel Jurafsky, James H. Martin , "Speech and Language Processing"(2nd and 3rd editions), Pearson Prentice Hall
2. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana," Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems " June 2020 Publisher(s): O'Reilly Media, Inc. ISBN: 9781492054054.

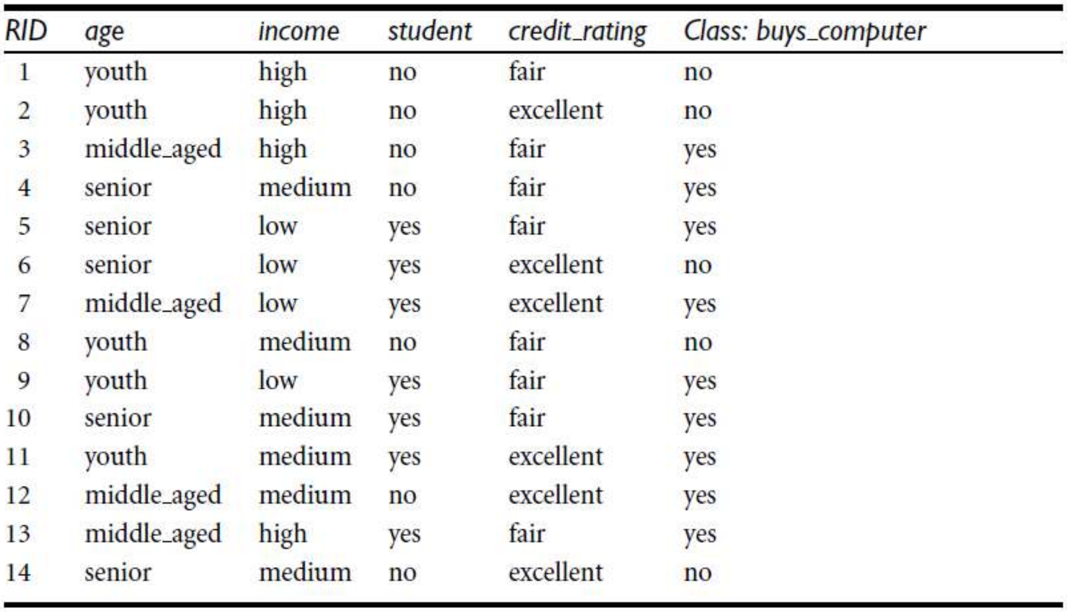
Reference Books

1. James Allen, "Natural Language Understanding", Second Edn , Pearson.
2. Christopher Manning and Hinrich Schutze, Statistical Natural Language Processing, MIT Press.

Course Level Assessment Questions

Course Outcome1 (CO1):

1. Explain the fundamental tasks that make up an NLP system.
2. Why is NLP considered a challenging problem domain?
3. The following table shows data about the profile of customers and whether they purchase computers or not. Given this data, use Naïve Bayes Classifier to classify the customer ***X*** (*age = youth, income = medium, student = yes, credit rating = fair*)



RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

4. Illustrate how linearly inseparable data can be made linearly separable by suitable mapping using kernel functions.

Course Outcome 2(CO2):

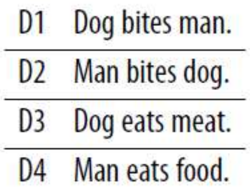
1. Mention two issues associated with sentence segmentation.

2. Show how is lemmatization done using Python Library.

3. Given a dataset of tweets, prepare the data for sentiment analysis by doing the following operations: conversion to lower casing, removal of punctuations, removal of stop-words, stemming, lemmatization, removal of emojis and removal of URLs. (Assignment Question)

Course Outcome 3(CO3):

1. Compare Bag-of-Words model and Bag-of-n-gram model.
2. Illustrate how TF-IDF model is used to represent text. Mention the advantage of TF-IDF over other models.
3. A corpus of data is given below :



D1	Dog bites man.
D2	Man bites dog.
D3	Dog eats meat.
D4	Man eats food.

Use one hot-encoding and Bag-of-words models to represent “dog bites man”.

Using the toy corpus given above, represent the sentence “Dog and Man eat meat” with TF-IDF model. Use python code for implementation. (Assignment Question)

Course Outcome 4(CO4): .

1. Given the following data about documents and contents, use tf-idf document scoring method to retrieve the document for the query “best game”

Doc 1	The game was so exciting. The players excelled in every department of the game.
Doc 2	It was an excellent game.
Doc 3	The game was not good. The moves were boring

2. A corpus of data is available from a social media platform that represents review of books. How can Naïve Bayes Classifier be used for sentiment analysis of the reviews? What changes can be made to this classifier to make it tuned for sentiment analysis.
3. Use python library to implement sentiment analysis of review of a book, given a toy corpus data set given below. (Assignment Question)

Document	Category
just plain boring	Negative
entirely predictable and lacks energy	Negative
no surprises and very few laughs	Negative
very powerful book	Positive
the best book of the summer	Positive

Course Outcome 5(CO5):

1. Explain lightly supervised approaches to relational analysis.

2. Explain a statistical algorithm for word alignment in Machine Translation.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST473

Course Name: Natural Language Processing

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Differentiate information extraction and information retrieval.

2. State Bayes' Theorem.

3. List three preprocessing steps that are necessary for an HTML file.

4. Differentiate CBOW and Skipgram models

5. Explain the role of support vectors in SVM Classification.

6. Explain challenges in Name Entity Recognition.

7. How is a Relational Analysis System evaluated?

8. Explain the need for an inverted index in an information retrieval system. Are there any more efficient data structures that serve the same purpose.

9. How do you extract answers to DEFINITION questions?

10. What are the components that make up a noisy channel model of statistical Machine Translation?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) How is classification done by SVM on linearly separable data? **(8)**

(b) What is a kernel function? What is the need for a kernel function? Can a kernel function be replaced by an ordinary mapping function? **(4)**

(c) Explain Heuristic-based NLP. **(2)**

OR

12. (a) Illustrate the steps involved in classification in Naïve Bayes Classifier. **(8)**

(b) Explain the fundamental tasks that make up an NLP system. **(6)**

13. (a) Supposing that a set of social media posts' dataset is available to do sentiment analysis. What pre-processing steps need to be done in order to use the data for generating a language model? Illustrate. **(8)**

(b) Illustrate Bag-of-ngrams model with an example. **(6)**

OR

14. (a) Explain the concept of word embeddings as a model for text representation. **(6)**

(b) Compare word embeddings model with vectorization approaches. **(4)**

(c) Explain the concept of feature engineering in NLP Systems. **(4)**

15. (a) 1. Given the following data about movie review and its classification, classify "predictable with no fun" to one of the classes using Naïve Bayes Classifier. **(10)**

Document	Category
just plain boring	Negative
entirely predictable and lacks energy	Negative
no surprises and very few laughs	Negative
very powerful	Positive
the most fun film of the summer	Positive

(b) Explain challenges in Name Entity Recognition. **(4)**

OR

16. (a) Explain Logistic Regression for Text Classification. **(6)**

(b) Explain Name Entity Recognition using Sequence Labeling. **(8)**

17. (a) Explain supervised approach to relation analysis. What are its limitations? **(10)**

(b) How is term selection done for indexing? **(4)**

OR

18. (a) Given the following data about documents and contents, use tf-idf document scoring method to retrieve the document for the query "sweet love". **(10)**

Doc 1	Sweet sweet nurse! Love
Doc 2	Sweet sorrow
Doc 3	How sweet is love?
Doc 4	Nurse!

(b) Explain the approaches to evaluate a relation analysis system. **(4)**

19. (a) Explain the phases of a factoid question-answering system. **(8)**

(b) Give an algorithm for word alignment in Machine Translation. **(6)**

OR

20. (a) How is decoding done in a Phrase-based Statistical Machine Translation System? **(10)**

(b) Explain the concept of Mean Reciprocal Rank. **(4)**

Teaching Plan

No	Contents	No of Lecture Hrs: 35
Module 1 : Introduction to NLP (7 hours)		
1.1	Introduction to NLP – Tasks and Applications	1
1.2	Language – Building Blocks, Challenges of NLP	1
1.3	Approaches to NLP - Heuristics-Based NLP, Machine Learning for NLP	1
1.4	Machine Learning for NLP – Naïve Bayes Classifier	1
1.5	Logistic Regression	1
1.6	Support Vector Machines – Linearly Separable Data	1
1.7	Support Vector Machines – Linearly Inseparable Data	1
Module 2 : Pre-processing and Representation Models(7 hours)		
2.1	NLP System Pipeline – Stages – Overview, Data Acquisition	1
2.2	NLP System Pipeline – Text Extraction and Cleanup	1
2.3	NLP System Pipeline – Preprocessing - Sentence segmentation, Word tokenization, Stemming and lemmatization	1
2.4	Feature Engineering, Model Building, Evaluation – Metrices, Post-modeling phase	1
2.5	Text Representation – Vector Space Model, Vectorization Approaches – One hot encoding, Bag of words	1
2.6	Bag of n-grams, TF-IDF	1
2.7	Word Embeddings – Word2Vec- CBOW, SkipGram models	1
Module 3: Classification and Information Extraction(7 hours)		
3.1	Text Classification--Text classification applications – Pipeline for building text classification systems	1
3.2	Sentiment Analysis using Naïve Bayes Classifier	1
3.3	Case Studies for Text Classification using Logistic Regression and	1

	Support Vector Machines	
3.4	Information Extraction (IE) and Applications, IE Tasks and the IE Pipeline	1
3.5	Named Entity Recognition (NER) – Ambiguity in NER	1
3.6	NER as Sequence Labeling	1
3.7	Evaluation of NER, Practical NER Systems	1
Module 4 : Relation Detection and Information Retrieval(5 hours)		
4.1	Relation Detection and Classification – Supervised Learning Approaches to Relation Analysis	1
4.2	Relation Detection and Classification – Lightly Supervised Approaches to Relation Analysis	1
4.3	Relation Detection and Classification -Evaluation of Relation Analysis systems	1
4.4	Information Retrieval – Term weighting and document scoring	1
4.5	Inverted Index, Evaluation of Information-Retrieval Systems	1
Module 5 : QA Systems and Machine Translation (9 hours)		
5.1	Question-Answering Systems – Factoid Question Answering, Question Processing	1
5.2	Passage Retrieval	1
5.3	Answer Processing, Evaluation of Factoid Answers	1
5.4	Machine Translation – Why Machine Translation is Hard	1
5.5	Classical Machine Translation	1
5.6	Statistical Machine Translation	1
5.7	The Phrase based Translation model	1
5.8	Alignment in Machine Translation	1
5.9	Decoding for Phrase-based Statistical MT	1

SEMESTER VII

OPEN ELECTIVE

CST415	INTRODUCTION TO MOBILE COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2019

Preamble: The purpose of this course is to prepare learners to understand the functionalities and design considerations of mobile computing. The course content is designed to cover the mobile computing architecture, features of different communication systems and major elements of mobile security and next generation computer systems. This course enables the learners to acquire advanced concepts on mobile and ad-hoc networks.

Prerequisite: A good knowledge of data communication and computer networks.

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Describe the mobile computing applications, services, design considerations and architectures(**Cognitive knowledge: Understand**)
CO2	Identify the technology trends for cellular wireless networks**(Cognitive knowledge:Understand)**
CO3	Summarize the Short Messaging Service and General Packet Radio Service **(Cognitive knowledge: Understand)**
CO4	Outline the LAN technologies used in mobile communication **(Cognitive knowledge: Understand)**
CO5	Describe the security protocols and apply suitable security algorithm to secure the communication **(Cognitive knowledge: Apply)**
CO6	Explain the fundamental concepts of next generation mobile networks**(Cognitive knowledge: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓									✓
CO6	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	50	50	50
Apply	20	20	20
Analyse			
Evaluate			

Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment Test	: **25 marks**
Continuous Assessment Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations shall be conducted for 50 marks. First series test shall be conducted preferably after completing the first half of the syllabus and the second series test shall be conducted preferably after completing the remaining part of the syllabus. There shall be two parts for the question paper: Part A and Part B. Part A shall contain five questions (preferably, two questions each from the fully completed modules and one question from the partly covered module), having three marks for each question adding up to 15 marks for part A. A student is expected to answer all questions from Part A. Part B shall contain seven questions (preferably, three questions each from the fully completed modules and one question from the partially completed module), each having seven marks. Out of the seven questions, a student is expected to answer any five.

End Semester Examination Pattern:

There shall be two parts; Part A and Part B. Part A shall contain 10 questions with 2 questions from each module, having 3 marks for each question. A student is expected to answer all questions from Part A. Part B shall contain 2 questions from each module, out of which a student is expected to answer any one. Each question shall have a maximum of two sub-divisions and shall carry 14 marks.

Syllabus

Module-1 (Mobile Computing Architecture)

Introduction to mobile computing – Functions, Devices, Middleware and gateways, Applications and services, Limitations. Mobile computing architecture – Internet: The ubiquitous network, Three-tier architecture, Design considerations for mobile computing.

Module-2 (Communication Systems)

Mobile computing through telephony - Evolution of telephony, Multiple access procedures - Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Space Division Multiple Access (SDMA). Satellite communication systems – Basics, Applications, Geostationary Earth Orbit (GEO), Low Earth Orbit (LEO), Medium Earth Orbit (MEO), Satellite phones. Mobile computing through telephone – Interactive Voice Response (IVR) architecture, Overview of voice software, Developing an IVR application. Global System for Mobile Communication (GSM) - Introduction, Architecture, Entities, Call routing, Mobility management, Frequency allocation, Authentication and security.

Module-3 (Short Messaging Service and General Packet Radio Service)

Short Message Service (SMS) – Strengths, Architecture, Value added services, Accessing the SMS bearer. General Packet Radio Service (GPRS) – Architecture, Network operations, Data services, Applications, Limitations, Billing and charging.

Module-4 (Wireless Local Area Networks)

Wireless Local Area Network (WLAN) - Advantages, Evolution, Applications, Architecture, Mobility, Security, Deploying WLAN. Wireless Local Loop (WLL) – Architecture. High Performance Radio Local Area Network (HIPERLAN). WiFi Vs 3G.

Module-5 (Mobile Security and Next Generation Networks)

Security issues in mobile computing - Information security, Security techniques and algorithms, Security protocols. Next generation networks – The Converged Scenario, Narrowband to broadband, Orthogonal Frequency Division Multiplexing (OFDM), Multi Protocol Label Switching (MPLS), Wireless Asynchronous Transfer Mode (WATM), Multimedia broadcast services.

Text Books

1. Asoke K. Talukder, Hasan Ahmad, Roopa R Yavagal, Mobile Computing Technology- Application and Service Creation, 2nd Edition, McGraw Hill Education.
2. Schiller J., Mobile Communications, 2/e, Pearson Education, 2009.

Reference Books

1. Andrew S. Tanenbaum, Computer Networks, 6/e, PHI.
2. Theodore S. Rappaport, Wireless Communications Principles and Practice, 2/e, PHI, New Delhi, 2004.
3. Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

Course Level Assessment Questions

Course Outcome 1 CO1):

1. Describe the design considerations in mobile computing.
2. Give five examples of mobile computing applications.

Course Outcome 2 (CO2):

1. Draw a call flow diagram for a theatre ticket booking system.
2. Illustrate the GSM architecture with figure.

Course Outcome 3 (CO3):

1. Illustrate the billing and charging services in GPRS.
2. Describe the SMS architecture.

Course Outcome 4 (CO4):

1. Compare IEEE 802.11, HIPERLAN with respect to their ad-hoc capabilities.
2. Discuss the security mechanism used in WLAN.

Course Outcome 5 (CO5):

1. With the help of a suitable example, show the working of Diffie-Hellman key exchange algorithm.
2. Bob chooses 7 and 11 as two prime numbers and chooses e as 13. Find an appropriate value for d and decrypt the plaintext 5 send by Alice to Bob.
3. Describe the security issues in mobile computing.

Course Outcome 6 (CO6):

1. Describe WATM and Multimedia broadcast services.
2. Describe the significance of Orthogonal Frequency Division Multiplexing (OFDM) in next generation networks.

Model Question Paper

QP CODE: **PAGES: 3**

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST415

Course Name : INTRODUCTION TO MOBILE COMPUTING

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Explain the different types of middleware and gateways required in mobile computing.
2. List any six limitations of mobile computing.
3. Compare and contrast the satellite systems – GEO, LEO and MEO.
4. How is frequency allocation done in GSM?
5. What are the various strengths of SMS?
6. How is billing and charging done in GPRS?
7. What are the different types of Wireless LANs?
8. Describe the architecture of a Wireless Local Loop.
9. Explain the key features of TLS protocol.
10. How are attacks classified?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Describe any four mobile computing functions. **(4)**

    (b) Explain the three-tier architecture of mobile computing with figure. **(10)**

OR

12. (a) Describe the significance and functions of core, edge and access network. **(6)**

(b) Explain the terms (i) Client Context Manager (ii) Policy Manager (iii) Security Manager (iv) Adaptability Manager **(8)**

13. (a) Why is multiple access important? With the help of suitable examples, explain the various multiple access techniques. **(7)**

(b) Describe the different algorithms used for security and authentication in GSM. **(7)**

OR

14. (a) Show how call routing is done in GSM. Give an example. **(7)**

(b) Explain the process of handover. How does handover differ from roaming? **(7)**

15. (a) With the help of neat sketches, explain the difference between Short Message Mobile Terminated (SM MT) and Short Message Mobile Originated (SM MO) messages. **(6)**

(b) Explain the network operations in GPRS. **(8)**

OR

16. (a) How does operator-centric pull differ from operator-independent push and pull? **(7)**

(b) Describe the data services and applications of GPRS. **(7)**

17. (a) Compare the HIPERLAN and OSI layered architecture. **(4)**

(b) Explain the 802.11 architecture. **(10)**

OR

18. (a) Compare 3G and WiFi. **(7)**

(b) Explain the HIPERLAN communication models with suitable diagrams. **(7)**

19. (a) Given p = 7, q = 17 and e = 5. Find the value of d and also encrypt the message P = 65 using RSA. **(7)**

(b) Explain the role of MPLS in service provisioning. **(7)**

OR

20. (a) With the help of a suitable example, show the working of Diffie-Hellman key exchange algorithm. **(7)**

(b) Explain the features of any three multimedia broadcast services. **(7)**

TEACHING PLAN

No	Contents	No.of Lecture Hrs (35 hrs)
	Module-1 (Mobile Computing Architecture) (6 hrs)	
1.1	Introduction to mobile computing – Functions, Devices, Middleware and gateways	1
1.2	Applications, services, limitations, Internet: The ubiquitous network	1
1.3	Three-tier architecture (Lecture 1)	1
1.4	Three-tier architecture (Lecture 2)	1
1.5	Design considerations for mobile computing (Lecture 1)	1
1.6	Design considerations for mobile computing (Lecture 2)	1
	Module-2 (Communication Systems) (7hrs)	
2.1	Evolution of telephony, Multiple access procedures – FDMA, TDMA, CDMA, SDMA	1
2.2	Satellite communication systems – GEO, MEO, LEO, Satellite phones	1
2.3	Interactive Voice Response (IVR) architecture, Overview of voice software, Developing an IVR application (Call flow diagram)	1
2.4	Introduction to GSM,Architecture	1
2.5	GSM entities, Call routing	1
2.6	Mobility management	1
2.7	Frequency allocation, Authentication and security	1
	Module-3 (Short Messaging Service and General Packet Radio Service) (8hrs)	
3.1	SMS Strengths, Architecture, Short Message Mobile Terminated (SM MT) and Short Message Mobile Originated (SM MO) messages	1
3.2	SMS Architecture - Operator-centric pull, operator-	1

	independent push/pull, Value added services	
3.3	Accessing the SMS bearer (Lecture 1)	1
3.4	Accessing the SMS bearer (Lecture 2)	1
3.5	GPRS architecture	1
3.6	Network operations	1
3.7	Data services, Applications	1
3.8	Limitations, Billing and charging	1
Module-4 (Wireless Local Area Networks) (7 hrs)		
4.1	WLAN Advantages, Evolution, Applications	1
4.2	WLAN Architecture (Lecture 1)	1
4.3	WLAN Architecture (Lecture 2)	1
4.4	Mobility, Security	1
4.5	Deploying WLAN	1
4.6	WLL Architecture, HIPERLAN	1
4.7	WiFi Vs 3G	1
Module-5 (Mobile Security and Next Generation Networks) (7hrs)		
5.1	Information security – Attacks, Components	1
5.2	Security techniques and algorithms – Stream Vs Block cipher, Symmetric Vs Asymmetric cryptography	1
5.3	Security techniques and algorithms – RSA, Diffie Hellman Key exchange	1
5.4	Security protocols – Secure Socket Layer, Transport Layer Security, Wireless Transport Layer Security	1
5.5	The Converged Scenario, Narrowband to broadband	1
5.6	Orthogonal Frequency Division Multiplexing (OFDM) and Multi Protocol Label Switching (MPLS)	1
5.7	Wireless Asynchronous Transfer Mode (WATM) and Multimedia broadcast services	1

CST425	INTRODUCTION TO DEEP LEARNING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2019

Preamble: This course aims to introduce the learner to an overview of the concepts and algorithms involved in deep learning. Basic concepts and application areas of machine learning, deep networks, convolutional neural network and recurrent neural network are covered in this course. This is a foundational program that will help students understand the capabilities, challenges, and consequences of deep learning and prepare them to participate in the development of leading-edge AI technology. They will be able to gain the knowledge needed to take a definitive step in the world of AI.

Prerequisite: Basics of linear algebra and probability.

Course Outcomes: After the completion of the course the student will be able to

CO1	Demonstrate basic concepts in machine learning.**(Cognitive Knowledge Level: Understand)**
CO2	Illustrate the validation process of machine learning models using hyper-parameters and validation sets. **(Cognitive Knowledge Level: Understand)**
CO3	Demonstrate the concept of the feed forward neural network and its training process. **(Cognitive Knowledge Level: Apply)**
CO4	Build CNN and Recurrent Neural Network (RNN) models for different use cases. **(Cognitive Knowledge Level: Apply)**
CO5	Use different neural network/deep learning models for practical applications. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	✓	✓										✓
CO2	✓	✓										✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		End Semester Examination Marks (%)
	Test 1 (%)	**Test 2 (%)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module-1 (Introduction)

Key components - Data, models, objective functions, optimization algorithms, Learning algorithms. Supervised learning- regression, classification, tagging, web search, page ranking, recommender systems, sequence learning, Unsupervised learning, Reinforcement learning, Historical Trends in Deep Learning. Other Concepts - overfitting, underfitting , hyperparameters and validation sets, estimators, bias and variance.

Module- 2 (Optimization and Neural Networks)

Neural Networks –Perceptron, Gradient Descent solution for Perceptron, Multilayer perceptron, activation functions, architecture design, chain rule, back propagation, gradient based learning. Introduction to optimization– Gradient based optimization, linear least squares. Stochastic gradient descent, Building ML algorithms and challenges.

Module -3 (Convolutional Neural Network)

Convolutional Neural Networks – convolution operation, motivation, pooling, Convolution and Pooling as an infinitely strong prior, variants of convolution functions, structured outputs, data types, efficient convolution algorithms.

Module- 4 (Recurrent Neural Network)

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs.

Module-5 (Application Areas)

Applications – computer vision, speech recognition, natural language processing. Research Areas – Autoencoders, Representation learning, Boltzmann Machines, Deep belief networks.

Text Book

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press 2015 ed.
2. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learning, August 2019.
3. Neural Networks and Deep Learning: A Textbook by Charu C. Aggarwal. Springer.1st edition, 2018.

Reference Books

1. Neural Smithing: Supervised Learning in Feed forward Artificial Neural Networks by Russell Reed, Robert J MarksII, 1st edition, 1999, MIT Press.
2. Practical Convolutional Neural Networks by Mohit Sewak, Md. Rezaul Karim, Pradeep Pujari, 1st edition, 2018, Packt Publishing Ltd.
3. Hands-On Deep Learning Algorithms with Python by Sudharsan Ravichandran, 1st edition, 2019, Packt Publishing Ltd.
4. Deep Learning with Python by Francois Chollet, 2nd edition, 2018, Manning Publications Co.

Sample Course Level Assessment Questions

Course Outcome1(CO1):

1. Compare regression and classification.
2. Define supervised learning? Distinguish between regression and classification.
3. Discuss the different learning approaches used in machine learning.
4. You train an initial model that achieves a 90% accuracy on the training dataset. What kind of problems your model is experiencing, and suggest a possible solution.
5. How does splitting a dataset into train, validation and test sets help identify overfitting?
6. Consider solving a classification task. You first train your network on 20 samples. Training converges, but the training loss is very high. You then decide to train this network on 10,000 examples. Is your approach to fixing the problem correct? If yes, explain the most likely results of training with 10,000 examples. If not, give a solution to this problem.

7. Describe one advantage of using mini-batch gradient descent instead of full-batch gradient descent.
8. Sketch the typical learning curves for the training and validation sets, for a setting where overfitting occurs at some point. Assume that the training set and the validation set are of the same size

Course Outcome 2(CO2):

1. What are hyperparameters? Why are they needed?
2. What issues are to be considered while selecting a model for applying machine learning in a given problem?
3. Update the parameters V11 in the given MLP using back propagation with learning rate as 0.5 and activation function as sigmoid. Initial weights are given as V11= 0.2, V12=0.1, V21=0.1, V22=0.3, V11=0.2, W11=0.5, W21=0.2
4. Draw the architecture of a multi-layer perceptron.
5. Derive update rules for parameters in the multi-layer neural network through the gradient descent.
6. Why is it important to place non-linearities between the layers of neural networks?
7. You design a fully connected neural network architecture where all activations are sigmoids. You initialize the weights with large positive numbers. Is this a good idea? Explain your answer.
8. You are doing full batch gradient descent using the entire training set (not stochastic gradient descent). Is it necessary to shuffle the training data? Explain your answer.
9. Consider training a fully-connected neural network with 5 hidden layers, each with 10 hidden units. The input is 20-dimensional and the output is a scalar. What is the total number of trainable parameters in your network?
10. Consider building a 10-class neural network classifier. Given a cat image, you want to classify which of the 10 cat breeds it belongs to. What loss function do you use? Introduce the appropriate notation and write down the formula of the loss function.
11. Why is the sigmoid activation function susceptible to the vanishing gradient problem?

Course Outcome 3 (CO3):

1. Give two benefits of using convolutional layers instead of fully connected ones for visual tasks.
2. Suppose that a CNN was trained to classify images into different categories. It performed well on a validation set that was taken from the same source as the training set but not on a testing set. What could be the problem with the training of such a CNN? How will you ascertain the problem? How can those problems be solved?
3. You are given a dataset of 10 x 10 grayscale images. Your goal is to build a 5-class classifier. You have to adopt one of the following two options: a) the input is flattened into a 100-dimensional vector, followed by a fully-connected layer with 5

neurons, b) the input is directly given to a convolutional layer with five 10 x 10 filters. Explain which one you would choose and why.

4. Weight sharing allows CNNs to deal with image data without using too many parameters. Does weight sharing increase the bias or the variance of a model?
5. Why do the layers in a deep architecture need to be non-linear?
6. A convolutional neural network has 4 consecutive layers as follows:
   3 x 3 conv (stride 2) - 2 x 2 Pool - 3 x 3 conv (stride 2) - 2 x 2 Pool
   How large is the set of image pixels which activate a neuron in the 4th non-image layer of this network?
7. Consider a convolution layer. The input consists of 6 feature maps of size 20 x 20. The output consists of 8 feature maps, and the filters are of size 5 x 5. The convolution is done with a stride of 2 and zero padding, so the output feature maps are of size 10 x 10. Determine the number of weights in this convolution layer

Course Outcome 4(CO4):

1. Explain how the cell state is updated in the LSTM model from Ct-1 to Ct
2. Show the steps involved in an LSTM to predict stock prices.
3. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.
4. If we have a recurrent neural network (RNN), we can view it as a different type of network by "unrolling it through time". Briefly explain what that means.
5. Briefly explain how “unrolling through time” is related to “weight sharing” in convolutional networks.
6. Explain how the cell state is updated in the LSTM model from Ct-1 to Ct
7. Show the steps involved in an LSTM to predict stock prices.Give one advantage of using an RNN rather than a convolutional network.

Course Outcome 5 (CO5):

1. Development a deep learning solution for problems in the domain i) natural language processing or ii Computer vision (Assignment)
2. Is an autoencoder for supervised learning or for unsupervised learning? Explain briefly.
3. Sketch the architecture of an autoencoder network.
4. Describe how to train an autoencoder network.
5. Write down the formula for the energy function (E) of a Restricted Boltzmann Machine (RBM).

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST425

Course Name: Introduction To Deep Learning

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Distinguish between supervised learning and Reinforcement learning. Illustrate with an example.
2. Differentiate classification and regression.
3. Compare overfitting and underfitting. How it can affect model generalization.
4. Why does a single perceptron cannot simulate simple XOR function? Explain how this limitation is overcome?
5. Illustrate the strengths and weaknesses of convolutional neural networks.
6. Illustrate convolution and pooling operation with an example
7. How many parameters are there in AlexNet? Why the dataset size (1.2 million) is important for the success of AlexNet?
8. Explain your understanding of unfolding a recursive or recurrent computation into a computational graph.
9. Illustrate the use of deep learning concepts in Speech Recognition.
10. What is an autoencoder? Give one application of an autoencoder

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) "A computer program is said to learn from experience E with respect to some class oftasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." What is your understanding of the terms task, performance and experience. Explain with two example **(10)**

    (b) "How does bias and variance trade-off affect machine learning algorithms? **(4)**

OR

12. (a) Illustrate the concepts of Web search, Page Ranking, Recommender systems with suitable examples. **(10)**

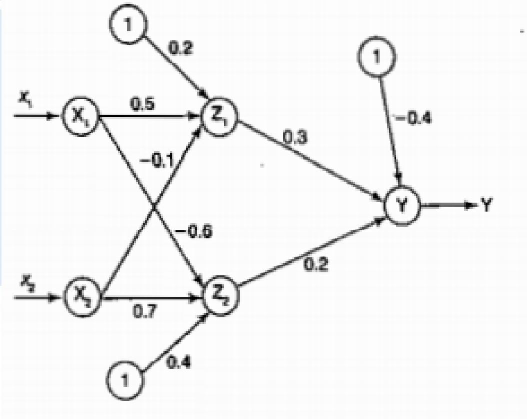
(b) List and discuss the different hyper parameters used in fine tuning the traditional machine learning models **(4)**

13. (a) How multilayer neural networks learn and encode higher level features from input features. **(7)**

(b) Explain gradient decent and delta rule? Why stochastic approximation to gradient descent is needed? **(7)**

OR

14. (a) Find the new weights for the network using backpropogation algorithm, the network is given with a input pattern[-1,1] and target output as +1, Use learning rate of alpha=0.3 and bipolar sigmoid function. **(7)**



(b) Write an algorithm for backpropagation which uses stochastic gradient descent method. Comment on the effect of adding momentum to the network. **(7)**

15. (a) Input to CNN architecture is a color image of size 112x112x3. The first convolution layer comprises of 64 kernels of size 5x5 applied with a stride of 2 and padding 0. What will be the number of parameters? **(5)**

(b) Let X=[-1, 0, 3, 5] W=[.3 ,.5 .2,.1] be the the input of ith layer of a neural network and to apply softmax function. What should be the output of it? **(4)**

(c) Draw and explain the architecture of convolutional network **(5)**

OR

16. (a) Explain the concept behind i) Early stopping ii) dropout iii) weight decay **(9)**

(b) How backpropagation is used to learn higher-order features in a convolutional Network? **(5)**

17. (a) Explain the working of RNN and discuss how backpropagation through time is used in recurrent networks. **(8)**

(b) Describe the working of a long short term memory in RNNs. **(6)**

OR

18. (a) What is the vanishing gradient problem and exploding gradient problem? **(8)**

(b) Why do RNNs have a tendency to suffer from exploding/vanishing gradient? How to overcome this challenge? **(6)**

19. (a) Explain any two word embedding techniques **(8)**

(b) Explain the merits and demerits of using Auto encoders in Computer Vision. **(6)**

OR

20. (a) Illustrate the use of representation learning in object classification. **(7)**

(b) Compare Boltzmann Machine with Deep Belief Network. **(7)**

Teaching Plan

No	Contents	No. of Lecture Hours (37 hrs)
Module 1 : Introduction (8 hours)		
1.1	Key components - Data, models, objective functions, optimization algorithms. (TB2: Section 1.1-1.2)	1
1.2	Learning algorithm (TB1: Section 5.1), Supervised learning- regression, classification, tagging, web search, page ranking (TB2: Section 1.3.1)	1
1.3	Recommender systems, Sequence learning, Unsupervised learning, Reinforcement learning(TB2: Section 1.3.2-1.3.4)	1
1.4	Historical Trends in Deep Learning (TB1: Section 1.2).	1
1.5	Concepts: overfit, underfit, hyperparameters and validation sets. (TB1: Section 5.2-5.3)	1
1.6	Concepts: Estimators, bias and variance. (TB1: Section 5.4)	1
1.7	Demonstrate the concepts of supervised learning algorithms using a suitable platform.	1
1.8	Demonstrate the concepts of unsupervised using a suitable platform.	1
Module 2 : Optimization and Neural Networks (9 hours)		
2.1	Perceptron, Stochastic Gradient descent, Gradient descent solution for perceptron (TB3: Section 1.1 - 1.2.1)	1
2.2	Multilayer perceptron (TB3: Section 1.2.2), (TB1: Section 6.1,6.3)	1
2.3	Activation functions- Sigmoid, tanh, Softmax, ReLU, leaky ReLU (TB3: Section 1.2.1.3 - 1.2.1.5)	1
2.4	Architecture design (TB1: Section 6.4, TB3: Section 1.6)	1
2.5	Chain rule, back propagation (TB3: Section 1.3)	1
2.6	Gradient based learning (TB1: Section 6.2)	1
2.7	Gradient based optimization (TB1: Section 4.3)	1
2.8	Linear least squares using a suitable platform. (TB1: Section 4.5)	1
2.9	Building ML Algorithms and Challenges (TB3: 1.4, TB1: 5.10-5.11)	1
Module 3 :Convolution Neural Network (8 hours)		
3.1	Convolution operation (TB1:Section 9.1)	1
3.2	Motivation, pooling (TB1:Section 9.2-9.3)	1

3.3	Convolution and Pooling as an infinitely strong prior (TB1: Section 9.4)	1
3.4	Variants of convolution functions – multilayer convolutional network, tensors, kernel flipping, downsampling, strides and zero padding. (TB1: Section 9.5)	1
3.5	Variants of convolution functions - unshared convolutions, tiled convolution, training different networks. (TB1: Section 9.5)	1
3.6	Structured outputs, data types (TB1: Section 9.6-9.7)	1
3.7	Efficient convolution algorithms. (TB1: Section 9.8,9.10)	1
3.8	Case Study: AlexNet, VGG, ResNet. (TB3: Section 8.4.1, 8.4.3, 8.4.5)	1
Module 4 :Recurrent Neural Network (7 hours)		
4.1	Computational graphs (TB1: Section 10.1)	1
4.2	RNN (TB1: Section 10.2-10.3)	1
4.3	Encoder – decoder sequence to sequence architectures. (TB1: Section 10.4)	1
4.4	Deep recurrent networks (TB1: Section 10.5)	1
4.5	Recursive neural networks , Modern RNNs, LSTM and GRU (TB1: Section 10.6, 10.10)	1
4.6	Practical use cases for RNNs. (TB1: Section 11.1-11.4)	1
4.7	Demonstrate the concepts of RNN using a suitable platform.	1
Module 5 : Applications and Research (5 hours)		
5.1	Computer vision. (TB1: Section 12.2)	1
5.2	Speech recognition. (TB1: Section 12.3)	1
5.3	Natural language processing. (TB1: Section 12.4)	1
5.4	Brief introduction on current research areas- Autoencoders, Representation learning. (TB1: Section 14.1-14.2, TB3: 9.3)	1
5.5	Brief introduction on current research areas- Boltzmann Machines, Deep belief networks. (TB1: Section 20.1, 20.3)	1

CST435	COMPUTER GRAPHICS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2019

Preamble: This course helps the learners to make awareness about strong theoretical concept in computer graphics. It covers the three-dimensional environment representation in a computer, transformation of 2D/3D objects and basic mathematical techniques and algorithms used to build applications. This course enables the learners to develop the ability to create image processing frameworks for different domains and develop algorithms for emerging display technologies.

Prerequisite: A sound knowledge of Mathematics and concepts of any programming language.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Describe the working principles of graphics devices**(Cognitive Knowledge level: Understand)**
CO2	Illustrate line drawing, circle drawing and polygon filling algorithms**(Cognitive Knowledge level: Apply)**
CO3	Demonstrate geometric representations and transformations on 2D & 3D objects **(Cognitive Knowledge level: Apply)**
CO4	Demonstrate the working of line and polygon clipping algorithms**(Cognitive Knowledge level: Apply)**
CO5	Summarize visible surface detection methods and illustrate projection algorithms. **(Cognitive Knowledge level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓										✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of SeriesTests1& 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one full question. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1(Basics of Computer graphics)

S

Basics of Computer Graphics and its applications. Video Display devices- Refresh Cathode Ray Tubes(CRT), Random Scan Displays and systems, Raster scan displays and systems, Color CRT displays, Flat panel display and its categories.

Module – 2 (Line drawing, Circle drawing and Filled Area Primitives)

Line drawing algorithms- DDA, Bresenham’s algorithm. Circle drawing algorithms- Midpoint Circle generation algorithm, Bresenham’s algorithm. Filled Area Primitives- Scan line polygon filling, Boundary filling and flood filling.

Module - 3 (Geometric transformations)

Two dimensional transformations-Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations.

Module - 4 (Clipping)

Window to viewport transformation. Cohen Sutherland and Midpoint subdivision line clipping algorithms, Sutherland Hodgeman and Weiler Atherton Polygon clipping algorithms.

Module - 5 (Three dimensional graphics)

Three dimensional viewing pipeline. Projections- Parallel and Perspective projections. Visible surface detection algorithms- Back face detection, Depth buffer algorithm, Scan line algorithm, A buffer algorithm

Text Book

1. Zhigang Xiang and Roy Plastock, Computer Graphics (Schaum's outline Series), McGraw Hill, 2019.
2. Donald Hearn and M. Pauline Baker, Computer Graphics, PHI, 2e, 1996

References

1. William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics. McGraw Hill, 2001
2. David F. Rogers , Procedural Elements for Computer Graphics, Tata McGraw Hill,2001.
3. Donald Hearn, M. Pauline Baker and Warren Carithers, Computer Graphics with OpenGL, PHI, 4e, 2013

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare the working principle of raster scan systems and random scan systems.
2. How much time is spent scanning across each row of pixels during screen refresh on a raster system with resolution of 1280*1024 and a refresh rate of 60 frames per second?

Course Outcome 2 (CO2):

1. Rasterize the line using Bresenham's line drawing algorithm with end points (2,3) and (5,8) accepted from the user and implement it using any appropriate programming language. (Assignment)
2. Illustrate how the 4-connected boundary filling approach differs from 8-connected boundary filling and implement it using any appropriate programming language. (Assignment)

Course Outcome 3 (CO3):

1. Rotate a triangle ABC 45 degree counter clockwise about the pivot point (10,3) , where the
   position vector of the coordinates is given as A(4,1), B(5,2) and C(4,3).
2. Implement the above transformation using any appropriate programming language with user inputs. (Assignment)
3. Illustrate the steps required for a general 3D rotation if the rotation axis is not parallel to any one of the principal axis. The rotation axis is defined by the points P1(x1,y1,z1) and P2(x2,y2,z2). Give its composite matrix representation.

Course Outcome 4 (CO4):

1. Given a clipping window A(20,20), B(60,20), C(60,40) and D(20,40). Using Cohen Sutherland algorithm, find the visible portion of the line segment joining the points P(40,80) and Q(120,30).

2. Implement Cohen Sutherland clipping algorithm using any appropriate programming language with user inputs. (Assignment)

Course Outcome 5 (CO5):

1. Explain scan line algorithm for detecting visible surfaces in an object.
2. Derive the matrix for performing perspective projection and parallel projection.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST435

Course Name: Computer Graphics

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Describe Flat Panel display and its categories.

2. Consider a raster system with a resolution of 1024*1024. Compute the size of the raster needed to store 4 bits per pixel? How much storage is needed if 8 bits per pixel are to be stored?

3. Justify the usage of integer arithmetic in Bresenham's line drawing algorithm.

4. How 8-way symmetry of circle can be used for developing circle drawing algorithms?

5. Show that two successive reflections about either of the coordinate axes is equivalent to a single rotation about the coordinate origin.

6. Determine a sequence of basic transformations that is equivalent to x-direction shearing.

7. Find the window to viewport normalization transformation with window lower left corner at (1,1) and upper right corner at (2,6).

8. How does Cohen Sutherland algorithm determine whether a line is visible, invisible or a candidate for clipping based on the region codes assigned to the end points of the line?

9. Define the terms (i) Centre of projection (ii) Principal vanishing point

10. Differentiate between the object space and image space method for the hidden surface removal of an image. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the working principle of beam penetration method and shadow mask method with suitable illustrations. **(8)**

(b) Draw the architecture of raster scan display systems and explain its working principle. **(6)**

OR

12. (a) Explain the working principle of a Refresh CRT monitor with suitable diagrams. **(8)**

(b) Describe random graphics system with suitable illustrations. **(6)**

13. (a) Differentiate between boundary fill and flood fill algorithms. **(5)**

(b) Derive the initial decision parameter of Bresenham's line drawing algorithm and rasterize a line with endpoints (2,2) and (10,10). **(9)**

OR

14. (a) Write Midpoint circle drawing algorithm and identify the points in the circle with radius as 20 and center at (50,30) using the algorithm. **(8)**

(b) Illustrate the working principle of scan line polygon filling algorithm. **(6)**

15. (a) Reflect a triangle ABC about the line 3x-4y+8=0, where the coordinates of the triangle are given as A(4,1), B(5,2) and C(4,3). **(8)**

(b) A diamond shaped polygon is located at P(-1,0), Q(0,-2), R(1,0) and S(0,2). Find the transformation matrix which would rotate the triangle by 90 degree counter clockwise about the point Q. Using the transformation matrix, find the coordinates of the rotated polygon**.** **(6)**

OR

16. (a) Describe the steps required for a general 3D rotation if the rotation axis is not parallel to any one of the principal axis. The rotation axis is defined by the points P1(x1,y1,z1) and P2(x2,y2,z2). Give its composite matrix representation. **(8)**

(b) Consider a triangle at (2,2), (10,2), (2,10). Perform the following 2D transformations in succession and find the resultant vertices. **(6)**

i) Scale with respect to (2,2) by scaling factors (2,2) along x and y directions.

ii) Rotate by 90 degree counter clockwise direction.

iii) Reflection based on y=x

17. (a) Illustrate Weiler – Atherton polygon clipping algorithm. **(6)**

(b) Explain Cohen-Sutherland line clipping algorithm. Use the algorithm to clip line with end points P1 (70, 20) and P2(100,10) against a window with lower left hand corner (50,10) and upper right hand corner (80,40). **(8)**

OR

18. (a) Describe Sutherland Hodgeman polygon clipping algorithm and list out its limitations. **(7)**

(b) Explain the steps involved in clipping a line using Mid point Subdivision algorithm. **(7)**

19. (a) Explain how visible surfaces can be detected using depth buffer algorithm. **(7)**

(b) Define parallel projection. Describe orthographic and oblique parallel projection. **(7)**

OR

20. (a) Illustrate the scan line method used in visible surface detection. **(7)**

(b) Derive the matrix needed for performing perspective projections. **(7)**

TEACHING PLAN

No	Contents	No of Lecture Hrs (35 hrs)
Module – 1 (Basics of Computer Graphics) (6 hrs)		
1.1	Basics of Computer Graphics and applications	1
1.2	Refresh Cathode Ray Tubes	1
1.3	Random Scan Displays and systems	1
1.4	Raster scan displays and systems	1
1.5	Color CRT displays	1
1.6	Flat panel display and its categories.	1
Module - 2 (Line drawing, Circle drawing and Filled Area Primitives) (7 hrs)		
2.1	DDA Line drawing Algorithm	1
2.2	Bresenham’s line drawing algorithm	1
2.3	Midpoint Circle generation algorithm	1
2.4	Bresenham’s Circle generation algorithm	1
2.5	Illustration of line drawing and circle drawing algorithms	1
2.6	Scan line polygon filling	1
2.7	Boundary filling and flood filling	1
Module - 3 (Geometric transformations) (8 hrs)		
3.1	Basic 2D transformations-Translation and Rotation	1
3.2	Basic 2D transformations- Scaling	1
3.3	Reflection and Shearing	1
3.4	Illustration of 2D Transformations	1
3.5	Composite transformations	1
3.6	Matrix representations and homogeneous coordinates	1
3.7	Basic 3D transformations	1
3.8	Illustration of basic 3D transformations	1
Module - 4 (2D Clipping) (6 hrs)		
4.1	Window to viewport transformation	1
4.2	Cohen Sutherland Line clipping algorithm	1
4.3	Midpoint subdivision Line clipping algorithm	1
4.4	Sutherland Hodgeman Polygon clipping algorithm	1
4.5	Weiler Atherton Polygon clipping algorithm	1
4.6	Practice problems on Clipping algorithms	1
Module - 5 (Three dimensional graphics)(8 hrs)		
5.1	Three dimensional viewing pipeline, Projections-Parallel projections	1

5.2	Projections- Perspective projections	1
5.3	Visible surface detection algorithms- Back face detection.	1
5.4	Depth buffer algorithm	1
5.5	Depth buffer algorithm	1
5.6	Scan line visible surface detection algorithm	1
5.7	Scan line visible surface detection algorithm	1
5.8	A buffer algorithm	1

CST445	PYTHON FOR ENGINEERS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2019

Preamble: The objective of the course is to provide learners an insight into Python programming in a scientific computation context and develop programming skills to solve engineering problems. It covers programming environment, important instructions, data representations, intermediate level features, Object Oriented Programming and file data processing of Python. This course lays the foundation to scientific computing, develop web applications, Machine Learning, and Artificial Intelligence-based applications and tools, Data Science and Data Visualization applications.

Prerequisite: NIL

Note : ***Students who have successfully completed CST 283 - Python for Machine Learning (Minor) are not eligible to opt this course.***

Course Outcomes: After the completion of the course the student will be able to

CO1	Write, test and debug Python programs **(Cognitive Knowledge level: Apply)**
CO2	Illustrate uses of conditional (if, if-else, if-elif-else and switch-case) and iterative (while and for) statements in Python programs **(Cognitive Knowledge level: Apply)**
CO3	Develop programs by utilizing the modules Lists, Tuples, Sets and Dictionaries in Python **(Cognitive Knowledge level: Apply)**
CO4	Implement Object Oriented programs with exception handling **(Cognitive Knowledge level: Apply)**
CO5	Analyze, Interpret, and Visualize data according to the target application (**Cognitive Knowledge level: Apply**)
CO6	Develop programs in Python to process data stored in files by utilizing the modules Numpy, Matplotlib, and Pandas **(Cognitive Knowledge level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓										✓
CO3	✓	✓										✓
CO4	✓	✓	✓		✓							✓
CO5	✓	✓	✓		✓							✓
CO6	✓	✓	✓		✓							✓

Abstract POs defined by National Board of Accreditation

#PO	Broad PO	#PO	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (*Marks in percentage*)	Test 2 (*Marks in percentage*)	End Semester Examination Marks
Remember	20	20	20
Understand	30	30	30
Apply	50	50	50
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 sub-divisions and carries 14 marks.

Syllabus

Module 1 (Basics of Python)

Getting Started with Python Programming - Running code in the interactive shell, Editing, Saving, and Running a script. Using editors - IDLE, Jupyter. Basic coding skills - Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions, Working with numeric data, Type conversions, Comments in the program, Input Processing, and Output, Formatting output. How Python works. Detecting and correcting syntax errors. Using built in functions and modules in math module. Control statements - Selection structure - if-else, if-elif-else. Iteration structure - for, while. Testing the control statements. Lazy evaluation.

Module 2 (Functions and Python Data Structures)

Functions - Hiding redundancy and complexity, Arguments and return values, Variable scopes and parameter passing, Named arguments, Main function, Working with recursion, Lambda functions. Strings - String function. Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension. Work with tuples. Sets. Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup.

Module 3 (Object Oriented Programming)

Design with classes - Objects and Classes, Methods, Instance Variables, Constructor, Accessors and Mutators. Structuring classes with Inheritance and Polymorphism. Abstract Classes. Exceptions - Handle a single exception, Handle multiple exceptions.

Module 4 (Visualization and File handling)

Plotting - An Interactive Session with PyPlot, Basic Plotting, Logarithmic Plots, More Advanced Graphical Output, Plots with multiple axes, Mathematics and Greek symbols, The Structure of matplotlib, Contour and Vector Field Plots. File Processing - The os and sys modules, Introduction to file I/O, Reading and writing text files, Working with CSV files.

Module 5 (Scientific Computing)

Numerical Routines. SciPy and NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Special Functions, Random Numbers, Linear Algebra, Solving Nonlinear Equations, Numerical Integration, Solving ODEs. Data Manipulation and Analysis – Pandas : Reading Data from Files Using Pandas, Data Structures: Series and DataFrame, Extracting Information from a DataFrame, Grouping and Aggregation.

Text Books:

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. David J. Pine, Introduction to Python for Science and Engineering, CRC Press, 2021

Reference Books:

1. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017
2. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
3. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016
4. David M.Baezly, Python Essential Reference. Addison-Wesley Professional; 4/e, 2009.
5. Charles Severance. Python for Informatics: Exploring Information,
6. http://swcarpentry.github.io/python-novice-gapminder/

Sample Course Level Assessment Questions

Course Outcome1(CO1):

1. What is type conversion? How is it done in Python?

Course Outcome 2(CO2):

1. Given is a list of of words, *wordlist*, and a string, *name.* Write a Python function which takes *wordlist* and *name* as input and returns a tuple. The first element of the output tuple is the number of words in the *wordlist* which have *name* as a substring in it. The second element of the tuple is a list showing the index at which the *name* occurs in each of the words of the *wordlist* and a 0 if it doesn't occur.

Course Outcome 3(CO3):

1. Write a Python program to implement the addition, subtraction, and multiplication of complex numbers using classes. Use constructors to create objects. The input to the program consist of real and imaginary parts of the complex numbers.

Course Outcome 4(CO4):

1. Plot the function $y = 3x^2$ for $-1 \leq x \leq 3$ as a continuous line. Include enough points so that the curve you plot appears smooth. Label the axes x and y

Course Outcome 5(CO5):

1. Given a file "auto.csv" of automobile data with the fields *index, company, body-style, wheel-base, length, engine-type, num-of-cylinders, horsepower, average-mileage,* and *price***,** write python code to
   1. Clean and Update the CSV file
   2. Print total cars of all companies
   3. Find the average mileage of all companies
   4. Find the highest priced car of all companies.

Model Question Paper

QP CODE: PAGES:

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR
Course Code: CST445

Course name : PYTHON FOR ENGINEERS

Max Marks: 100 **Duration: 3 Hours**

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Explain the basic data types available in Python, with examples.
2. Write a Python program to reverse a number and also find the sum of digits of the number. Prompt the user for input.
3. Compare tuples, lists, and dictionaries.
4. Explain the concept of scope and lifetime of variables in Python programming language, with a suitable example.
5. What is polymorphism? Give an example in the context of OOP in Python.
6. How is exception handling accomplished in Python programs?
7. Describe the characteristics of the CSV format.

8. Plot the function y = $3x^2$ for $-1 \leq x \leq 3$ as a continuous line. Include enough points so that the curve you plot appears smooth. Label the axes x and y

9. Describe random number generation using Python

10. How can a generalized eigen value problem can be solved using Python?

PART-B

(Answer any one full question from each module)

Module -1

11. (a) Compare and contrast interpreted languages and compiled languages. How does it affect the quality of program development and execution of the program? (6)

(b) What are the possible errors in a Python program. Write a Python program to print the value of $2^{2n}+n+5$ for *n* provided by the user. (8)

OR

12. (a) Describe Arithmetic operators, Assignment operators, Comparison operators, Logical operators, and Bitwise operators in detail with examples. (6)

(b) Input 4 integers (+ve and −ve). Write a Python code to find the sum of negative numbers, positive numbers, and print them. Also, find the averages of these two groups of numbers and print (8)

Module -2

13. (a) Write a Python code to create a function called *list_of_frequency* that takes a string and prints the letters in non-increasing order of the frequency of their occurrences. Use dictionaries. (5)

(b) Write a Python program to read a list of numbers and sort the list in a non-decreasing order without using any built in functions. Separate function should be written to sort the list wherein the name of the list is passed as the parameter. (9)

OR

14. (a) Illustrate the following Set methods with an example. (8)

i. *intersection( )* ii. *Union( )* iii. *Issubset( )* iv. *Difference( )* v. *update( )* vi. *discard( )*

(b) Write a Python program to check the validity of a password given by the user. (6)

The Password should satisfy the following criteria:

1. Contains at least one letter between a and z

2. Contains at least one number between 0 and 9

3. Contains at least one letter between A and Z

4. Contains at least one special character from $, #, @

5. Minimum length of password: 6

Module -3

15. (a) How can a class be instantiated in Python? Write a Python program to express the instances as return values to define a class RECTANGLE with parameters *height, width, corner_x,* and *corner_y* and member functions to find center, area, and perimeter of an instance. (5)

(b) Explain inheritance in Python. Give examples for each type of inheritance. (9)

OR

16. (a) Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a given circle (6)

(b) Define a class in Python to store the details of a ship (name, source,destination) with the following methods: (8)

i) *get_details( )* - to assign values to class attributes

ii) *print_details( )* - to display the attribute values

Create an object of the class and invoke the methods

Module -4

17. (a) Plot the functions *sin x* and *cos x* vs *x* on the same plot with x going from −π to π. Make sure the limits of the *x*-axis do not extend beyond the limits of the data. Plot *sin x* in the color orange and *cos x* in the color green and include a legend to label the two curves. Place the legend within the plot, but such that it does not cover either of the sine or cosine traces. Draw thin gray lines behind the curves, one horizontal at $y = 0$ and the other vertical at $x = 0$. (10)

(b) Explain semi-log plots and log-log plots along with the functions used in creating such plots. (4)

OR

18. (a) Explain how *matplotlib* can be used to create dimensional contour plots and vector field plots. (6)

(b) Given a file "*auto.csv*" of automobile data with the fields *index, company, body-style, wheel-base, length, engine-type, num-of-cylinders, horsepower, average-mileage,* and *price,* write Python codes using Pandas to (8)

1) Clean and Update the CSV file
2) Print total cars of all companies
3) Find the average mileage of all companies
4) Find the highest priced car of all companies.

Module -5

19. (a) Write python program to solve the following system of equations (4)

$x_1 - 2x_2 + 9x_3 + 13x_4 = 1$

$-5x_1 + x_2 + 6x_3 - 7x_4 = -3$

$4x_1 + 8x_2 - 4x_3 - 2x_4 = -2$

$8x_1 + 5x_2 - 7x_3 + x_4 = 5$

(b) Given the sales information of a company as CSV file with the following fields *month_number, facecream, facewash, toothpaste, bathingsoap, shampoo, moisturizer, total_units, total_profit*. Write Python codes to visualize the data as follows (10)

1) Toothpaste sales data of each month and show it using a scatter plot
2) Face cream and face wash product sales data and show it using the bar chart

Calculate total sale data for last year for each product and show it using a Pie chart.

OR

20. (a) Write Python program to write the data given below to a CSV file. (9)

SN	Name	Country	Contribution	Year
1	Linus Torvalds	Finland	Linux Kernel	1991
2	Tim Berners-Lee	England	World Wide Web	1990
3	Guido van Rossum	Netherlands	Python	1991

(b) Explain how integration is performed with SciPy. Illustrate the same with the two sample integrals using SciPy function. (5)

Teaching Plan

Sl No	Contents	Number of Hours (35 Hrs)
Module 1: Basics of Python (8 hours)		
1.1	Getting Started with Python Programming: Running code in the interactive shell Editing, Saving, and Running a script	1 hour
1.2	Using editors: IDLE, Jupyter	1 hour
1.3	Basic coding skills: Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions,	1 hour
1.4	Working with numeric data, Type conversions, Comments in the program, Input Processing, and Output. Formatting output	1 hour
1.5	How Python works. Detecting and correcting syntax errors. Using built in functions and modules in math module.	1 hour
1.6	Control statements : Selection structure, if-else, if ~~elif~~else	1 hour
1.7	Iteration structure - for, while	1 hour
1.8	Testing the control statements, Lazy evaluation.	1 hour
Module 2: Functions and Python Data Structures (8 hours)		
2.1	Functions: Hiding redundancy and complexity, Arguments and return values	1 hour
2.2	Variable scopes and parameter passing	1 hour
2.3	Named arguments, Main function,	1 hour
2.4	Working with recursion, Lambda functions	1 hour
2.5	Strings - String function	1 hour
2.6	Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension.	1 hour
2.7	Work with tuples. Sets.	1 hour
2.8	Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, Accessing and replacing values, traversing dictionaries, reverse lookup	1 hour
Module 3: Object Oriented Programming (6 hours)		
3.1	Design with classes : Objects and Classes, Methods, Instance Variables	1 hour
3.2	Constructor, Accessors, and Mutators	1 hour
3.3	Structuring classes with Inheritance	1 hour
3.4	Polymorphism	1 hour
3.5	Abstract Classes	1 hour
3.6	Exceptions: Handle a single exception, Handle multiple exception	1 hour
Module 4: Visualization and File handling (6 hours)		

4.1	Plotting - An Interactive Session with PyPlot, Basic Plotting,	1 hour
4.2	Logarithmic Plots, More Advanced Graphical Output	1 hour
4.3	Plots with multiple axes, Mathematics and Greek symbols	1 hour
4.4	The Structure of matplotlib, Contour and Vector Field Plots	1 hour
4.5	File Processing -The *os* and *sys* modules, Introduction to file I/O, Reading and writing text files	1 hour
4.6	Working with CSV files	1 hour
Module 5: Scientific Computing (7 hours)		
5.1	Numerical Routines: SciPy and NumPy - Basics, Creating arrays, Arithmetic, Slicing	1 hour
5.2	Matrix Operations, Special Functions, Random Numbers	1 hour
5.3	Linear Algebra, Solving Nonlinear Equations	1 hour
5.4	Numerical Integration, Solving ODEs	1 hour
5.5	Data Manipulation and Analysis: Pandas - Reading Data from Files Using Pandas	1 hour
5.6	Data Structures - Series and DataFrame	1 hour
5.7	Extracting Information from a DataFrame, Grouping and Aggregation	1 hour

CST455	OBJECT ORIENTED CONCEPTS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2019

Preamble: The purpose of this course is to enable learners to solve problems by breaking it down to object level while designing software and to implement it using Java. This course covers Object Oriented Principles, Object Oriented Programming in Java, Exception handling, Event handling, multithreaded programming and working with window-based graphics. This course provides learners the basics to develop Mobile applications, Enterprise Applications, Scientific Applications and Web based Applications.

Prerequisite: A sound background in any of the programming languages like C, C++, Python etc is mandatory. Students who completed the minor stream course CST 281 Object Oriented Programming are not allowed to choose this Open Elective Course.

Course Outcomes: After the completion of the course the student will be able to

CO1	Develop Java programs using the object-oriented concepts - classes, objects, constructors, data hiding, inheritance and polymorphism **(Cognitive Knowledge Level: Apply)**
CO2	Utilise data types, operators, control statements, built in packages & interfaces, Input/Output Streams and Files in Java to develop programs **(Cognitive Knowledge Level: Apply)**
CO3	Illustrate how robust programs can be written in Java using exception handling mechanism **(Cognitive Knowledge Level: Apply)**
CO4	Develop application programs in Java using multithreading **(Cognitive Knowledge Level: Apply)**
CO5	Develop Graphical User Interface based application programs by utilising event handling features and Swing in Java **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation

PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	**20**	**20**	**20**
Understand	**40**	**40**	**40**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests1&2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Object Orientation and Java basics)

Object Orientation Principles – Object and Class, Data abstraction and Encapsulation, Inheritance, Polymorphism, Dynamic binding, Message communication, Benefits of using Object orientation.

Introduction to Java - Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. Java Virtual Machine (JVM), Java compiler, Bytecode, Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues.

Primitive Data types - Integers, Floating Point Types, Characters, Boolean. Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.

Module – 2 (Core Java Fundamentals)

Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.

Control Statements - Selection Statements, Iteration Statements and Jump Statements.

Object Oriented Programming in Java - Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods, Constructors, ***this*** Keyword, Method Overloading, Using Objects as Parameters, Returning Objects, Recursion, Access Control, Static Members, Command-Line Arguments, Variable Length Arguments.

Module - 3 (More features of Java)

Inheritance - Super Class, Sub Class, The Keyword ***super***, protected Members, Calling Order of Constructors, Method Overriding, the Object class, Abstract Classes and Methods, Using ***final*** with Inheritance.

Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages, Interfaces.

Exception Handling - Checked Exceptions, Unchecked Exceptions, ***try*** Block and ***catch*** Clause, Multiple ***catch*** Clauses, Nested ***try*** Statements, ***throw***, ***throws*** and ***finally***.

Module - 4 (Advanced features of Java)

Input/Output - I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class, Reading and Writing Files.

Java Library - String Handling – String Constructors, String Length, Special String Operations - Character Extraction, String Comparison, Searching Strings, Modifying Strings, Using valueOf(), Comparison of String Buffer and String.

Module - 5 (GUI Programming, Event Handling and Multithreaded Programming)

Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread, Creating Multiple Threads, Suspending, Resuming and Stopping Threads.

Event Handling - Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Model.

Swing Fundamentals - Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Exploring Swing - JFrame, JLabel, JButton, JTextField.

Text Books

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.
2. Balagurusamy E., Programming JAVA a Primer, 5/e, McGraw Hill, 2014.

Reference Books

1. Paul Deitel, Harvey Deitel, Java How to Program, Early Objects 11/e, Pearson, 2018.
2. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.
3. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.
4. Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
5. Sierra K., Head First Java, 2/e, O'Reilly, 2005.

Course Level Assessment Questions

Course Outcome1 (CO1):

1. Three types of employees work in an organization: Regular, Contract and Hourly. Regular employees are permanent workers of the organization. Their salary is computed as the sum of basic pay, DA (50% of basic pay) and HRA. Contract employees work for the organization only for the contract period and earn a fixed salary. Hourly employees work for a fixed number of hours each day. Their salary is computed based on the total number of hours worked.

   Using object oriented principles, write a Java program to prepare pay roll of the organization.

2. Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Square, Triangle and Circle with proper class hierarchy. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

Course Outcome 2(CO2):

1. Write a Java program that reads a file and displays the file on the screen, with a line number before each line.
2. Write a Java program to prepare the rank list of computer science students based on their performance in the first Semester B.Tech. Degree examination at APJ Abdul Kalam Technological University. The output should be stored in a file.

Course Outcome 3(CO3):

1. Write a program to demonstrate the use of *throws* clause to handle an exception occurred within a method.
2. Write a program to demonstrate how exception handling is supported in Java.

Course Outcome 4(CO4):

1. Write a program to compute the sum of elements in an array using two threads in a parallel way. The first thread sums up the first half of the array and the second thread sums up the second half of the array. Finally, the main thread adds these partial sums and prints the result.
2. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and ifthe value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

Course Outcome 5(CO5):

1. Write a GUI based program to convert temperature from degree Celsius to Fahrenheit.
2. Write a java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with buttons. On selecting a button, an appropriate message with "stop" or "ready" or "go" should appear above the buttons in a selected color. Initially there is no message shown.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES :4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST455

Course Name: Object Oriented Concepts

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Java is considered to be secure and portable. Justify this statement.

2. Describe the concept of dynamic binding.

3. Explain the different arithmetic operators in Java.

4. What does the following Java function compute? Justify your answer.

```
intgreater(int a, int b)
{
while(a!=b)
{
   if(a>b)
a=a-b;
   else
b=b-a;
}
return a;
}
```

5. Explain the use of CLASSPATH with an example.

6. What are the different types of exceptions?

7. Explain file handling features available in Java.

8. Write a simple program to read an integer value from console and print it.

9. Explain the concept of *main thread* in multi-threading.

10. Explain any two Event classes in Java. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Describe in detail polymorphism, abstraction and inheritance with suitable examples. **(9)**

    (b) What is Java Virtual Machine? **(5)**

OR

12. (a) Explain the salient features of Java language. How does Java Enterprise Edition (J2EE) differ from Java Standard Edition (Java SE)? **(9)**

    (b) Explain the declaration and use of multi-dimensional array variables in Java, with example. **(5)**

13. (a) Explain iteration control statements in Java. Give examples. **(8)**

(b) Write a recursive program to compute the factorial of a number. **(6)**

OR

14. (a) Using a suitable Java program, explain the concept of methods and constructors. **(6)**

(b) Write a Java program that prompts the user for an integer and then prints out all the prime numbers up to that number. **(8)**

15. (a) In a table format, show the effect of access specifiers within and outside packages in Java. **(6)**

(b) Describe exception handling using **try** block and **catch** clause in Java with the help of a suitable Java program. **(8)**

OR

16. (a) What is an interface in Java? Explain with a suitable example. **(6)**

(b) Write a program that perform integer divisions. The user enters two input data (any data type) through console into variables Num1 and Num2. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the appropriate exception or result. **(8)**

17. (a) Write a Java program that displays the number of characters, lines and words in a text file. **(8)**

(b) Explain any three String constructors with the help of sample code for each. **(6)**

OR

18. (a) Write a program to demonstrate the usage of the *PrintWriter* class. **(7)**

(b) Write a Java program for sorting a given list of names in ascending order. **(7)**

19. (a) Explain Delegation Event model for event handling in Java. **(7)**

(b) Write a program to compute the sum of elements in an array using two **(7)** threads in a parallel way. The first thread sums up the first half of the array and the second thread sums up the second half of the array. Finally, the main thread adds these partial sums and prints the result.Use Runnable interfacefor the creation of a thread.

OR

20. (a) What are the differences between a process and a thread? **(4)**

(b) Write a Graphical User Interface (GUI) based Java program to implement a simple calculator supporting the operations addition, subtraction, multiplication and division. Use Swing controls to implement GUI. There may be three text boxes, the first two for accepting the operands and the last for displaying the result. Add four buttons for the above operations. Write neat comments in your program to show how you handle events. **(10)**

Teaching Plan

No	**Contents**	**No. of Lecture Hours** **(36hrs)**
Module – 1 (Object Orientation and Java basics) (7 hrs)		
1.1	Object Orientation Principles – Object and Class, Data abstraction and Encapsulation	1 hour
1.2	Inheritance, Polymorphism	1 hour
1.3	Dynamic binding, Message communication, Benefits of using Object orientation.	1 hour
1.4	Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. JVM, Java compiler, Bytecode	1 hour
1.5	Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues	1 hour
1.6	Primitive Data types - Integers, Floating Point Types, Characters, Boolean	1 hour
1.7	Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector	1 hour

	class.	
Module - 2 (Core Java Fundamentals) (7 hrs)		
2.1	Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.	1 hour
2.2	Control Statements - Selection Statements, Iteration Statements and Jump Statements.	1 hour
2.3	Object Oriented Programming in Java - Class Fundamentals, Declaring Objects	1 hour
2.4	Object Reference, Introduction to Methods, Constructors, ***this*** Keyword	1 hour
2.5	Method Overloading, Using Objects as Parameters, Returning Objects	1 hour
2.6	Recursion, Access Control, static Members	1 hour
2.7	Command-Line Arguments, Variable Length Arguments	1 hour
Module - 3 (More features of Java) (8 hrs)		
3.1	Inheritance - Super class, Sub class, the keyword super, protected Members	1 hour
3.2	Calling Order of Constructors, Method Overriding, the Object class	1 hour
3.3	Abstract Classes and Methods, Using final with Inheritance	1 hour
3.4	Packages and Interfaces - Defining Package, CLASSPATH, Access Protection	1 hour
3.5	Importing Packages, Interfaces	1 hour
3.6	Exception Handling **-** Checked Exceptions, Unchecked Exceptions, ***try*** Block and ***catch*** Clause	1 hour
3.7	Multiple ***catch*** Clauses, Nested ***try*** Statements	1 hour
3.8	***throw***, ***throws*** and ***finally***	1 hour
Module - 4 (Advanced features of Java) (6 hrs)		
4.1	Input/Output - I/O Basics, Reading Console Input	1 hour
4.2	Writing Console Output, PrintWriter Class	1 hour
4.3	Working with Files (Lecture-1)	1 hour

4.4	Working with Files (Lecture-2)	1 hour
4.5	Java Library - String Handling – String Constructors, String Length	1 hour
4.6	Special String Operations - Character Extraction, String Comparison, Searching Strings, Modifying Strings, Using valueOf( ), Comparison of StringBuffer and String.	1 hour
	Module - 5 (GUI Programming, Event Handling and Multithreaded Programming) (8hrs)	
5.1	Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread	1 hour
5.2	Creating Multiple Threads	1 hour
5.3	Suspending, Resuming and Stopping Threads.	1 hour
5.4	Event handling **-** Event Handling Mechanisms, Delegation Event Model	1 hour
5.5	Event Classes,Sources of Events, Event Listener Interfaces	1 hour
5.6	Using the Delegation Model, Swing fundamentals, Swing Key Features	1 hour
5.7	Model View Controller (MVC), Swing Controls, Components and Containers	1 hour
5.8	Exploring Swing –JFrame, JLabel, JButton, JTextField	1 hour

SEMESTER VII

MINOR

CSD481	MINI PROJECT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	0	3	4	2019

Preamble: The objective of this course is to apply the fundamental concepts of different courses learned in respective Minor Streams: Software Engineering, Machine Learning and Networking. This course helps the learners to get an exposure to the development of application software/hardware solutions/ software simulations in the field of Computer Science and Engineering. It enables the learners to understand the different steps to be followed such as literature review and problem identification, preparation of requirement specification &design document, testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

Prerequisite:

A sound knowledge in courses studied in respective minor stream.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Identify technically and economically feasible problems **(Cognitive Knowledge Level: Apply)**
CO2	Identify and survey the relevant literature for getting exposed to related solutions. **(Cognitive Knowledge Level: Apply)**
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (**Cognitive Knowledge Level: Apply)**
CO4	Prepare technical report and deliver presentation **(Cognitive Knowledge Level: Apply)**
CO5	Apply engineering and management principles to achieve the goal of the project **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Mark Distribution

Total Marks	CIE Marks	ESE Marks
150	75	75

Continuous Internal Evaluation Pattern:

Attendance **10 marks**

Project Guide **15 marks**

Project Report **10 marks**

Evaluation by the Committee (will be evaluating the level of completion
and demonstration of functionality/specifications, presentation,
oral examination, work knowledge and involvement) : **40 marks**

Student Groups with 4 or 5 members should identify a topic of interest in consultation with a Faculty Advisor/Project Coordinator/Guide. Review the literature and gather information pertaining to the chosen topic. State the objectives and develop a methodology to achieve the objectives. Carryout the design/fabrication or develop codes/programs to achieve the objectives by strictly following steps specified in the teaching plan. Innovative design concepts, performance, scalability, reliability considerations, aesthetics/ergonomic, user experience and security aspects taken care of in the project shall be given due weight.

The progress of the mini project is evaluated based on a minimum of two reviews. The review committee may be constituted by a senior faculty member, Mini Project coordinator and project guide. The internal evaluation shall be made based on the progress/outcome of the project, reports and a viva-voce examination, conducted internally by a 3-member committee. A project report is required at the end of the semester. The project has to be demonstrated for its full design specifications.

End Semester Examination Pattern:

The marks will be distributed as

Presentation	: **30 marks**
Demo	: **20 marks**
Viva	: **25 marks.**
Total	: **75 marks**.

TEACHING PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc.
3. Create Requirements Specification
4. Create Design Document . This may include designs like,
    a. System Architecture Design
    b. Application Architecture Design
    c. GUI Design
    d. API Design
    e. Database Design
    f. Technology Stack
5. Deployment, Test Run & Get Results
6. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman18, Bold; Heading 2 – Times New Roman16, Bold; Heading 3 – Times New Roman14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.

- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figure title under the figure and table title above the table.

- **Suggestive order of documentation:**
  i. Top Cover
  ii. Title page
  iii. Certification page
  iv. Acknowledgement
  v. Abstract
  vi. Table of Contents
  vii. List of Figures and Tables
  viii. Chapters
  ix. Appendices, if any
  x. References/Bibliography

SEMESTER VII

HONOURS

CST495	CYBER FORENSICS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		VAC	3	1	0	4	2019

Preamble: The course on Cyber Forensics aims at exploring the basics of Cyber Forensics and Cyber security, the forensic investigation process and principles and the different types of cybercrimes and threats. This course also focuses on the forensic analysis of File systems, the Network, the Windows and Linux Operating systems. The course gives a basic understanding of the forensics analysis tools and a deep understanding of Anti forensics practices and methods. All the above aspects are dealt with case studies of the respective areas.

Prerequisite: Knowledge in File Systems, Operating systems, Networks and a general awareness on Cyber Technologies.

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain thebasic concepts in Cyber Forensics, Forensics Investigation Process and Cyber security(**Cognitive Knowledge Level: Understand**)
CO2	Infer the basic concepts of File Systems and its associated attribute definitions (**Cognitive Knowledge Level: Understand)**
CO3	Utilize the methodologies used in data analysis and memory analysis for detection of artefacts**(Cognitive Knowledge Level: Apply)**
CO4	Identify web attacks and detect artefacts using OWASP and penetration testing. **(Cognitive Knowledge Level: Apply)**
CO5	Summarize anti-forensics practices and data hiding methods **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓				✓						✓
CO2	✓	✓										✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓			✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Tests	: 25 marks
Continuous Assessment Assignment	: 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1(Cyber Forensics and Cyber Security)

Computer Forensics: History of computer forensics, preparing for computer investigations, understanding Public and private investigations- Forensics Investigation Principles - Forensic Protocol for Evidence Acquisition - Digital Forensics -Standards and Guidelines - Digital Evidence – Data Acquisition - storage formats for digital evidence, determining the best acquisition method, contingency planning for image acquisitions, Cyber Forensics tools- Challenges in Cyber Forensics, Skills Required to Become a Cyber Forensic Expert

Cyber Security: Cybercrimes, Types of Cybercrimes - Recent Data Breaches - Recent Cyber security Trends - Case Study: Sim Swapping Fraud, ATM Card Cloning, Hacking email for money, Google Nest Guard, Email Crimes, Phishing, Types of Phishing.

Module-2 (File System Forensics)

File system Analysis: FAT and NTFS concepts and analysis -File system category, Content category, Metadata category, File name category, Application category,Application-level search techniques, Specific file systems, File recovery, Consistency check. FAT data structure-Boot sector, FAT 32 FS info, directory entries, Long file name directory entries

Module-3 (Operating System Forensics)

Windows Forensics: Live Response- Data Collection- Locard's Exchange Principle, Order of Volatility Volatile and Non Volatile Data Live-Response Methodologies: Data Analysis- Agile Analysis, Windows Memory Analysis, Rootkits and Rootkit detection.

Linux Forensics: Live Response Data Collection- Prepare the Target Media, Format the Drive, Gather Volatile Information, Acquiring the Image, Initial Triage, Data Analysis- Log Analysis, Keyword Searches, User Activity, Network Connections, Running Processes, Open File Handlers, The Hacking Top Ten, Reconnaissance Tools

Module-4 (Network Forensics)

The OSI Model, Forensic Footprints, Seizure of Networking Devices, Network Forensic Artifacts, ICMP Attacks, Drive-By Downloads, Network Forensic Analysis Tools, Case Study: Wireshark. Web Attack Forensics: OWASP Top 10, Web Attack Tests, Penetration Testing.

Module-5 (Anti-Forensics)

Anti-forensic Practices - Data Wiping and Shredding- Data Remanence, Degaussing, Case Study: USB Oblivion, Eraser - Trail Obfuscation: Spoofing, Data Modification, Case Study: Timestamp – Encryption, Case Study: VeraCrypt, Data Hiding: Steganography and Cryptography, Case Study: SilentEye, Anti-forensics Detection Techniques, Case Study: Stegdetect

Text Books

1. Bill Nelson, Amelia Phillips and Christopher Steuart, Computer forensics - Guide to Computer Forensics and Investigations, 4/e, Course Technology Inc.
2. Brian Carrier, File System Forensic Analysis, Addison Wesley, 2005.
3. Harlan Carvey, Windows Forensic Analysis DVD Toolkit, 2/e, Syngress.
4. Cory Altheide, Todd Haverkos, Chris Pogue,Unix and Linux Forensic Analysis DVD Toolkit, 1/e, Syngress.
5. William Stallings,Network Security Essentials Applications and Standards, 4/e, Prentice Hall
6. Eric Maiwald, Fundamentals of Network Security, McGraw-Hill, 2004.

References

1. Michael. E. Whitman, Herbert. J. Mattord, Principles of Information Security, Course Technology, 2011.
2. William Stallings, Cryptography and Network Security Principles and Practice, 4/e, Prentice Hall.
3. Niranjan Reddy, Practical Cyber Forensics: An Incident-Based Approach to Forensic Investigations, Apress, 2019.

Sample Course Level Assessment Questions

CourseOutcome1(CO1): Explain the Forensics principles and protocols for evidence acquisition.

Discuss the different cyber forensics tools used for image acquisition.

CourseOutcome2(CO2): Explain the pros and cons of NTFS and FAT File systems. Also give the challenges the investigators would face in extracting evidences from these file systems.

CourseOutcome3 (CO3): Apply any memory forensics methodologies/tools to extract volatile and nonvolatile data from a Windows based system.

CourseOutcome4 (CO4): Use web attacks test tools like netcraft to identify web application vulnerabilities of a particular site say **www.xyz.com**

Course Outcome 5 (CO5): Explain the different anti-forensics practices used to destroy or conceal data in order to prevent others from accessing it.

Model Question Paper

QP CODE:

Reg No: _______________

Name: ___________________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST495

Course Name: Cyber Forensics

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Distinguish between public and private investigations.

2. What are the three computer forensics data acquisitions formats?

3. List any three features of NTFS which are not in FAT.

4. Define the terms file slack, RAM slack and drive slack.

5. What is Locard's exchange principle? Why is it important in forensic investigations?

6. Why would you conduct a live response on a running system?

7. What are the different tools used in Network Forensics?

8. Explain how Risk Analysis and Penetration Testing are different.

9. Why we are using Steganography?

10. How is data wiping done in hard drive?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Discuss the different types of Cybercrimes. List the tools used for identifying Cyber Crimes. **(8)**

(b) Differentiate between Static acquisition and Live acquisition with example. **(6)**

OR

12. (a) Explain the principles of Digital Forensic Investigation? Why is it important? Comment. **(8)**

(b) When you perform an acquisition at a remote location, what should you consider preparing this task? **(6)**

13. (a) Discuss the FAT File Structure. **(8)**

(b) Does Windows NT use FAT or NTFS? Explain. **(6)**

14. (a) What is Metadata? Discuss the first 16 metadata records you would find in the MFT? **(6)**

(b) Explain the different data categories in a File System. **(8)**

15. (a) What is Agile requirement analysis? **(6)**

(b) Explain the different types of volatile information in a live response system. List any two tools used for obtaining volatile information. **(8)**

16. (a) What are the main live response methodologies? **(6)**

(b) What is Physical Memory Dump? Explain how a physical memory dump is analysed. **(8)**

17. (a) What is OWASP? Also mention the Top 10 web application vulnerabilities in 2021. **(8)**

(b) How would you setup Wireshark to monitor packets passing through aninternet router? **(6)**

18. (a) What are the goals of conducting a pentesting exercise? **(3)**

(b) Discuss the types of penetration testing methodologies. **(5)**

(c) Define OSI Layers. **(6)**

19. (a) How is Steganography done? **(7)**

(b) Why does data need Cryptography? **(4)**

(c) What is the difference between a Cryptographer and a Crypter? **(3)**

OR

20. (a) Explain the different types of Anti-forensics Detection Techniques. **(8)**

(b) What is Spoofing? How to prevent spoofing attack? **(6)**

TEACHING PLAN

Sl.No.	**Contents**	**No of Lecture Hrs (44hrs)**
Module-1 (Cyber Forensics and Cyber Security) (11 Hrs)		
1.1	History of computer forensics, preparing for computer investigations	1 hour
1.2	Understanding Public and private investigations- Forensics Investigation Principles	1 hour
1.3	Forensic Protocol for Evidence Acquisition	1 hour
1.4	Digital Forensics -Standards and Guidelines - Digital Evidence	1 hour
1.5	Data Acquisition - storage formats for digital evidence, determining the best acquisition method	1 hour
1.6	Contingency planning for image acquisitions, Cyber Forensics tools	1 hour
1.7	Challenges in Cyber Forensics, Skills Required to Become a Cyber Forensic Expert	1 hour
1.8	Cybercrimes, Types of Cybercrimes - Recent Data Breaches - Recent Cyber security Trends	1 hour
1.9	Case Study: Sim Swapping Fraud, ATM Card Cloning	1 hour
1.10	Case Study:Hacking email for money, Google Nest Guard	1 hour
1.11	Email Crimes, Phishing, Types of Phishing	1 hour
Module-2 (File System Forensics) (9 Hrs)		

2.1	FAT and NTFS concepts and analysis	1 hour
2.2	File system category, Content category	1 hour
2.3	Metadata category	1 hour
2.4	File name category,Application category	1 hour
2.5	Application-level search techniques	1 hour
2.6	Specific file systems, File recovery, Consistency check	1 hour
2.7	FAT data structure-Boot sector	1 hour
2.8	FAT 32 FS info, directory entries	1 hour
2.9	Long file name directory entries	1 hour
Module-3 (Operating System Forensics) (11 Hrs)		
3.1	Live Response- Data Collection- Locard's Exchange Principle	1 hour
3.2	Order of Volatility, Volatile and Non Volatile Data	1 hour
3.3	Live-Response Methodologies: Data Analysis- Agile Analysis	1 hour
3.4	Windows Memory Analysis	1 hour
3.5	Rootkits and Rootkit detection	1 hour
3.6	Linux Forensics: Live Response Data Collection	1 hour
3.7	Prepare the Target Media, Format the Drive, Gather Volatile Information	1 hour
3.8	Acquiring the Image, Initial Triage	1 hour
3.9	Data Analysis- Log Analysis, Keyword Searches, User Activity	1 hour

3.10	Data Analysis- Network Connections, Running Processes, Open File Handlers	1 hour
3.11	The Hacking Top Ten, Reconnaissance Tools	1 hour
Module-4 (Network Forensics) (7 Hrs)		
4.1	OSI Model	1 hour
4.2	Forensic Footprints, Seizure of Networking Devices, Network Forensic Artifacts	1 hour
4.3	ICMP Attacks, Drive-By Downloads, Network Forensic Analysis Tools	1 hour
4.4	Web Attack Forensics	1 hour
4.5	OWASP Top 10, Web Attack Tests	1 hour
4.6	Penetration Testing-1	1 hour
4.7	Penetration Testing.-2	1 hour
Module – 5 (Anti-Forensics) (6 Hrs)		
5.1	Anti-forensic Practices - Data Wiping and Shredding	1 hour
5.2	Data Remanence, Degaussing	1 hour
5.3	Trail Obfuscation: Spoofing, Data Modification	1 hour
5.4	Role of Encryption in Forensics	1 hour
5.5	Data Hiding: Steganography and Cryptography	1 hour
5.6	Anti-forensics Detection Techniques	1 hour

CST497	REINFORCEMENT LEARNING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		VAC	3	1	0	4	2019

Preamble: This course covers fundamental principles and techniques in reinforcement learning. Reinforcement learning is concerned with building programs that learn how to predict and act in a stochastic environment, based on past experience. Applications of reinforcement learning range from classical control problems, such as power plant optimization or dynamical system control, to game playing, inventory control, and many other fields. Topics include Markov decision process, dynamic programming, Monte Carlo, temporal difference, function approximation reinforcement learning algorithms, and applications of reinforcement learning. This course enables the leaners to apply reinforcement learning on real world applications and research problems.

Prerequisite: A pass in CST 294(Computational Fundamentals for Machine Learning)

Course Outcomes: After the completion of the course the student will be able to

CO 1	Solve computational problems using probability and random variables. **(Cognitive Knowledge Level: Apply)**
CO 2	Apply policy iteration and value iteration reinforcement learning algorithms. **(Cognitive Knowledge Level: Apply)**
CO 3	Employ Monte Carlo reinforcement learning algorithms. **(Cognitive Knowledge Level: Apply)**
CO 4	Apply temporal-difference reinforcement learning algorithms.(**Cognitive Knowledge Level: Apply)**
CO 5	Apply on-policy and off-policy reinforcement learning algorithms with function approximation. (**Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	✓	✓	✓									✓
CO 2	✓	✓	✓	✓								✓
CO 3	✓	✓	✓	✓								✓
CO 4	✓	✓	✓	✓								✓
CO 5	✓	✓	✓	✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	**30%**	**30%**	**30%**
Understand	**30%**	**30%**	**30%**
Apply	**40%**	**40%**	**40%**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance : **10 marks**

Continuous Assessment Tests : **25 marks**

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer anyone. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module 1 (Review Of Probability Concepts)

Probability concepts review - Axioms of probability, concepts of random variables, probability mass function, probability density function, cumulative density functions, Expectation. Concepts of joint and multiple random variables, joint, conditional and marginal distributions. Correlation and independence.

Module 2 (Markov Decision Process)

Introduction to Reinforcement Learning(RL) terminology - Examples of RL, Elements of RL, Limitations and Scope of RL.

Finite Markov Decision Processes - The Agent–Environment Interface, Goals and Rewards, Returns and Episodes, Policies and Value Functions, Optimal Policies and Optimal Value Functions.

Module 3 (Prediction And Control)

Dynamic Programming - Policy Evaluation (Prediction), Policy Improvement, Policy Iteration, Value Iteration.

Monte Carlo (MC) for model free prediction and control - Monte Carlo Prediction, Monte

Carlo Estimation of Action Values, Monte Carlo Control, Monte Carlo Control without Exploring Starts, Off-policy Prediction via Importance Sampling, Incremental Implementation, Off-policy Monte Carlo Control.

Module 4 (Temporal-Difference (TD) Methods For Model Free Prediction And Control)

TD Methods - TD Prediction, Advantages of TD Prediction Methods, Optimality of TD(0), Sarsa: On-policy TD Control, Q-learning: Off-policy TD Control, Expected Sarsa.

n-step Bootstrapping- n-step TD Prediction, n-step Sarsa, step Off-policy Learning, Off-policy Learning Without Importance Sampling: The n-step Tree Backup Algorithm.

Module 5 (Function Approximation Method)

On-policy Prediction with Approximation - Value-function Approximation, The Prediction Objective, Stochastic-gradient Methods, Linear Methods.

Eligibility Traces - The λ-return, TD(λ), n-step Truncated λ-return Methods, Sarsa(λ).

Policy Gradient Methods - Policy Approximation and its Advantages, The Policy Gradient Theorem, REINFORCE: Monte Carlo Policy Gradient, REINFORCE with Baseline, Actor–Critic Methods.

Text book:

1 Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, , 2nd Edition

2 Alberto Leon-Garcia, Probability, Statistics, and Random Processes for Electrical Engineering, 3rd Edition,

Reference books:

1 Reinforcement Learning: State-of-the-Art, Marco Wiering and Martijn van Otterlo, Eds

2 Algorithms for Reinforcement Learning, Szepesvari (2010), Morgan & Claypool.

3 Artificial Intelligence: A Modern Approach, Stuart J. Russell and Peter Norvig

4 Mathematical Statistics and Data Analysis by John A. Rice,University of California, Berkeley, Third edition, published by Cengage.

5 Machine Learning: A Probabilistic Perspective, Kevin P. Murphy

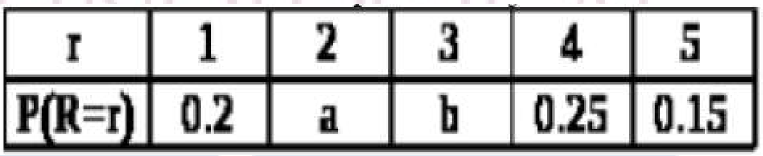
Sample Course Level Assessment Questions.

Course Outcome 1 (CO1):

1 Let $J$and Tbe independent events, where $P(J)=0.4$and $P(T)=0.7$. Find $P(J \cap T)$, $P(J \cup T)$ and $P(J \cap T')$

2 Let A and B be events such that $P(A)=0.45$, $P(B)=0.35$ and $P(A \cup B)=0.5$ Find $P(A \mid B)$

3 A random variable **R**has the probability distribution as shown in the following table:



r	1	2	3	4	5
P(R=r)	0.2	a	b	0.25	0.15

Given that $E(R)=2.85$, find a and $b$and $P(R>2)$.

4 A biased coin (with probability of obtaining a head equal to $p > 0$) is tossed repeatedly and independently until the first head is observed. Compute the probability that the first head appears at an even numbered toss.

5 Two players A and B are competing at a quiz game involving a series of questions. On any individual question, the probabilities that A and B give the correct answer are p and q respectively, for all questions, with outcomes for different questions being independent. The game finishes when a player wins by answering a question correctly. Compute the probability that A wins if

(i) A answers the first question,

(ii) B answers the first question.

6 A coin for which $P(heads) = p$is tossed until two successive tails are obtained. Find the probability that the experiment is completed on the$n$thtoss.

7 An urn contains **p** black balls, **q** white balls, and **r** red balls; and **n** balls are chosen without replacement.

i. Find the joint distribution of the numbers of black, white, and red balls in the sample.

ii. Find the joint distribution of the numbers of black and white balls in the sample.

iii. Find the marginal distribution of the number of white balls in the sample.

8 Suppose that two components have independent exponentially distributed lifetimes, $T1$and T2, with parameters $\alpha$and β, respectively. Find (a) $P(T1> T2)$and (b) $P(T1> 2 T2)$.

9 Let ***Z1*** and ***Z2*** be independent random variables each having the standard normal distribution. Define the random variables ***X*** and ***Y*** by ***X = Z1 + 3Z2*** and Y = ***Z1+ Z2***. Argue that the joint distribution of ***(X, Y)*** is a bivariate normal distribution. What are the parameters of this distribution?

10 Given a continuous random variable ***x***, with cumulative distribution function ***Fx(x)***, show that the random variable ***y = Fx(x)*** is uniformly distributed.

11 ou roll a fair dice twice. Let the random variable ***X*** be the product of the outcomes of the two rolls. What is the probability mass function of ***X***? What are the expected values and the standard deviation of ***X***?

12 Show that if two events ***A***and ***B***are independent, then ***A***and ***B'***are independent

13 Prove that ***X***and ***Y***are independent if and only if ***fX|Y (x|y) = fX(x)***for all ***x***and ***y***

14 A random square has a side length that is a uniform [0, 1] random variable. Find the expected area of the square.Let X be a continuous random variable with the density function f (x) = 2x, 0 ≤ x ≤ 1

i. Find ***E(X)***.

ii. Find ***E(X²)***and ***Var(X)***.

Course Outcome 2 (CO2):

1 What are the main differences between supervised learning and reinforcement learning?

2 Give examples of Markovian and non-Markovian environments?

3 What are the advantages and disadvantages of value methods vs policy methods?

4 Define the optimal state-value function $\mathbf{V^*(s)}$ for an MDP.

5 Imagine that the rewards are at most 1 everywhere. What is the maximum value that the discounted return can attain ? Why ?

6 Write down the Bellman optimality equation for state-value functions

7 Suppose that you are in a casino. You have Rs 20 and will play until you lose it all or as soon as you double your money. You can choose to play two slot machines: 1) slot machine A costs Rs 10 to play and will return Rs 20 with probability 0.05 and Rs 0 otherwise; and 2) slot machine B costs Rs 20 to play and will return Rs30 with probability 0.01 and Rs 0 otherwise. Until you are done, you will choose to play machine A or machine B in each turn. Describe the state space, action space, rewards and transition probabilities. Assume the discount factor $\gamma = 1$. Rewards should yield a higher reward when terminating with Rs 40 than when terminating with Rs 0. Also, the reward for terminating with Rs 40 should be the same regardless of how we got there (and equivalently for Rs 0).

Course Outcome 3 (CO3):

1 Explain policy iteration and value iteration? What are their similarities and differences?

2 Why Monte Carlo methods for learning value functions require episodic tasks? How is it that n-step TD methods avoid this limitation and can work with continuing tasks?

3 List any three uses of the depth parameter in the Monte-Carlo tree search procedure.

4 Given that $\mathbf{q_\pi(s, a) > v_\pi(s)}$, can we conclude that $\boldsymbol{\pi}$ is not an optimal policy. Justify

Course Outcome 4 (CO4):

1 Draw the backup diagram for 2-step Sarsa. Write the corresponding learning rule for 2-step Sarsa.

2 Why is Sarsa an on-policy algorithm while Q-learning is an off-policy algorithm?

3 How would you differentiate between learning algorithms using on-policy from those that use off-policy?

4 When using Temporal Difference learning, why is it better to learn action values (Q-values) rather than state values (V-values)?

5 Supose that a Q-learning agent always chooses the action which maximizes the Q-value. What is one potential problem with that approach?

6 Describe any two ways that will force a Q-learning agent to explore.

7 Why and when do we need importance sampling?

Course Outcome 5 (CO5):

1 How do you deal with a large possible action space in reinforcement learning?

2 List any two benefits of policy gradient methods over value function based methods.

3 What is the relation between Q-learning and policy gradients methods?

4 Consider a five state random walk. There are five states, $\mathbf{s_1, s_2, ..., s_5,}$ in a row with two actions each, left and right. There are two terminal states at each end, with a reward of **+1** for terminating on the right, after $\mathbf{s_5}$ and a reward of **0** for all other transitions, including the one terminating on the left after $\mathbf{s_1}$. In designing a linear function approximator, what is the least number of state features required to represent the value of the equi-probable random policy?

Model Question paper

QP Code : **Total Pages: 4**

Reg No.:_______________ Name:__________________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION (HONOURS), MONTH and YEAR

Course Code: CST497

Course Name: REINFORCEMENT LEARNING

Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks.

1 The first three digits of a telephone number are 452. If all the sequences of the remaining four digits are equally likely, what is the probability that a randomly selected telephone number contains seven distinct digits?

2 If X is a discrete uniform random variable, i.e., $P(X = k) = 1/n$ for $k = 1, 2, ... , n$, find $E(X)$and $Var(X)$.

3 Define the discounted return $\mathbf{G_t}$. Give an expression for $\mathbf{G_t}$in terms of $\mathbf{G_{t+1}}$.

4 Write down the Bellman expectation equation for state-value functions.

5 Suppose that we are doing value iteration with $\gamma = 0$. How many iterations will it take for value iteration to converge to the optimal value function?

6 List any three advantages of Monte Carlo methods over dynamic programming techniques?

7 Draw the backup diagram for 2-step Q-learning. Write the corresponding learning rule for 2-step Q-learning.

8 Why Monte Carlo methods for learning value functions require episodic tasks. How does **n**-step TD methods avoid this limitation and can work with continuing tasks?

9 In using policy gradient methods, if we make use of the average reward formulation rather than the discounted reward formulation, then is it necessary to consider, for problems that do not have a unique start state, a designated start state, $\mathbf{s_0}$? Justify.

10 Value function based methods are oriented towards finding deterministic

policies whereas policy search methods are geared towards finding stochastic policies. True or false? Justify.

10 x 3 = 30

PART B

Answer any one Question from each module. Each question carries 14 Marks

11 a) Three players play 10 independent rounds of a game, and each player has probability 1/3 of winning each round. Find the joint distribution of the numbers of games won by each of the three players. **(7)**

b) Find the joint density of $\boldsymbol{X + Y}$ and $\boldsymbol{X/Y}$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent exponential random variables with parameter $\boldsymbol{\lambda}$. Show that $\boldsymbol{X + Y}$ and $\boldsymbol{X/Y}$are independent. **(7)**

OR

12 a) An experiment consists of throwing a fair coin four times. Find the probability mass function and the cumulative distribution function of the following random variables: (7)

i the number of heads before the first tail

ii the number of heads following the first tail

iii the number of heads minus the number of tails

iv the number of tails times the number of heads.

b) Let $\boldsymbol{X}$ be a continuous random variable with probability density function on $\boldsymbol{0 <= x <= 1}$ defined by $\boldsymbol{f(x) = 3x^2}$. Find the pdf of $\boldsymbol{Y = X^2}$. **(7)**

13 a) What is the difference between a state value function **V(s)** and a state-action value function **Q(s,a)**? **(4)**

b) Consider designing a recycling robot whose job is to collect empty bottles around the building. The robot has a sensor to detect when a bottle is in front of it, and a gripper to pick up the bottle. It also senses the level of its battery. The robot can navigate, as well as pick up a bottle and throw a bottle it is holding in the trash. There is a battery charger in the building, and the robot should not run out of battery. **(10)**

i. Describe this problem as an MDP. What are the states and actions?

ii. Suppose that you want the robot to collect as many bottles as possible, while not running out of battery. Describe what rewards would enable it to achieve this task.

OR

14 a) Define the state-value function $\mathbf{V_\pi(s)}$ for a discounted MDP. **(5)**

b) Consider a 4x4 gridworld where the agent starts in the top left, the bottom rigl state is terminal, rewards are always **-1**, γ **= 1**, and state transitions aı deterministic. Consider the policy that always chooses the action to move dowı except when it is on the bottom row, at which point it chooses the action to mov right. Starting with $\mathbf{v_0(s) = 0}$ for all **s**, compute $\mathbf{v_1, v_2, ... , v_7}$. **(10)**

15 a) During a single iteration of the Value Iteration algorithm, we typically iterate over the states in **S** in some order to update $\mathbf{V_t(s)}$ to $\mathbf{V_{t+1}(s)}$ for all states **s**. Is it possible to do this iterative process in parallel? Explain why or why not. **(5)**

b) Consider an undiscounted Markov Reward Process with two states A and B. The transition matrix and reward function are unknown, but you have observed two sample episodes: **(9)**

A +3 --> A +2 --> B -4 --> A +4 --> B -3

B -2 --> A +3 --> B -3

i. Using first-visit Monte-Carlo evaluation, estimate the state-value function **V(A),V(B)**.

ii. Using every-visit Monte-Carlo evaluation, estimate the state-value function **V(A),V(B).**

iii. Draw a diagram of the Markov Reward Process that best explains these two episodes. Show rewards and transition probabilities on your diagram.

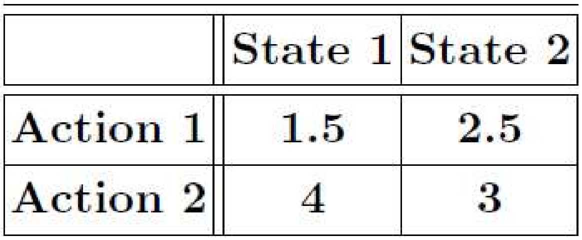
OR

16 a) Suppose you are given a finite set of transition data. Assuming that the Markov model that can be formed with the given data is the actual MDP from which the data is generated, will the value functions calculated by the MC and TD methods necessarily agree? Justify. **(4)**

b) With respect to the expected Sarsa algorithm, is exploration required as it is in the normal Sarsa and Q-learning algorithms? Justify. **(5)**

c) For a specific MDP, suppose we have a policy that we want to evaluate through the use of actual experience in the environment alone and using Monte Carlo methods. We decide to use the first-visit approach along with the technique of always picking the start state at random from the available set of states. Will this approach ensure complete evaluation of the action value function corresponding to the policy? **(5)**

17 a) Consider the following **Q[S,A]** table **(9)**



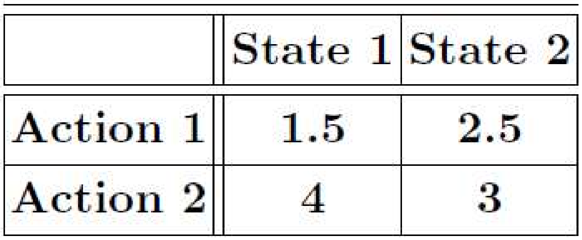
	State 1	State 2
Action 1	1.5	2.5
Action 2	4	3

Assume the discount factor, γ= **0.5**, and the step size, $\boldsymbol{\alpha}$ = **0.1**. After the experience **(s, a, r, s')=(1, 1, 5, 2),** which value of the table gets updated and what is its new value?

b) What is the difference between Q-learning and Sarsa? **(5)**

OR

18 a) Consider the following **Q[S,A]** table **(9)**



	State 1	State 2
Action 1	1.5	2.5
Action 2	4	3

Assume the discount factor, γ= **0.5**, and the step size, $\boldsymbol{\alpha}$ = **0.1**. After the experience **(s, a, r, s', a')=(1, 1, 5, 2, 1),** which value of the table gets updated and what is its new value?

b) For Q-learning to converge we need to correctly manage the exploration vs. exploitation tradeoff. What property needs to be hold for the exploration strategy? **(5)**

19 a) Given the following sequence of states observed from the beginning of an episode, $s_2, s_1, s_3, s_2, s_1, s_2, s_1, s_6$, what is the eligibility value, $\mathbf{e_7(s_1)}$, of state $\mathbf{s_1}$at time step 7 given trace decay parameter λ, discount rate γ, and initial value, $\mathbf{e_0(s_1) = 0}$, when accumulating traces are used? What is the eligibility value if replacing traces are used? **(8)**

b) Suppose that we are using a policy gradient method to solve a reinforcement learning problem and the policy returned by the method is not optimal. Give three plausible reasons for such an outcome? **(6)**

OR

20 a) Suppose that we have a Q-value function represented as a sigmoid function of a set of features: **(8)**

$$Q(\phi, a) = \frac{1}{1 + e^{\theta^T \phi}}$$

Write down the update rule that Sarsa would give for this function.

b) Suppose that in a particular problem, the agent keeps going back to the same state in a loop. What is the maximum value that can be taken by the eligibility trace of such a state if we consider accumulating traces with **λ = 0.25** and **γ = 0.8**? **(6)**

Teaching Plan

No	Topic	No. of Lectures (42)
	Module-1 (Review Of Probability Concepts) TB-2(Ch 2,3,4,5) (8 hours)	
1.1	Axioms of probability, concepts of random variables	1 hour
1.2	Probability mass function	1 hour
1.3	Probability density function	1 hour
1.4	Cumulative density functions	1 hour
1.5.	Expectation of random variables	1 hour
1.6.	Joint and multiple random variables	1 hour
1.7	Conditional and marginal distributions	1 hour
1.8	Correlation and independence	1 hour
	Module-2 (Markov Decision Process) TB-1(Ch 1,3)(8 hours)	
2.1.	Introduction to Reinforcement Learning(RL) terminology - Examples of RL, Elements of RL, Limitations and Scope of RL	1 hour
2.2	Finite Markov Decision Processes	1 hour
2.3	The Agent–Environment Interface	1 hour
2.4.	Goals and Rewards	1 hour
2.5.	Returns and Episodes	1 hour
2.6.	Policies and Value Functions	1 hour
2.7	Optimal Policies and Optimal Value Functions	1 hour
2.8	Optimal Policies and Optimal Value Functions	1 hour
	Module-3 (Prediction And Control) TB-1(Ch 4,5) (9 hours)	

3.1	Policy Evaluation (Prediction)	1 hour
3.2	Policy Improvement	1 hour
3.3	Policy Iteration, Value Iteration	1 hour
3.4	Monte Carlo Prediction	1 hour
3.5	Monte Carlo Estimation of Action Values	1 hour
3.6	Monte Carlo Control, Monte Carlo Control without Exploring Starts	1 hour
3.7	Off-policy Prediction via Importance Sampling	1 hour
3.8	Incremental Implementation	1 hour
3.9	Off-policy Monte Carlo Control	1 hour
Module-4 (Temporal-Difference(Td) Methods) TB-1 (Ch 6,7) (8 hours)		
4.1	TD Prediction, Advantages of TD Prediction Methods	1 hour
4.2	Optimality of TD(0)	1 hour
4.3	Sarsa: On-policy TD Control	1 hour
4.4	Q-learning: Off-policy TD Control	1 hour
4.5	Expected Sarsa	1 hour
4.6	n-step TD Prediction, n-step Sarsa	1 hour
4.7	n-step Off-policy Learning	1 hour
4.8	Off-policy Learning Without Importance Sampling: The n-step Tree Backup Algorithm	1 hour
Module-5 (Function Approximation Method) TB-1 (Ch 9,12,13) (9 hours)		
5.1	Value-function Approximation	1 hour
5.2	The Prediction Objective	1 hour
5.3	Stochastic-gradient Methods	1 hour
5.4	Linear Methods	1 hour
5.5	The Lambda-return , TD(Lambda)	1 hour
5.6	n-step Truncated Lambda-return Methods, Sarsa(Lambda)	1 hour
5.7	Policy Approximation and its Advantages	1 hour
5.8	The Policy Gradient Theorem, REINFORCE: Monte Carlo Policy Gradient	1 hour
5.9	REINFORCE with Baseline, Actor–Critic Methods	1 hour

CST499	LOGIC FOR COMPUTER SCIENCE	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		VAC	3	1	0	4	2019

Preamble: This course enables the learners to understand the concepts of various logics used in computer science. The course covers the standard and most popular logics such as propositional logic, predicate logic, linear temporal logic, computation tree logic, Hoare logic and modal logic. This course helps the students to develop solutions for specification and verification of real world systems.

Prerequisite: Nil

Mapping of course outcomes with program outcomes

CO1	Explain the concepts of Predicate Logic, Propositional Logic, Linear Temporal Logic, Computation Tree Logic, Hoare Logic and Modal Logic as a formal language. **(Cognitive Knowledge Level: Understand)**
CO2	Develop proofs to show the satisfiability, validity and equivalence of logic formulas.**(Cognitive Knowledge Level: Apply)**
CO3	Illustrate model checking and program verification to prove correctness of systems.**(Cognitive Knowledge Level: Apply)**
CO4	Demonstrate *Alloy Analyzer* to model and analyze software systems. **(Cognitive Knowledge Level: Apply)**
CO5	Demonstrate *New Symbolic Model Verifier (NuSMV)* as a model checking tool to check the validity of temporal logic formulas.**(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓							✓

CO5	✓	✓	✓	✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests1&2)	**25 marks**
Continuous AssessmentAssignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Propositional Logic)

Declarative Sentences, Natural Deduction, Propositional Logic as a Formal Language, Semantics of Propositional Logic, Normal Forms, SAT Solvers.

Module– 2(Predicate Logic)

The Need of a Richer Language, Predicate Logic as a Formal Language, Proof Theory of Predicate Logic, Semantics of Predicate Logic, Undecidability of Predicate Logic, Expressiveness of Predicate Logic.

Module - 3 (Verification by Model Checking)

Motivation for Verification, Linear Time Temporal Logic (LTL), Model Checking Systems, Tools, Properties, Branching Time Logic, Computation Tree Logic (CTL) and the Expressive Powers of LTL and CTL, Model Checking Algorithms, The Fixed Point Characterization of CTL.

Module–4 (Program Verification)

Why Should We Specify and Verify Code, A Framework for Software Verification, Proof Calculus for Partial Correctness, Proof Calculus for Total Correctness, Programming by Contract.

Module–5 (Modal Logics and Agents)

Modes of Truth, Basic Modal Logic, Logic Engineering, Natural Deduction, Reasoning about Knowledge in a Multi-Agent System.

Text Books

1. Michael Huth and Mark Ryan, Logic in Computer Science, 2/e, Cambridge University Press, 2004.

Reference Books

1. Daniel Jackson, Software Abstractions, MIT Press, 2011.
2. Roberto Cavada, Alessandro Cimatti, Gavin Keighren, Emanuele Olivetti, Marco Pistore and Marco Roveri, NuSMV 2.6 Tutorial (available at *https://nusmv.fbk.eu*).
3. Tutorial for Alloy Analyzer 4.0 (available at *https://alloytools.org/tutorials/online/*).

Course Level Assessment Questions

Course Outcome1 (CO1):

1. Express the following statements as appropriate logic formulas.
   a. If the barometer falls, either it will rain or it will snow.
   b. No student attended every lecture.
   c. Once you are on the field, you keep on playing until the game is over.
   d. There are eight planets in the solar system.
2. Explain Horn Clause and Horn Formula.
3. Explain modal logic.

Course Outcome 2(CO2):

1. Prove the validity of the following sequents.

   $(p \wedge q) \wedge r, s \wedge t \vdash q \wedge s$

2. Prove the validity of

   (a) $\forall x \, \forall y \, P(x, y) \vdash \forall u \, \forall v \, P(u, v)$
   (b) $\exists x \, \exists y \, F(x, y) \vdash \exists u \, \exists v \, F(u, v)$
   (c) $\exists x \, \forall y \, P(x, y) \vdash \forall y \, \exists x \, P(x, y)$.

3. Prove that for all paths π of all models, $\pi \vDash \phi \, W \, \psi \wedge F \, \psi$ implies $\pi \vDash \phi \, U \, \psi$.

Course Outcome 3(CO3):

1. Consider an LTL formula $\phi \equiv (a\ U\ b) \longrightarrow F\ b$. Is ϕ valid? If yes, give an automata-theoretic proof of validity (i.e., construct a suitable NBA and use nested DFS to check an appropriate persistence condition). Otherwise, give a transition system that violates the formula. Illustrate the constructions clearly.

2. A familiar command missing from the core language (described in the text book) is the *for-statement*. It may be used to sum the elements in an array, for example, by programming as follows:

```
s = 0;
for (i = 0; i <= max; i = i+1) {
        s = s + a[i];
}
```

After performing the initial assignment s = 0, this executes i = 0 first, then executes the body *s = s + a[i]* and the incrementation *i = i + 1* continually untili*<= max* becomes false. Explain how ***for(C1;B;C2) {C3}*** can be defined as a derived program in our core language.

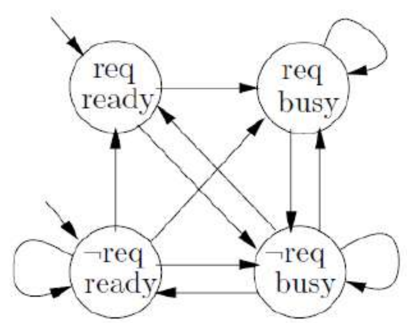
Course Outcome 4(CO4): .

1. Use *Alloy Analyzer* to model and solve the following problem.

A farmer is on one shore of a river and has with him a fox, a chicken, and a sack of grain. He has a boat that fits one object besides himself. In the presence of the farmer nothing gets eaten, but if left without the farmer, the fox will eat the chicken, and the chicken will eat the grain. How can the farmer get all three possessions across the river safely?

Course Outcome 5(CO5):

1. Simulate the following system using NuSMV..



Verify that $G\ (req\ \longrightarrow F\ busy)$ holds in all initial states.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SEVENTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST499

Course Name: Logic for Computer Science

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Check the validity of the following sequents.
   a. $\sim p \longrightarrow \sim q \ \vdash q \longrightarrow p$
   b. $\sim(\sim p \vee q\ ) \ \vdash p)$

2. For the formula $\phi = p \wedge \sim(q \vee \sim p)$, we compute the inductively defined translation as $T(\phi) = p \wedge \sim\sim(\sim q \wedge \sim\sim p)$. Draw the parse tree of $T(\phi)$.

3. Translate the following into predicate logic.
   a. All red things are in the box.
   b. No animal is both a cat and a dog.

4. Let ϕ be $\exists x\ (P(y,z) \wedge\ (\forall y\ (\sim Q(y,x) \vee\ P(y,z))))$, where P and Q are predicate symbols with two arguments. Identify all bound and free variables in ϕ.

5. Show the syntax of Computation Tree Logic (CTL).

6. Prove that the LTL equivalence between $\phi\ U\ \psi$and$\sim\left(\sim\psi\ U\ (\sim\phi\ \wedge \sim\psi)\right) \wedge F\psi$.

7. Explain the need of specification and verification of code.

8. In what circumstances would $if\ (B)\{C1\}\ else\ \{C2\}$ fail to terminate?

9. Illustrate Kripke model.

10. With an example, explain the equivalences between modal formulas.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Give the rules for Natural Deduction in propositional logic. **(6)**

(b) Use Natural Deduction to show the equivalence of the following formulas. **(8)**

a. $(p \wedge q) \wedge r , s \wedge t \ \vdash q \wedge s$

b. $(q \longrightarrow r) \longrightarrow ((\sim q \longrightarrow \sim p) \longrightarrow (p \longrightarrow r))$

OR

12. (a) What is a Horn Formula? How do you decide the satisfiability of a Horn formula. **(6)**

(b) Check the satisfiability of the following Horn Formulas. **(8)**

a. $(p \wedge q \wedge s \longrightarrow p) \wedge (q \wedge r \longrightarrow p) \wedge (p \wedge s \longrightarrow s)$

b. $(T \longrightarrow q) \wedge (T \longrightarrow s) \wedge (w \longrightarrow \perp) \wedge (p \wedge q \wedge s \longrightarrow v) \wedge (v \longrightarrow s) \wedge (T \longrightarrow r) \wedge (r \longrightarrow p)$

13. (a) Use Natural Deduction to prove the following equivalences. **(8)**

a. $\forall x\big(Q(x) \rightarrow R(x)\big), \exists x(P(x) \wedge Q(x)) \ \vdash \ \exists x(P(x) \wedge R(x))$

b. $\exists x P(x), \forall x\big(P(x) \longrightarrow Q(x)\big) \ \vdash \ \forall y Q(y)$

(b) Illustrate how Quantifier Equivalences can be used to check the equivalence of predicate logic formulas. **(6)**

OR

14. (a) Model the following system in *Alloy Analyzer*. **(7)**

There is an entity named **Person**, **Man** and **Woman** are two specializations of the entity **Person**. Every **Person** has a **Father** (a **Man**) and a **Mother** as **Parent**. The **Parent**s of a **Person** should be married. A **Man**'s **spouse** should be a **Woman** and a **Woman**'s **spouse** should be a **Man**. The **spouse** relation is symmetric.

Add a predicate to check whether marriage between siblings is possible in the above system.

(b) Explain Existential Second Order Logic and Universal Second Order Logic. **(7)**

15. (a) Model the Ferryman problem using New Symbolic Model Verifier (NuSMV). **(7)**

(b) Construct a Generalized Buchi Automaton for the LTL formula Oa. **(7)**

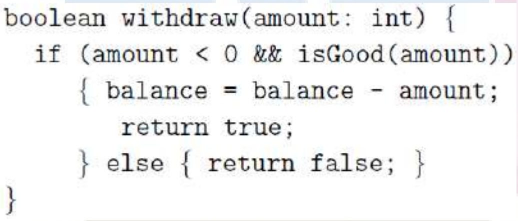
OR

16. (a) Show the closure of the LTL formula $\sim p\ U\ (F\ r \lor G \sim q\ \longrightarrow q\ W \sim r)$. **(7)**

(b) Explain the Fixed Point Characterization of CTL. **(7)**

17. (a) Illustrate partial correctness and total correctness in program verification. **(7)**

(b) **(7)**



```
boolean withdraw(amount: int) {
  if (amount < 0 && isGood(amount))
     { balance = balance - amount;
       return true;
     } else { return false; }
}
```

Consider the method named *withdraw* which attempts to withdraw amount from an integer field *balance* of the class within which the method *withdraw* lives. This method makes use of another method *isGood* which returns true iff the value of *balance* is greater than or equal to the value of *amount*.

Write a contract for the method *isGood*. Use that contract to show the validity of the contract for *withdraw*:

Method name: *withdraw*

Input: *amount* of type int

Assumes: *0 <= balance*

Guarantees: *0 <= balanace*

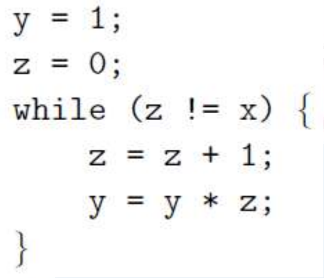
Output: of type boolean

Modifies only: *balance*

Upon validation, this contract establishes that all calls to *withdraw* leave *0<=balance* invariant.

OR

18. (a) Consider the program for computing the factorial of a number as given below. **(7)**

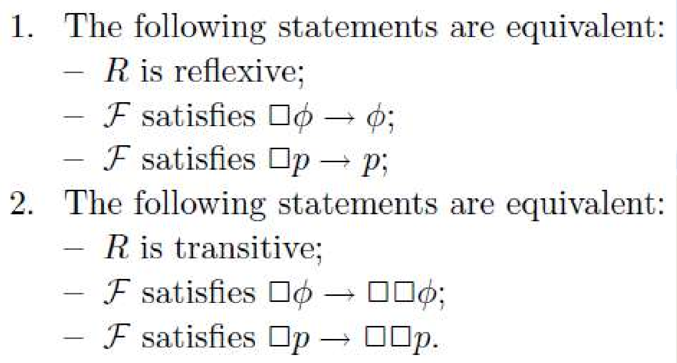


```
y = 1;
z = 0;
while (z != x) {
    z = z + 1;
    y = y * z;
}
```

Find a partial correctness proof for the above program.

(b) Explain the proof calculus for total correctness. **(7)**

19. (a) Let $\mathcal{F} = (W, R)$ be a frame. Prove the two claims given below. **(7)**

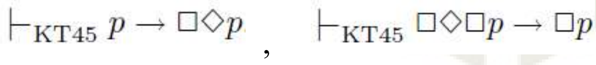


1. The following statements are equivalent:
   - R is reflexive;
   - $\mathcal{F}$ satisfies $\Box\phi \to \phi$;
   - $\mathcal{F}$ satisfies $\Box p \to p$;
2. The following statements are equivalent:
   - R is transitive;
   - $\mathcal{F}$ satisfies $\Box\phi \to \Box\Box\phi$;
   - $\mathcal{F}$ satisfies $\Box p \to \Box\Box p$.

(b) Explain the modal logics *K*, *KT45* and *KT4*. **(7)**

OR

20. (a) Prove the following using Natural Deduction. **(8)**



$\vdash_{\mathrm{KT45}} p \to \Box\Diamond p$, $\vdash_{\mathrm{KT45}} \Box\Diamond\Box p \to \Box p$

(b) Find a modal logic to formalize and solve *The Wise-Men Puzzle*. **(6)**

Teaching Plan

No	Contents	No. of Lecture Hours (45 hrs)
	Module-1(Propositional Logic) (8 hours)	
1.1	Declarative Sentences, Natural Deduction	1 hour
1.2	Rule for Natural Deduction	1 hour
1.3	Derived Rules, Natural Deduction in Summary	1 hour
1.4	Provable Equivalence, Proof by Contradiction. Propositional Logic as a Formal language	1 hour
1.5	Semantics of Propositional Logic – The Meaning of Logical Connectives, Soundness of Propositional Logic, Completeness of Propositional Logic (Proof not required)	1 hour
1.6	Semantic Equivalence, Satisfiability and Validity	1 hour
1.7	Normal Forms – Conjunctive Normal Forms and Validity, Horn Clauses and Satisfiability	1 hour
1.8	SAT Solvers – A Linear Solver, A Cubic Solver	1 hour
	Module-2(Predicate Logic) (7 hours)	
2.1	The Need of a Richer language, Predicate Logic as a Formal Language – Terms, Formulas, Free and Bound Variables, Substitution	1 hour
2.2	Proof Theory of Predicate Logic – Natural Deduction Rules	1 hour
2.3	Proof Theory of Predicate Logic – Quantifier Equivalences	1 hour
2.4	Semantics of Predicate Logic – Models, Semantic Entailment, The Semantics of Equality	1 hour
2.5	Undecidabilty of Predicate Logic (*no proof required*), Expressiveness of Predicate Logic – Existential Second Order Logic, Universal Second Order Logic	1 hour
2.6	Micromodels of Software – State Machines, A Software Micromodel (*Alloy*) (Lecture 1)	1 hour
2.7	A Software Micromodel (*Alloy*) (Lecture 2)	1 hour
	Module-3(Verification by Model Checking) (13 hours)	

3.1	Motivation for Verification, Linear Time Temporal Logic (LTL) - Syntax	1 hour
3.2	Semantics of LTL – Practical Patterns of Specifications, Important Equivalences between LTL Formulas, Adequate Sets of Connectives for LTL	1 hour
3.3	Introduction to model checking	1 hour
3.4	Model Checking Systems, Tools, Properties	1 hour
3.5	Model checking example: Mutual Exclusion	1 hour
3.6	The New Symbolic Model Verifier (NuSMV) Model Checker – Introduction, Mutual Exclusion Revisited	1 hour
3.7	The NuSMV Model Checker – The Ferryman, The Alternating Bit Protocol	1 hour
3.8	Branching Time Logic – Syntax of Computation Tree Logic (CTL), Semantics of CTL	1 hour
3.9	Practical Patterns of Specification, Important Equivalences between CTL Formulas, Adequate Sets of CTL Connectives	1 hour
3.10	CTL and the Expressive Powers of LTL and CTL – Boolean Combinations of Temporal Formulas in CTL	1 hour
3.11	Model-Checking Algorithms – The CTL Model Checking Algorithm	1 hour
3.12	CTL Model Checking with Fairness	1 hour
3.13	The LTL Model Checking Algorithm (Algorithm only)	1 hour
Module-4 (Program Verification) (8 hours)		
4.1	Introduction to Program Verification, Need of Specification and Verification of Code	1 hour
4.2	A Framework for Software Verification – A Core Programming Language, Hoare Triples	1 hour
4.3	A Framework for Software Verification – Partial and Total Correctness, Program Variables and Logical Variables	1 hour
4.4	Proof Calculus for partial Correctness – Proof Rules	1 hour
4.5	Proof Calculus for partial Correctness – Proof Tableaux	1 hour
4.6	Proof Calculus for partial Correctness – A Case Study: Minimal-Sum Section	1 hour
4.7	Proof Calculus for Total Correctness	1 hour
4.8	Programming by Contract	1 hour
Module-5 (Modal Logics and Agents) (9 hours)		

5.1	Modes of Truth, basic Modal Logic - Syntax	1 hour
5.2	Basic Modal Logic - Semantics	1 hour
5.3	Logic Engineering – The Stock of Valid Formulas, Important Properties of the Accessibility Relation	1 hour
5.4	Logic Engineering – Correspondence Theory, Some Modal Logics	1 hour
5.5	Natural Deduction	1 hour
5.6	Reasoning about Knowledge in a Multi-Agent System –Examples (The Wise - Man Puzzle, The Muddy – Children Puzzle)	1 hour
5.7	The Modal Logic KT45n	1 hour
5.8	Natural Deduction for KT45n	1 hour
5.9	Formalizing the Examples (The Wise - Man Puzzle, The Muddy – Children Puzzle)	1 hour

SEMESTER VIII

CST402	DISTRIBUTED COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	2	1	0	3	2019

Preamble: The purpose of this course is to understand the system models, algorithms and protocols that allow computers to communicate and coordinate their actions to solve a problem. This course helps the learner to understand the distributed computation model and various concepts like global state, termination detection, mutual exclusion, deadlock detection, shared memory, failure recovery, consensus, file system. It helps the learners to develop solutions to problems in distributed computing environment.

Prerequisite: Basic knowledge in data structures and operating systems.

Course Outcomes: After the completion of the course the student will be able to

CO1	Summarize various aspects of distributed computation model and logical time. **(Cognitive Knowledge Level: Understand)**
CO2	Illustrate election algorithm, global snapshot algorithm and termination detection algorithm. **(Cognitive Knowledge Level: Apply)**
CO3	Compare token based, non-token based and quorum based mutual exclusion algorithms. **(Cognitive Knowledge Level: Understand)**
CO4	Recognize the significance of deadlock detection and shared memory in distributed systems. **(Cognitive Knowledge Level: Understand)**
CO5	Explain the concepts of failure recovery and consensus. **(Cognitive Knowledge Level: Understand)**
CO6	Illustrate distributed file system architectures. **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓									✓
CO6	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	**30**	**30**	**30**
Understand	**50**	**50**	**50**
Apply	**20**	**20**	**20**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests1&2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Distributed systems basics and Computation model)

Distributed System – Definition, Relation to computer system components, Motivation, Primitives for distributed communication, Design issues, Challenges and applications. A model of distributed computations – Distributed program, Model of distributed executions, Models of communication networks, Global state of a distributed system, Cuts of a distributed computation, Past and future cones of an event, Models of process communications.

Module – 2 (Election algorithm, Global state and Termination detection)

Logical time – A framework for a system of logical clocks, Scalar time, Vector time. Leader election algorithm – Bully algorithm, Ring algorithm. Global state and snapshot recording algorithms – System model and definitions, Snapshot algorithm for FIFO channels – Chandy Lamport algorithm. Termination detection – System model of a distributed computation, Termination detection using distributed snapshots, Termination detection by weight throwing,Spanning-tree-based algorithm.

Module – 3 (Mutual exclusion and Deadlock detection)

Distributed mutual exclusion algorithms – System model, Requirements of mutual exclusion algorithm. Lamport's algorithm, Ricart–Agrawala algorithm, Quorum-based mutual exclusion algorithms – Maekawa's algorithm. Token-based algorithm – Suzuki–Kasami's broadcast algorithm. Deadlock detection in distributed systems – System model, Deadlock handling strategies, Issues in deadlock detection, Models of deadlocks.

Module – 4 (Distributed shared memory and Failure recovery)

Distributed shared memory – Abstraction and advantages. Shared memory mutual exclusion – Lamport's bakery algorithm. Check pointing and rollback recovery – System model, consistent and inconsistent states, different types of messages, Issues in failure recovery, checkpoint based recovery, log based roll back recovery.

Module – 5 (Consensus and Distributed file system)

Consensus and agreement algorithms – Assumptions, The Byzantine agreement and other problems, Agreement in (message-passing) synchronous systems with failures – Consensus algorithm for crash failures. Distributed file system – File service architecture, Case studies: Sun Network File System, Andrew File System, Google File System.

(Note: Proof of correctness and performance analysis are not expected for any of the algorithms in the syllabus).

Text Books

1. Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2011.

Reference Books

1. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair. Distributed Systems: Concepts and Design, Addison Wesley, Fifth edition.
2. Kai Hwang, Geoffrey C Fox, Jack J Dongarra, Distributed and Cloud Computing – From Parallel Processing to the Internet of Things, Morgan Kaufmann Publishers, 2012.
3. Sukumar Ghosh, Distributed Systems: An Algorithmic Approach, CRC Press, Second edition, 2015.
4. Maarten Van Steen, Andrew S. Tanenbaum, Distributed Systems, Prentice Hall of India,Third edition, 2017.
5. Randy Chow and Theodore Johnson, Distributed Operating Systems and Algorithm Analysis, Pearson Education India, First edition, 2009.
6. Valmir C. Barbosa, An Introduction to Distributed Algorithms, MIT Press, 2003.

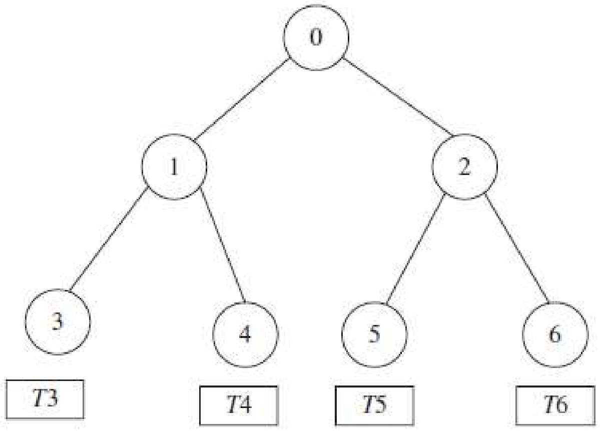
Course Level Assessment Questions

Course Outcome1 (CO1):

1. Define logical clock and explain the implementation of the logical clock.
2. Explain different forms of load balancing.

Course Outcome 2(CO2):

1. Apply ring-based leader election algorithm with 10 processes in the worst-performing case. Count the number of messages needed.
2. Apply spanning tree-based termination detection algorithm in the following scenario. The nodes are processes 0 to 6. Leaf nodes 3, 4, 5, and 6 are each given tokens T3, T4, T5 and T6 respectively. Leaf nodes 3, 4, 5 and 6 terminate in the order, but before terminating node 5,it sends a message to node 1.



Course Outcome 3(CO3):

1. What are the requirements of mutual exclusion algorithms?
2. Illustrate Suzuki- Kasami's broadcast algorithm.

Course Outcome 4(CO4):

1. Compare different models of deadlocks.
2. Illustrate the detailed abstraction of distributed shared memory and interaction with application processes.

Course Outcome 5(CO5):

1. Explain how consensus problem differs from the Byzantine agreement problem.
2. Classify different log based roll back recovery techniques.

Course Outcome 6 (CO6):

1. Explain the directory service and its interface operations in a file service architecture.
2. Describe the architecture of Google file system.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST402

Course Name: Distributed Computing

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Identify any three distributed applications and for each application, determine which all motivating factors are important for building an application over a distributed system.

2. Assume that the surface of the past cone form a consistent cut. Does it mean that all events on the surface of the past cone are always concurrent? Demonstrate with the help of an example.

3. Specify the issues in recording a global state.

4. Explain the rules used to update clocks in scalar time representation.

5. Describe how quorum-based mutual exclusion algorithms differ from the other categories of mutual exclusion algorithms.

6. Explain with example, how wait-for-graphs can be used in deadlock detection.

7. List any three advantages of using distributed shared memory.

8. Define the no-orphans consistency condition.

9. Define Byzantine agreement problem.

10. Differentiate between whole file serving and whole file caching in Andrew file system **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the three different models of the service provided by communication networks. **(6)**

(b) Explain how the causal dependency between events in distributed execution is defined using Lamport's happened before relationship. **(8)**

OR

12. (a) Address the various strategies that can be adopted to satisfy the requirements of a reliable and fault tolerant distributed system. **(6)**

(b) Which are the different versions of send and receive primitives for distributed communication? Explain. **(8)**

13. (a) Illustrate bully algorithm for electing a new leader. Does the algorithm meet liveness and safety conditions? **(7)**

(b) Clearly mentioning assumptions, explain the rules of termination detection using distributed snapshots. **(7)**

OR

14. (a) In Chandy-Lamport algorithm for recording global snapshots, explain how the recorded local snapshots can be put together to create the global snapshot. Can multiple processes initiate the algorithm concurrently? **(6)**

(b) Illustrate the working of spanning tree based termination detection algorithm. **(8)**

15. (a) Explain and illustrate Lamport's mutual exclusion algorithm. **(8)**

(b) Discuss the three types of messages required for deadlock handling in Maekawa's algorithm. Explain how Maekawa's algorithm handles deadlocks. **(6)**

OR

16. (a) Explain and illustrate Ricart- Agrawala algorithm for achieving mutual exclusion. **(8)**

(b) Explain any three different models of deadlock. **(6)**

17. (a) What are the issues in failure recovery? Illustrate with suitable examples. **(7)**

(b) Show that Lamport's Bakery algorithm for shared memory mutual exclusion, satisfy the three requirements of critical section problem. **(7)**

OR

18. (a) Differentiate consistent and inconsistent states with examples. **(4)**

(b) What is check point-based rollback-recovery? Explain the three classifications of check point-based rollback-recovery. **(10)**

19. (a) Explain consensus algorithm for crash failures under synchronous systems. **(6)**

(b) Summarize distributed file system requirements. **(8)**

OR

20. (a) Differentiate Andrew file system and NFS. **(4)**

(b) Explain Sun NFS architecture with diagram. **(10)**

Teaching Plan

No	Contents	No. of Lecture Hours (35 hours)
Module – 1(Distributed systems basics and Computation model) (7 hours)		
1.1	Distributed System – Definition, Relation to computer system components	1 hour
1.2	Primitives for distributed communication.	1 hour
1.3	Design issues, challenges and applications.	1 hour
1.4	Design issues, challenges and applications.	1 hour
1.5	A model of distributed computations – Distributed program, Model of distributed executions	1 hour
1.6	Models of communication networks, Global state of a distributed system, Cuts of a distributed computation	1 hour
1.7	Cuts of a distributed computation, Past and future cones of an event, Models of process communications.	1 hour
Module – 2 (Election algorithm, Global state and Termination detection) (8 hours)		
2.1	Logical time – A framework for a system of logical clocks, Scalar time	1 hour
2.2	Vector time.	1 hour
2.3*	Leader election algorithm – Bully Algorithm, Ring Algorithm	1 hour
2.4	Global state and snapshot recording algorithms – System model and definitions	1 hour
2.5*	Snapshot algorithm for FIFO channels – Chandy Lamport algorithm.	1 hour
2.6	Termination detection – System model of a distributed computation	1 hour
2.7*	Termination detection using distributed snapshots	1 hour
2.8*	Termination detection by weight throwing, Spanning tree-based algorithm	1 hour
Module – 3 (Mutual exclusion and Deadlock detection) (6 hours)		
3.1*	Distributed mutual exclusion algorithms – System model, Lamport's algorithm	1 hour

3.2*	Ricart–Agrawala algorithm	1 hour
3.3*	Quorum-based mutual exclusion algorithms – Maekawa's algorithm	1 hour
3.4*	Token-based algorithm – Suzuki–Kasami's broadcast algorithm.	1 hour
3.5	Deadlock detection in distributed systems – System model, Deadlock handling strategies, Issues in deadlock detection	1 hour
3.6	Models of deadlocks	1 hour
Module – 4 (Distributed shared memory and Failure recovery) (7 hours)		
4.1	Distributed shared memory – Abstraction and advantages.	1 hour
4.2*	Shared memory mutual exclusion – Lamport's bakery algorithm.	1 hour
4.3	Checkpointing and rollback recovery – System model, consistent and inconsistent states	1 hour
4.4	different types of messages, Issues in failure recovery	1 hour
4.5	checkpoint based recovery	1 hour
4.6	log based roll back recovery.	1 hour
4.7	log based roll back recovery.	1 hour
Module – 5(Consensus and Distributedfile system) (7 hours)		
5.1	Consensus and agreement algorithms – Assumptions, The Byzantine agreement and other problems	1 hour
5.2	Agreement in (message-passing) synchronous systems with failures – Consensus algorithm for crash failures	1 hour
5.3*	Agreement in (message-passing) synchronous systems with failures – Consensus algorithm for crash failures	1 hour
5.4	Distributed File System – File Service Architecture	1 hour
5.5	Case Studies: Sun Network File System	1 hour
5.6	Andrew File System	1 hour
5.7	Google File System.	1 hour

* **Proof of correctness and performance analysis are not expected for this algorithm**.

CST404	COMPREHENSIVE COURSE VIVA	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	1	0	0	1	2019

The objective of this Course viva is to ensure the basic knowledge of each student in the most fundamental core courses in the curriculum. The viva voce shall be conducted based on the core subjects studied from third to eighth semester. This course helps the learner to become competent in placement tests and other competitive examinations.

Guidelines

1. The course should be mapped with a faculty and classes shall be arranged for practicing questions based on the core courses listed in the curriculum.
2. The viva voce will be conducted by the same three member committee assigned for final project phase II evaluation. It comprises of Project coordinator, expert from Industry/research Institute and a senior faculty from a sister department.
3. The pass minimum for this course is 25.
4. The mark will be treated as internal and should be uploaded along with internal marks of other courses.
5. Comprehensive Viva should be conducted along with final project evaluation by the three member committee.

Mark Distribution

Total marks: 50, only CIE, minimum required to pass : **25**

CSD416	PROJECT PHASE II	CATEGORY	L	T	P	CREDIT
		PWS	0	0	12	4

Preamble: The course 'Project Work' is mainly intended to evoke the innovation and invention skills in a student. The course will provide an opportunity to synthesize and apply the knowledge and analytical skills learned, to be developed as a prototype or simulation. The project extends to 2 semesters and will be evaluated in the 7th and 8th semester separately, based on the achieved objectives. One third of the project credits shall be completed in 7th semester and two third in 8th semester. It is recommended that the projects may be finalized in the thrust areas of the respective engineering stream or as interdisciplinary projects. Importance should be given to address societal problems and developing indigenous technologies.

Course Objectives

- To apply engineering knowledge in practical problem solving.
- To foster innovation in design of products, processes or systems.
- To develop creative thinking in finding viable solutions to engineering problems.

Course Outcomes [COs]: After successful completion of the course, the students will be able to:

CO1	Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: **Apply**).
CO2	Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: **Apply**).
CO3	Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: **Apply**).
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: **Apply**).
CO5	Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: **Analyze**).
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: **Apply**).

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	1	2	2	2	1	1	1	1	2
CO2	2	2	2		1	3	3	1	1		1	1
CO3									3	2	2	1
CO4					2			3	2	2	3	2
CO5	2	3	3	1	2							1
CO6					2			2	2	3	1	1

Abstract POs defined by National Board of Accreditation			
PO #	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO0	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

PROJECT PHASE II

Phase 2 Targets

- In depth study of the topic assigned in the light of the report prepared under Phase - I;
- Review and finalization of the approach to the problem relating to the assigned topic.
- Preparing a detailed action plan for conducting the investigation, including teamwork.
- Detailed Analysis/ Modeling / Simulation/ Design/ Problem Solving/Experiment as needed.
- Final development of product/ process, testing, results, conclusions and future directions.
- Preparing a paper for Conference Presentation/ Publication in Journals, if possible.
- Presenting projects in Project Expos conducted by the University at the cluster level and/ or state level as well as others conducted in India and abroad.
- Filing Intellectual Property Rights (IPR) if applicable.
- Preparing a report in the standard format for being evaluated by the Department Assessment Board.
- Final project presentation and viva voce by the assessment board including the external expert.

Evaluation Guidelines & Rubrics

Total: 150 marks (Minimum required to pass: 75 marks).

- Project progress evaluation by guide: 30 Marks.
- Two interim evaluations by the Evaluation Committee: 50 Marks (25 marks for each evaluation).
- Final evaluation by the Final Evaluation committee: 40 Marks
- Quality of the report evaluated by the evaluation committee: 30 Marks

(The evaluation committee comprises HoD or a senior faculty member, Project coordinator and project supervisor. The final evaluation committee comprises of Project coordinator, expert from Industry/research/academic Institute and a senior faculty from a sister department).

Evaluation by the Guide

The guide/supervisor must monitor the progress being carried out by the project groups on regular basis. In case it is found that progress is unsatisfactory it should be reported to the Department Evaluation Committee for necessary action. The presence of each student in the group and their involvement in all stages of execution of the project shall be ensured by the guide. Project evaluation by the guide: 30 Marks. This mark shall be awarded to the students in his/her group by considering the following aspects:

Project Scheduling & Distribution of Work among Team members: Detailed and extensive Scheduling with timelines provided for each phase of project. Work breakdown structure well defined. (5)

Literature survey: Outstanding investigation in all aspects. (4)

Student's Diary/ Daily Log: The main purpose of writing daily diary is to cultivate the habit of documenting and to encourage the students to search for details. It develops the students' thought process and reasoning abilities. The students should record in the daily/weekly activity diary the day to day account of the observations, impressions, information gathered and suggestions given, if any. It should contain the sketches & drawings related to the observations made by the students. The daily/weekly activity diary shall be signed after every day/week by the guide. (7)

Individual Contribution: The contribution of each student at various stages. (9)

Completion of the project: The students should demonstrate the project to their respective guide. The guide shall verify the results and see that the objectives are met. (5)

EVALUATION RUBRICS for PROJECT Phase II: Interim Evaluation - 1

No.	Parameters	Marks	Poor	Fair	Very Good	Outstanding
2-a	Novelty of idea, and Implementation scope [CO5] [Group Evaluation]	**5**	The project is not addressing any useful requirement. The idea is evolved into a non-implementable one. The work presented so far is lacking any amount of original work by the team.	Some of the aspects of the proposed idea can be implemented. There is still lack of originality in the work done so far by the team. The project is a regularly done theme/topic without any freshness in terms of specifications, features, and/or improvements.	Good evidence of an implementable project. There is some evidence for the originality of the work done by the team . There is fresh specifications/ features/improvements suggested by the team. The team is doing a design from fundamental principles, and there is some independent learning and engineering ingenuity.	The project has evolved into incorporating an outstandingly novel idea. Original work which is not yet reported anywhere else. Evidence for ingenious way of innovation which is also Implementable. Could be a patentable / publishable work.
			(0 – 1 Marks)	(2 – 3 Marks)	(4 Marks)	(5 Marks)
2-b	Effectiveness of task distribution among team members. [CO3] [Group Evaluation]	**5**	No task distribution of any kind. Members are still having no clue on what to do.	Task allocation done, but not effectively, some members do not have any idea of the tasks assigned. Some of the tasks were identified but not followed individually well.	Good evidence of task allocation being done, supported by project journal entries, identification of tasks through discussion etc. However, the task distribution seems to be skewed, and depends a few members heavily than others. Mostly the tasks are being followed by the individual members.	Excellent display of task identification and distribution backed by documentary evidence of team brainstorming, and project journal entries. All members are allocated tasks according to their capabilities, and as much as possible in an equal manner. The individual members are following the tasks in an excellent manner.
			(0 – 1 Marks)	(2 – 3 Marks)	(4 Marks)	(5 Marks)
2-c	Adherence to project schedule. [CO4] [Group Evaluation]	**5**	Little or no evidence of continued planning or scheduling of the project. The students did not stick to the plan what they were going to build nor plan on what materials / resources to use in the project. The students do not have any idea on the budget required even after the end of phase - I. No project journal kept or the journal.	There is some improvement in the primary plan prepared during phase I. There were some ideas on the materials /resources required, but not really thought out. The students have some idea on the finances required, but they have not formalized a budget plan. Schedules were not prepared. The project journal has no useful details on the project.	Good evidence of planning done and being followed up to a good extent after phase I. Materials were listed and thought out, but the plan wasn't followed completely. Schedules were prepared, but not detailed, and needs improvement. Project journal is presented but it is neither complete nor updated regularly.	Excellent evidence of enterprising and extensive project planning and follow-up since phase I. Continued use of project management/version control tool to track the project. Material procurement if applicable is progressing well. Tasks are updated and incorporated in the schedule. A well-kept project journal showed evidence for all the above, in addition to the interaction with the project guide.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)

2-d	Interim Results. [CO6] [Group assessment]	**5**	There are no interim results to show.	The team showed some interim results, but they are not complete / consistent to the current stage, Some corrections are needed.	The interim results showed were good and mostly consistent/correct with respect to the current stage. There is room for improvement.	There were significant interim results presented which clearly shows the progress.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)
2-e	Presentation [Individual assessment]	**5**	Very poor presentation and there is no interim results. The student has no idea about the project proposal.	Presentation is average, and the student has only a feeble idea about the team work.	Good presentation. Student has good idea about the team's project. The overall presentation quality is good.	Exceptionally good presentation. Student has excellent grasp of the project. The quality of presentation is outstanding.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)
Phase-II Interim Evaluation - 1 Total Marks: 25						

EVALUATION RUBRICS for PROJECT Phase II: Interim Evaluation – 2

No	Parameters	Marks	Poor	Fair	Very Good	Outstanding
2-f	Application of engineering knowledge [CO1] [Individual Assessment]	10	The student does not show any evidence of applying engineering knowledge on the design and the methodology adopted. The student's contribution in application of engineering knowledge in the project is poor.	The student appears to apply some basic knowledge, but not able to show the design procedure and the methodologies adopted in a comprehensive manner.	The student is able to show some evidence of application of engineering knowledge in the design and development of the project to good extent.	Excellent knowledge in design procedure and its adaptation. The student is able to apply knowledge from engineering domains to the problem and develop solutions.
			(0 – 3 Marks)	(4 – 6 Marks)	(7 - 9 Marks)	(10 Marks)
2-g	Involvement of individual members [CO3] [Individual Assessment]	5	No evidence of any Individual participation in the project work.	There is evidence for some amount of individual contribution, but is limited to some of the superficial tasks.	The individual contribution is evident. The student has good amount of involvement in core activities of the project.	Evidence available for the student acting as the core technical lead and has excellent contribution to the project.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)
2-h	Results and inferences upon execution [CO5] [Group Assessment]	5	None of the expected outcomes are achieved yet. The team is unable to derive any inferences on the failures/ issues observed. Any kind of observations or studies are not made.	Only a few of the expected outcomes are achieved. A few inferences are made on the observed failures/issues. No further work suggested.	Many of the expected outcomes are achieved. Many observations and inferences are made, and attempts to identify the issues are done. Some suggestions are made for further work.	Most of the stated outcomes are met. Extensive studies are done and inferences drawn. Most of the failures are addressed and solutions suggested. Clear and valid suggestions made for further work.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)
2-i	Documentation and presentation. .[CO6] [Individual assessment]	5	The individual student has no idea on the presentation of his/her part. The presentation is of poor quality.	Presentation's overall quality needs to be improved.	The individual's presentation performance is satisfactory.	The individual's presentation is done professionally and with great clarity. The individual's performance is excellent.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)

Phase-II Interim Evaluation - 2 Total Marks: 25

EVALUATION RUBRICS for PROJECT Phase II: Final Evaluation

No	Parameters	Marks	Poor	Fair	Very Good	Outstanding
2-j	Engineering knowledge. [CO1] [Group Assessment]	10	The team does not show any evidence of applying engineering knowledge on the design and the methodology adopted.	The team is able to show some of the design procedure and the methodologies adopted, but not in a comprehensive manner.	The team is able to show evidence of application of engineering knowledge in the design and development of the project to good extent. There is scope for improvement.	Excellent knowledge in design procedure and its adaptation. The team is able to apply knowledge from engineering domains to the problem and develop an excellent solution.
			(0 – 3 Marks)	(4 – 6 Marks)	(7 - 9 Marks)	(10 Marks)
2-k	Relevance of the project with respect to societal and/or industrial needs. [Group Assessment] [CO2]	5	The project as a whole do not have any societal / industrial relevance at all.	The project has some relevance with respect to social and/or industrial application. The team has however made not much effort to explore further and make it better.	The project is relevant to the society and/or industry. The team is mostly successful in translating the problem into an engineering specification and managed to solve much of it.	The project is exceptionally relevant to society and/or industry. The team has made outstanding contribution while solving the problem in a professional and/ or ethical manner.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)
2-i	Innovation / novelty / Creativity [CO5] [Group Assessment]	5	The project is not addressing any useful requirement. The idea is evolved into a non-implementable one. The work presented so far is lacking any amount of original work by the team.	Some of the aspects of the proposed idea appears to be practical. There is still lack of originality in the work done. The project is a regularly done theme/topic without any freshness in terms of specifications, features, and/ or improvements.	Good evidence of an implementable project. There is some evidence for the originality of the work done by the team. There is fresh specifications/ features/improvements suggested by the team. The team is doing a design from fundamental principles, and there is some independent learning and engineering ingenuity. Could be translated into a product / process if more work is done.	The project has evolved into incorporating an outstandingly novel idea. Original work which is not yet reported anywhere else. Evidence for ingenious way of innovation which is also Implementable. Could be a patentable publishable work.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)
2-m	Quality of results / conclusions / solutions. [CO1] [Group Assessment]	10	None of the expected outcomes are achieved. The team is unable to derive any inferences on the failures/issues observed. Any kind of observations or studies is not made.	Only a few of the expected outcomes are achieved. A few inferences are made on the observed failures/issues. No further work suggested.	Many of the expected outcomes are achieved. Many observations and inferences are made, and attempts to identify the issues are done. Some suggestions are made for further work.	Most of the stated outcomes are met. Extensive studies are done and inferences drawn. Most of the failures are addressed and solutions suggested. Clear and valid suggestions made for further work.
			(0 – 3 Marks)	(4 – 6 Marks)	(7 - 9 Marks)	(10 Marks)

2-n	Presentation - Part I Preparation of slides. [CO6] [Group Assessment].	5	The presentation slides are shallow and in a clumsy format. It does not follow proper organization.	Presentation slides follow professional style formats to some extent. However, its organization is not very good. Language needs to be improved. All references are not cited properly, or acknowledged. Presentation slides needs to be more professional.	Presentation slides follow a good style format and there are only a few issues. Organization of the slides is good. Most of references are cited properly. The flow is good and team presentation is neatly organized. Some of the results are not clearly shown. There is room for improvement.	The presentation slides are exceptionally good. Neatly organized. All references cited properly. Diagrams/Figures, Tables and equations are properly numbered, and listed. Results/ inferences clearly highlighted and readable.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)
	Presentation - Part II: Individual Communication [CO6] [Individual Assessment].	5	The student is not communicating properly. Poor response to questions.	The student is able to explain some of the content. The student requires a lot of prompts to get to the idea. There are language issues.	Good presentation/ communication by the student. The student is able to explain most of the content very well. There are however, a few areas where the student shows lack of preparation. Language is better.	Clear and concise communication exhibited by the student. The presentation is outstanding. Very confident and tackles all the questions without hesitation. Exceptional traits of communicator.
			(0 - 1 Marks)	(2 - 3 Marks)	(4 Marks)	(5 Marks)
Phase-II Final Evaluation, Marks: 40						

EVALUATION RUBRICS for PROJECT Phase II: Report Evaluation						
Sl. No.	**Parameters**	**Marks**	**Poor**	**Fair**	**Very Good**	**Outstanding**
2-o	Report [CO6]	30	The prepared report is shallow and not as per standard format. It does not follow proper organization. Contains mostly unacknowledged content. Lack of effort in preparation is evident. References are not cited. Unprofessional and inconsistent formatting.	Project report follows the standard format to some extent. However, its organization is not very good. Language needs to be improved. All references are not cited properly in the report. There is lack of formatting consistency.	Project report shows evidence of systematic documentation. Report is mostly following the standard style format and there are only a few issues. Organization of the report is good. Mostly consistently formatted. Most of references/sources are cited/ acknowledged properly.	The report is exceptionally good. Neatly organized. All references cited properly. Diagrams/Figures, Tables and equations are properly numbered, and listed and clearly shown. Language is excellent and follows professional styles. Consistent formatting and exceptional readability.
			(0 - 11 Marks)	(12 - 18 Marks)	(19 - 28 Marks)	(29 - 30 Marks)
Phase - II Project Report Marks: 30						

SEMESTER VIII

PROGRAM ELECTIVE III

CST414	DEEP LEARNING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: Deep Learning is the recently emerged branch of machine learning, particularly designed to solve a wide range of problems in Computer Vision and Natural Language Processing. In this course, the building blocks used in deep learning are introduced. Specifically, neural networks, deep neural networks, convolutional neural networks and recurrent neural networks. Learning and optimization strategies such as Gradient Descent, Nesterov Accelerated Gradient Descent, Adam, AdaGrad and RMSProp are also discussed in this course. This course will helps the students to attain sound knowledge of deep architectures used for solving various Vision and NLP tasks. In future, learners can master modern techniques in deep learning such as attention mechanisms, generative models and reinforcement learning.

Prerequisite: Basic understanding of probability theory, linear algebra and machine learning

Course Outcomes: After the completion of the course, the student will be able to

CO1	Illustrate the basic concepts of neural networks and its practical issues **(Cognitive Knowledge Level: Apply)**
CO2	Outline the standard regularization and optimization techniques for deep neural network **(Cognitive Knowledge Level: understand)**
CO3	Implement the foundation layers of CNN (pooling, convolutions) **(Cognitive Knowledge Level: Apply)**
CO4	Implement a sequence model using recurrent neural networks **(Cognitive Knowledge Level: Apply)**
CO5	Use different neural network/deep learning models for practical applications. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓		✓									✓
CO2	✓	✓	✓		✓							✓
CO3	✓		✓		✓							✓
CO4	✓	✓	✓		✓							✓
CO5	✓	✓	✓	✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Neural Networks)

Introduction to neural networks -Single layer perceptrons, Multi Layer Perceptrons (MLPs), Representation Power of MLPs, Activation functions - Sigmoid, Tanh, ReLU, Softmax. , Risk minimization, Loss function, Training MLPs with backpropagation, Practical issues in neural network training - The Problem of Overfitting, Vanishing and exploding gradient problems, Difficulties in convergence, Local and spurious Optima, Computational Challenges. Applications of neural networks.

Module-2 (Deep learning)

Introduction to deep learning, Deep feed forward network, Training deep models, Optimization techniques - Gradient Descent (GD), GD with momentum, Nesterov accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam. Regularization Techniques - L1 and L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout, Parameter initialization.

Module-3 (Convolutional Neural Network)

Convolutional Neural Networks – convolution operation, motivation, pooling, Convolution and Pooling as an infinitely strong prior, variants of convolution functions, structured outputs, data types, efficient convolution algorithms.

Module- 4 (Recurrent Neural Network)

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU.

Module-5 (Application Areas)

Applications – computer vision, speech recognition, natural language processing, common word embedding: continuous Bag-of-Words, Word2Vec, global vectors for word representation (GloVe). Research Areas – autoencoders, representation learning, boltzmann machines, deep belief networks.

Text Books

1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Neural Networks and Deep Learning, Aggarwal, Charu C.
3. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms (1st. ed.). Nikhil Buduma and Nicholas Locascio. 2017. O'Reilly Media, Inc.

Reference Books

1. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Michael Nielsen, Neural Networks and Deep Learning, 2018

Course Level Assessment Questions

Course Outcome1 (CO1):

1. Suppose you have a 3-dimensional input x = (x1, x2, x3) = (2, 2, 1) fully connected to 1 neuron which is in the hidden layer with activation function sigmoid. Calculate the output of the hidden layer neuron.

2. Design a single layer perceptron to compute the **NAND (not-AND)** function. This function receives two binary-valued inputs $\mathbf{x_1}$ and $\mathbf{x_2}$, and returns 0 if both inputs are 1, and returns 1 otherwise.

3. Suppose we have a fully connected, feed-forward network with no hidden layer, and 5 input units connected directly to 3 output units. Briefly explain why adding a hidden layer with 8 linear units does not make the network any more powerful.

4. Briefly explain one thing you would use a validation set for, and why you can't just do it using the test set.

5. Give a method to fight vanishing gradients in fully-connected neural networks. Assume we are using a network with Sigmoid activations trained using SGD.

6. You would like to train a fully-connected neural network with 5 hidden layers, each with 10 hidden units. The input is 20-dimensional and the output is a scalar. What is the total number of trainable parameters in your network?

Course Outcome 2(CO2):

1. Derive a mathematical expression to show L2 regularization as weight decay. Explain how L2 regularization improves the performance of deep feed forward neural networks.

2. In stochastic gradient descent, each pass over the dataset requires the same number of arithmetic operations, whether we use minibatches of size 1 or size 1000. Why can it nevertheless be more computationally efficient to use minibatches of size 1000?

3. State how to apply early stopping in the context of learning using Gradient Descent. Why is it necessary to use a validation set (instead of simply using the test set) when using early stopping?

4. Suppose that a model does well on the training set, but only achieves an accuracy of 85% on the validation set. You conclude that the model is overfitting, and plan to use L1 or L2 regularization to fix the issue. However, you learn that some of the examples in the data may be incorrectly labeled. Which form of regularisation would you prefer to use and why?

5. Describe one advantage of using Adam optimizer instead of basic gradient descent.

Course Outcome 3(CO3):

1. Draw and explain the architecture of convolutional neural networks.

2. Consider a convolution layer. The input consists of 6 feature maps of size 20×20. The output consists of 8 feature maps, and the filters are of size 5 x 5. The convolution is done with a stride of 2 and zero padding, so the output feature maps are of size 10 x 10.

   a. Determine the number of weights in this convolution layer.

   b. Determine the number of weights if we made this a fully connected layer, but the number of input and output units are kept the same as in the network.

3. Suppose two people A and B have implemented two neural networks for recognizing handwritten digits from 16 x 16 grayscale images. Each network has a single hidden layer, and makes predictions using a softmax output layer with 10 units, one for each digit class.

   a. A’s network is a convolutional net. The hidden layer consists of three 16 x 16 convolutional feature maps, each with filters of size 5 x 5, and uses the logistic nonlinearity. All of the hidden units are connected to all of the output units.

   b. B’s network is a fully connected network with no weight sharing. The hidden layer consists of 768 logistic units (the same number of units as in A’s convolutional layer).

4. Briefly explain one advantage of A’s approach and one advantage of B’s approach.

5. Why do the layers in a deep architecture need to be non-linear?

6. Give two benefits of using convolutional layers instead of fully connected ones for visual tasks.

7. You have an input volume of 32 x 32 x 3. What are the dimensions of the resulting volume after convolving a 5 x 5 kernel with zero padding, stride of 1, and 2 filters?

Course Outcome 4(CO4): .

1. Draw and explain the architecture of LSTM.
2. Name at least one benefit of the LSTM model over the bag-of-vectors model.
3. Give one advantage of GloVe over Skipgram/CBOW models.
4. What are two ways practitioners deal with having two different sets of word vectors **U** and **V** at the end of training both Glove and word2vec?
5. If we have a recurrent neural network (RNN), we can view it as a different type of network by "unrolling it through time". Briefly explain what that means.
6. Briefly explain how "unrolling through time" is related to "weight sharing" in convolutional networks.

Course Outcome 5(CO5):

1. Development a deep learning solution for problems in the domain i) natural language processing or ii) Computer vision
2. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.
3. Is an autoencoder for supervised learning or for unsupervised learning? Explain briefly.
4. Sketch the architecture of an autoencoder network.
5. Describe how to train an autoencoder network.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST414

Course Name: Deep Learning

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Discuss the limitation of a single layer perceptron with an example.

2. List the advantages and disadvantages of sigmoid and ReLU activation functions.

3. Derive weight updating rule in gradient descent when the error function is a) mean squared error b) cross entropy.

4. Discuss methods to prevent overfitting in neural networks.

5. What happens if the stride of the convolutional layer increases? What can be the maximum stride? Explain.

6. Draw the architecture of a simple CNN and write short notes on each block.

7. How does a recursive neural network work?

8. List down the differences between LSTM and RNN.

9. Illustrate the use of deep learning concepts in Speech Recognition.

10. What is an autoencoder? Give one application of an autoencoder

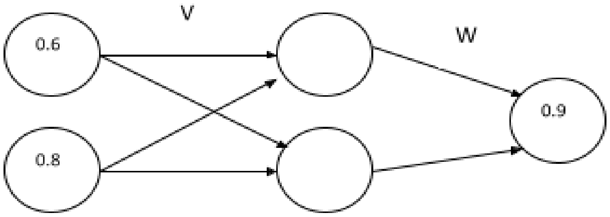
(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Update the parameters in the given MLP using gradient descent with learning rate as 0.5 and activation function as ReLU. Initial weights are given as **(10)**

$V =$ 0.1 0.2 0 .1 0.1 W= 0.1 0.1

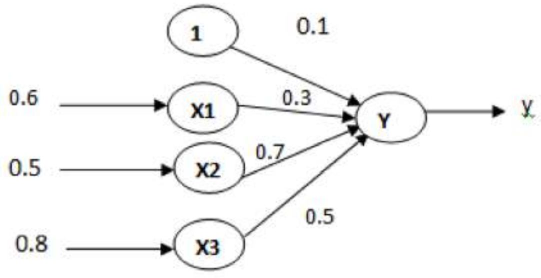


(b) Explain the importance of choosing the right step size in neural networks. **(4)**

OR

12. (a) Draw the architecture of a multi-layer perceptron. Derive update rules for parameters in the multi-layer neural network through the gradient descent **(10)**

(b) Calculate the output of the following neuron Y if the activation function is a **(4)**



binary sigmoid.

13. (a) Explain, what might happen in ADAGRAD, where momentum is expressed as $\Delta w_t = -\eta g_t / \sqrt{}(\sum_{\tau=1}^{t} \quad g_\tau^2$) where the denominator computes the L2 norm of all previous gradients on a per-dimension basis and □ is a global learning rate shared by all dimensions. **(6)**

(b) Differentiate gradient descent with and without momentum. Give equations for weight updation in GD with and without momentum. Illustrate plateaus, saddle points and slowly varying gradient. **(8)**

OR

14. (a) Suppose a supervised learning problem is given to model a deep feed forward neural network. Suggest solutions for the following a) small sized dataset for training b) dataset with unlabeled data c) large data set but data from different distribution. **(9)**

(b) Describe the effect in bias and variance when a neural network is modified with more number of hidden units followed with dropout regularization **(5)**

15. (a) Draw and explain the architecture of Convolutional Neural Networks **(8)**

(b) Suppose that a CNN was trained to classify images into different categories. It performed well on a validation set that was taken from the same source as the training set but not on a testing set, which comes from another distribution. What could be the problem with the training of such a CNN? How will you ascertain the problem? How can those problems be solved? **(6)**

OR

16. (a) What is the motivation behind convolution neural networks? **(4)**

(b) Discuss all the variants of the basic convolution function. **(10)**

17. (a) Describe how an LSTM takes care of the vanishing gradient problem. Use some hypothetical numbers for input and output signals to explain the concept. **(8)**

(b) Draw and explain the architecture of Recurrent Neural Networks **(6)**

OR

18. (a) Explain the application of LSTM in Natural Language Processing. **(8)**

(b) Discuss the architecture of GRU. **(6)**

19. (a) Explain any two word embedding techniques **(8)**

(b) Explain the merits and demerits of using Autoencoders in Computer Vision. **(6)**

OR

20. (a) Illustrate the use of representation learning in object classification. **(7)**

(b) Compare Boltzmann Machine with Deep Belief Network. (7)

Teaching Plan

No	**Contents**	**No. of Lecture Hours** **(36 hrs)**
	Module-1 (Neural Networks) (7 hours)	
1.1	Introduction to neural networks -Single layer perceptrons	1
1.2	Multi Layer Perceptrons (MLPs), Representation Power of MLPs	1
1.3	Activation functions - Sigmoid, Tanh, ReLU, Softmax. , Risk minimization, Loss function	1
1.4	Training MLPs with backpropagation	1
1.5	Illustration of back propagation algorithm	1
1.6	Practical issues in neural network training - The Problem of Overfitting, Vanishing and exploding gradient problems	1
1.7	Difficulties in convergence, Local and spurious Optima, Computational Challenges.	1
	Module-2 (Deep learning) (9 hours)	
2.1	Introduction to deep learning, Deep feed forward network	1
2.2	Training deep models, Concepts of Regularization and optimization,	1

2.3	Gradient Descent (GD), GD with momentum,	1
2.4	Nesterov accelerated GD, Stochastic GD,	1
2.5	AdaGrad, RMSProp, Adam,	1
2.6	L1 and L2 regularization, Early stopping, Dataset augmentation,	1
2.7	Parameter sharing and tying, Injecting noise at input, Ensemble methods	1
2.8	Parameter sharing and tying, Injecting noise at input, Ensemble methods	1
2.9	Dropout, Parameter initialization.	
Module-3 (Convolutional Neural Network) (6 hours)		
3.1	Convolutional Neural Networks – convolution operation	1
3.2	motivation, pooling	1
3.3	Convolution and Pooling as an infinitely strong prior	1
3.4	Variants of convolution functions	1
3.5	structured outputs, data types.	1
3.6	Efficient convolution algorithms.	1
Module- 4 (Recurrent Neural Network) (5 hours)		
4.1	Recurrent neural networks – Computational graphs, RNN design	1
4.2	Encoder – decoder sequence to sequence architectures	1
4.3	Deep recurrent networks, recursive neural networks	1
4.4	Modern RNNs LSTM	1
4.5	GRU	1
Module-5 (Application Areas)(9 hours)		
5.1	Computer vision. (TB1: Section 12.2)	1
5.2	Speech recognition. (TB1: Section 12.3)	1
5.3	Natural language processing. (TB1: Section 12.4)	1
5.4	Common Word Embedding - Continuous Bag-of-Words, Word2Vec (TB3: Section 2.6)	1

5.5	Common Word Embedding - Global Vectors for Word Representation(GloVe) (TB3: Section 2.9.1- Pennigton 2014)	1
5.6	Brief introduction on current research areas - Autoencoders, Representation learning. (TB3: Section 4.10)	1
5.7	Brief introduction on current research areas - representation learning. (TB3: Section 9.3)	1
5.8	Brief introduction on current research areas - Boltzmann Machines, Deep belief networks. (TB1: Section 20.1, TB3 Section 6.3)	1
5.9	Brief introduction on current research areas - Deep belief networks. (TB1: Section 20.3)	1

CST424	PROGRAMMING PARADIGMS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: The course provides the learners a clear understanding of the main constructs of contemporary programming languages and the various systems of ideas that have been used to guide the design of programming languages. This course covers the concepts of Names, Bindings & Scope, Statement-Level Control Structures, Sub Programs, Support for Object Oriented Programming, Exception Handling, Concurrency Control, Functional Programming and Logic Programming. This course helps the learners to equip with the knowledge necessary for the critical evaluation of existing and upcoming programming languages. It also enables the learner to choose the most appropriate language for a given programming task, apply that language's approach to structure or organize the code, classify programming languages based on their features and to design new generation languages.

Prerequisite: Sound knowledge in Programming in C and Object-Oriented Programming.

Mapping of course outcomes with program outcomes

CO1	Explain the criteria for evaluating programming languages and compare Imperative, Functional and Logic programming languages (**Cognitive Knowledge Level: Understand**)
CO2	Illustrate the characteristics of data types and variables (**Cognitive Knowledge Level: Apply**)
CO3	Comprehend how control flow structures and subprograms help in developing the structure of a program to solve a computational problem (**Cognitive Knowledge Level: Apply)**
CO4	Explain the characteristics of Object-Oriented Programming Languages (**Cognitive Knowledge Level: Understand)**
CO5	Compare concurrency constructs in different programming languages (**Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓										✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓										✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	40	40	40

Apply	**30**	**30**	**30**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the two completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed two modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Course Level Assessment Questions

Course Outcome1 (CO1):

1. Compare any three programming languages based on the language evaluation criteria. Prepare a list of characteristics that affect the language evaluation criteria.
2. Identify the advantages and disadvantages of imperative, functional and logic programming languages.

Course Outcome 2 (CO2):

1. Two most important design issues that are specific to character string types are
   (1) whether a string is simply a special kind of character array or a primitive type.
   (2) whether strings have static or dynamic length.
   Identify the implementations options for the above two cases.
2. Consider the following records of a particular language. Let the size of each char variable be 1 byte, int be 4 bytes and and Boolean be 1 bit.

```
Struct Student
{
        int id;
        char name[2];
        int age;
        boolean scholarship;
}
```

   Draw and comment on the possible memory layouts for the record for a 32-bit aligned machine

Course Outcome 3(CO3):

1. Explain three situations where a combined counting and logical looping statement is needed.
2. Describe the ways that aliases can occur with pass-by-reference parameters.
3. Identify the two fundamental design considerations for parameter-passing methods.
4. What will be the output of the given program segment if it uses the following parameter passing mechanisms:
   a) call by reference
   b) call by value

```
x : integer – – global
procedure foo(y : integer)
y := 3
print x
. . .
```

```
x := 2
foo(x)
print x
```

Course Outcome 4 (CO4):

1. Describe the role of a virtual method table in implementing dynamic method binding.
2. Identify the merits and demerits of inheritance.

Course Outcome 5 (CO5):

1. Evaluate the use of semaphores and monitors for providing competition synchronization and cooperation synchronization.

Syllabus

Module – 1

Introduction – Role of Programming Languages, Programming Domains, Language Evaluation Criteria, Influence on Language Design, Language Design Trade-offs, Implementation Methods. Names, Bindings & Scope – Names, Variables, Concept of Binding, Scope and Lifetime, Referencing Environments.

Module - 2

Data Types – Primitive Data Types, Character String Types, User-Defined Ordinal Types, Array Types, Record Types, List Types, Pointer & Reference Types, Type Checking, Strong Typing, Type Equivalence. Expressions – Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation. Assignment - Assignment Statements, Mixed-mode Assignment.

Module - 3

Statement-Level Control Structures – Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands. Subprograms – Design Issues of Subprograms, Local Referencing Environments, Parameter Passing Methods, Subprograms as Parameters, Overloaded Subprograms, Closures, Co-routines

Module - 4

Support for Object Oriented Programming – Inheritance, Dynamic Binding, Design Issues for Object Oriented Languages, Support for Object Oriented Programming in C++, Implementation of Object-oriented Constructs. Exception Handling – Basic Concepts, Design Issues.

Module - 5

Concurrency – Subprogram Level Concurrency, Semaphores, Monitors, Message Passing. Functional Programming Languages – Introduction to LISP and Scheme, Comparison of

Functional and Imperative Languages. Logic Programming Languages – Basic Elements of Prolog, Applications of Logic Programming.

Text Books

1. Robert W Sebesta, Concepts of Programming Languages, 10th Edition, Pearson.
2. Scott M L, Programming Language Pragmatics, 3rd Edition, Morgan Kauffman Publishers.

ReferenceBooks

1. Kenneth C. Louden, Programming Languages: Principles and Practice, 2nd Edition, Cengage Learning.
2. Tucker A. B. and R. E. Noonan, Programming Languages: Principles and Paradigms, 2nd Edition. –TMH.
3. Ravi Sethi, Programming Languages: Concepts & Constructs, 2nd Edition., Pearson Education.
4. David A. Watt, Programming Language Design Concepts, Wiley Dreamtech.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST424

Course Name: Programming Paradigms

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Differentiate between readability and writability.
2. Define binding and binding time.
3. What are the advantages of user-defined enumeration types?
4. Define narrowing and widening conversions.
5. Why for statement in C language is more flexible than that of older languages?

6. What are the advantages and disadvantages of dynamic local variables i subprograms?

7. Illustrate the concept of dynamic method binding with an example.

8. Is it mandatory to use constructors in object-oriented languages? Justify your answer.

9. What are the applications of logic programming languages?

10. Explain the working of let and let-rec constructs in Scheme.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain different criteria used for evaluating languages. **(7)**

(b) Consider the following pseudocode: **(7)**

```
x : integer := 3
y : integer := 4
procedure add
x := x + y
procedure second(P : procedure)
x : integer := 5
P()
procedure first
y : integer := 6
second(add)
first()

write integer(x)
```

(a) What does this program print if the language uses static scoping? Give reasons.

(b) What does it print if the language uses dynamic scoping? Give reasons.

OR

12. (a) With respect to storage binding, explain the meanings, purposes, advantages and disadvantages of four categories of scalar variables. **(7)**

(b) What is meant by referencing environment of a statement? Show the **(7)** referencing environment at the indicated program points (1), (2), (3) & (4) for the following program segment. Assume that the programming language is statically scoped.

```
program example;
        var a, b : integer;
        procedure sub1;
            var x, y: integer;
                begin { sub1 }
                 ……….                  (1)
                end { sub1 }
        procedure sub2;
            var x : integer;
            ……..
            procedure sub3;
                var x: integer;
                    begin { sub3 }
                        ……….       (2)
                    end { sub3 }
                begin { sub2 }
                    ……….            (3)
                end { sub2}
        begin {example}
                     ……….           (4)
        end {example }
```

13. (a) Explain any two issues associated with the pointer data types and also indicate how dangling pointer problem can be solved. **(7)**

(b) Describe the lazy and eager approaches for reclaiming garbage. **(7)**

OR

14. (a) What is meant by side effect and illustrate the advantages of referential transparency? **(8)**

(b) Explain the terms: compound assignment operator, coercion and short circuit evaluation. **(6)**

15. (a) Illustrate the different categories of iteration control statements. **(8)**

(b) Explain the techniques used for identifying the correct referencing environment for a subprogram that was sent as a parameter. **(6)**

OR

16. (a) Describe the implementation models of Parameter passing. **(10)**

(b) Differentiate coroutines from conventional subprograms. **(4)**

17. (a) What is meant by an exception handler? Explain how exceptions are handled in object-oriented languages. **(7)**

(b) Describe the design issues in object-oriented languages. **(7)**

OR

18. (a) Illustrate how a virtual method table can be used for implementing dynamic method binding. **(7)**

(b) Explain the different categories, merits and demerits of inheritance. **(7)**

19. (a) Compare functional and imperative programming languages. **(7)**

(b) Explain the role of monitors in concurrency. **(7)**

OR

20. (a) Explain the searching strategies used in Prolog. Why backward chaining is preferred over forward chaining in Prolog? **(10)**

(b) **(4)**

```
(let ((a 6)
(b 8)
(square (lambda (x) (* x x)))
(plus +))
(sqrt (plus (square a) (square b))))
```

Write the output of the above code? Explain how let and lambda construct works?

Teaching Plan

No	Contents	No. of Lecture Hours (36 hrs.)
Module-1 (7 hours)		
1.1	Introduction: Reasons for studying Concepts of programming languages, Programming Domains	1 hour
1.2	Language Evaluation Criteria	1 hour
1.3	Influence on Language Design, Language Design Trade-offs	1 hour
1.4	Implementation Methods	1 hour
1.5	Names, Variables	1 hour
1.6	Concept of Binding	1 hour
1.7	Scope and Lifetime, Referencing Environments	1 hour
Module-2 (7 hours)		
2.1	Primitive Data Types, Character String Types	1 hour
2.2	User-Defined Ordinal Types, Array Types	1 hour
2.3	Record Types, List Types, Pointer and Reference Types	1 hour
2.4	Implementation of pointer and reference types, Type Checking, Strong Typing, Type Equivalence	1 hour
2.5	Expressions and Assignment Statements, Arithmetic Expressions	1 hour
2.6	Overloaded Operators, Type Conversions	1 hour
2.7	Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed-mode Assignment	1 hour
Module-3 (8 hours)		
3.1	Selection Statements, Iterative Statements	1 hour
3.2	Unconditional Branching	1 hour

3.3	Guarded Commands	1 hour
3.4	Subprograms: Design Issues of Subprograms	1 hour
3.5	Local Referencing Environments	1 hour
3.6	Parameter Passing Methods	1 hour
3.7	Subprograms as Parameters, Overloaded Subprograms	1 hour
3.8	Closures, Co-routines	1 hour
Module-4 (7 hours)		
4.1	Inheritance	1 hour
4.2	Dynamic Binding	1 hour
4.3	Design Issues for Object Oriented Languages	1 hour
4.4	Support for Object Oriented Programming in C++	1 hour
4.5	Implementation of Object-Oriented Constructs	1 hour
4.6	Exception Handling – Basic Concepts	1 hour
4.7	Exception Handling - Design Issues	1 hour
Module-5 (7 hours)		
5.1	Subprogram Level Concurrency	1 hour
5.2	Semaphores, Monitors	1 hour
5.3	Message Passing	1 hour
5.4	Introduction to LISP and Scheme	1 hour
5.5	Comparison of Functional and Imperative Languages	1 hour
5.6	Basic Elements of Prolog	1 hour
5.7	Applications of Logic Programming	1 hour

CST434	NETWORK SECURITY PROTOCOLS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learners to explore various network and system security protocols. This course covers authentication protocols, firewalls and security protocols from different layers such as data link, network, transport and application. The concepts covered in this course enable the learners in effective use of security protocols for securing network applications.

Prerequisite: A fundamental knowledge in the concepts of Security in Computing.

Course Outcomes: After the completion of the course, the student will be able to

CO1	Explain authentication protocols, X.509 authentication service and Public Key Infrastructure (PKI).**(Cognitive Knowledge Level: Understand)**
CO2	Identify the security mechanisms in E mail security services. **(Cognitive Knowledge Level: Understand)**
CO3	Summarize the network and transport layer security services provided in a secure communication scenario. **(Cognitive Knowledge Level: Apply)**
CO4	Describe real time communication security and application layer security protocols. **(Cognitive Knowledge Level: Apply)**
CO5	Explain the concepts of firewalls and wireless network security. **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓		✓									✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓			✓						✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Test 1 (%)**	**Test 2 (%)**	**End Semester Examination (%)**
Remember	20	20	20
Understand	50	50	50
Apply	30	30	30
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment Test	: **25 marks**
Continuous Assessment Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Authentication Protocols)

Authentication Protocols – Mutual authentication, One way authentication. Kerberos – Kerberos Version 4, Kerberos Version 5. X.509 Authentication service. Public Key Infrastructure (PKI) – Trust models, Revocation.

Module-2 (E-mail Security)

Pretty Good Privacy (PGP) – Operational Description, Cryptographic keys and key rings, Message format, PGP message generation, PGP message reception, Public key management. S/MIME – Functionality, Messages, Certificate processing, Enhanced security services.

Module-3 (Network Layer Security and Web Security)

Internet Protocol Security (IPSec) – Overview, IP security architecture, Authentication Header (AH), Encapsulating Security Payload (ESP), Combining Security Associations, Key management. Internet Key Exchange (IKE) - Phases. Web Security – Web security considerations. Secure Socket Layer and Transport Layer Security (SSL/TLS) – SSL Architecture, SSL protocols, Cryptographic computations, Transport layer security.

Module-4 (Real-time Security and Application Layer Security)

Real-time communication security – Perfect Forward Secrecy (PFS), Denial-of-Service protection, Endpoint identifier hiding, Live partner reassurance. Hypertext Transfer Protocol Secure (HTTPS) – Connection initiation, Closure. Secure Shell (SSH) – Transport layer protocol, User authentication protocol, Connection protocol. Secure Electronic Transaction (SET) – Overview, Features, Participants, Dual signature, Payment processing.

Module-5 (System Security and Wireless Security)

Firewalls – Firewall characteristics, Types of Firewalls, Firewall configurations, Encrypted Tunnels, Trusted systems – Data access control, The concept of Trusted Systems, Trojan horse defense. IEEE 802.11i wireless LAN security - Services, Phases of operation, Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2.

Text Books

1. William Stallings, Cryptography and Network Security Principles and Practice, 4/e, Pearson Ed.
2. C. Kaufman, R. Perlman and M. Speciner, “Network Security: Private Communication in a Public World”, 2/e, PHI.

References

1. Behrouz A. Forouzan, DebdeepMukhopadhyay, “Cryptography and Network Security”, 3/e, Tata McGraw Hill.
2. Tyler Wrightson, “Wireless Network Security A Beginner’s Guide”, 2012, Tata McGraw Hill.
3. William Stallings, “Network Security Essentials: Applications and Standards”, 4/e, Prentice Hall.
4. Schiller J., Mobile Communications, 2/e, Pearson Education.
5. Roberta Bragg et. al., “Network Security: The Complete Reference”, Tata McGraw Hill

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Identify the threats associated with user authentication over a network or Internet.
2. In the context of Kerberos, mention the significance of a realm.

Course Outcome 2 (CO2):

1. Mention the use of R64 conversion for an e-mail application.
2. Give the general structure of Private and Public Key rings in PGP.

Course Outcome 3 (CO3):

1. In AH protocol, identify the fields in an IP header which are included in MAC calculation. For each of the fields in the IP header, indicate whether the field is immutable, mutable but predictable, or mutable. Justify your decision for each field.

2. Is it possible for the receiver to reorder SSL record blocks that arrive out of order? If so, explain how it can be done. If not, why?

Course Outcome 4 (CO4):

1. Devise a protocol based on a pre-shared secret key that hides identities and gives Perfect Forward Secrecy (PFS) for identity hiding. Make two variants, one in which an active attacker can learn only the initiator's identity, and one in which an active attacker can learn only the target's identity.
2. Explain the tasks performed by the payment gateway during Payment Authorization in SET.

Course Outcome 5 (CO5):

1. List the weaknesses of a packet-filtering router.
2. Give the relevance of pair wise keys and group keys in IEEE 802.11i.
3. State the design goals of firewalls.

Model Question Paper

QP CODE: **PAGES: ___**

Reg No:_______________
Name:__________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST434
Course Name: NETWORK SECURITY PROTOCOLS

Max Marks: 100 **Duration: 3 Hours**

PART A
(Answer All Questions. Each question carries 3 marks)

1. List any three requirements of Kerberos.

2. Specify the significance of key pair recovery. When is the key pair updated?

3. Why does PGP generate signature before applying compression?

4. List the four principal services provided by S/MIME.

5. Explain the significance of Alert protocol in SSL and list out any three Alert messages with their uses.

6. Specify the purpose of MAC during the change cipher spec TLS exchange.

7. What is the advantage, if any, of not including the MAC in the scope of packet encryption in SSH packets?

8. Givethe significance of dual signature in SET.

9. List the IEEE 802.11i services.

10. How is the concept of association related to that of mobility in wireless networks? **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Describe the requirements for a public-key certificate scheme. **(8)**

    (b) Explain the significance of chain of certificates. **(6)**

OR

12. (a) Specify the purpose of the X.509 standard. How is an X.509 certificate revoked? **(8)**

    (b) Describe the management functions of a PKI. What is a cross certificate? **(6)**

13. (a) List the services provided by PGP and explain how authentication and confidentiality are provided. **(8)**

    (b) Explain the functionalities provided by S/MIME. **(6)**

OR

14. (a) Give the format of a PGP message and specify the significance of each field in the message. **(8)**

    (b) Explain the enhanced security services provided in S/MIME. **(6)**

15. (a) Explain the parameters that identify an SSL session state. **(8)**

    (b) Differentiate between transport mode and tunnel mode in IPSec. **(6)**

OR

16. (a) The IPsec architecture document states that when two transport mode SAs arebundled to allow both AH and ESP protocols on the same end-to-end flow, only one ordering of security protocols seems appropriate: performing the ESP protocol before performing the AH protocol. Why is this approach **(8)**

recommended rather than authentication before encryption?

(b) List and explain the purpose each Alert Codes supported by SSL. **(6)**

17. (a) Illustrate the significance ofperfect forward secrecy. **(6)**

(b) Explain the key features provided by SET. **(8)**

OR

18. (a) List and explain the SSH protocols. **(8)**

(b) "The HTTPS capability is built into all modern web browsers". Justify. **(6)**

19. (a) Explain the phases of operations in IEEE 802.11i. **(8)**

(b) Give the significances of Encrypted Tunnels **(6)**

OR

20. (a) Compare the features of three types of firewalls. **(8)**

(b) Compare the Wireless LAN protocols WEP, WPA and WPA2 **(6)**

TEACHING PLAN

No	Contents	No.of Lecture Hours (35 Hrs)
Module-1 (Authentication Protocols)(7hrs)		
1.1	Authentication Protocols – Mutual authentication, One way authentication	1
1.2	Kerberos –Version 4	1
1.4	Differences between Kerberos Version 4 and Version 5, Kerberos Version 5	1
1.5	X.509 Authentication service – Certificates, Authentication Procedures, X.509 Version 3	1
1.6	Public Key Infrastructure (PKI) – Trust models	1
1.7	Public Key Infrastructure (PKI) – Revocation	1

Module-2 (E-mail Security) (6 hrs)		
2.1	Pretty Good Privacy (PGP) – Operational Description	1
2.2	Cryptographic keys and key rings, Message format	1
2.3	PGP message generation, PGP message reception	1
2.4	PGP -Public key management	1
2.5	S/MIME – Overview of MIME, Functionality, Messages	1
2.6	S/MIME - Certificate processing, Enhanced security services	1
Module-3 (Network Layer Security and Web Security)(8 hrs)		
3.1	Internet Protocol Security (IPSec) – Overview, IP security architecture	1
3.2	Authentication Header (AH)	1
3.3	Encapsulating Security Payload (ESP)	1
3.4	Combining Security Associations, Key management	1
3.5	Internet Key Exchange (IKE) – Phases	1
3.6	Web Security – Web security considerations. Secure Socket Layer and Transport Layer Security (SSL/TLS) – SSL Architecture	1
3.7	SSL Protocols - Record Protocol, Change Cipher Spec Protocol, Alert Protocol	1
3.8	SSL Handshake Protocol, Cryptographic computations, Transport Layer Security	1
Module-4 (Real-time Security and Application Layer Security) (8hrs)		
4.1	Real-time communication security – Perfect Forward Secrecy (PFS)	1
4.2	Denial-of-Service protection, Endpoint identifier hiding, Live partner reassurance	1
4.3	Hypertext Transfer Protocol Secure (HTTPS) – Connection initiation, Closure	1
4.4	Secure Shell (SSH) – Transport layer protocol	1
4.5	User authentication protocol	1

4.6	Connection protocol	1
4.7	Secure Electronic Transaction (SET) – Overview, Features, Participants	1
4.8	Dual signature, Payment processing	1
Module-5 (System Security and Wireless Security) (6 hrs)		
5.1	Firewalls – Firewall characteristics, Types of Firewalls	1
5.2	Firewalls – Firewall configurations, Encrypted Tunnels	1
5.3	Trusted systems – Data Access Control, The Concept of Trusted Systems, Trojan Horse Defense	1
5.4	IEEE 802.11i wireless LAN security - Services, Phases of operation	1
5.5	Wired Equivalent Privacy (WEP)	1
5.6	Wi-Fi Protected Access (WPA), WPA2	1

CST444	SOFT COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course enables the learners to understand the concepts of Soft Computing techniques and its applications. It covers Artificial Neural Networks, operations and models of fuzzy logic, genetic algorithms and multi objective optimization techniques. This course helps the students to develop algorithms and solutions for different real world applications.

Prerequisite: NIL.

Mapping of course outcomes with program outcomes

CO1	Describe soft computing techniques and the basic models of Artificial Neural Network **(Cognitive Knowledge Level: Understand)**
CO2	Solve practical problems using neural networks **(Cognitive Knowledge Level: Apply)**
CO3	Illustrate the operations, model and applications of fuzzy logic **(Cognitive Knowledge Level: Apply)**
CO4	Illustrate the concepts of Genetic Algorithm **(Cognitive Knowledge Level: Apply)**
CO5	Describe the concepts of multi-objective optimization models and the need for using hybrid soft computing approaches**(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	**30**	**30**	**30**
Understand	**30**	**30**	**30**
Apply	**40**	**40**	**40**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

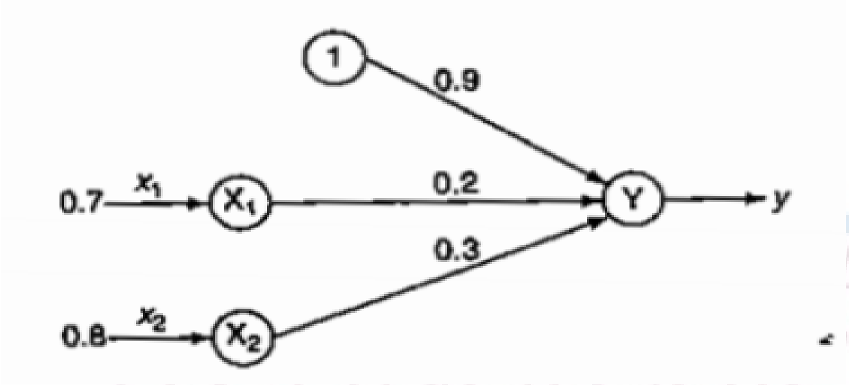
End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Describe the necessity of Activation function? Examine the various aspects of sigmoidal activation function. List the drawbacks. Calculate the net output of the following neural network using the bipolar and binary sigmoidal activation function.



2. Explain the architecture of McCulloch-Pitts Neuron network model. Implement NAND(NOT-AND) gate function using M-P Neuron Model(with binary input).

Course Outcome 2(CO2):

1. Find the weights required to perform classification of patterns shown below using perceptron network. The patterns (1,1,-1) and (1, -1,-1) are belonging to the target class -1. The patterns (-1,1,1) and (-1,-1,1) are belonging to the target class +1. Assume suitable learning rate and initial weights.
2. Explain the architecture and training algorithm of Adaline network . Use Adaline nerwork to train NOR logic function with bipolar inputs and targets. Perform 2 epochs of training.

Course Outcome 3(CO3):

1. There is an imprecise relationship between the ambient temperature for clay masonry bricks and their compressive strengths. Let X be a fuzzy set of fracture strengths and Y be a fuzzy set of temperatures with the following membership functions:

$$X=\left\{\frac{1.0}{1500}+\frac{0.8}{2175}+\frac{0.6}{7000}+\frac{0.5}{12750}+\frac{0.3}{16500}+\frac{0.1}{20000}\right\}$$

$$Y=\left\{\frac{0.2}{20}+\frac{0.4}{25}+\frac{0.5}{32}+\frac{1.0}{50}+\frac{0.6}{90}+\frac{0.3}{105}\right\}$$

(a) Find the Cartesian Product of X and Y and represent it as relation R.
Suppose there is a second fuzzy set of masonry lengths given as

$$Z=\left\{\frac{0.4}{1500}+\frac{0.5}{2175}+\frac{0.6}{7000}+\frac{0.8}{12750}+\frac{0.9}{16500}+\frac{1.0}{20000}\right\}$$

(b) Find S=ZoR using max-min composition (c) Find T=ZoR using max-product composition

2. Given two universes X={x1,x2,x3,x4,x5} and Y={y1,y2,y3,y4,y5},the fuzzy sets A defined on X and fuzzy set B defined on Y are given below:

   $A = \left\{\frac{0.4}{x1} + \frac{0.7}{x2} + \frac{1}{x3} + \frac{0.8}{x4} + \frac{0.6}{x5}\right\} B = \left\{\frac{0.2}{y1} + \frac{0.6}{y2} + \frac{1}{y3} + \frac{0.9}{y4} + \frac{0.7}{y5}\right\}$

   (i)Find the relation R = A × B

   Consider another fuzzy set C defined on the universe V={v1,v2,v3}, $C = \left\{\frac{0.4}{v1} + \frac{1}{v2} + \frac{0.8}{v3}\right\}$

   (ii) Find P = B × C. Using max-min composition, Find RoP.

Course Outcome 4(CO4):

1. Illustrate the various types of cross over with suitable examples.
2. Using Genetic algorithm with Roulette wheel selection method maximize the function f(x)=x2 over {0, 1, 2, ..., 31} with initial x values of (13, 24, 8, 19). Show one crossover and mutation.

Course Outcome 5(CO5):

1. Explain strong dominance and weak pareto-optimality.
2. What are the different classifications of neuro-fuzzy hybrid systems?

Syllabus

Module – 1 (Introduction to Soft Computing & Artificial Neural Network)

Introduction to Soft Computing. Difference between Hard Computing & Soft Computing. Applications of Soft Computing. Artificial Neurons Vs Biological Neurons. Basic models of artificial neural networks – Connections, Learning, Activation Functions. McCulloch and Pitts Neuron. Hebb network.

Module – 2 (Supervised Learning Network)

Perceptron Networks– Learning rule, Training and testing algorithm. Adaptive Linear Neuron– Architecture, Training and testing algorithm. Back propagation Network – Architecture, Training and testing algorithm.

Module - 3 (Fuzzy Logic & Defuzzification)

Fuzzy sets – properties, operations on fuzzy set. Fuzzy membership functions, Methods of membership value assignments – intuition, inference, Rank Ordering. Fuzzy relations– operations on fuzzy relation. Fuzzy Propositions. Fuzzy implications. Defuzzification– Lamda cuts, Defuzzification methods.

Module - 4 (Fuzzy Inference System & Genetic Algorithm)

Fuzzy Inference Systems - Mamdani and Sugeno types. Fuzzy Logic Controller. Concepts of genetic algorithm. Operators in genetic algorithm - coding, selection, cross over, mutation. Stopping condition for genetic algorithm.

Module - 5 (Multi Objective Optimization & Hybrid Systems)

Multi objective optimization problem. Principles of Multi- objective optimization, Dominance and pareto-optimality. Optimality conditions. Neuro-fuzzy hybrid systems. Genetic – neuro hybrid systems.

Text Books

1. S.N.Sivanandam and S.N. Deepa, Principles of Soft Computing , 2ndEdition, John Wiley & Sons.
2. Kalyanmoy Deb, Multi-objective Optimization using Evolutionary Algorithms, 1st Edition, John Wiley & Sons.

ReferenceBooks

1. Timothy J Ross, Fuzzy Logic with Engineering Applications, John Wiley & Sons, 2016.
2. T.S.Rajasekaran, G.A.Vijaylakshmi Pai “Neural Networks, Fuzzy Logic & Genetic Algorithms Synthesis and Applications”, Prentice-Hall India.
3. Simon Haykin, “Neural Networks- A Comprehensive Foundation”, 2/e, Pearson Education.
4. Zimmermann H. J, “Fuzzy Set Theory & Its Applications”, Allied Publishers Ltd.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 444

Course Name: Soft Computing

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Explain the architecture of a simple Artificial Neural network? Compare it with a biological neuron.

2. A 4-input neuron has weights 1, 2, 3 and 4. The transfer function is linear with the constant of proportionality being equal to 2. The inputs are 4, 10, 5 and 20 respectively. Predict the output?

3. Explain the Widrow-Hoff learning rule for supervised learning in neural networks with help of an example. Why is it sometimes called the LMS learning rule?

4. Implement one epoch of Adaline algorithm for AND logic function with binary inputs and bipolar outputs. Initial weights are w1=0.2, w2=0.1 and learning rate parameter η=0.2.

5. Consider two fuzzy sets $A = \left\{\frac{0.2}{0} + \frac{0.3}{1} + \frac{1}{2} + \frac{0.1}{3} + \frac{0.5}{4}\right\} B = \left\{\frac{0.1}{0} + \frac{0.25}{1} + \frac{0.9}{2} + \frac{0.7}{3} + \frac{0.3}{4}\right\}$ Find the following: (a) Algebraic sum (b) Algebraic product(c) Bounded sum.

6. Using your own intuition and definition of universe of discourse, plot membership

functions for liquid level (Empty, very less, less, full, very full) in a tank.

7. Explain Stochastic Universal Sampling with an example.

8. Explain any two mutation methods.

9. Differentiate between linear and nonlinear Multi Objective Optimization Problem.

10. What are the characteristics of neuro fuzzy hybrid systems?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Implement XOR function using M-P Neuron Model (with binary input).Why M-P neuron is widely used in processing binary data? **(8)**

(b) Using Hebb Network calculate the weight required to perform the following classification of given input pattern. **(6)**

L □belongs to the members of the class(+)□target value +1

U□does not belongs to members of class(.)□target value -1

+	.	.
+	.	.
+	+	+

+	.	+
+	.	+
+	+	+

L U

OR

12. (a) Compare the three learning approaches in Artificial Neural Network. How is the critic information used in learning process. **(8)**

(b) Define Hebb Law. Design a Hebb Network to implement logical AND function. Use bipolar input and targets. **(7)**

13. (a) Discuss the training algorithm and explain the weight updates in back propagation networks. **(10)**

(b) Implement one epoch of Perceptron training algorithm for OR logic function with binary input and bipolar output. **(4)**

OR

14. (a) Explain how synaptic weights are adapted iteration by iteration using error correction rule in Perceptron convergence algorithm for an OR gate with bipolar inputs and outputs. Initial weights are all zero and learning rate parameter η=0.1. **(10)**

(b) Explain Perceptron convergence theorem and discuss Perceptron algorithm based on XOR logic function. **(4)**

15. (a) Three fuzzy sets are defined as follows: **(10)**

$A = \left\{\frac{0.1}{30} + \frac{0.2}{60} + \frac{0.3}{90} + \frac{0.4}{120}\right\}, B = \left\{\frac{1}{1} + \frac{0.2}{2} + \frac{0.5}{3} + \frac{0.7}{4} + \frac{0.3}{5} + \frac{0}{6}\right\},$

$$C = \left\{\frac{0.33}{100} + \frac{0.65}{200} + \frac{0.92}{300} + \frac{0.21}{400}\right\}$$

Find: (i)$R = A \times B$ (ii)$S = B \times C$ (iii)$T = RoS$, using Max-Min composition (iv)$T = RoS$, using Max-Product composition.

(b) For the fuzzy sets given $A = \left\{\frac{0.5}{x_1} + \frac{0.2}{x_2} + \frac{0.9}{x_3}\right\}$ and$B = \left\{\frac{1}{y_1} + \frac{0.5}{y_2} + \frac{1}{y_3}\right\}$. Find relation R by performing Cartesian product over the given fuzzy sets. **(4)**

OR

16. (a) Using inference approach, find the membership values for each of the triangular shapes (I, R, IR, T) for a triangle with angles 120°, 50°, 10°. **(8)**

(b) Using Zadeh's notation, determine the ʎ - cut sets for the given fuzzy sets: **(6)**

$$S_1 = \left\{\frac{0}{0} + \frac{0.5}{20} + \frac{0.65}{40} + \frac{0.85}{60} + \frac{1.0}{80} + \frac{1.0}{100}\right\}$$

$$S_2 = \left\{\frac{0}{0} + \frac{0.45}{20} + \frac{0.6}{40} + \frac{0.8}{60} + \frac{0.95}{80} + \frac{1.0}{100}\right\}$$

Express the following for ʎ = 0.5: a) $S_1 \cup S_2$ b) $\acute{S}_2$ c) $S_1 \acute{\cap} S_2$

17. (a) Differentiate between value encoding and permutation encoding. **(8)**

(b) Explain the stopping conditions for genetic algorithm. **(6)**

OR

18. (a) Apply Mamdani fuzzy model to design a controller to determine the wash time of a domestic washing machine. Assume input is dirt and grease of the cloth. Use three descriptors for input variable and five descriptors for output variables .Derive the set of rules for controller action and defuzzification. Design should be supported by figure wherever possible. **(10)**

(b) Explain Single-Point Crossover and Two-Point Crossover with example. **(4)**

19. (a) Explain convex and non convex MOOP? How to find a non dominated set. **(10)**

(b) What are the properties of dominance relation? **(4)**

OR

20. (a) Explain Genetic Neuro-Hybrid System with block diagram. Also write the advantages of Genetic- Neuro Hybrid systems. **(8)**

(b) Discuss the classification of Neuro-Fuzzy Hybrid System. **(6)**

Teaching Plan

No	Contents	No. of Lecture Hours (35 hrs)
Module-1 (Introduction to Soft Computing & Artificial Neural Network) (6 hours)		
1.1	Introduction to Soft Computing	1 hour
1.2	Difference between Hard Computing & Soft Computing & Applications of Soft Computing	1 hour
1.3	Artificial Neurons Vs Biological Neurons, Basic models of artificial neural networks	1 hour
1.4	Activation Functions	1 hour
1.5	McCulloch and Pitts Neuron	1 hour
1.6	Hebb network	1 hour
Module-2 (Supervised Learning Network) (7 hours)		
2.1	Perceptron networks – Learning rule, Training and testing algorithm	1 hour
2.2	Perceptron networks – Problems	1 hour
2.3	Adaptive Linear Neuron (Lecture I)	1 hour
2.4	Adaptive Linear Neuron (Lecture II)	1 hour
2.5	Adaptive Linear Neuron-Problems (Lecture III)	1 hour
2.6	Back propagation Network (Lecture I)	1 hour
2.7	Back propagation Network (Lecture II)	1 hour
Module-3 (Fuzzy Logic & Defuzzification) (8 hours)		
3.1	Introduction to Fuzzy Set, Properties & operations on fuzzy sets	1 hour
3.2	Fuzzy membership functions, Fuzzification	1 hour
3.3	Methods of membership value assignments	1 hour
3.4	Fuzzy relations, Operations on Fuzzy Relation	1 hour

3.5	Fuzzy Propositions & Fuzzy Implications	1 hour
3.6	Lamda cuts for fuzzy sets	1 hour
3.7	Defuzzification methods(Lecture I)	1 hour
3.8	Defuzzification methods(Lecture II)	1 hour
Module-4 (Fuzzy Inference System & Genetic Algorithm) (6 hours)		
4.1	Fuzzy Inference Systems - Mamdani type	1 hour
4.2	Fuzzy Inference Systems - Sugeno type	1 hour
4.3	Fuzzy Logic Controller	1 hour
4.4	Introduction to genetic algorithm, operators in genetic algorithm - coding	1 hour
4.5	Selection, Cross over	1 hour
4.6	Mutation, stopping condition for genetic algorithm	1 hour
Module-5 (Multi-Objective Optimization & Hybrid System) (8 hours)		
5.1	MOOP-Linear &Non linear, Convex & Non Convex	1 hour
5.2	Principles of MOO-Illustrating Pareto Optimal Solutions, Objectives in MOO	1 hour
5.3	Dominance & Pareto-Optimality-Concept of Domination	1 hour
5.4	Properties of Dominance Relation, Pareto Optimality	1 hour
5.5	Procedure for finding a non dominated set	1 hour
5.6	Optimality Conditions	1 hour
5.7	Neuro Fuzzy hybrid system-Classification& characteristics	1 hour
5.8	Genetic –neuro hybrid systems	1 hour

CST454	FUZZY SET THEORY AND APPLICATIONS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course equips the students to understand the concepts of fuzziness and its use in building better solutions to problems. The course covers basic concepts of fuzzy sets, fuzzy relations, fuzzy logic and building of fuzzy approximation-based solutions. It helps students to design and develop fuzzy based solutions to real world applications.

Prerequisite: Basic knowledge in set theory.

Course Outcomes: After the completion of the course, the student will be able to

CO1	Explain fuzzy logic based problem solving **(Cognitive Knowledge Level: Understand)**
CO2	Summarize the concepts of crisp sets, crisp relations, crisp logic with fuzzy sets, fuzzy relations and fuzzy logic**(Cognitive Knowledge Level: Apply)**
CO3	Develop fuzzy systems by selecting appropriate membership functions, fuzzification and defuzzification methods **(Cognitive Knowledge Level: Apply)**
CO4	Develop solutions using graphical and rule-based methods**(Cognitive Knowledge Level: Apply)**
CO5	Make use of fuzzy logic inference to solve real world problems**(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓							✓

CO5	✓	✓		✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**20**	**20**	**20**
Understand	**50**	**50**	**50**
Apply	**30**	**30**	**30**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests1&2)	**25 marks**
Continuous AssessmentAssignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Basic Fuzzy Set Theory)

The case for imprecision, Utility and Limitations of Fuzzy Systems, Fuzzy Sets and Membership, Classical Sets – Properties, Operations, Fuzzy Sets – Properties and Operations, Classical Relations – Cartesian Product, Operations and Properties of Crisp Relations, Composition, Fuzzy Relations – Cardinality, Operations, Properties, Fuzzy Cartesian Product and Composition.

Module – 2 (Fuzzy Membership Functions)

Tolerance and Equivalence Relations – Crisp and Fuzzy, Similarity Methods – Cosine, Min-max, Fuzzy Membership Functions – Features, Fuzzification, Defuzzification to Crisp Sets, λ-Cutsfor Fuzzy Relations, Linguistic Hedges.

Module - 3 (Fuzzification and Defuzzification Methods)

Development of Membership Functions –Intuition, Inference, Rank ordering, Inductive reasoning. Defuzzification to Scalars - Max membership principle, Centroid method, Weighted average method, Mean max membership, Center of sums, Center of largest area, First (or last) of maxima.

Module - 4 (Fuzzy Inference)

Classical Logic, Fuzzy Logic, Approximate Reasoning, Fuzzy (Rule-Based) Systems - Multiple conjunctive antecedents, Multiple disjunctive antecedents, Aggregation of fuzzy rules, Graphical Techniques of Inference.

Module - 5 (Fuzzy Applications)

Applications of Fuzzy Systems - Fuzzy Classification, Fuzzy Pattern Recognition, Fuzzy Control Systems, Fuzzy Systems and Neural Networks, Fuzzy Clustering, Fuzzy Databases and Information retrieval systems.

Text Books

1. Fuzzy Logic with Engineering Applications – Timothy J. Ross, Third Edition, John Wiley and Sons, 2010
2. Fuzzy Sets and Fuzzy Logic: Theory and Applications - George J. Klir and Bo Yuan , Prentice Hall, 1995.

Reference Books

1. Kenneth H. Rosen, Discrete Mathematics and Its Applications with Combinatorics and GraphTheory, Seventh Edition, MGH,2011
2. Trembly J.P and Manohar R, "Discrete Mathematical Structures with Applications to Computer Science", TataMc Graw Hill Pub. Co. Ltd., New Delhi,2003.
3. Bernard Kolman, Robert C. Busby, Sharan Cutler Ross, "Discrete Mathematical Structures", Pearson Education Pvt Ltd., New Delhi,2003
4. Kenneth H .Rosen, "Discrete Mathematics and its Applications", 5/e, TataMc Graw Hill Pub. Co. Ltd, New Delhi2003

Course Level Assessment Questions

Course Outcome1 (CO1):

1. What are the limitations of crisp systems?
2. Explain the difference between randomness and fuzziness.
3. Find some examples of prospective fuzzy variables in daily life.

Course Outcome 2(CO2):

1. The strength of two types of concrete needs to be compared. Four concrete masonry units (CMUs) from each type of concrete are stressed until they fail. The lowest stress at failure of a CMU is denoted 1, and the highest stress at failure is denoted 4,so the CMUs are rank ordered by failure stress, that is, X = {1, 2, 3, 4}. Since "failure" of CMUs is fuzzy, the membership value for a specific CMU represents the judgment that the CMU really failed. The following fuzzy sets represent the failure estimates for the two different concrete types:

$$A = \left\{\frac{0\cdot 15}{1} + \frac{0.25}{2} + \frac{0\cdot 6}{3} + \frac{0.9}{4}\right\}$$

$$B = \left\{\frac{0.2}{1} + \frac{0.3}{2} = + \frac{0.5}{3} + \frac{0.8}{4}\right\}$$

Calculate the union, intersection and difference for the two concrete types.

2. An engineer is testing the properties, strength and weight of steel. Suppose he has two fuzzy sets A , defined on a universe of three discrete strengths, $\{s1, s2, s3\}$, and B, defined on a universe of three discrete weights, $\{w1, w2, w3\}$. Suppose A and B represent a “high-strength steel” and a “near-optimum weight,” respectively, as shown below

$$A = \left\{\frac{1}{s_1} + \frac{0.5}{s_2} + \frac{0.2}{s_3}\right\}$$

$$B = \left\{\frac{1}{w_1} + \frac{0.5}{w_2} + \frac{0.2}{w_3}\right\}$$

a) Find the fuzzy relation for the Cartesian product, R, of A and B

b) Introducing another fuzzy set, C, which represents a set of “moderately good” steel strengths

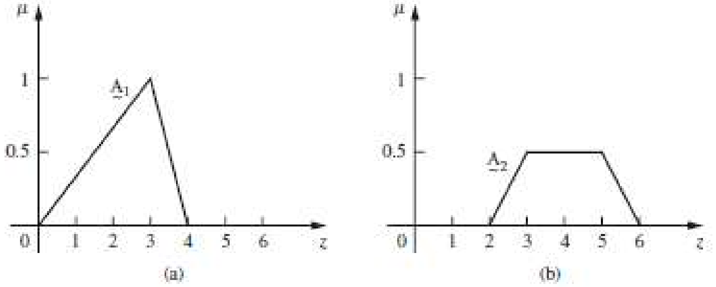
$$C = \left\{\frac{0.1}{s_1} + \frac{0.6}{s_2} + \frac{1}{s_3}\right\}$$

Find C◦R using max–min composition

Course Outcome 3(CO3):

1. Using your own intuition and your own definitions of the universe of discourse,plot fuzzy membership functions for “age of people” who are:
   (i) very young
   (ii) young
   (iii) middle-aged
   (iv) old
2. a) Define membership functions for approximately isosceles triangle, approximately equilateral and approximately right-angled triangles.

   b) Find the membership value for the triangle represented by the angles 80°, 75°, 25°, in the above triangles.
3. In metallurgy, materials are made with mixtures of various metals and other elements to achieve certain desirable properties. In a particular preparation of steel, three elements, namely, iron, manganese, and carbon, are mixed in two different proportions. The samples obtained from these two different proportions are placed on a normalized scale and are represented as fuzzy sets A1 and A2. Do a logical union of the membership functions A1 and A2 and find the defuzzified value of the resulting membership function.



(a) (b)

Course Outcome 4(CO4): .

1. Consider the following two discrete fuzzy sets, which are defined on universeX = {−5, 5}:

$$A = \text{"}zero\text{"} = \left\{\frac{0}{-2} + \frac{0.5}{-1} + \frac{1}{0} + \frac{0.5}{1} + \frac{0}{2}\right\}$$

$$B = \text{"}positive\ medium\text{"} = \left\{\frac{0}{0} + \frac{0.6}{1} + \frac{1}{2} + \frac{0.6}{3} + \frac{0}{4}\right\}$$

   Construct the relation for IF *x* is “zero” THEN *y* is “positive medium”

2. A metro train system uses fuzzy logic in ensuring smooth ride on the train. The metro train system has fixed stops and the distance between the stops are known. The system uses fuzzy logic in deciding the pressure applied on the brakes. The amount of pressure applied depends on the distance to the next stop and the speed of the train. Design appropriate membership functions for the input and illustrate the use of Mamdani Inference in arriving at the brake pressure.

Course Outcome 5(CO5):

1. A fuzzy systems needs to be designed to provide a rating for a web store as “excellent”, “good” or “poor”. The web store can be rated based on the products available, the customer service and the discount provided. Design appropriate membership functions and fuzzy rules for generating the fuzzy based rating system.

2. Design a fuzzy control system for an air-conditioning application. Make appropriate decisions regarding inputs and outputs.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST454

Course Name: Fuzzy Set Theory and Applications

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Illustrate where a fuzzy logic based application is suitable.

2. Consider a LAN using Ethernet protocol with maximum bandwidth of 10 Mbps. Traffic rates can be represented using two fuzzy variables, Quiet and Congested. If the universal set X = {0,1,2,3,4,5,6,7,8,9,10} represents bandwidth usage in Mbps, then draw possible membership functions for the fuzzy variables.

3. Define fuzzy tolerance and equivalence relations.

4. Given two data points, illustrate how a similarity measure between them can be computed.

5. Define a convex normalized fuzzy set.

6. How does augmented query help in information retrieval.

7. Given the propositions

   (i) $C \vee D$

   (ii) $\sim H \Rightarrow (A \wedge \sim B)$

   (iii) $(C \vee D) \Rightarrow \sim H$

(iv) $(A \wedge \sim B) \Rightarrow (R \vee S)$

Infer $(R \vee S)$ from the above propositions and state the tautologies used.

8. Write a predicate logic statement for “Ram likes all kinds of food”.

9. Given the relation R below, find λ-cut for the relation using suitable λ value.

$$\begin{bmatrix} 1 & 0.8 & 0 & 0.1 & 0.2 \\ 0.8 & 1 & 0.4 & 0 & 0.9 \\ 0 & 0.4 & 1 & 0 & 0 \\ 0.1 & 0 & 0 & 1 & 0.5 \\ 0.2 & 0.9 & 0 & 0.5 & 1 \end{bmatrix}$$

10. Define maximum approaching degree.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) An engineer is testing the properties, strength and weight of steel. Suppose he has two fuzzy sets A, defined on the universe of three discrete strengths { s1, s2, s3 } and B, defined on the universe of discrete weights { w1, w2, w3}. Suppose A represents a “high-strength steel” and B a “near-optimum weight”. **(4)**

$$A = \left\{\frac{1}{s1} + \frac{0.5}{s2} + \frac{0.2}{s3}\right\},\ B = \left\{\frac{1}{w1} + \frac{0.5}{w2} + \frac{0.3}{w3}\right\}$$

Find fuzzy Cartesian product, R, of A and B.

(b) Let a fuzzy set C= $\left\{\frac{0.1}{s1} + \frac{0.6}{s2} + \frac{1}{s3}\right\}$ be introduced, which represents a set of “moderately good” steel strength. Find the max-min composition of C and R. **(5)**

(c) Define 5 operations associated with crisp relations. **(5)**

OR

12. (a) How is excluded middle axiom different for crisp and fuzzy sets? **(4)**

(b) Differentiate between crisp and fuzzy sets with respect to their membership functions. **(4)**

(c) Illustrate any 4 operations associated with a fuzzy relation. **(6)**

13. (a) A structural designer is considering four different kinds of structural beams { S1, S2, S3, S4} for a new building. Laboratory experiments on the deflection resistance for these four kinds of beams have been performed, and the engineer wants to determine their suitability in the new structure. The following data have been observed based on the overall deflection capacity of each beam type: **(10)**

		S1	**S2**	**S3**	**S4**
No deflection	X_1	0.3	0.6	0.5	0.8
Some deflection	X_2	0.6	0.3	0.5	0.2
Excessive deflection	X_3	0.1	0.1	0	0

Use cosine amplitude method to determine the similarity of the four beam types.

(b) Given a fuzzy set "tall" = $\left\{\frac{0.1}{s1} + \frac{0.6}{s2} + \frac{1}{s3}\right\}$, illustrate how the fuzzy set "very tall" be defined? **(4)**

OR

14. (a) Define tolerance and equivalence relations. Check whether the relation R given below is tolerance or equivalence relation. **(4)**

$$\begin{bmatrix} 1 & 0.8 & 0 & 0.1 & 0.2 \\ 0.8 & 1 & 0.4 & 0 & 0.9 \\ 0 & 0.4 & 1 & 0 & 0 \\ 0.1 & 0 & 0 & 1 & 0.5 \\ 0.2 & 0.9 & 0 & 0.5 & 1 \end{bmatrix}$$

(b) Given the following data regarding three cities and the quality of their bridges, find the similarity between the cities using max-min method. **(10)**

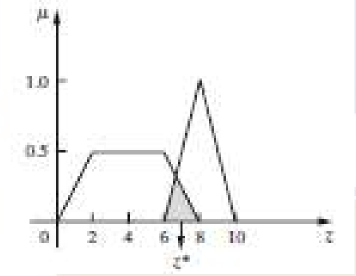
		C1	**C2**	**C3**
Poor	Q_1	0.00	0.10	0.10
Fair	Q_2	0.04	0.04	0.08
Good	Q_3	0.02	0.04	0.06

15. (a) Explain the process of developing membership functions using the inference method. **(6)**

(b) The following raw data were determined in a pair wise comparison of new premium car preferences in a poll of 100 people. When it was compared with a Porsche (P), 79 of those polled preferred a BMW (B), 85 preferred a Mercedes (M), 59 preferred a Lexus (L), and 67 preferred an Infinity (I). When a BMW was compared, the preferences were 21 – P, 23 – M, 37 – L, and 45 – I. When a Mercedes was compared, the preferences were 15 – P, 77 – B, 35 – L, and 48 – I. When a Lexus was compared, the preferences were 41 – P, 63 – B, 65 – M, and 51 – I. Finally, when an Infinity was compared, the preferences were 33 – P, 55 – B, 52 – M, and 49 – L. Using rank ordering, plot the membership function for "most preferred car." **(8)**

OR

16. (a) **1.** Defuzzify the following region using centroid method. **(9)**



(b) 2. Defuzzify the region given in 16(a) using weighted average method. **(5)**

17. (a) For a distillation process, the objective is to separate components of a mixture in the input stream. The relationship between the input variable, temperature, and the output variable, distillate fractions, is not precise but the human operator of this process has developed an intuitive understanding of this relationship. The universe for each of these variables is **(8)**

X = universe of temperatures (degree fahrenheit) = {160, 165, 170, 175, 180, 185, 190, 195}.
Y = universe of distillate fractions (percentage) = {77, 80, 83, 86, 89, 92, 95, 98}.
Given two fuzzy sets
A = "temperature of input steam is hot"= $\left\{\frac{0}{175}+\frac{0.7}{180}+\frac{1}{185}+\frac{0.4}{190}\right\}$

B = "separation of mixture is good" $=\left\{\frac{0}{89}+\frac{0.5}{92}+\frac{0.8}{95}+\frac{1}{98}\right\}$.

Find the fuzzy relation corresponding to " IF x is $\tilde{A}$, THEN y is $\tilde{B}$

(b) Show how inference is done using Generalized Modus Ponens **(6)**

OR

18. (a) Illustrate how graphical inference is done using Mamdani method. **(6)**

(b) A restaurant uses a fuzzy inference system to calculate the tips given to its employees. The tips are based on the timeliness of service and quality of service of the waiters. Design appropriate membership functions for the input and illustrate the use of Sugeno Inference in arriving at the tip amount. **(8)**

19. (a) Explain fuzzy pattern recognition using multiple features. **(7)**

(b) Describe how fuzziness in information retrieval can enhance the quality of search results. **(7)**

OR

20. (a) Design a fuzzy control system for an air-conditioning system. **(7)**

(b) Illustrate how the join operation is performed in fuzzy databases. **(7)**

Teaching Plan

No	Contents	No. of Lecture Hours (36 hrs)
	Module-1(Basic Fuzzy Set Theory) (6 hours)	
1.1	Introduction to Fuzzy Concepts – Case for imprecision- utility and limitations of Fuzzy Systems	1 hour
1.2	Classical Sets – Properties, Operations	1 hour
1.3	Fuzzy Sets – Properties, Operations	1 hour
1.4	Classical Relations – Properties, Operations – Cartesian Product,	1 hour

	Composition	
1.5	Fuzzy Relations – Properties, Operations, Cardinality	1 hour
1.6	Fuzzy Cartesian Product, Fuzzy Composition	1 hour
Module-2 (Fuzzy Membership Functions) (6 hours)		
2.1	Tolerance and Equivalence Relations - Crisp	1 hour
2.2	Tolerance and Equivalence Relations - Fuzzy	1 hour
2.3	Similarity Methods – Cosine, Minmax	1 hour
2.4	Fuzzy Membership Functions- Features	1 hour
2.5	Fuzzification, Defuzzification to crisp sets – λ-cuts	1 hour
2.6	Linguistic Hedges	1 hour
Module-3 (Fuzzification and Defuzzification Methods) (7 hours)		
3.1	Development of Membership Functions – Intuition, Inference	1 hour
3.2	Development of Membership Functions – Rank Ordering	1 hour
3.3	Development of Membership Functions – Inductive reasoning	1 hour
3.4	Defuzzification – Max membership principle, weighted average method, mean max membership	1 hour
3.5	Defuzzification – Centroid method	1 hour
3.6	Defuzzification – Center of Sums, Center of Largest area, First/Last of maxima	1 hour
3.7	Defuzzification - exercises	1 hour
Module-4 (Fuzzy Inference) (9 hours)		
4.1	Classical Logic – Propositional Logic	1 hour
4.2	Classical Logic – Predicate Logic	1 hour
4.3	Fuzzy Logic	1 hour
4.4	Fuzzy Approximation based reasoning	1 hour
4.5	Fuzzy Rule based systems	1 hour
4.6	Multiple conjunctive and disjunctive antecedents, aggregation	1 hour
4.7	Graphical Techniques for Inference	1 hour
4.8	Illustration of Graphical Techniques for Inference	1 hour

4.9	Fuzzy Inference - Exercises	1 hour
Module-5 (Fuzzy Applications) (8 hours)		
5.1	Fuzzy Control Systems	1 hour
5.2	Illustration of Fuzzy Control Systems	1 hour
5.3	Fuzzy Classification	1 hour
5.4	Fuzzy Pattern Recognition	1 hour
5.5	Fuzzy Systems and Neural Networks	1 hour
5.6	Fuzzy Clustering	1 hour
5.7	Fuzzy Databases	1 hour
5.8	Fuzzy Information Retrieval Systems	1 hour

CST464	**EMBEDDED SYSTEMS**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**	**YEAR OF INTRODUCTION**
		PEC	**2**	**1**	**0**	**3**	**2019**

Preamble: The objective of this course is to familiarize learners with the technologies behind embedded computing systems. This course introduces and explains the role of different hardware, software, and firmware components involved in the design and development of embedded systems. It discusses how real time operating systems incorporate specific features to ensure timeliness of critical tasks. The course also aims to provide insights about the design followed in several real-world embedded devices and expose the recent trends in embedded system design to the students.

Prerequisite: Basic knowledge of Operating Systems, Computer Organization and Architecture.

Course Outcomes: After the completion of the course the student will be able to

CO1	Describe the characteristics of different hardware/software components of an embedded system. **(Cognitive Knowledge Level: Understand)**
CO2	Map the design of an embedded system to an appropriate computational model. **(Cognitive Knowledge Level: Apply)**
CO3	Recommend appropriate process synchronization / communication / scheduling mechanisms for specific system scenarios. **(Cognitive Knowledge Level: Analyze)**
CO4	Describe the role of real-time operating systems in embedded devices. **(Cognitive Knowledge Level: Understand)**
CO5	Make use of design strategies for developing real-world embedded systems. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓								✓

CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**20**	**20**	**20**
Understand	**50**	**50**	**50**
Apply	**30**	**30**	**30**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus

Module – 1 (Introduction to Embedded Systems)

Embedded Systems – Definitions, Embedded Systems vs. General Computing Systems, History, Classification, Application Areas, Purpose. Building Blocks of a Typical Embedded System – System Core (Microprocessors, Microcontrollers, DSP, ASICs, PLDs), Memory (Different ROMs and RAMs), Sensors and Actuators, I/O Subsystem Interface, Communication Interface, Embedded Firmware, Other System Components (Reset and Brown-out Protection Circuits, Oscillator Unit, Real-Time Clock, Watchdog Timer), Printed Circuit Board. Embedded System Design Process – Requirements, Specification, Architecture Design, Designing Hardware and Software Components, System Integration.

Module - 2 (System Modeling and Hardware Software Co-Design)

Computational Models in Embedded Design – Data Flow Graph, Control Data Flow Graph, State Machine Model, Sequential Program Model, Concurrent Process Model, Object-Oriented Model. Hardware Software Co-Design – Traditional Embedded Development Cycle, History**,** Advantages of the Co-Design Methodology, The Co-Design Process, Fundamental Issues in Hardware Software Co-Design. Hardware software trade-offs.

Module - 3 (Real-Time Embedded System Design)

Prerequisite Topics: Operating System – Basics, Types. Basics of Tasks, Process and Threads. Multiprocessing and Multitasking. Task Scheduling – Non-Preemptive (FIFO, LIFO, SJF) and Preemptive (SRT, RR, Priority-based, Rate-based).

Task Communication – Shared Memory, Message Passing, Remote Procedure Call and Sockets. Task Synchronization – Synchronization Issues – Race Condition, Deadlock, Priority Inversion, Priority Inheritance, Priority Ceiling. Synchronization Techniques – Spin Lock, Sleep & Wakeup, Semaphores. Selection of an RTOS for an Embedded Design – Functional and Non-Functional Requirements.

Module 4 – (Embedded Firmware Design and Development, and EDLC)

Embedded Firmware Design and Development – Firmware Design Approaches, Firmware Development Languages. Integration of Embedded Hardware and Firmware.

Embedded Product Development Life Cycle – Objectives, Different Phases, Modeling Techniques – Waterfall Model, Incremental Model, Evolutionary Model, Spiral Model.

Module 5 (Embedded System Industry – Case Studies and Applications)

Design Case Studies – Battery Operated Smart Card Reader, Automated Meter Reading System, Smart Watch.

Automotive and Aerospace Systems – Networked Control Systems in Cars and Airplanes, Vehicular Networks – CAN bus, Time-triggered Architecture, FlexRay and LIN.

Internet of Things Systems – IoT System Architectures - Use Cases (Smart Appliance, Monitoring and Control Systems). Networks for IoT – Networking concepts, Bluetooth, Bluetooth Low Energy, 802.15.4, ZigBee and WiFi. Databases and Timewheels. Smart Home Example.

Text Books:

1. K. V. Shibu, *Introduction to Embedded Systems*, McGraw Hill Education, Second Edition, 2017.
2. James K. Peckol, *Embedded Systems: A Contemporary Design Tool,* John Wiley & Sons, Second Edition, 2019.
3. Marilyn Wolf, *Computers as Components-Principles of Embedded Computing System Design,* Morgan Kaufmann, Elsevier, Fourth Edition, 2016.

References:

1. Jorgen Staunstrup and Wayne Wolf, *Hardware/Software Co-Design: Principles and Practice,* Springer Science & Business Media, 2013.
2. Raj Kamal, *Embedded Systems: Architecture, Programming and Design*. Tata McGraw-Hill Education, 2011.
3. Daniel D. Gajski, Samar Abdi, Andreas Gerstlauer, and Gunar Schirner, *Embedded System Design: Modeling, Synthesis and Verification,* Springer Science & Business Media, 2009.
4. Peter Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things,* Springer, 2017.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are Sensors? Explain its role in Embedded System Design. Illustrate with an example.
2. How do sensors communicate data with other computing devices in an embedded system?
3. What are the advantages and disadvantages of using a microprocessor as the compute engine of an embedded device?
4. What is a watchdog timer? What can go wrong if your system does not have one?
5. Elaborate the steps involved in the design of an embedded system with appropriate diagrams.

Course Outcome 2 (CO2):

1. Draw a class diagram to represent a smart eyeglass that automatically changes the glass shade according to the external light.
2. Design a Seat Belt Warning System and explain its working using a state machine model.

3. Represent the authentication process of an ATM machine using one of the following computational models:
    1. Control Data Flow Graph.
    2. State Transition Diagram.
4. Draw the Finite State Machine diagram for an automated tea/coffee vending machine.
5. Draw a CDFG for the following program fragment.

```
fun0();
if (cond1) fun1();
else fun2();
fun3();
switch (test1) {
        case 1 : fun4();
                break;
        case 2: fun5();
                break;
        case 3: fun6();
                break;
}
fun7();
```

Course Outcome 3(CO3):

1. Explain the working of spin locks. When are spin locks preferred to sleep and wake up mechanisms to synchronize process execution?
2. Assume 3 tasks with the following characteristics:

Task	Arrival Time	Execution Time
A	0	7
B	2	9
C	4	6

- A goes for I/O for 5 units after 4 units of execution time in CPU.
- B and C are CPU bound tasks.
- Find out if a First Come First Serve or Shortest Job First scheduling strategy will offer the shortest average waiting time for the above scenario.

3. An organization maintains energy smart buildings with the help of different types of computing devices spread across different levels of a building. What process communication mechanism do you recommend? Why?
4. With an example, illustrate how priority inversion can cause high priority tasks to miss deadlines.
5. Consider the following scenario:

   ◦ There are three tasks, H, M and L with high, medium, and low priority. Task L and task H share a resource. Shortly after Task L takes the resource, Task H becomes ready to run. However, Task H must wait for Task L to finish with the resource, so it pends. Before Task L finishes with the resource, Task M becomes ready to run, preempting Task L. While Task M runs, Task H, the highest-priority task in the system, remains in a pending state.

   How can you prevent the high priority task H, from missing any deadlines?

Course Outcome 4 (CO4): .

1. Differentiate General Purpose Operating System (GPOS) from Real-Time Operating System (RTOS).
2. Explain briefly the functional and non-functional requirements that need to be addressed in the selection of an RTOS.
3. What are the characteristics of a real-time system?
4. How does a hard real-time system differ from a soft-real time system? Explain with an example.
5. Identify the reasons for choosing an operating system-based firmware to a super-loop model-based firmware in an embedded device.

Course Outcome 5(CO5):

1. Prepare a requirement chart for a robotic vacuum cleaner. Fine tune the requirement chart to list the product specifications (Assignment, preferably group work).
2. Suppose you want to make an old building energy smart. Analyze existing systems and prepare a report on how this can be achieved following an Embedded product Development Life Cycle (EDLC) model (Assignment, preferably group work). (Allow groups to make necessary assumptions such as prevailing climatic conditions).
3. Design a baby monitoring system with suitable active devices and networking components. Represent your system with a suitable diagram.
4. Identify the components required to build a battery-operated smart card reader and design the system using these components.

5. Draw a sequential diagram representing the working of an automated energy metre reading system.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST464

Course Name: Embedded Systems

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Differentiate between PLD and ASIC.
2. List the factors that need to be considered in the selection of memory for embedded systems.
3. How does building an embedded system become a hardware-software co-design problem?
4. If the time to market of an embedded device is critical, would you prefer a high level language or assembly language for developing the firmware? Justify.
5. Explain the various factors that need to be considered for the selection of a scheduling algorithm.
6. What is the difference between ‘Hard’ and ‘Soft’ real-time systems? Give an example for ‘Hard’ and ‘Soft’ real-time kernels.
7. When will you choose a concurrent process model for an embedded device?

8. What are the three primary objectives of EDLC? List the different phases of this development life cycle.

9. Why are traditional databases not preferred in embedded devices?

10. Draw the functional block diagram of a fully automatic washing machine.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) What is embedded firmware? Which are the different approaches used for embedded firmware development? **(6)**

(b) List out the requirements for a smartwatch that can show time as well as measure the user's vitals such as blood pressure, pulse rate and body temperature. Think of the physical and functional aspects of the product and also what extra features can be added over the basic functions. **(8)**

OR

12. (a) Describe various on-board communication interfaces used in embedded systems. **(10)**

(b) Explain the role of a watchdog timer in an embedded system. **(4)**

13. (a) Explain the object oriented program model for embedded system design. Under which circumstances is this model considered as the best? **(6)**

(b) Design and draw a concurrent program model for the Seat Belt Warning System of an automobile. Clearly specify your assumptions in the design. **(8)**

OR

14. (a) Suppose you are designing a digital motion camera. What are the tradeoffs in implementing the multimedia codec part of the camera in hardware and in software? **(6)**

(b) Draw the state machine diagram of an automatic dawn-dusk solar street light with a motion sensor. **(8)**

- The light automatically turns on when the ambient light is below ‘x’ lumens.
- The streetlight operates at low power when no motion is detected for 30 secs.

15. (a) Identify the type of synchronization mechanism best suited for each of the following scenarios and explain why. **(6)**

    1. Updating a single byte memory location shared by different tasks, some of which are time-critical.
    2. Updating a block of memory that is shared between multiple low priority tasks.
    3. Multiple instances of a resource shared between tasks.

    (b) Under what circumstances would you use the following task communication mechanisms? **(8)**

    1. Sockets.
    2. Remote Procedure Call
    3. Shared Memory

OR

16. (a) What is priority inversion? In case of priority inversion, what techniques can be adopted to ensure that the critical tasks are able to meet their deadlines? **(7)**

    (b) Three processes with process IDs P1, P2, P3 with estimated completion time 6, 2, 4 milliseconds respectively, enters the ready queue together in the order P1, P3, P2. Process P4 with estimated execution time 4 milliseconds entered the ready queue 3 milliseconds later the start of execution of P1. Calculate the waiting time and Turn Around Time (TAT) for each process and the Average waiting time and Turn Around Time (Assuming there is no I/O waiting for the processes) in RR algorithm with Time slice = 2 ms. **(7)**

17. (a) An embedded product under consideration is very complex in nature and there is a possibility for change in requirements of the product. Also the risk associated with the development of this product is very high. Which is the best suited life cycle method to handle this product development? Justify your answer. **(8)**

    (b) Explain the similarities and differences between iterative and incremental life cycle models. **(6)**

OR

18. (a) When do you prefer a super-loop-based firmware design approach over an RTOS based approach? What are the limitations of the super-loop based approach and how do you overcome them? **(8)**

    (b) Briefly explain the different approaches used for embedding firmware into the hardware of an embedded device. **(6)**

19. (a) Identify the components required to build an automated energy metre reading system and design it with these components **(6)**

(b) List any four wireless standards used for building IoT networks and compare their characteristics **(8)**

OR

20. (a) Draw a sequential diagram representing the working of a battery-operated smart card reader. **(6)**

(b) Explain the different communication buses used in automotive applications. **(8)**

Teaching Plan

No	Contents	No. of Lecture Hours (35 hrs)
Module-1 (Introduction to Embedded Systems) (8 hours)		
1.1	Introduction–Embedded Systems, Characteristics and Quality Attributes of Embedded Systems	1 hour
1.2	System Core (Microprocessors, Microcontrollers, DSP, ASICs, PLDs)	1 hour
1.3	System Core (Microprocessors, Microcontrollers, DSP, ASICs, PLDs)	1 hour
1.4	Memory (Different ROMs and RAMs), Sensors and Actuators, I/O Subsystem Interface	1 hour
1.5	Communication Interface	1 hour
1.6	Embedded Firmware, Other System Components (Reset and Brown-out Protection Circuits, Oscillator Unit, Real-Time Clock, Watchdog Timer), Printed Circuit Board.	1 hour
1.7	Embedded System Design Process – Requirements, Specification, Architecture Design.	1 hour
1.8	Embedded System Design Process–Designing Hardware and Software Components, System Integration.	1 hour
Module-2 (System Modeling and Hardware Software Co-Design) (6 hours)		
2.1	Computational Models in Embedded Design – Data Flow Graph, Control Data Flow Graph	1 hour
2.2	Computational Models in Embedded Design – State Machine Model, Sequential Program Model	1 hour

2.3	Computational Models in Embedded Design – Concurrent Process Model, Object-Oriented Model.	1 hour
2.4	Hardware Software Co-Design – Traditional Embedded Development Cycle, History, Advantages of the Co-Design Methodology	1 hour
2.5	The Co-Design Process	1 hour
2.6	Fundamental Issues in Hardware Software Co-Design. Hardware software trade-offs.	1 hour
Module-3 (Real-Time Embedded System Design) (7 hours)		
3.1	Task Communication–Shared Memory, Message Passing	1 hour
3.2	Task Communication–Remote Procedure Call and Sockets	1 hour
3.3	Task Synchronization–Synchronization Issues – Race Condition, Deadlock	1 hour
3.4	Task Synchronization–Synchronization Issues – Priority Inversion, Priority Inheritance, Priority Ceiling.	1 hour
3.5	Synchronization Techniques – Spin Lock, Sleep & Wakeup, Semaphores	1 hour
3.6	Synchronization Techniques – Spin Lock, Sleep & Wakeup, Semaphores	1 hour
3.7	Selection of an RTOS for an Embedded Design – Functional and Non-Functional Requirements	1 hour
Module-4 (Embedded Firmware Design and Development, and EDLC) (6 hours)		
4.1	Firmware Design Approaches	1 hour
4.2	Firmware Development Languages	1 hour
4.3	Firmware Development Languages	1 hour
4.4	Integration of Embedded Hardware and Firmware	1 hour
4.5	Embedded Product Development Life Cycle–Objectives, Different Phases	1 hour
4.6	Embedded Product Development Life Cycle – Modeling Techniques – Waterfall Model, Incremental Model, Evolutionary Model, Spiral Model (Review Only)	1 hour
Module-5 (Embedded System Industry – Case Studies and Applications) (8 hours)		
5.1	Design Case Studies–Battery Operated Smart Card Reader	1 hour
5.2	Design Case Studies–Automated Meter Reading System	1 hour
5.3	Design Case Studies–Smart Watch	1 hour

5.4	Automotive and Aerospace Systems – Networked Control Systems in Cars and Airplanes	1 hour
5.5	Automotive and Aerospace Systems – Vehicular Networks – CAN bus, Time-triggered Architecture, FlexRay and LIN	1 hour
5.6	Internet of Things Systems – IoT System Architectures – Use Cases (Smart Appliance, Monitoring and Control Systems)	1 hour
5.7	Internet of Things Systems – Networks for IoT – Networking concepts, Bluetooth, Bluetooth Low Energy, 802.15.4, ZigBee and WiFi.	1 hour
5.8	Internet of Things Systems – Databases and Timewheels, Smart Home Example	1 hour

CST474	COMPUTER VISION	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs. The curriculum covers the basics of image formation, key computer vision concepts, methods, techniques, pattern recognition, various problems in designing computer vision and object recognition systems. This course enables the learners to understand the fundamentals of computer vision and develop applications in computer vision.

Prerequisite: Nil

Course Outcomes: After the completion of the course, the student will be able to

CO1	Summarize basic concepts, terminology, theories, models and methods in the field of computer vision. **(Cognitive Knowledge Level: Understand)**
CO2	Explain basic methods of computer vision related to multi-scale representation, edge detection, detection of other primitives, stereo, motion and object recognition. **(Cognitive Knowledge Level: Understand)**
CO3	Describe principles of Segmentation, Motion Segmentation and Classification **(Cognitive Knowledge Level: Understand)**
CO4	Select appropriate object Tracking and detection methods for computer vision applications **(Cognitive Knowledge Level: Understand).**
CO5	Implement a computer vision system for a specific problem **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓		✓									✓
CO2	✓		✓									✓
CO3	✓		✓									✓

CO4	✓		✓									✓
CO5	✓	✓	✓	✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**30**	**30**	**30**
Understand	**50**	**50**	**50**
Apply	**20**	**20**	**20**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests1&2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Image Formation and Filtering)

Geometric Camera Models - Pinhole perspective, Intrinsic and Extrinsic Parameters, Geometric Camera Calibration. Linear Filters**-** Linear Filters and Convolution, Shift Invariant Linear Systems. Filters as Templates - Normalized Correlation and Finding Patterns.

Module - 2(Local Image Features and Stereo Vision)

Image Gradients - Computing the Image Gradient, Gradient Based Edge and Corner Detection. Stereopsis- Binocular Camera Geometry, Epipolar Constraint, Binocular Reconstruction, Local Methods for Binocular Fusion, Global Methods for Binocular Fusion.

Module - 3 (Segmentation)

Segmentation - Background subtraction, Interactive segmentation, Forming image regions. Segmentation by clustering - Watershed Algorithm. Motion Segmentation by Parameter Estimation- Optical Flow and Motion, Flow Models, Motion Segmentation with Layers.

Module- 4 (Classification and Tracking)

Classification - Classification Basics, Two-class and Multiclass classifiers, Error, Overfitting and Regularization, Cross Validation, Classifying Images of Single Objects.
Tracking - Tracking Basics, Simple Tracking Strategies, Tracking by detection, Tracking Linear Dynamical models with Kalman filters.

Module - 5 (Finding Objects and other Applications)

Object detection - The Sliding Window Method. Object Recognition -Goals of Object Recognition System. Applications - Robot Navigation by stereo vision, Face detection, Face recognition, Activity Recognition, Tracking people.

Text Books

1. Forsyth, David, and Jean Ponce. Computer vision: A modern approach. Prentice hall, 2011.

Reference Books

1. Szeliski, Richard, Computer vision: algorithms and applications. Springer Science & Business Media, 2010.
2. Medioni, Gerard, Emerging topics in computer vision. and Sing Bing Kang. Prentice Hall PTR, 2004.
3. Trucco, Emanuele, and Alessandro Verri, Introductory techniques for 3-D computer vision. Vol. 201. Englewood Cliffs: Prentice Hall, 1998.
4. Faugeras, Olivier, and Olivier Autor Faugeras, Three-dimensional computer vision: a geometric viewpoint. MIT press, 1993.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the relationship between coordinates involved in a pinhole camera imaging setup.
2. Explain the basic principle behind geometric camera calibration**.**
3. Describe how linear filters can be used for smoothing digital images.
4. How does normalised correlation help in matching patterns in images?

Course Outcome 2 (CO2):

1. Describe edge detection methods for computer vision.
2. List any five applications of object recognition.
3. Explain how the epipolar constraint simplifies the correspondence search between two stereo images.
4. List and explain the different methods used for binocular fusion.
5. Explain the different corner detection methods.

Course Outcome 3 (CO3):

1. Explain the principle of background subtraction.
2. Describe the watershed algorithm for image segmentation.
3. What is meant by optical flow? How can it be utilized for segmenting images?
4. Describe motion segmentation with layers.
5. What is overfitting in the context of classification?
6. Explain the principle behind classification of single images.

Course Outcome 4 (CO4):

1. Explain 'Mean Shift Algorithm' to track an object using matching.
2. Describe an algorithm to track a moving object (dynamic object).
3. Explain the sliding window method for object detection.
4. Assume that we have the dynamics

$$x_i \sim N(d_i x_{i-1}, \sigma^2_{d_i})$$

$$y_i \sim N(m_i x_i, \sigma^2_{m_i})$$

a. $P(x_i|x_{i-1})$ is a normal density with mean $d_i x_{i-1}$ and variance $\sigma^2_{d_i}$. Whatis$(x_{i-1}|x_i)$?

b. Show how to obtain a representation of $P(x_i|y_{i+1},\dots,y_N)$ usinga KalmanFilter.

Course Outcome 5(CO5):

1. Explain how to implement a computer vision system.
2. Illustrate a computer vision system with the help of a neat diagram.
3. Discuss the components of a computer vision system for object recognition.
4. Explain how activity recognition can be done using computer vision.
5. Illustrate a face recognition system with the help of a diagram.

Assignment Questions

6. Implement a voxel-based approach to visual hull construction.
7. Implement a computer vision system for object recognition.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST474

Course Name: COMPUTER VISION

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. State three properties of shift invariant linear systems.
2. Explain the term normalized correlation.
3. What is image rectification? Mention its significance?
4. Illustrate epipolar geometry and showepipolar lines and epipoles.
5. Explain the term flow model.
6. How does background subtraction help in segmenting an image?
7. What is a Kalman filter? Give its applications.
8. State any three simple tracking strategies.
9. State the goals of an object recognition system.
10. Explain the task of face recognition. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Demonstrate the relationship between a point in the world coordinate frame and its corresponding image point using camera parameters. **(9)**

(b) Show that convolving a function with a δ function simply reproduces the original function. **(5)**

OR

12. (a) What is linear filtering? Explain two applications of linear filtering to image processing. **(7)**

(b) Explain an application of normalised correlation to find patterns. **(7)**

13. (a) Show that smoothing an image and then computing the gradient is same as convolving an image with the derivative of a smoothing function. **(5)**

(b) State the epipolar constraint and derive its representations using the Essential matrix and the Fundamental matrix. **(9)**

OR

14. (a) Explain the algorithm for computing edges using gradients. **(9)**

(b) Define binocular fusion. Explain two local methods for binocular fusion. **(5)**

15. (a) Discuss the different interactive segmentation approaches. **(7)**

(b) What is meant by optical flow? How can it be utilized for segmenting images? **(7)**

OR

16. (a) Explain the Watershed algorithm. **(7)**

(b) How can we perform motion segmentation by parameter estimation? **(7)**

17. (a) Explain tracking algorithm using Kalman filtering. **(7)**

(b) Illustrate the tracking by detection algorithm. **(7)**

OR

18. (a) Explain the various kinds of errors in classification and the relationship between them. **(7)**

(b) What is overfitting and how does regularization help to minimise it? **(7)**

19. (a) Explain human activity recognition with appearance features. **(7)**

(b) Describe the Sliding window method for detecting objects in images. **(7)**

OR

20. (a) Explain the principle of detecting faces in an image. **(7)**

(b) What are the various strategies for object recognition? **(7)**

Teaching Plan

No	Contents	No. of Lecture Hours (36hrs)
	Module 1 Image Formation and Filtering (7)	
1.1	Geometric Camera model - Pinhole perspective	1
1.2	Geometric Camera model - Intrinsic Parameters	1
1.3	Geometric Camera model - Extrinsic Parameters	1
1.4	Geometric Camera Calibration – Linear Approach	1
1.5	Linear Filters and Convolution	1
1.6	Shift Invariant Linear Systems - Discrete convolution	1
1.7	Normalized Correlation and Finding patterns	1
	Module 2 Local Image Features and Stereo Vision (8)	
2.1	Local Image Features - Computing the Image Gradient	1
2.2	Gradient Based Edge Detection	1
2.3	Gradient Based Corner Detection	1
2.4	Stereopsis - Binocular Camera Geometry and Epipolar Constraint	1
2.5	Essential Matrix and Fundamental Matrix	1
2.6	Binocular Reconstruction	1
2.7	Local Methods for Binocular Fusion	1
2.8	Global Methods for Binocular Fusion	1
	Module 3 Segmentation (6)	

3.1	Segmentation basics	1
3.2	Applications - Background Subtraction, Interactive Segmentation	1
3.3	Forming Image Regions	1
3.4	Segmentation by clustering - The Watershed Algorithm	1
3.5	Motion Segmentation by Parameter Estimation - Optical Flow and Motion	1
3.6	Flow Models and Motion Segmentation with Layers	1
Module 4 Classification and Tracking (8)		
4.1	Classification Basics, Two-class and Multiclass classifier	1
4.2	Error, Overfitting and Regularization	1
4.3	Cross Validation, Classifying Images of Single Objects	1
4.4	Tracking Basics, Simple Tracking Strategies	1
4.5	Tracking by detection	1
4.6	Linear Dynamical models	1
4.7	The Kalman Filter background	1
4.8	Kalman filter algorithm	1
Module 5 Finding Objects and other Applications (7)		
5.1	Detecting Objects in Images- The Sliding Window Method	1
5.2	Object Recognition - Goals of Object Recognition System	1
5.3	Application of binocular stereo vision - Robot Navigation	1
5.4	Face detection	1
5.5	Face recognition	1
5.6	Activity recognition	1
5.7	Tracking people	1

SEMESTER VIII

PROGRAM ELECTIVE IV

CST416	FORMAL METHODS AND TOOLS IN SOFTWARE ENGINEERING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: The course enables the learners to apply formal methods for modelling, validation and verification of software systems. It covers a series of advanced tools that address challenges faced in design, coding and verification. This includes both an introduction to the theoretical underpinnings of these tools, as well as hands-on exploration.

Pre-requisites: Fundamental knowledge in Formal Methods and Software Engineering

Course Outcomes: After the completion of the course, the student will be able to

CO1	Explain the need and use of formal methods and tools in software engineering. **(Cognitive Knowledge Level: Understand)**
CO2	Demonstrate conceptual modelling of systems using *Alloy*.**(Cognitive Knowledge Level: Apply)**
CO3	Illustrate the process of proving correctness of code using Hoare-Triple based weakest precondition analysis.**(Cognitive Knowledge Level: Apply)**
CO4	Demonstrate program verification using VCC. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓	✓	✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests1&2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Introduction)

Stages in software development; software defects –causes of software defects; techniques for dealing with software defects-Testing and verification, formal methods and tools.

Module– 2 (Ensuring reliability in the design phase)

Conceptual modelling, the tool Alloy, conceptual modelling in Alloy, Analysing Alloy models, Fixing bugs in modelling, How Alloy works? Show that the Konigsberg Bridge Problem has no solution.

Module - 3 (Verification by Model Checking)

Verifier for Concurrent C (VCC): a Hoare-Triple- based tool for Verifying Concurrent C, intra-procedure verification of programs, ghost statements.

Module–4 (Program Verification)

Inter-procedure verification of programs in VCC, function contracts, pure functions, loop-invariants, proving total correctness of programs in VCC.

Module–5 (Ghost Language and Ownership in VCC)

Ghost Language of VCC, modelling programs in the ghost language, verification of a C program with respect to a ghost model, ownerships in VCC, Refinement for proving correctness, Proving refinements in VCC, Example problems

Text Books

1. Daniel Jackson, Software Abstractions, MIT Press, 2011.

Reference Materials

1. Tutorial for Alloy Analyzer 4.0
2. E. Cohen, M. A., Hillebrand, S. Tobies, M. Moskal, W. Schulte, Verifying C Programs: A VCC Tutorial, Working draft, version 0.2, July 10, 2015.
3. The VCC Manual, Working draft, version 0.2, April 7, 2016.

Course Level Assessment Questions

Course Outcome1 (CO1):

1. Is the following code segment to increment a number safe? If not, provide the reason.

```
int increment(int x)
{
  return ++x;
}
```

Course Outcome 2 (CO2):

1. A farmer is on one shore of a river and has with him a fox, a chicken, and a sack of grain. He has a boat that fits one object besides himself. In the presence of the farmer nothing gets eaten, but if left without the farmer, the fox will eat the chicken, and the chicken will eat the grain. How can the farmer get all three possessions across the river safely?Solve the problem by modeling it in Alloy and using the analyzer to find a solution.

Course Outcome 3 (CO3):

1. Find an inductive loop invariant and show the correctness proof for the statement

$$while\ i < n\ do\ \{a[i] := 0; i := i + 1;\}$$

with the precondition $i = 0 \wedge n > 0$and the post condition $\forall j, 0 \leq j < n \rightarrow a[j] = 0$.

Course Outcome 4 (CO4):

1. Write and verify a program that sorts the elements of an array in non-decreasing order. Use VCC for verification.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST416

Course Name: Formal Methods and Tools in Software Engineering

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Give an example of a software defect.

2. Explain the benefits of formal models in software engineering.

3. Draw the architecture of *Alloy*.

4. With an example, explain a predicate in *Alloy*.

5. Define Loop Invariant. Give an example.

6. Describe the purpose of Hoare Logic.

7. State a sufficient condition required in the assert statement below to provide the

correctness of the function *smallest*.

```
int smallest(int x, int y)
{
  int val;
  if (x<=y)
    val = x;
  else
    val = y;
assert(....);
  return val;
}
```

8. Explain inter-procedure verification of programs in VCC.

9. Why do you need a specification language like Ghost for VCC?

10. Give the difference between *\writable* and *\writes* in VCC.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. a Explain the stages in software development. **(10)**

    b Differentiate between testing and verification. **(4)**

OR

12. (a) Explain the important causes of software defects. **(7)**

    (b) Describe the terms *Formal Methods* and *Formal Verification***.** **(7)**

13. (a) Model the following system in *Alloy Analyzer*. **(7)**

    There is an entity named *Person*. *Man* and *Woman* are two specializations of it. Every *Person* has a *Father*(a *Man*) and a *Mother*(a Woman) as *Parent*. The *Parents* of a *Person* should be married. A *Man's spouse* should be a *Woman* and a *Woman's spouse* should be a *Man*. The *spouse* relation is symmetric.

    Add a predicate to check whether marriage between siblings is possible in the above system.

(b) Model the *stack* data structure in *Alloy* with support for *Push* and *Pop* operations. Treat each operation that modifies the stack's state as an event. The stack should begin empty, and you should not be able to Pop from an empty stack. **(7)**

Write three assertions verifying the behaviour of your stack in the following cases:

- A Pop followed by a Push of the same element
- The number of Push events is exactly equal to the number of Pop events
- A Pop from a stack with no elements

14. (a) Model an operating system as follows. **(7)**

```
// The following signature models the set of all tasks in the system
sig Task {}
// Following signature models the operating system
one sig OS {
  free: set Task, // set of free tasks in the OS, which are ready to be created
  ready: set Task, // set of ready tasks
  deleted: set Task, // set of deleted tasks
  running: one Task // the currently running task
}
```

Add the following properties

P1: The running task is also in the ready state.

P2: There is at least one free task (in order to enable task creation)

P3: Each task is in exactly one of the three states free, ready, deleted.

Create a predicate *noP1* that looks for instances that do not satisfy property *P1*, and a corresponding*run* statement. Similarly, create predicates *noP2* and *noP3*.

(b) Show a model in Alloy to prove that the Konigsberg bridge problem has no solution. **(7)**

15. (a) Use Hoare Logic to prove **(6)**

(i) $\{z = 2\}\, y := x\, \{y = x\}$

(ii) $\{true\}\, x := 2;\ y := x\, \{y = 2 \wedge x = 2\}$

(b) Differentiate between partial correctness and total correctness in Hoare Logic. **(8)**

16. (a) Use Hoare Logic with the help of loop invariant to prove **(8)**

    (i) $\{x \ \leq n\}\, while\ x < n\ do\ x{:} = x + 1\ \{x \ \geq n\}$

    (ii) $\{i \ = 0 \wedge j = 0 \wedge n = 5\}\, while\ i < n\ do\ i \coloneqq i + 1; j \coloneqq j + 1\ \{j = 15\}$

    (b) Illustrate assertions and assumptions in VCC. **(6)**

17. (a) Write a function to find the largest of three numbers and prove using function contract that your program is correct. Also write another function that calls the former and assert that the latter returns the correct result. **(10)**

    (b) Write notes on inter-procedure verification of programs in VCC. **(4)**

OR

18. (a) Write and verify a non-recursive program that takes an array and checks whether it contains any duplicate elements. **(10)**

    (b) Distinguish between sequential and atomic memory access in VCC. **(4)**

19. (a) Explain how the refinement conditions can be phrased in VCC. **(7)**

    (b) Illustrate how refinements are proved in VCC. **(7)**

OR

20. (a) Explain the Ghost language of VCC **(7)**

    (b) Illustrate the verification of a C program with respect to its Ghost model. **(7)**

Teaching Plan

No	Contents	No. of Lecture Hours (36 hrs)
	Module-1(Introduction) (5 hours)	
1.1	Stages in software development.	1 hour
1.2	Software defects and causes of software defects.	1 hour
1.3	Techniques for dealing with software defects.	1 hour
1.4	Testing and verification.	1 hour
1.5	Formal methods and formal verification.	1 hour
	Module-2 (Conceptual Modelling in Alloy) (7 hours)	
2.1	Introduction to Conceptual modelling.	1 hour
2.2	Overview of Alloy, Architecture of alloy.	1 hour
2.3	Conceptual modelling in Alloy.	1 hour
2.4	Analysing Alloy models.	1 hour
2.5	Fixing bugs in modelling.	1 hour
2.6	How Alloy works?	1 hour
2.7	Show that the Konigsberg Bridge Problem has no solution.	1 hour
	Module-3 (Hoare Logic and Introduction to VCC) (11 hours)	
3.1	Introduction to VCC.	1 hour
3.2	Verifying C programs in VCC- Assertions, Logical Operators and Quantifiers, Assumptions, Overflows and unchecked arithmetic.	1 hour
3.3	Hoare Logic - Simple Imperative Programming Language, Partial Correctness Specification.	1 hour
3.4	Meaning of Hoare Triples, Hoare-Triple- based tool for Verifying Concurrent C.	1 hour

3.5	Partial vs. Total Correctness, Proving Partial Correctness.	1 hour
3.6	Inference Rules for the Simple Imperative Programming Language (Lecture 1).	1 hour
3.7	Inference Rules for the Simple Imperative Programming Language (Lecture 2).	1 hour
3.8	Weakest Precondition.	1 hour
3.9	Invariant vs. Inductive Invariant.	1 hour
3.10	Intra-procedure verification of programs.	1 hour
3.11	Verification of Hoare Triples.	1 hour
Module-4 (Program Verification) (6 hours)		
4.1	Inter-procedure verification of programs in VCC.	1 hour
4.2	Function contracts.	1 hour
4.3	Pure functions.	1 hour
4.4	Quantifiers, loop-invariants and Object invariant.	1 hour
4.5	Triggers in VCC.	1 hour
4.6	Proving total correctness of programs in VCC.	1 hour
Module-5 (Ghost Language and Ownership in VCC) (7 hours)		
5.1	Ghost Language of VCC.	1 hour
5.2	Modelling programs in the ghost language.	1 hour
5.3	Verification of a C program with respect to a ghost model.	1 hour
5.4	Ownerships in VCC.	1 hour
5.5	Phrasing refinement conditions in VCC.	1 hour
5.6	Proving refinements in VCC, Example problems (Lecture 1).	1 hour
5.7	Proving refinements in VCC, Example problems (Lecture 2).	1 hour

CST426	CLIENT SERVER ARCHITECTURE	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates to build effective Client/Server applications. This course aims at providing a foundation in decentralized computer systems, using the client/server model. The course content is decided to cover the essential fundamentals which can be taught within the given slots in the curriculum.

Prerequisite: Computer Networks

Course Outcomes: After the completion of the course the student will be able to

CO 1	Explain the basics of client/server systems and the driving force behind the development of client/server systems **(Cognitive Knowledge Level: Understand)**
CO 2	Outline the architecture and classifications of client/server systems **(Cognitive Knowledge Level: Understand)**
CO 3	Choose the appropriate client/server network services for a typical application **(Cognitive Knowledge Level: Understand)**
CO 4	Describe management services and issues in network **(Cognitive Knowledge Level: Understand)**
CO 5	Compare and summarize the web extensions and choose appropriate web services standards for an application **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓										✓
CO3	✓	✓			✓							✓
CO4	✓											✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test 1 (Marks)	**Test 2 (Marks)**	
Remember	30	30	30
Understand	70	70	50
Apply			
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test 1 (for theory, for 2 hrs)	: 20 marks
Continuous Assessment Test 2 (for lab, internal examination, for 2hrs)	: 20 marks

Internal Examination Pattern: There will be two parts; Part A and Part B. Part A contains 5 questions with 2 questions from each module (2.5 modules x 2 = 5), having 3 marks for each question. Students should answer all questions. Part B also contains 5 questions with 2 questions from each module (2.5 modules x 2 = 5), of which a student should answer any one. The questions should not have sub- divisions and each one carries 7 marks.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Explain the driving forces behind the development of Client/ Server system from different perspectives.

Course Outcome 2 (CO2):

1. How two-tier computing and three-tier computing improves the performance of Client/Server system.

Course Outcome 3(CO3):

1. Explain the role of client in Client/Server computing and also explain the various services provided by client.
2. What is the primary motivation behind the RPC facility ? How does a RC facility makes the job of distributed application programmers simpler?
3. Implement RPC concept using suitable language/tool(Assignment)

Course Outcome 4 (CO4):

1. Explain Connectivity and Communication Interface Technology in Client/Server application. How does transmission protocol work in Client/Server application?

Course Outcome 5 (CO5):

1. Discuss the role of web browser for providing web service in Client/Server environment.
2. Identify and explain the social relevance of web services (Assignment)

Syllabus

Module – 1 (Introduction)

Introduction to Client/Server computing - Driving forces behind Client/ Server, Client/ Server development tools, Development of client/server systems, Client/Server security, Organizational Expectations, Improving performance of client/server applications, Single system image, Downsizing and Rightsizing, Advantages of client server computing, Applications of Client/Server.

Module -2 (Client/Server Application Components)

Classification of Client/Server Systems- Two-Tier Computing, Middleware, Three-Tier Computing- Model View Controller (MVC), Principles behind Client/Server Systems. Client/Server Topologies. Existing Client/Server Architecture. Architecture for Business Information System.

Module -3 (Client/Server Network)

Client- Services, Request for services, RPC, Windows services, Print services, Remote boot services, other remote services, Utility Services. Dynamic Data Exchange (DDE). Object Linking and Embedding (OLE). Common Object Request Broker Architecture (CORBA).

Server- Detailed server functionality, Network operating system, Available platforms, Server operating system.

Module -4 (Client/ Server Systems Development)

Services and Support- System administration, Availability, Reliability, Scalability, Observability, Agility, Serviceability. Software Distribution, Performance, Network management. Remote Systems Management- RDP, Telnet, SSH, Security. LAN and Network Management issues, Training, Connectivity, Communication interface technology, Interprocess communication, Wide area network technologies, Network Acquisition, PC-level processing unit, X-terminals, Server hardware.

Module -5 (Client/Server Technology and Web Services)

Web Services History. Web Server Technology- Web Server, Web Server Communication, Role of Java for Client/Server on Web. Web Services- MicroServices, APIs, API Gateway, Authentication of users/clients, Tokens/Keys for Authentication, Service Mesh, Message Queues, SaaS, Web Sockets.

Client/Server/Browser – Server Technology, Client/Server Technology and Web Applications, Balanced Computing and the Server’s Changing Role. Thin client computing - Computing models-Comparison-Computing Environment.

Future of client/ server Computing Enabling Technologies, Transformational system.

Text Books

1. Patrick Smith & Steave Guengerich, "Client / Server Computing", PHI
2. Dawna Travis Dewire, "Client/Server Computing", TMH

Reference Books

1. Jeffrey D.Schank, "Novell's Guide to Client-Server Application & Architecture" Novell Press
2. Robert Orfali, Dan Harkey, Jeri Edwards, Client/Server Survival Guide, Wiley-India Edition, Third Edition
3. W. H. Inman, Developing Client Server Applications, BPB

Model Question Paper

QP CODE:

Reg No: _______________

Name: __________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST426

Course Name : Client Server Architecture

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. How client/server computing environment is different from mainframe based computing environment?
2. Write short notes on single system image and downsizing.
3. Discuss the topologies of Clients/Server system with suitable examples.
4. Discuss the relevance of Clients/Server system in adopting open system standards. Justify your answer.

5. Enumerate the services provided in a client/server system.

6. List out the features of network operating system.

7. How interposes communication is established?.

8. Write short note on x-terminals.

9. Explain the history of web services.

10. With an example, explain the role of java for client/server on web **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the driving forces behind the development of Client/ Server system from different perspectives. **(10)**

    (b) Explain the various Clients/Server system development tools. **(4)**

OR

12. (a) Explain Client/Server System development methodology and explain various phases and their activities involved in System Integration Life Cycle. **(10)**

    (b) Write short notes on the following. (a) Single system image. (b) Downsizing and Client/Server computing. **(4)**

13. (a) How two-tier computing and three-tier computing improves the performance of Client/Server system. **(10)**

    (b) List out the principles behind client/server systems . **(4)**

OR

14. (a) Explain the architecture of Business Information System. **(10)**

    (b) Explain different ways to improve performance in Client/Server developed applications. **(4)**

15. (a) In Client/Server computing, explain the following with example in detail (a) Dynamic Data Exchange (b) RPC (c) Remote Boot Service (d) Object-linking and embedding. **(10)**

(b) Explain the role of client in Client/Server computing and also explain the various services provide by client. **(4)**

OR

16. (a) Explain the architecture of CORBA. **(10)**

(b) Explain the server functionality in detail, for Client/Server computing. **(4)**

17. (a) Explain Connectivity and Communication Interface Technology in Client/Server application. How does transmission protocol work in Client/Server application? **(10)**

(b) Comment on the network service acquisition mechanism for the client/service model. **(4)**

OR

18. (a) In client server architecture, what do you mean by Availability, Reliability, Serviceability and Security? Explain with examples **(10)**

(b) How remote systems management security is ensured in a Client/Server application. **(4)**

19. (a) What is the future of Client/Server computing in the following technologies (i) Electronic Document Management. (ii) Full Text Retrieval. (iii) Geographic Information System. **(10)**

(b) Discuss the role of web browser for providing web service in Client/Server environment. **(4)**

OR

20. (a) Explain end-to-end working of Client/Server web model. **(10)**

(b) Explain the architecture of Transformational system. **(4)**

Teaching Plan

Sl No	Contents	No. of Lecture Hours (35)
	Module- 1(Introduction) (7 hours)	
1.1	Driving forces behind Client/ Server	1 hour
1.2	Client Server development tools	1 hour
1.3	Development of client/server systems	1 hour
1.4	Client/Server security, Organizational Expectations	1 hour
1.5	Improving performance of client/server applications	1 hour
1.6	Single system image, Downsizing and Rightsizing	1 hour
1.7	Advantages and Applications of client server computing	1 hour
	Module- 2(Client/Server Application Components) (8 hours)	
2.1	Classification of Client/Server Systems	1 hour
2.2	Open System Standards	1 hour
2.3	Two-Tier Computing	1 hour
2.4	Three-Tier Computing, Middleware	1 hour
2.5	Principles behind Client/Server Systems	1 hour
2.6	Client/Server Topologies	1 hour
2.7	Existing Client/Server Architecture	1 hour
2.8	Architecture for Business Information System	1 hour
	Module- 3(Client/Server Network) (6 hours)	
3.1	The client: Services, Request for services, RPC, Windows services, Print services	1 hour
3.2	Remote boot services, Utility Services & Other Services	1 hour
3.3	Dynamic Data Exchange (DDE), Object Linking and Embedding (OLE)	1 hour
3.4	Common Object Request Broker Architecture (CORBA)	1 hour

3.5	The server: Detailed server functionality, the network operating system	1 hour
3.6	Available platforms, the server operating system	1 hour
Module- 4(Client Server Systems Development) (7 hours)		
4.1	Services and Support, System administration	1 hour
4.2	Availability, Reliability, Scalability, Observability, Agility Serviceability, Software Distribution, Performance	1 hour
4.3	Network management, Remote Systems Management, RDP,Telnet,SSH	1 hour
4.4	Security ,LAN and Network Management issues	1 hour
4.4	Training, Connectivity, Communication interface technology	1 hour
4.5	Interposes communication, wide area network technologies	1 hour
4.6	Network Acquisition, PC-level processing unit, x-terminals, server Hardware	1 hour
Module -5(Client/Server Technology And Web Services) (7 hours)		
5.1	Web Services History , Web Server Technology , Web Server	1 hour
5.2	Web Server Communication , Role of Java for Client/Server on Web	1 hour
5.3	Web Services , MicroServices, APIs, API Gateway, Authentication of users/clients	1 hour
5.4	Tokens/Keys for Authentication ,Service Mesh, Message Queues	1 hour
5.5	SaaS, Web Sockets ,Client/Server Technology and Web Applications	1 hour
5.6	Balanced Computing and the Server's Changing Role ,Thin client computing , Computing models, Computing Environment	1 hour
5.7	Future of client/ server Computing Enabling Technologies, Transformational system	1 hour

CST436	PARALLEL COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learners to understand basic and advanced concepts of parallel computing. It covers Principles of Parallel Algorithm Design, Communication operations, Programming Using the Message Passing Paradigm, Programming Shared Address Space Platforms Thread Basics, and GPU Programming. This course enables a learner to design solutions to complex real world problems using parallel computing paradigms including thread parallelism, shared memory program, message passing interfaces, and vector processing.

Prerequisite: Knowledge in Computer Organization and Architecture.

Course Outcomes: After the completion of the course the students will be able to

CO1	Summarize the key parallel computational models **(Cognitive Knowledge Level : Understand)**
CO2	Appreciate and apply parallel and distributed algorithms in problem Solving **(Cognitive Knowledge Level :Apply)**
CO3	Appreciate the communication models for parallel algorithm development (**Cognitive Knowledge Level : Understand**)
CO4	Develop parallel algorithms using message passing paradigm **(Cognitive Knowledge Level : Apply)**
CO5	Formulate parallel algorithms for shared memory architectures. **(Cognitive Knowledge Level : Apply)**
CO6	Demonstrate the fundamental skills of heterogeneous computing with GPUs**(Cognitive Knowledge Level : Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓									✓
CO3	✓	✓										✓
CO4	✓	✓	✓	✓	✓							✓

CO5	✓	✓	✓	✓	✓							✓
CO6	✓	✓	✓	✓	✓							✓

Abstract POs Defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and teamwork
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Blooms Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Percentage)	Test 2 (Percentage)	
Remember	30	20	20
Understand	50	40	40
Apply	20	40	40

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 Hours**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus, and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have a maximum 2 subdivisions and carries 14 marks.

Syllabus

Module- 1 (Principles of Parallel Algorithm Design)

Basic Introduction to Parallel Processing platforms. Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models.

Module- 2 (Communication Operations)

Basic Communication Operations **-** One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations, Scatter and Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operation

Module-3 (Programming Using the Message Passing Paradigm)

Principles of Message-Passing Programming, The Building Blocks: Send and Receive Operations, MPI: The Message Passing Interface, Overlapping Communication with Computation, Collective Communication and Computation Operations, Groups and Communicators.

Module 4 (Programming Shared Address Space Platforms Thread Basics)

Thread Basics, Why Threads? The POSIX Thread Application Programme Interface, Synchronization Primitives in POSIX, Controlling Thread and Synchronization Attributes, Thread Cancellation, Composite Synchronization Constructs, OpenMP: a Standard for Directive Based Parallel Programming, Specifying Concurrent Tasks in OpenMP, Synchronization Constructs in OpenMP, Data Handling in OpenMP, OpenMP Library Functions, OpenMP Applications: Parallel algorithm development for Matrix multiplication

Module 5 (GPU Programming)

Heterogeneous Parallel Computing, Architecture of a Modern GPU, Speeding up Real Applications, Data parallel computing, CUDA C Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading, Kernel Launch, CUDA Thread Organization, Mapping Threads to Multidimensional Data, Synchronization and Transparent Scalability, Resource Assignment, Querying Device Properties, Thread Scheduling and Latency Tolerance, Importance of Memory Access Efficiency, Cuda Memory Types, Tiling for Reduced Memory Traffic, Tiled Matrix Multiplication Kernel, Boundary Checks

Text Books

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, 2nd Ed, Addison-Wesley, 2003
2. David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, 3rd Ed., Morgan Kaufman, 2016.

References

1. Steven Brawer, Introduction to Parallel Computing, Academic Press, (1989)
2. Barbara Chapman, Gabriele Jost, Ruud van der Pas, Using OpenMP: Portable Shared Memory Paralwlel Programming , MIT Press, 2008.
3. William Gropp, Ewing Lusk, Anthony Skjellum Using MPI: Portable Parallel Programming with the Message-Passing Interface, 3rd Ed, MIT Press, 2014.
4. Thomas Rauber, Gudula Rünger, Parallel Programming for Multicore and Cluster Systems, Springer, 2010

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Differentiate between static and dynamic task mapping
2. Explain partitioning of data with an example

Course Outcome 2 (CO2):

1. Explain the handshaking sequence of Blocking Non-Buffered Send/Receive operation with a neat diagram.
2. In the algorithm, assume a decomposition such that each execution of Line 7 is a task. Draw a task-dependency graph and a task-interaction graph.

```
1. procedure FFT_like_pattern(A, n)
2. begin
3.     m := $\log_2 n$;
4.     for j := 0 to m - 1 do
5.         k := 2j;
6.         for i := 0 to n - 1 do
7.             A[i] := A[i] + A[i XOR 2j];
8.     end  // for
9. end    // FFT_like_pattern
```

Course Outcome 3 (CO3):

1. Write a procedure for performing all-to-all reduction on a mesh
2. Give a hypercube algorithm to compute prefix sums of n numbers if *p* is the number of nodes and n/p is an integer greater than 1. Assuming that it takes time t_{add} to add two numbers and time ts to send a message of unit length between two directly-connected nodes, give an exact expression for the total time taken by the algorithm.

Course Outcome 4(CO4):

1. Show how the two-dimensional matrix-vector multiplication program needs to be changed so that it will work correctly for a matrix of size *n x m* on a *q x r* process grid
2. One of the advantages of non-blocking communication operations is that they allow the transmission of the data to be done concurrently with computations. Discuss the type of restructuring that needs to be performed on a program to allow for the maximal overlap of computation with communication. Is the sending process in a better position to benefit from this overlap than the receiving process

Course Outcome 5(CO5):

1. Implement a multi-access threaded queue with multiple threads inserting and multiple threads extracting from the queue. Use mutex-locks to synchronize access to the queue. Document the time for 1000 insertions and 1000 extractions each by 64 insertion threads (producers) and 64 extraction threads (consumers).
2. Implement a producer-consumer framework in OpenMP using sections to create a single producer task and a single consumer task. Ensure appropriate synchronization using locks.

Course Outcome 6 (CO6):

1. Consider a hypothetical block with 8 threads executing a section of code before reaching a barrier. The threads require the following amount of time (in microseconds) to execute the sections: 2.0, 2.3, 3.0, 2.8, 2.4, 1.9, 2.6, and 2.9 and to spend the rest of their time waiting for the barrier. What percentage of the total execution time of the thread is spent waiting for the barrier?
2. Write and explain the CUDA program for vector addition.

Model Question Paper

QP CODE: PAGES :3

Reg No: _______________

Name: _________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST436

Course Name: PARALLEL COMPUTING

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Explain partitioning of data with an example

2. Which are the characteristics of tasks influencing the selection of mapping scheme?

3. Describe the scatter - gather communication.

4. Explain the Circular Shift operation.

5. Explain the handshaking sequence of Blocking Non-Buffered Send/Receive operation with a neat diagram.

6. Describe the six fundamental routines of MPI.

7. Explain thread cancellation.

8. Explain how concurrent tasks are specified in openMP

9. Explain the architecture of modern GPU with a diagram.

10. Describe how the data transfer between GPU device and the host memories are managed. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Describe recursive decomposition with an example. **(8)**

(b) Compare various parallel algorithm models **(6)**

OR

12. (a) Differentiate between static and dynamic task mapping **(8)**

(b) In the algorithm, assume a decomposition such that each execution of Line 7 is a task. Draw a task-dependency graph and a task-interaction graph. **(6)**

```
1. procedure FFT_like_pattern(A, n)
2. begin
3. m := log2 n;
4. for j := 0 to m - 1 do
5. k := 2j;
6. for i := 0 to n - 1 do
7. A[i] := A[i] + A[i XOR 2j];
8. end  // for
9. end    // FFT_like_pattern
```

13. (a) Illustrate the All-to-All Broadcast and Reduction with an example **(8)**

(b) Explain any three techniques to improve the speed of communication operations **(6)**

OR

14. (a) Explain the One-to-All Broadcast and All-to-One Reduction with an example **(8)**

(b) Explain the Ring and Mesh techniques of All-to-All Personalized communication. **(6)**

15. (a) Explain Collective Communication and Computation Operations in MPI **(9)**

(b) Show the impact of finite buffers in message passing. **(5)**

OR

16. (a) Write algorithm for Collective Communication and Computation Operations **(9)**

using MPI.

(b) How is deadlock avoided in *MPI_Send* and *MPI_Recv* **(5)**

17. (a) Explain how mutual exclusion for shared variables are accomplished in threads. **(6)**

(b) Explain the nesting of parallel directives with a suitable example. **(8)**

OR

18. (a) Explain the compilation operations of an example openMP program along with its *pThread* translations. **(4)**

(b) Explain the parallel matrix multiplication using openMP **(10)**

19. (a) Describe the CUDA Kernel functions. **(6)**

(b) How is synchronization between CUDA threads achieved? **(8)**

OR

20. (a) Explain the two-level hierarchical organization of CUDA threads. **(10)**

(b) Write and explain the CUDA program for vector addition. **(4)**

TEACHING PLAN

No	Contents	No of Lecture Hrs (37)
Module – 1 (Basic Introduction to Parallel Processing) (TB-1, Ch. 3) (7 hrs)		
1.1	Basic Introduction to Parallel Processing platforms. Preliminaries	1
1.2	Decomposition Techniques – Recursive, Data	1
1.3	Decomposition Techniques – Exploratory, Speculative, Hybrid	1
1.4	Characteristics of Tasks and Interactions	1
1.5	Mapping Techniques for Load Balancing -Static	1
1.6	Mapping Techniques for Load Balancing - Dynamic	1
1.7	Methods for Containing Interaction Overheads, Parallel Algorithm Models.	1
Module- 2 (Basic Communication Operations) (TB-1, Ch. 4) (6hrs)		
2.1	One-to-All Broadcast and All-to-One Reduction	1
2.2	All-to-All Broadcast and Reduction	1
2.3	All-Reduce and Prefix-Sum Operations, Scallter Gather	1
2.4	All-to-All Personalized Communication	1
2.5	Circular Shift	1
2.6	Improving the Speed of Some Communication Operation	1
Module- 3 (Programming Using the Message Passing Paradigm) (TB-1, Ch. 6) (7 hrs)		
3.1	Principles of Message-Passing Programming, The Building Blocks: Send and Receive Operations	1
3.2	MPI: The Message Passing Interface	1
3.3	MPI: The Message Passing Interface : Illustration	1

3.4	Overlapping Communication with Computation	1
3.5	Overlapping Communication with Computation : Illustration	1
3.6	Collective Communication and Computation Operations	1
3.7	Collective Communication and Computation Operations : Illustration	1
Module 4 (Programming Shared Address Space Platforms) (TB-1, Ch. 7, 8) (8hrs)		
4.1	Thread Basics, Why Threads? The POSIX Thread API	1
4.2	Synchronization Primitives in POSIX	1
4.3	Controlling Thread and Synchronization Attributes	1
4.4	Thread Cancellation, Composite Synchronization Constructs,	1
4.5	OpenMP: a Standard for Directive Based Parallel Programming	1
4.6	Specifying Concurrent Tasks in OpenMP, Synchronization Constructs in OpenMP	1
4.7	Data Handling in OpenMP, OpenMP Library Functions	1
4.8	OpenMP Applications: Parallel algorithm development for Matrix multiplication	1
Module 5 (GPU Programming) (TB-2, Ch. 1, 2) (9 hrs)		
5.1	Heterogeneous Parallel Computing, Architecture of a Modern GPU, Speeding up Real Applications	1
5.2	Data parallel computing – CUDA C Program Structure	1
5.3	Vector Addition Kernel, Device Global Memory and Data Transfer	1
5.4	Kernel Functions and Threading, Kernel Launch	1

5.5	CUDA Thread Organization, Mapping Threads to Multidimensional Data	1
5.6	Synchronization and Transparent Scalability, Resource Assignment, Querying Device Properties, Thread Scheduling and Latency Tolerance	1
5.7	Importance of Memory Access Efficiency, Cuda Memory Types	1
5.8	Tiling for Reduced Memory Traffic	1
5.9	Tiled Matrix Multiplication Kernel, Boundary Checks	1

CST446	DATA COMPRESSION TECHNIQUES	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learners to understand compression techniques on text, image, audio and video data. It covers lossy &lossless compression, RLE, JPEG, MPEG and its variants. This course enables the students to develop and implement compression algorithms on different domains.

Prerequisite: Knowledge of probability theory, computation on matrices, basic topics in data structures, storage and efficiency

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Describe the fundamental principles of data compression**(Cognitive Knowledge level: Understand)**
CO2	Make use of statistical and dictionary based compression techniques for various applications **(Cognitive Knowledge level: Apply)**
CO3	Illustrate various image compression standards. **(Cognitive Knowledge level: Apply)**
CO4	Summarize video compression mechanisms to reduce the redundancy in video.**(Cognitive Knowledge level: Understand)**
CO5	Use the fundamental properties of digital audio to compress audio data.**(Cognitive Knowledge level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											✓
CO2	✓	✓	✓		✓							✓
CO3	✓	✓	✓		✓							✓
CO4	✓											✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of SeriesTests1& 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Modelling and types of compression)) 1

Introduction to Compression Techniques- Lossy compression & Lossless compression, Measures of Performance, Modeling and coding. Mathematical modelling for Lossless and lossy compression - Physical models and probability models.

Module – 2 (Basic Compression Methods)

Basic Compression Technique- Run length encoding, RLE Text compression. Statistical Methods- Prefix Codes, Binary Huffman coding, non-binary Huffman Algorithms, Arithmetic Coding.

Module - 3 (Text & Image Compression)

Dictionary based Coding- LZ77, LZ78 and LZW compression.Image Compression- Image standards, JPEG image Compression- Baseline JPEG, JPEG-LS.

Module - 4 (Video Compression)

Video Compression- Analog video, Digital Video, Motion Compensation. MPEG standards- MPEG 1, MPEG 4

Module - 5 (Audio Compression)

Audio Compression- Basics of Digital Audio, Basic Audio Compression Techniques, MPEG Audio Compression-Layer 1 coding, Layer 2 coding and Layer 3 coding.

Text Book

1. David Solomon, Data compression: the complete reference, 4/e, Springer, January 2007
2. Khalid Sayood, Introduction to data compression, Morgan Kaufmann Publishers,2003.

References

1) Stephen Welstead, Fractal and wavelet Image Compression techniques, PHI, 1999.
2) Sleinreitz, Multimedia System, Addison Wesley.
3) Mark Nelson and Jean-loup Gailly, The Data Compression Book, M&T Books.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Discuss different types of compression performance metrics
2. Explain mathematical model for lossless compression

Course Outcome 2 (CO2):

1. Explain RLE based text compression and identify a example with compression ratio of 2.
2. Given the eight symbols A, B, C, D, E, F, G, and H with probabilities 1/30, 1/30, 1/30, 2/30, 3/30, 5/30, 5/30, and 12/30, draw three different Huffman trees with heights 5 and 6 for these symbols and calculate the average code size for each tree.

Course Outcome 3 (CO3):

1. Differentiate the LZ77 and LZ78 performance with the input given as 'sirsideastmaneasilyteasesseasickseals'
2. Explain why the continuous-tone images is required for JPEG and the main steps used in image compression.

Course Outcome 4 (CO4):

1. Briefly explain MPEG-4 video compression standard
2. How H.261 video compression is completed.

Course Outcome 5 (CO5):

1. Explain critical bands, thresholding and masking related to audio compression
2. Explain the working of -law encoder and decoder with an example

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 2**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST446

Course Name: Data Compression Techniques

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Specify different quantities used to measure the performance of a data compression technique
2. Explain mathematical model for lossless compression
3. State and prove Kraft-McMillan inequality
4. Compare Huffman and Arithmetic coding
5. Describe LZ77 approach of encoding a string with the help of an example
6. Compare and contrast JPEG and JPEG-LS differences in working.
7. Discuss different components of video
8. Identify the advantage of MPEG-4 over MPEG
9. Explain critical bands, thresholding and masking related to audio compression
10. Explain the working of -law encoder and decoder with an example

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain mathematical model for lossy compression and lossless compression **(10)**

(b) Define compression ratio with an example **(4)**

OR

12. (a) Discuss any probability model and identify the shortcoming of the solution. **(7)**

(b) Identify the mathematical preliminaries for Lossless Compression **(7)**

13. (a) With a help of flowchart discuss the RLE text compression for text data given below **(10)**

‘*ABBBBBBBBBCDEEEEF’*

(b) calculate the compression ratio for the example while taking repetitions = 4 **(4)**

OR

14. (a) Illustrate with a example why Huffman coding is preferred than Shannon Fano Algorithm for compression **(10)**

(b) How Huffman coding is handling the unpredictability of input data stream **(4)**

15. (a) Explain in detail the working of LZ78 with example and dictionary Tree **(10)**

(b) Illustrate with example, how the compression factor LZW differ from the LZ78 **(4)**

OR

16. (a) How quantization and coding helps in compression and their role in JPEG. **(6)**

(b) With the help of the given example illustrate the compression ratio of JPEG and JPEG-LS **(8)**

17. (a) With the help of equations discuss Composite and Components Video **(7)**

(b) Differentiate the major changes in MPEG - 2 and MPEG-4 Video **(7)**

OR

18. (a) Describe in details about functionalities for MPEG-4 **(8)**

(b) How Motion Compensation help in video compression **(6)**

19. (a) How The Human Auditory System limitations can be taken in audio **(7)**

compressions

(b) Discuss the complexity of Layer III compared to others in MPEG Audio Coding **(7)**

OR

20. (a) Discuss Format of Compressed Data and encoding in layer I and II **(9)**

(b) Differentiate Spectral and Temporal Masking **(5)**

TEACHING PLAN

No	**Contents**	**No of Lecture Hrs (36 Hours)**
Module – 1 (Modelling and types of compression) (7 hrs)		
1.1	Introduction to Compression Techniques- Lossy compression & Lossless compression, Measures of Performance	2
1.2	Modelling and coding.	1
1.3	Physical model for lossless compression	1
1.4	Physical model for lossy compression	1
1.5	Probability model for lossless compression	1
1.6	Probability model for lossly compression	1
Module - 2 (Basic Compression Methods) (8 hrs)		
2.1	Run length encoding, RLE Text compression	1
2.2	Statistical methods-Prefix Codes	1
2.3	Binary Huffman coding	1
2.4	Illustration of Binary Huffman coding	1
2.5	Non-binary Huffman Algorithms	1
2.6	Arithmetic Coding algorithm	1
2.7	Illustration of Arithmetic Coding algorithm	2

Module - 3 (Text & Image Compression) (8 hrs)		
3.1	LZ77 compression	2
3.2	LZ78 Compression	1
3.3	LZW Compression	1
3.4	Basics of Image compression and Image standards	1
3.5	Baseline JPEG Image compression	1
3.6	JPEG-LS Image compression	1
Module - 4 (Video Compression) (7 hrs)		
4.1	Basics of Video Compression- Analog video and Digital Video.	2
4.2	Motion Compensation	1
4.3	MPEG-1 standard and Video Syntax	1
4.4	MPEG-1 Pel Reconstruction	1
4.5	MPEG-4 standard	1
4.6	Functionalities for MPEG-4	1
Module - 5 (Audio Compression) (6 hrs)		
5.1	Basics of Audio Compression, Digital Audio	1
5.2	Basic Audio Compression Techniques	1
5.3	MPEG Audio Compression basics- Frequency Domain Coding	1
5.4	Encoding: Layers I and II	1
5.5	Encoding: Layer II -Psychoacoustic Models	1
5.6	Psychoacoustic Models - Encoding: Layer III	1

CST466	DATA MINING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learner to understand the concepts of data mining and data warehousing. It covers the key processes of data mining, data preprocessing techniques, fundamentals and advanced concepts of classification, clustering, association rule mining, web mining and text mining. It enables the learners to develop new data mining algorithms and apply the existing algorithms in real-world scenarios.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Employ the key process of data mining and data warehousing concepts in application domains. (**Cognitive Knowledge Level: Understand**)
CO2	Make use of appropriate preprocessing techniques to convert raw data into suitable format for practical data mining tasks (**Cognitive Knowledge Level**: **Apply**)
CO3	Illustrate the use of classification and clustering algorithms in various application domains (**Cognitive Knowledge Level: Apply**)
CO4	Comprehend the use of association rule mining techniques. (**Cognitive Knowledge Level: Apply)**
CO5	Explain advanced data mining concepts and their applications in emerging domains (**Cognitive Knowledge Level: Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓	✓							✓

CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓										✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**20**	**20**	**20**
Understand	**30**	**30**	**30**
Apply	**50**	**50**	**50**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Test(Average of Internal Test1&2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the seven questions, a student should answer any five.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have a maximum 2 subdivisions and carries 14 marks.

Syllabus

Module – 1 (Introduction to Data Mining and Data Warehousing)

Data warehouse-Differences between Operational Database Systems and Data Warehouses, Multidimensional data model- Warehouse schema, OLAP Operations, Data Warehouse Architecture, Data Warehousing to Data Mining, Data Mining Concepts and Applications, Knowledge Discovery in Database Vs Data mining, Architecture of typical data mining system, Data Mining Functionalities, Data Mining Issues.

Module - 2 (Data Preprocessing)

Data Preprocessing-Need of data preprocessing, Data Cleaning- Missing values, Noisy data, Data Integration and Transformation, Data Reduction-Data cube aggregation, Attribute subset selection, Dimensionality reduction, Numerosity reduction, Discretization and concept hierarchy generation.

Module - 3 (Advanced classification and Cluster analysis)

Classification- Introduction, Decision tree construction principle, Splitting indices -Information Gain, Gini indexDecision tree construction algorithms-ID3, Decision tree construction with presorting-SLIQ, Classification Accuracy-Precision, Recall.

Introduction to clustering-Clustering Paradigms, Partitioning Algorithm- PAM, Hierarchical Clustering-DBSCAN, Categorical Clustering-ROCK

Module 4: (Association Rule Analysis)

Association Rules-Introduction, Methods to discover Association rules, Apriori(Level-wise algorithm), Partition Algorithm, Pincer Search Algorithm, Dynamic Itemset Counting Algorithm, FP-tree Growth Algorithm.

Module 5 (Advanced Data Mining Techniques)

Web Mining - Web Content Mining, Web Structure Mining- Page Rank, Clever, Web Usage Mining- Preprocessing, Data structures, Pattern Discovery, Pattern Analysis. Text Mining-Text Data Analysis and information Retrieval, Basic measures for Text retrieval, Text Retrieval methods, Text Indexing Techniques, Query Processing Techniques.

Text Books

1. Dunham M H, “Data Mining: Introductory and Advanced Topics”, Pearson Education, New Delhi, 2003.
2. Arun K Pujari, “Data Mining Techniques”, Universities Press Private Limited,2008.
3. Jaiwei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, Elsevier, 2006

Reference Books

1. M Sudeep Elayidom, “Data Mining and Warehousing”, 1st Edition, 2015, Cengage Learning India Pvt. Ltd.
2. MehmedKantardzic, “Data Mining Concepts, Methods and Algorithms”, John Wiley and Sons, USA, 2003.
3. Pang-Ning Tan and Michael Steinbach, “Introduction to Data Mining”, Addison Wesley, 2006.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. (a) Explain the OLAP operations in a multidimensional model.
   (b) Compare the techniques used in ROLAP, MOLAP and HOLAP
2. Explain the various data mining issues with respect to mining methodology, user interaction and diversity of data types.
3. Suppose that a data warehouse consists of the three dimensions time, doctor, and patient, and the two measures count and charge, where charge is the fee that a doctor charges a patient for a visit.
   a) Draw star and snowflake schema diagrams for the data warehouse.
   b) Starting with the base cuboid [day; doctor; patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2004?

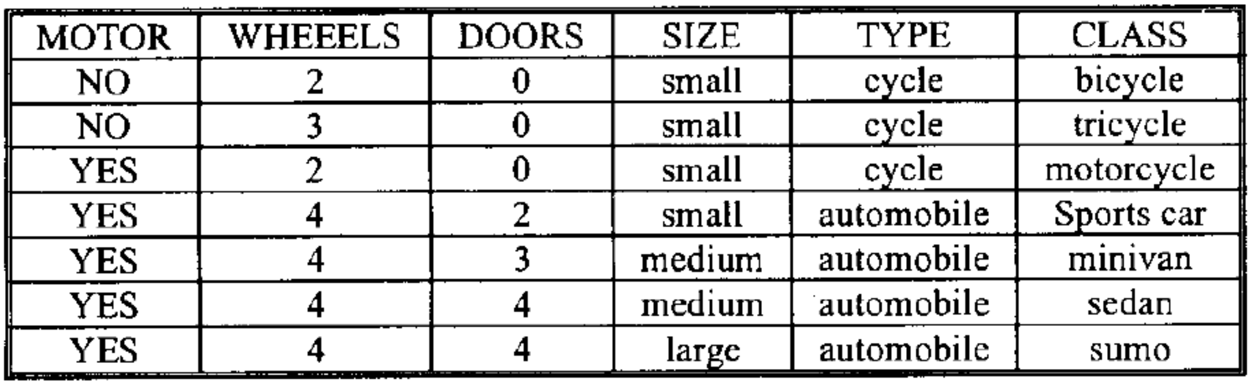
Course Outcome 2 (CO2):

1. Use the methods below to normalize the following group of data:100, 200, 300, 400,550, 600, 680, 850, 1000
   (a) min-max normalization by setting min = 0 and max = 1
   (b) z-score normalization
   (c) Normalization by decimal scaling

   Comment on which method you would prefer to use for the given data, givingreasons as to why.
2. Identify a suitable dataset from any available resources and apply different preprocessing steps that you have learned. Observe and analyze the output obtained. (Assignment)

Course Outcome 3 (CO3):

1. Illustrate the working of ID3 algorithm with the following example



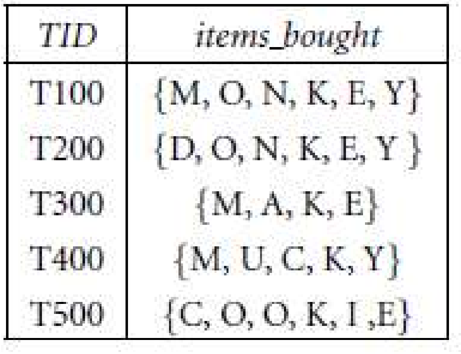
MOTOR	WHEEELS	DOORS	SIZE	TYPE	CLASS
NO	2	0	small	cycle	bicycle
NO	3	0	small	cycle	tricycle
YES	2	0	small	cycle	motorcycle
YES	4	2	small	automobile	Sports car
YES	4	3	medium	automobile	minivan
YES	4	4	medium	automobile	sedan
YES	4	4	large	automobile	sumo

2. Illustrate the working of K medoid algorithm for the given dataset. A1=(3,9), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

3. Take a suitable dataset from available resources and apply all the classification and clustering algorithms that you have studied on original and preprocessed datasets. Analyze the performance variation in terms of different quality metrics. Give a detailed report based on the analysis. (Assignment)

Course Outcome 4 (CO4):

1. A database has five transactions. Let min sup = 60% and min con f = 80%.



TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y }
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I ,E}

a) Find all frequent item sets using Apriori and FP-growth, respectively. Compare the efficiency of the two mining processes.

b) List all of the strong association rules (with support s and confidence c) matching the following metarule, where X is a variable representing customers, and$item_i$denotes variables representing items (e.g., “A”, “B”, etc.)

$$\forall x \in transaction, buys(X, item_1) \wedge \ buys(X, item_2) \Rightarrow buys\ (X, item_3) \quad [s, c]$$

2. Identify and list some scenarios in which association rule mining can be used, and then use at least two appropriate association rule mining techniques in one of the two scenarios. (Assignment)

Course Outcome 5 (CO5):

1. Consider an e-mail database that stores a large number of electronic mail (e-mail) messages. It can be viewed as a semi structured database consisting mainly of text data. Discuss the following.
   a. How can such an e-mail database be structured so as to facilitate multidimensional search, such as by sender, by receiver, by subject, and by time?
   b. What can be mined from such an e-mail database?
   c. Suppose you have roughly classified a set of your previous e-mail messages as junk, unimportant, normal, or important. Describe how a data mining system may take this as the training set to automatically classify new e-mail messages or unclassified ones.
2. Precision and recall are two essential quality measures of an information retrieval system.
   (a) Explain why it is the usual practice to trade one measure for the other.
   (b) Explain why the F-score is a good measure for this purpose.

(c) Illustrate the methods that may effectively improve the F-score in an information retrieval system.

3. Explain HITS algorithm with an example.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST466

Course Name: Data Mining

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

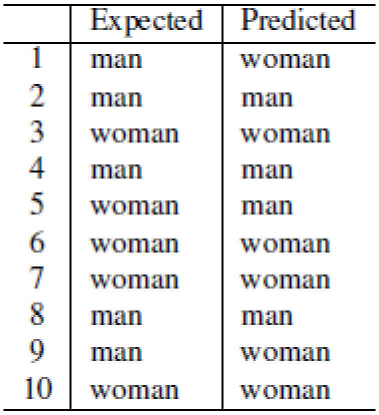
1. Differentiate between OLTP and OLAP.

2. Compare the techniques of ROLAP, MOLAP and HOLAP

3. Explain Concept hierarchy with an example.

4. Explain heuristic methods of attribute subset selection techniques.

5. Consider a two-class classification problem of predicting whether a photograph contains a man or a woman. Suppose we have a test dataset of 10 records with expected outcomes and a set of predictions from our classification algorithm.



	Expected	Predicted
1	man	woman
2	man	man
3	woman	woman
4	man	man
5	woman	man
6	woman	woman
7	woman	woman
8	man	man
9	man	woman
10	woman	woman

Calculate precision, recall of the data.

6. Given two objects represented by the tuples (22,1,42,10) and (20,0, 36,8). Compute the Euclideanand Manhattan distance between the two objects.

7. The pincer search algorithm is a bi-directional search, whereas the level wise algorithm is a unidirectional search. Express your opinion about the statement.

8. Define support, confidence and frequent set in association data mining context.

9. Distinguish between focused crawling and regular crawling.

10. Describe any two-text retrieval indexing techniques. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Suppose a data warehouse consists of three measures: customer, account and branch and two measures count (number of customers in the branch) and balance. Draw the schema diagram using snowflake schema and star schema. **(7)**

(b) Explain three- tier data warehouse architecture with a neat diagram. **(7)**

OR

12 (a) Illustrate different OLAP operations in multidimensional data model **(7)**

(b) Describe different issues in data mining **(7)**

13 (a) Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70. **(8)**

(a) Use min-max normalization to transform the value 35 for age onto

the
range [0-1].

(b) Use z-score normalization to transform the value 35 for age, where the standard deviation of age is 12.94 years.

(c) Use normalization by decimal scaling to transform the value 35 for age.

(d) Use smoothing by bin means to smooth the above data, using a bin depth of 3. Illustrate your steps. Comment on the effect of this technique for the given data.

(b) With proper illustration, explain how PCA can be used for dimensionality reduction? Explain **(6)**

OR

14 (a) Suppose a group of 12 sales price records has been sorted as follows: 5, 10, 11, 13, 15, 35, 50, 55, 72, 92, 204, 215. Sketch examples of each of the following sampling techniques: SRSWOR, SRSWR, cluster sampling, stratified sampling. Use samples of size 5 and the strata "youth," "middle-aged," and "senior." **(8)**

(b) Partition the above data into three bins by each of the following methods: **(6)**
(i) equal-frequency (equi-depth) partitioning
(ii) equal-width partitioning

15 (a) Explain the concept of a cluster as used in ROCK. Illustrate with examples **(9)**

(b) Consider the following dataset for a binary classification problem. **(5)**

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	-
T	T	+
F	F	-
F	F	-
F	F	-
T	T	-
T	F	-

Calculate the gain in Gini index when splitting on A and B respectively. Which attribute would the decision tree induction algorithm choose?

OR

16 (a) For a sunburn dataset given below, find the first splitting attribute for the decision tree by using the ID3 algorithm. **(10)**

Name	**Hair**	**Height**	**Weight**	**Lotion**	**Class**
Sarah	Blonde	Average	Light	No	Sunburn
Dana	Blonde	Tall	Average	Yes	None
Alex	Brown	Tall	Average	Yes	None
Annie	Blonde	Short	Average	No	Sunburn
Emily	Red	Average	Heavy	No	Sunburn
Pete	Brown	Tall	Heavy	No	None
John	Brown	Average	Heavy	No	None
Katie	Blonde	Short	Light	Yes	None

(b) Explain the working of SLIQ algorithm. **(4)**

17 (a) Illustrate the working of Pincer Search Algorithm with an example. **(7)**

(b) Describe the working of dynamic itemset counting technique? Specify when to move an itemset from dashed structures to solid structures? **(7)**

OR

18 (a) A database has six transactions. Let min_sup be 60% and min_conf be 80%. **(9)**

TID	*items_bought*
T1	I1, I2, I3
T2	I2, I3, I4
T3	I4, I5
T4	I1, I2, I4
T5	I1, I2, I3, I5
T6	I1, I2, I3, I4

Find frequent itemsets using FP Growth algorithm and generate strong association rules from a three item dataset.

(b) Write partitioning algorithm for finding large itemset and compare its efficiency with apriori algorithm **(5)**

19 (a) Describe web content mining techniques. **(7)**

(b) Write an algorithm to find maximal frequent forward sequences to mine log traversal patterns. Illustrate the working of this algorithm. **(7)**

OR

20 (a) Explain how web structure mining is different from web usage mining and web content mining? Write a CLEVER algorithm for web structure mining. **(7)**

(b) Describe different Text retrieval methods. Explain the relationship between text mining and information retrieval and information extraction. **(7)**

Teaching Plan

No	Contents	No. of lecture hours (36 Hrs)
Module 1(Introduction to Data Mining and Data Warehousing) (Text3) (6 hours)		
1.1	Data warehouse-Differences between Operational Database Systems and Data Warehouses, Multidimensional data model- Warehouse schema	1
1.2	OLAP Operations	1
1.3	DataWarehouse Architecture, Data Warehousing to Data Mining	1
1.4	Datamining Concepts and Applications, Knowledge Discovery in Database Vs Data mining	1
1.5	Architecture of typical data mining system,Data Mining Functionalities	1
1.6	Data Mining Functionalities, Data Mining Issues	1
Module 2(Data Preprocessing) (6 hours) (Text3)		
2.1	Data Preprocessing: Need of Data Preprocessing, Data Cleaning- Missing values, Noisy data.	1
2.2	Data integration	1
2.3	Data transformation	1
2.4	Data Reduction-Data cube aggregation, Attribute subset selection	1
2.5	Data Reduction-Dimensionality reduction	1

2.6	Numerosity reduction, Discretization and concept hierarchy generation	1
Module 3(Advanced classification and Cluster analysis)(9 hours)(Text2,Text3)		
3.1	Classification- Introduction, Decision tree construction principle, Splitting indices-Information Gain, Gini index	1
3.2	Decision Tree- ID3	1
3.3	Decision Tree- ID3	1
3.4	Decision tree construction with presorting- SLIQ	1
3.5	Accuracy and error measures, evaluation	1
3.6	Introduction to clustering, Clustering Paradigms	1
3.7	Partitioning Algorithm- PAM	1
3.8	Hierarchical Clustering-DBSCAN	1
3.9	Categorical Clustering-ROCK	1
Module 4(Association Rule Analysis) (8 hours) (Text2,Text3,Text1)		
4.1	Association Rules: Introduction, Methods to discover association rules	1
4.2	A priori algorithm (Level-wise algorithm)	1
4.3	A priori algorithm (Level-wise algorithm)	1
4.4	Partition Algorithm	1
4.5	Pincer Search Algorithm	1
4.6	Pincer Search Algorithm	1
4.7	Dynamic Itemset Counting Algorithm	1
4.8	FP-tree Growth Algorithm	1
Module 5(Advanced Data Mining Techniques) (7 hours) (Text1, Text3		
5.1	Web Mining - Web Content Mining	1
5.2	Web Structure Mining- Page Rank	1
5.3	Web Structure Mining –Clever algorithm	1
5.4	Web Usage Mining- Preprocessing, Data structures	1

5.5	Web Usage Mining -Pattern Discovery, Pattern Analysis	1
5.6	Text Mining-Text Data Analysis and information Retrieval, Basic measures for Text retrieval	1
5.7	Text Retrieval methods, Text Indexing Techniques Query Processing Techniques	1

CST476	MOBILE COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: The course is designed with the view of preparing the engineering students capable of understanding the communication protocols, various architectures and security features used in mobile computing. This course covers basics of mobile computing, architecture of wireless transmission systems and next generation networks. This course enables the learners to acquire advanced concepts on wireless communication systems and mobile ad-hoc networks.

Prerequisite: A sound knowledge of computer networks.

Course Outcomes: After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Explain the various mobile computing applications, services, design considerations and architectures (**Cognitive knowledge: Understand**)
CO2	Describe the various technology trends for next generation cellular wireless networks and use the spreading concept on data transmission **(Cognitive knowledge: Apply)**
CO3	Summarize the architecture of various wireless LAN technologies **(Cognitive knowledge: Understand)**
CO4	Identify the functionalities of mobile network layer and transport layer **(Cognitive knowledge: Understand)**
CO5	Explain the features of Wireless Application Protocol **(Cognitive knowledge: Understand)**
CO6	Interpret the security issues in mobile computing and next generation technologies **(Cognitive knowledge: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓		✓							✓

CO5	✓	✓	✓									✓
CO6	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Test 1 (%)**	**Test 2 (%)**	**End Semester Examination (%)**
Remember	30	30	30
Understand	50	50	50
Apply	20	20	20
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	: **10 marks**
Continuous Assessment Test	: **25 marks**
Continuous Assessment Assignment	: **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Mobile Computing Basics)

Introduction to mobile computing – Functions, Middleware and Gateways, Application and services. Mobile computing architecture – Internet: The Ubiquitous network, Three-tier architecture for Mobile Computing, Design considerations for mobile computing.

Module – 2 (Wireless Transmission and Communication Systems)

Spread spectrum – Direct sequence, Frequency hopping. Medium Access Control – Space Division Multiple Access (SDMA), Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA). Satellite Systems – Basics, Applications, Geostationary Earth Orbit (GEO), Low Earth Orbit (LEO), Medium Earth Orbit (MEO), Routing, Localization, Handover. Telecommunication Systems - Global System for Mobile Communication (GSM) services, Architecture, Handover, Security.

Module – 3 (Wireless LANs)

Wireless LAN - Advantages, Design goals, Applications, Infrastructure Vs Ad-hoc mode, IEEE 802.11 System Architecture, Protocol Architecture, Physical layer, Medium Access Control layer, HIPERLAN-1, Bluetooth.

Module – 4 (Mobile Network and Transport Layer)

Mobile network layer – Mobile Internet Protocol (IP), Dynamic Host Configuration Protocol (DHCP), Mobile ad-hoc networks – Routing, Dynamic Source Routing (DSR), Destination Sequence Distance Vector (DSDV), Ad-hoc routing protocols. Mobile transport layer – Traditional Transmission Control Protocol (TCP), Improvements in Classical TCP. Wireless Application Protocol (WAP) - Architecture, Wireless Datagram Protocol (WDP), Wireless Transport Layer Security (WTLS), Wireless Transaction Protocol (WTP), Wireless Session Protocol (WSP).

Module – 5 (Mobile Security and Next Generation Networks)

Security issues in mobile computing - Information security, Security techniques and algorithms, Security models. Next generation networks - Orthogonal Frequency Division Multiplexing (OFDM), Wireless Asynchronous Transfer Mode (WATM), Multi Protocol Label Switching (MPLS), 10 pillars of 5G, Security for 5G communication.

Text Books

1. Asoke K. Talukder, Hasan Ahmad, Roopa R Yavagal, Mobile Computing Technology-Application and Service Creation, 2/e, McGraw Hill Education.
2. Jochen Schiller, Mobile Communications, Pearson Education Asia, 2008.
3. Jonathan Rodriguez, Fundamentals of 5G Mobile Networks, Wiley Publishers, 2015.

Reference Books

1. Raj Kamal, Mobile Computing, 2/e, Oxford University Press.
2. Andrew S. Tanenbaum, Computer Networks, PHI, 3/e, 2003
3. Theodore S. Rappaport, Wireless Communications Principles and Practice, 2/e, PHI, New Delhi, 2004.
4. Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Give examples for five mobile computing applications.
2. Identify any three differences between middleware and gateways.

Course Outcome 2 (CO2):

1. There are four stations sending data 1,1,1,0 respectively. Station 3 receives station 1's

data. Show the encoding, decoding and channel sharing mechanisms using CDMA.

2. Compare the influence of near/far effect and its countermeasures in TDMA and CDMA systems.

Course Outcome 3 (CO3):

1. Compare IEEE 802.11 and Bluetooth with respect to their ad-hoc capabilities.
2. Describe with neat sketch the major baseband states of a Bluetooth device.

Course Outcome 4 (CO4):

1. With the help of an example, show how routing process is handled by Dynamic Source Routing protocol.
2. Describe the major differences between AODV and the standard Distance Vector Routing algorithm. Why are extensions needed?
3. Simulate routing protocols using NS2.

Course Outcome 5 (CO5):

1. How does WAP push operation differ from pull operation?
2. With the help of a neat sketch explain the secure session establishment using WTLS.

Course Outcome 6 (CO6):

1. Explain the 3GPP security framework for mobile security.
2. Explain the features of policy-based security model.

Model Question Paper

QP CODE: **PAGES: 3**

Reg No:_______________
Name:_________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST476
Course Name : Mobile Computing

Max Marks: 100 **Duration: 3 Hours**

PART-A
(Answer All Questions. Each question carries 3 marks)

1. Explain the different types of middleware and gateways in the architecture of mobile computing.

2. Explain the major segments to support mobile computing functions.

3. Compare and contrast the satellite systems – GEO, LEO and MEO.

4. Assume all stations can hear all other stations. One station wants to transmit and senses the carrier idle. Why can a collision still occur after the start of transmission?

5. List any three advantages and disadvantages of wireless LANs.

6. Compare the features of infrastructure and ad-hoc networks.

7. Mention the basic purpose of DHCP. Also list the entities of DHCP.

8. Identify the benefits of location information for routing in ad-hoc networks.

9. List any six pillars of 5G.

10. How does multifactor security model provide security in a mobile network?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Describe the design considerations of three tier architecture of mobile computing. **(6)**

(b) Explain any four functions and applications of mobile computing. **(8)**

OR

12. (a) Explain Internet-Ubiquitous networks mentioning the significance and functions of core, edge and access network. **(6)**

(b) With the help of a neat sketch explain the three-tier architecture of mobile computing. **(8)**

13. (a) Check to see if the following set of chips can belong to an orthogonal system. **(6)**

[+1, +1, +1, +1] , [+1, -1, -1, +1] , [-1, +1, +1, -1] , [+1, -1, -1, +1]

(b) Summarize the routing and localization process in satellite systems. **(8)**

OR

14. (a) Apply Direct Sequence Spread Spectrum to the data 101 using Barker sequence 10110111000. Show the encoding and decoding steps. **(6)**

(b) Describe the system architecture of GSM networks. **(8)**

15. (a) How is Quality-of-Service provided in Bluetooth? **(6)**

(b) Explain the phases in Elimination-yield non-preemptive priority multiple access of HIPERLAN-1. **(8)**

OR

16. (a) Describe the protocol architecture of IEEE 802.11. **(6)**

(b) Explain the Medium Access Control management features provided in an IEEE 802.11 station. **(8)**

17. (a) With the help of an example, show the routing table creation using Destination Sequence Distance Vector Routing protocol in mobile ad-hoc networks. **(7)**

(b) Describe the router discovery methods used in mobile IP. **(7)**

OR

18. (a) Compare the features of flat routing and hierarchical routing. **(6)**

(b) List the entities of a mobile IP. With the help of an example, explain how packet delivery is done to and from a fixed node. **(8)**

19. (a) How is orthogonality helpful in Orthogonal Frequency Division Multiplexing? **(4)**

(b) Explain the functioning of Multi Protocol Label Switching technology. **(10)**

OR

20. (a) Describe the services of Wireless Asynchronous Transfer Mode. **(6)**

(b) Explain the different security models in mobile computing. **(8)**

TEACHING PLAN

No	**Contents**	**No.of Lecture Hrs (35 hrs)**
Module – 1 (Mobile Computing Basics) (6 hrs)		
1.1	Introduction to mobile computing – Functions	1
1.2	Middleware and Gateways	1
1.3	Application and services	1
1.4	Internet: The Ubiquitous network	1

1.5	Three-tier architecture for Mobile Computing	1
1.6	Design considerations for mobile computing	1
Module – 2 (Wireless Transmission and Communication Systems) (8 hrs)		
2.1	Direct sequence spread spectrum, Frequency hopping spread spectrum	1
2.2	Space Division Multiple Access (SDMA), Frequency Division Multiple Access (FDMA)	1
2.3	Time Division Multiple Access (TDMA)	1
2.4	Code Division Multiple Access (CDMA)	1
2.5	Satellite Systems Basics, Applications, Geostationary Earth Orbit (GEO), Low Earth Orbit (LEO), Medium Earth Orbit (MEO)	1
2.6	Routing, Localization, Handover	1
2.7	Global System for Mobile Communication (GSM) services, Architecture	1
2.8	Handover, Security	1
Module - 3 (Wireless LANs) (7 hrs)		
3.1	Wireless LAN - Advantages, Design goals, Applications, Infrastructure Vs Ad-hoc mode	1
3.2	IEEE 802.11 System Architecture	1
3.3	Protocol Architecture	1
3.4	Physical layer	1
3.5	Medium Access Control layer	1
3.6	HIPERLAN-1	1
3.7	Bluetooth	1
Module - 4 (Mobile Network and Transport Layer) (8 hrs)		
4.1	Mobile Internet Protocol (IP), Dynamic Host Configuration Protocol (DHCP)	1
4.2	Mobile ad-hoc networks – Routing, Dynamic Source Routing (DSR)	1

4.3	Destination Sequence Distance Vector (DSDV)	1
4.4	Ad-hoc routing protocols	1
4.5	Traditional Transmission Control Protocol (TCP), Improvements in Classical TCP	1
4.6	Wireless Application Protocol (WAP) – Architecture, Wireless Datagram Protocol (WDP)	1
4.7	Wireless Transport Layer Security (WTLS)	1
4.8	Wireless Transaction Protocol (WTP), Wireless Session Protocol (WSP)	1
Module - 5 (Mobile Security and Next Generation Networks) (6 hrs)		
5.1	Information security, Security techniques	1
5.2	Security algorithms, Security models	1
5.3	Introduction to Next generation networks, Orthogonal Frequency Division Multiplexing (OFDM)	1
5.4	Wireless Asynchronous Transfer Mode (WATM)	1
5.5	Multi Protocol Label Switching (MPLS)	1
5.6	10 pillars of 5G, Security for 5G communication	1

SEMESTER VIII

PROGRAM ELECTIVE V

CST418	HIGH PERFORMANCE COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learners to understand the different architectural features of high-end processors. This course discusses the Basics of high-end processors Architecture, Instruction-Level Parallelism, Data-Level Parallelism, Thread Level Parallelism, and GPU Architectures. This course enables the students to provide solutions to real-world problems making use of the capabilities of HPC systems.

Prerequisite: Basic knowledge in Computer System architecture, Microprocessors, Operating systems, and System software.

Course Outcomes: After the completion of the course the student will be able to

CO1	Describe different types of modern processing environments and parallel computing hardware (**Cognitive Knowledge Level: Understand**)
CO2	Summarize the concepts of Instruction Level Parallelism (**Cognitive Knowledge Level: Understand**)
CO3	Appreciate the idea of Data Level Parallelism (**Cognitive Knowledge Level: Apply**)
CO4	Demonstrate the concept of Thread Level Parallelism (**Cognitive Knowledge Level: Apply**)
CO5	Describe the basics of GPU architecture. (**Cognitive Knowledge Level: Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓										✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓									✓
C05	✓	✓										✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and teamwork
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	20	20	20
Understand	50	50	50
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks.	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations have to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which students should answer anyone. Each question can have a maximum of 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Basics of Architecture)

Classes of Computers - Classes of Parallelism and Parallel Architectures – Defining Computer Architecture – Dependability – Quantitative Principles of Computer Design – Basics of Memory Hierarchies – Virtual Memory and Virtual Machines – Pipelining

Module-2 (Instruction-Level Parallelism)

Instruction-Level Parallelism: Concepts and Challenges – Basic Compiler Techniques for Exposing ILP – Reducing Branch Costs With Advanced Branch Prediction – Hardware-Based Speculation – Multithreading: Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput

Module-3 (Data-Level Parallelism)

Vector Architecture – SIMD Instruction Set Extensions for Multimedia – Graphics Processing Units – Detecting and Enhancing Loop-Level Parallelism

Module-4 (Thread Level Parallelism)

Multiprocessor Architecture: Issues and Approach – Centralized Shared-Memory Architectures – Performance of Symmetric Shared-Memory Multiprocessors– Distributed Shared-Memory and Directory-Based Coherence – Synchronization: The Basics – Introduction to Memory Consistency

Module-5 (GPU Architectures)

The CPU-GPU system as an accelerated computational platform – The GPU and the thread engine – Characteristics of GPU memory spaces – The PCI bus: CPU to GPU data transfer overhead – Multi-GPU platforms – Potential benefits of GPU – accelerated platforms

Text Books

1. John L. Hennessy, David A. Patterson Computer Architecture, Sixth Edition A Quantitative Approach, Morgan Kaufman, Fifth Edition, 2012.
2. Robert Robey, Yuliana Zamora, Parallel and High-Performance Computing, Manning Publications, First Edition, 2021.

Reference Books

1. Thomas Sterling, Matthew Anderson, and MaciejBrodowicz, High-Performance Computing – Modern Systems and Practices, First Edition, 2017.
2. Charles Severance, Kevin Dowd,High-Performance Computing, O'Reilly Media, Second Edition, 1998.
3. Kai Hwang, Faye Alaye Briggs, Computer Architecture and Parallel Processing, McGraw-Hill, 1984.

Course Level Assessment Questions

Course Outcome 1 (CO1):

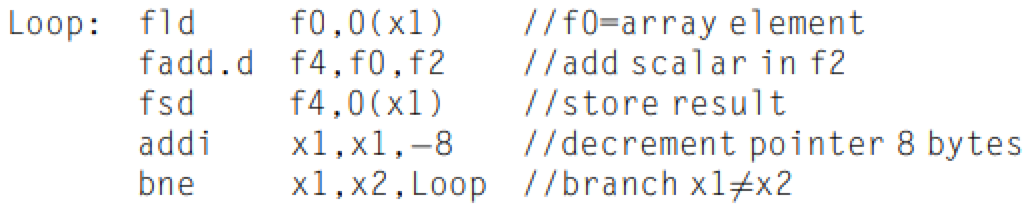
1. Differentiate different classes of computer-based on features like microprocessor cost, system cost, and system design issues.
2. Explain the different methods by which computer hardware exploits application-level parallelism.
3. Explain in detail the instruction set architecture
4. Describe the encoding scheme specified as part of ISA

Course Outcome 2 (CO2):

1. Differentiate data, name, and control dependencies with suitable examples.
2. Explain loop unrolling with suitable coding demonstration
3. Explain in detail about Tournament Predictors.
4. Describe the unique features of very long instruction word processors.

Course Outcome 3 (CO3):

1. What are the three things conveyed through a data dependence? Explain the Data Dependencies of the following code:

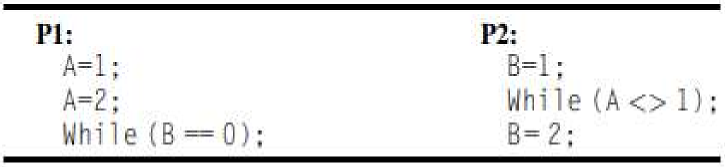


```
Loop:  fld     f0,0(x1)     //f0=array element
       fadd.d  f4,f0,f2     //add scalar in f2
       fsd     f4,0(x1)     //store result
       addi    x1,x1,-8     //decrement pointer 8 bytes
       bne     x1,x2,Loop   //branch x1≠x2
```

2. Assume a single-issue pipeline. Unroll the loop as many times as necessary to schedule it without any stalls, collapsing the loop overhead instructions. How many times must the loop be unrolled? Show the instruction schedule. What is the execution time per element of the result?
3. Explain the SIMD Instruction Set Extensions for Multimedia.

Course Outcome 4 (CO4):

1. With the help of a neat diagram illustrate a single-chip multicore with a distributed cache.
2. Demonstrate the Implementation of cache coherence in a distributed-memory multiprocessor by adding a directory to each node with a suitable diagram.
3. Consider the following code segments running on two processors P1 and P2. Assume A, and B, are initially 0. Explain how an optimizing compiler might make it impossible for B to be ever set to 2 in a sequentially consistent execution model.



```
P1:                          P2:
  A=1;                         B=1;
  A=2;                         While (A <> 1);
  While (B == 0);              B= 2;
```

Course Outcome 5 (CO5):

1. Explain the benefits of potential GPU.
2. Illustrate GPU system as an accelerated computational platform.
3. Discuss CPU to GPU data transfer overhead.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST418

Course Name: High Performance Computing

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Differentiate between Data level parallelism and Task level parallelism

2. Explain the principle of locality

3. Define Instruction Level Parallelism with an example.

4. Devise the importance of loop unrolling with an example.

5. What is the equation of CPI (cycles per instruction) for a pipelined processor? How can we set the ideal pipeline CPI?

6. Explain the two types of name dependencies between an instruction i that precedes instruction j in program order.

7. Differentiate between module reliability and module availability measures with suitable examples.

8. Why SMP architectures are called UMA multiprocessors and DSM multiprocessors as NUMA processors.

9. Explain the need for GPU.

10. List the characteristics of GPU memory spaces.

3x10=30

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Describe the quantitative principle of computer design with Amdahl's law. (8)

(b) Discuss in detail the importance of considering processor performance for the design of an efficient computer system. (6)

OR

12. (a) Illustrate how processes are protected with the help of virtual memory. (7)

(b) Discuss the role played by virtual machines in providing protection for processes. (7)

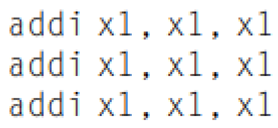
13. (a) Explain in detail data dependence and hazards. **(8)**

(b) With neat sketches explain how data-level parallelism is achieved in vector, and SIMD architectures. **(6)**

OR

14. (a) Describe the unique features of very long instruction word processors. **(8)**

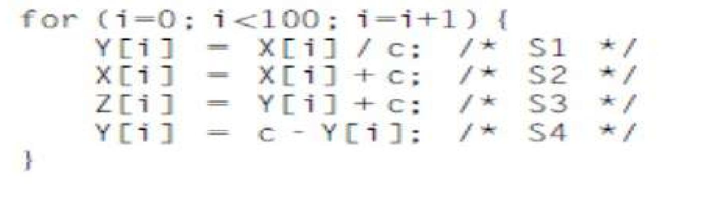
(b) Consider a three-way superscalar machine renaming these three instructions concurrently: **(6)**



```
addi x1, x1, x1
addi x1, x1, x1
addi x1, x1, x1
```

If the value of x1 starts at 5, then what will be its value when after this sequence is executed?

15. (a) The following loop has multiple types of dependences. Find all the true dependences, output dependencies, and anti-dependencies, and eliminate the output dependencies and anti-dependencies by renaming. **(8)**



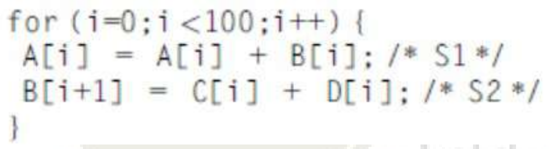
```
for (i=0; i<100; i=i+1) {
    Y[i]  =  X[i] / c;   /*  S1  */
    X[i]  =  X[i] + c;   /*  S2  */
    Z[i]  =  Y[i] + c;   /*  S3  */
    Y[i]  =  c - Y[i];   /*  S4  */
}
```

(b) Describe the limitations of Symmetric Shared-Memory Multiprocessors and Snooping Protocols **(6)**

OR

16. (a) Demonstrate the different types of hardware approaches required for the working of multithreading. **(8)**

(b) Consider the following loop: **(6)**



```
for (i=0;i<100;i++) {
 A[i] = A[i] + B[i]; /* S1*/
 B[i+1] = C[i] + D[i]; /* S2*/
}
```

Are there exist dependencies between S1 and S2? Determine whether the above loop is parallel? If not, show how to make it parallel.

17. (a) Consider an 8-processor multicore where each processor has its own L1 and L2 caches. Here snooping is performed on a shared bus among the L2 caches. Assume that the average L2 request is 15 cycles for a coherence miss or other miss and a clock rate of 3.0 GHz, a CPI of 0.7, and a load/store frequency of 40%. If the goal set is that no more than 50% of the L2 bandwidth is consumed by coherence traffic, then what is the maximum coherence miss rate per processor? **(8)**

(b) Explain the basic structure of a centralized shared-memory multiprocessor **(6)** based on a multicore chip.

OR

18. (a) Suppose an application running on a 100-processor multiprocessor use 1, 50, or 100 processors. If for 95% of the time all 100 processors are used, illustrate how the remaining 5% of the execution time employs 50 processors for a speedup of 80? **(6)**

(b) With a neat diagram, demonstrate invalidate cache coherence protocol for a private write-back cache, showing the states and state transitions for each block in the cache. **(8)**

19. (a) Explain the multi-GPU platform. **(8)**

(b) Explain some of the benefits of GPU. **(6)**

OR

20. (a) Discuss in detail the characteristics of GPU memory spaces. **(8)**

(b) Explain about GPU thread engine. **(6)**

Teaching Plan

No	**Contents**	**No. of Lecture Hours (36 hrs)**
Module 1 - Basics of Architecture (7 hours)		
1.1	Classes of Computers	1 hour
1.2	Classes of Parallelism and Parallel Architectures	1 hour
1.3	Dependability	1 hour
1.4	Quantitative Principles of Computer Design.	1 hour

1.5	Basics of Memory Hierarchies	1 hour
1.6	Virtual Memory and Virtual Machines	1 hour
1.7	Pipelining	1 hour
Module -2 (Introduction to Syntax Analysis) (7 hours)		
2.1	Instruction-Level Parallelism: Concepts and Challenges	1 hour
2.2	Basic Compiler Techniques for Exposing ILP	1 hour
2.3	Reducing Branch Costs With Advanced Branch Prediction	1 hour
2.4	Hardware-Based Speculation	1 hour
2.5	Multithreading	1 hour
2.6	Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput – Lecture 1.	1 hour
2.7	Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput – Lecture 2.	1 hour
Module- 3 - Data-Level Parallelism (7 hours)		
3.1	Vector Architecture -Lecture 1	1 hour
3.2	Vector Architecture -Lecture 2	1 hour
3.3	SIMD Instruction Set Extensions for Multimedia – Lecture 1	1 hour
3.4	SIMD Instruction Set Extensions for Multimedia – Lecture 2	1 hour
3.5	Graphics Processing Units	1 hour
3.6	Detecting and Enhancing Loop-Level Parallelism – Lecture 1	1 hour
3.7	Detecting and Enhancing Loop-Level Parallelism – Lecture 2	1 hour
Module 4– Thread Level Parallelism (8 hours)		
4.1	Multiprocessor Architecture: Issues and Approach	1 hour
4.2	Centralized Shared-Memory Architectures – Lecture 1	1hour
4.3	Centralized Shared-Memory Architectures – Lecture 2	1hour
4.4	Performance of Symmetric Shared-Memory Multiprocessors	1hour
4.5	Distributed Shared-Memory	1hour
4.6	Directory-Based Coherence	1hour
4.7	Synchronization	1hour

4.8	Introduction to Memory Consistency	1hour
Module 5 – GPU Architectures (7 hours)		
5.1	The CPU-GPU system as an accelerated computational platform	1 hour
5.2	The GPU and the thread engine – Lecture 1	1 hour
5.3	The GPU and the thread engine – Lecture 2	1 hour
5.4	Characteristics of GPU memory spaces	1hour
5.5	PCI bus: CPU to GPU data transfer overhead	1hour
5.6	Multi-GPU platforms	1hour
5.7	Potential benefits of GPU-accelerated platforms	1hour

CST428	BLOCKCHAIN TECHNOLOGIES	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: The purpose of this course is to create awareness and understanding among students on the foundation of blockchain technology. The course introduces the cryptographic principles behind blockchain and helps the students understand concepts like consensus, crypto-currency, smart contracts, use cases etc. The course enables students to develop simple decentralized applications using blockchain networks such as Ethereum.

Prerequisite: Basic knowledge in data structures and operating systems.

Course Outcomes: After the completion of the course the student will be able to

CO1	Illustrate the cryptographic building blocks of blockchain technology. **(Cognitive Knowledge Level: Understand)**
CO2	Explain the fundamental concepts of blockchain technology. **(Cognitive Knowledge Level: Understand)**
CO3	Summarize the classification of consensus algorithms. (**Cognitive Knowledge Level: Understand)**
CO4	Explain the concepts of first decentralized cryptocurrency bitcoin. (**Cognitive Knowledge Level: Understand)**
CO5	Explain the use of smart contracts and its use cases. **(Cognitive Knowledge Level: Understand)**
CO6	Develop simple applications using Solidity language on Ethereum platform. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓										✓

CO3	✓	✓										✓
CO4	✓	✓										✓
CO5	✓	✓										✓
CO6	✓	✓	✓	✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**30**	**30**	**30**
Understand	**50**	**50**	**50**
Apply	**20**	**20**	**20**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module – 1 (Fundamentals of Cryptography)

Introduction to Cryptography, Symmetric cryptography – AES. Asymmetric cryptography – RSA. Elliptic curve cryptography, Digital signatures – RSA digital signature algorithms. Secure Hash Algorithms – SHA-256. Applications of cryptographic hash functions – Merkle trees, Distributed hash tables.

Module – 2 (Fundamentals of Blockchain Technology)

Blockchain – Definition, architecture, elements of blockchain, benefits and limitations, types of blockchain. Consensus – definition, types, consensus in blockchain.

Decentralization – Decentralization using blockchain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization.

Module - 3 (Consensus Algorithms and Bitcoin)

Consensus Algorithms, Crash fault-tolerance (CFT) algorithms – Paxos, Raft. Byzantine fault-tolerance (BFT) algorithms – Practical Byzantine Fault Tolerance (PBFT), Proof of work (PoW), Proof of stake (PoS), Types of PoS.
Bitcoin – Definition, Cryptographic keys – Private keys, public keys, addresses. Transactions – Lifecycle, coinbase transactions, transaction validation. Blockchain – The genesis block.
Mining – Tasks of miners, mining algorithm, hash rate. Wallets – Types of wallets.

Module - 4 (Smart Contracts and Use cases)

Smart Contracts – Definition, Smart contract templates, Oracles, Types of oracles, Deploying smart contracts. Decentralization terminology – Decentralized applications, Decentralized Autonomous Organizations.
Use cases of Blockchain technology – Government, Health care, Finance, Supply chain management.
Blockchain and allied technologies – Blockchain and Cloud Computing, Blockchain and Artificial Intelligence.

Module - 5 (Ethereum and Solidity)

Ethereum – The Ethereum network. Components of the Ethereum ecosystem – Keys and addresses, Accounts, Transactions and messages. The Ethereum Virtual Machine, Blocks and blockchain.
The Solidity language – The layout of a Solidity source code, Structure of a smart contract, variables, data types, control structures, events, inheritance, libraries, functions, error handling.
Smart contracts Case study: Voting, Auction.

Text Book

1. Imran Bashir, Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, Packt Publishing, Third edition, 2020.

References

2. Ritesh Modi, Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain, Packt Publishing, First edition, 2018.

3. Kumar Saurabh, Ashutosh Saxena, Blockchain Technology: Concepts and Applications, First Edition, Wiley Publications, First edition, 2020.

4. Chandramouli Subramanian, Asha A George, et al, Blockchain Technology, Universities Press (India) Pvt. Ltd, First edition, August 2020.

5. Lorne Lantz, Daniel Cawrey, Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications, O'Reilly Media, First edition, 2020.

6. Andreas M. Antonopoulos, Gavin Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, First edition, 2018.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Distinguish between Symmetric cryptography and asymmetric cryptography.
2. Explain the working of AES algorithm.

Course Outcome 2 (CO2):

1. Categorize consensus mechanism used in blockchain.
2. Define Blockchain. Explain how decentralization of computing or processing power is achieved by a blockchain.

Course Outcome 3 (CO3):

1. Explain how Proof of Stake can achieve consensus among peers.
2. Explain the working of Raft protocol.

Course Outcome 4 (CO4):

1. Describe the use of genesis block.
2. Explain the mining algorithm used in bitcoin.

Course Outcome 5 (CO5):

1. Illustrate how blockchain technology can be used in supply chain management.
2. What are oracles in a blockchain ecosystem? Explain the generic data flow from a smart contract to an oracle.

Course Outcome 6 (CO6):

1. Develop a smart contract for voting process. In this application, delegated voting is allowed and the counting is automatic and completely transparent at the same time.
2. Develop a smart contract for auction process. The contract should be a blind auction where it is not possible to see the actual bid until the bidding period ends.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 2**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST428

Course Name: BLOCK CHAIN TECHNOLOGIES

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Discuss the role of secure hash functions in blockchain.

2. List out the properties of digital signatures.

3. Illustrate the blockchain based decentralized system.

4. Explain how Proof of Stake can achieve consensus among peers.

5. If your blockchain network has 5 Byzantine nodes, what is the minimum number of nodes that are required to ensure Byzantine fault tolerance using PBFT protocol?

6. How are transactions verified in a Bitcoin network?

7. Explain how smart contracts can be used for enforcing agreements between parties in the form of business logic.

8. Explain the concept of blockchain-based digital identity cards.

9. Explain error handling in Solidity language.

10. With the help of a figure show the relationship between the transaction, transaction trie, and block header in Ethereum. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the design of SHA-256 and its compression function using a diagram. **(9)**

    (b) Explain how hash functions are used to build Merkle trees in blockchain. **(5)**

OR

12. (a) Explain public and private keys. Perform encryption and decryption using RSA for p=3, q=11, e= 7 and M=5. **(7)**

    (b) Explain elliptic curve digital signature algorithm. **(7)**

13. (a) Illustrate and explain how blockchain works using a neat diagram. **(7)**

    (b) Explain the benefits, features and limitations of blockchain. **(7)**

OR

14. (a) Explain consensus mechanisms used in blockchain. List out any six consensus algorithms used in the context of blockchain. **(7)**

    (b) Define blockchain. Explain how decentralization of computing or processing power is achieved by a blockchain. **(7)**

15. (a) Explain and illustrate how Paxos protocol can be used to achieve consensus. **(7)**

    (b) Show how Practical Byzantine Fault Tolerance can achieve consensus in the presence of Byzantine faults. **(7)**

OR

16. (a) Describe the various fields that make up a transaction in Bitcoin. **(7)**

    (b) What is the role of a Bitcoin miner? Explain the mining algorithm used in **(7)**

Bitcoin with the help of a flowchart.

17. (a) Illustrate how blockchain technology can be implemented in finance sector. **(7)**

(b) Discuss oracles in a blockchain ecosystem. Explain the generic data flow from a smart contract to an oracle. **(7)**

OR

18. (a) Explain the design process of decentralized applications with diagrams. **(7)**

(b) Explain the use of blockchain technology in supply chain management. **(7)**

19. (a) Using Solidity language, create a simple bank contract that allows a user to deposit, withdraw and view balance. **(7)**

(b) Define block difficulty. Explain how block difficulty is adjusted in Ethereum blockchain network. **(7)**

OR

20. (a) Using Solidity language, create a simple voting smart contract where a chairperson will give the right to vote to each address individually. **(7)**

(b) Explain the concept of Gas in Ethereum. Explain how transaction cost can be calculated in an Ethereum blockchain network. **(7)**

Teaching Plan

No	Contents	No. of Lecture Hours (35 hours)
Module-1 (Fundamentals of Cryptography) (7 hours)		
1.1	Introduction to cryptography	1 hour
1.2	Symmetric cryptography, AES	1 hour
1.3	Asymmetric cryptography, RSA	1 hour
1.4	Elliptic curve cryptography	1 hour
1.5	Digital signatures – RSA digital signature algorithm	1 hour
1.6	Secure Hash Algorithms – SHA-256	1 hour
1.7	Applications of cryptographic hash functions – Merkle trees, Distributed hash tables	1 hour
Module-2 (Fundamentals of Blockchain Technology) (6 hours)		
2.1	Blockchain – definition and architecture	1 hour
2.2	Elements of blockchain.	1 hour
2.3	Blockchain – benefits and limitations, types.	1 hour
2.4	Consensus – definition, types, consensus in blockchain	1 hour
2.5	Decentralization using blockchain, Methods of decentralization	1 hour
2.6	Routes to decentralization, Blockchain and full ecosystem decentralization	1 hour
Module-3 (Consensus Algorithms and Bitcoin) (7 hours)		
3.1	Consensus Algorithms – Crash fault-tolerance (CFT) algorithms – Paxos, Raft (working is expected).	1 hour
3.2	Byzantine fault-tolerance (BFT) algorithms – Practical Byzantine Fault Tolerance (PBFT) (working is expected).	1 hour
3.3	Proof of work (PoW), Proof of stake (PoS), Types of PoS	1 hour
3.4	Bitcoin – Definition, Cryptographic keys – Private keys, public keys, addresses.	1 hour
3.5	Transactions – Lifecycle, coinbase transactions, transaction validation	1 hour

3.6	Blockchain – The genesis block. Mining – Tasks of miners	1 hour
3.7	Mining – mining algorithm, hash rate. Wallets – Types of wallets.	1 hour
Module-4 (Smart Contracts and Use cases) (6 hours)		
4.1	Smart Contracts – Definition, Smart contract templates	1 hour
4.2	Oracles, Types of oracles, Deploying smart contracts.	1 hour
4.3	Decentralization terminology –Decentralized applications, Decentralized Autonomous Organizations.	1 hour
4.4	Use cases of Blockchain technology – Government, Health care.	1 hour
4.5	Use cases of Blockchain technology – Finance, Supply chain management.	1 hour
4.6	Blockchain and Allied Technologies – Blockchain and Cloud Computing, Blockchain and Artificial Intelligence.	1 hour
Module-5 (Ethereum and Solidity) (9 hours)		
5.1	Ethereum - The Ethereum network, Components of the Ethereum ecosystem – Keys and addresses, Accounts	1 hour
5.2	Components of the Ethereum ecosystem – Transactions and messages	1 hour
5.3	The Ethereum Virtual Machine	1 hour
5.4	Ethereum Blocks and blockchain	1 hour
5.5	The Solidity language – The layout of a Solidity source code, Structure of a smart contract, variables, data types	1 hour
5.6	The Solidity language – control structures, events, inheritance, libraries	1 hour
5.7	The Solidity language – functions, error handling.	1 hour
5.8	Smart contracts Case study: Voting.	1 hour
5.9	Smart contracts Case study: Auction.	1 hour

CST438	IMAGE PROCESSING TECHNIQUE	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course helps the learners understand the core concepts and applications of Digital Image Processing. It covers Digital Image Fundamentals, Image Transforms, Image Enhancement in Spatial and Frequency Domain, Image Restoration & Image Segmentation and Morphological Operations & Representation and Description. The learners will be able to develop new algorithms, tools, and application software for real-world applications involving image processing.

Prerequisite: A basic knowledge of Computer Graphics and Image representation

Course Outcomes: After the completion of the course, the student will be able to

CO1	Explain the concepts of image formation and the basis of digital image processing. **(Cognitive Knowledge Level: Understand)**
CO2	Demonstrate the role of image transforms in representing, highlighting, and modifying image features. **(Cognitive Knowledge Level: Apply)**
CO3	Solve image enhancement problems using spatial and frequency domain techniques. **(Cognitive Knowledge Level: Apply)**
CO4	Make use of the concept of image restoration and image segmentation techniques in real-world problems. **(Cognitive Knowledge Level: Apply)**
CO5	Interpret morphological operations, image representation, and description techniques. **(Cognitive Knowledge Level: Understand)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓			✓							✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓										✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and teamwork
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks (%)**
	Test 1 (%)	**Test 2 (%)**	
Remember	**30**	**30**	**30**
Understand	**40**	**40**	**40**
Apply	**30**	**30**	**30**
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of Internal Tests 1 & 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which students should answer anyone. Each question can have a maximum 2 subdivisions and carries 14 marks.

Syllabus

Module – 1 (Digital Image Fundamentals)

Elements of Visual Perception, A Simple Image Formation Model. Spatial and Intensity Resolution. Image Interpolation. Classification of Digital Images. Image Types. Image Storage Mechanisms. Arithmetic and Logical Operations. Geometric Spatial Transformations and Image Registration. Image File Formats. Colour Fundamentals and Colour Models.

Module - 2 (Image Transforms)

Basic concept of spatial domain and frequency domain, Unitary transform, Discrete Fourier Transform- 2D DFT, 4 order DFT Transform coefficients, Forward and inverse transform, Discrete Cosine Transform- 2D DCT, 4 order DCT Transform Coefficients(No derivation needed), Forward and Inverse DCT, Hadamard Transform.

Module - 3 (Image Enhancement in Spatial and Frequency Domain)

Point operations- Clipping and Thresholding, Digital Negative, Intensity Level Slicing, Bit Extraction, Range Compression. Spatial Operations- Fundamentals of spatial convolution and correlation, Spatial averaging and spatial Low pass filtering, Directional Smoothing, Median Filtering, Unsharp masking and Crispening.

Basics of Filtering in Frequency Domain, Filters, Smoothing Frequency Domain Filters-Sharpening Frequency Domain Filters

Module - 4 (Image Restoration & Image Segmentation)

Image degradation model, Noise models, Mean Filters, Order Statistic filter, Adaptive filters. Edge Detection, gradient operators, Laplace operators and zero crossings. Thresholding, Basic Global Thresholding, Optimum global thresholding using Otsu method, Multiple thresholds, Variable thresholding, Multivariable thresholding. Region-Based Approach to Segmentation.

Module - 5 (Morphological Operations & Representation and Description)

Structuring Element, Dilation and Erosion, Opening and Closing, Hit or Miss Transformation.

Boundary Following. Chain Codes. Polygonal Approximation. Boundary Descriptors. Regional Descriptors. Relational Descriptors.

Text Books

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing (English) 3rd Edition, Pearson India, 2013
2. A K. Jain, Fundamentals of digital image processing, Prentice Hall of India, 1989.

Reference Books

1. Al Bovik, The Essential Guide to Image Processing, Academic Press, 2009.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis, and Machine Vision, Thomson Learning, 2008.
3. S Jayaraman, S Esakkirajan and T Veerakumar, Digital Image Processing, McGraw Hill Education, 2009.

Course Level Assessment Questions

Course Outcome1 (CO1) :

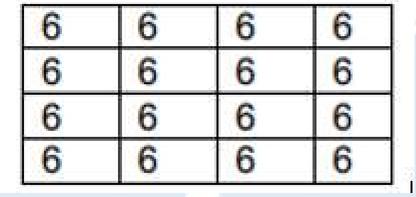
1. Find the number of bits required to store a 256 X 256 image with 32 gray levels.
2. Explain the reasons for blocking artifacts and false contours in an image.

Course Outcome 2 (CO2) :

1. Compare different image transforms based on their roles, properties and applications.
2. Compute the inverse 2D DFT of the transform coefficients F(k,l) given below.

$$F(k,l) = \begin{pmatrix} 64 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

3. Use Discrete Fourier transform to construct 2D DFT for a 4x4 image given below. Assume that indices start from (0,0)



6	6	6	6
6	6	6	6
6	6	6	6
6	6	6	6

Course Outcome 3 (CO3) :

1. Perform intensity level slicing on the 3 BPP (Bit Per Pixel) image. Let r1=3 and r2=5. Draw the modified image with/without background transformations.

$$\begin{bmatrix} 2 & 1 & 2 & 2 & 1 \\ 2 & 3 & 4 & 5 & 2 \\ 6 & 2 & 7 & 6 & 0 \\ 2 & 6 & 6 & 5 & 1 \\ 0 & 3 & 2 & 2 & 1 \end{bmatrix}$$

2. Let y(m) = {2,3,8,4,2}. Obtain the median filter output for the window W = [-1,0,1,2] and show how salt and pepper noise is reduced.
3. Consider a 3*3 spatial mask that averages the four closest neighbors of a point(x,y), but excludes the point itself from the average.

(a) Find the equivalent filter H(u,v) in the frequency domain.

(b) Show that H(u,v) is a lowpass filter (ASSIGNMENT)

Course Outcome 4 (CO4) :

1. Compare Region and Edge-based techniques in segmentation.

2. Consider a noisy image that is restored using arithmetic mean filter of size 3x3 and using the geometric mean filter of the same size. Which image will be less blurred and why?
3. Suppose that you want to help a radiologist to extract the tumor portion from an MRI image for volumetric analysis. This volumetric analysis determines the effect of treatment on the patient, which can be judged from the extracted size and shape of the abnormal portion. Manual tracing of the tumor regions is very difficult since the tumor portion on the MRI image is inhomogeneous, with complex shapes and ambiguous boundaries. Suggest a sequence of steps that you may use to automate this process as an image processing student. (ASSIGNMENT)

Course Outcome 5 (CO5) :

4. Explain the significance of structuring elements in morphological operations with example.
5. Explain how chain codes are used to represent boundaries of a region with examples.

Model Question Paper

QP CODE:

Reg No: _______________

Name: _________________ **PAGES : 4**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST438

Course Name: IMAGE PROCESSING TECHNIQUE

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Give an image representation model and describe how the representation changes in different types of images.
2. Describe any three types of color models.
3. Obtain the HADAMARD basis matrix for N=8.
4. Prove that DFT is a unitary transform.
5. Sketch perspective plot of a 2-D ideal low pass filter transfer function and filter cross-section. List its usefulness in Image enhancement.
6. Explain the significance of directional smoothing technique.
7. Specify the significance of the Zero crossing detector.
8. Describe region growing technique for image segmentation.
9. Define ‘Structuring Element’ used in morphological operations. Give samples for Structuring Elements.
10. Explain image boundary representation using polygonal approximation.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain a Simple Image Formation Model with the help of a neat diagram. **(7)**

(b) Explain the relationship between image size, spatial resolution, and image quality. Compare gray level and intensity resolution. **(7)**

OR

12. (a) Describe arithmetic, logical and geometrical operations on Image. **(7)**

(b) Explain the significance of image interpolation and describe its various types. **(7)**

13. (a) State the advantages of Discrete Cosine Transform over Discrete Fourier Transform. **(4)**

(b) You are given a 4 X 4 image patch Compute 2D DCT for the image patch. Reconstruct the original image patch by neglecting the last four coefficients in 2D DCT. Comment on the observed result. **(10)**

$$\begin{bmatrix} 12 & 4 & 2 & 6 \\ 5 & 10 & 12 & 24 \\ 6 & 8 & 10 & 12 \\ 14 & 12 & 8 & 10 \end{bmatrix}$$

OR

14. (a) Discuss the concept of sequency in Hadamard transform. **(4)**

(b) Find the 2D forward DFT of the image segment **(10)**

1 1 1 1

1 1 1 1

1 1 1 1

Prove the unitary property of the given image segment.

15. (a) Explain the output and application of the following point processing techniques **(9)**

(i)Range Compression (ii) Bit Extraction (iii) Thresholding

(b) State and explain the features of median filtering. Compute the output of the median filtering for Y(m)={2,4,8,3,2} , w={-1,0,1,2} where Y(m) is an array and w is a window. **(5)**

OR

16. (a) Describe the role of Unsharp masking with its applications **(4)**

(b) Explain and compare the basic frequency domain filters for image sharpening **(10)**

17. (a) A 4×4 image is given by **(8)**

2 4 8 7
12 6 9 8
13 7 4 3
8 12 4 9

Filter the above image using

(a) MIN filter (b) MAX filter using the filter mask

0 1 0
1 1 1
0 1 0

(Assume replicate padding of the input image)

(b) Explain any two types of thresholding techniques. Describe the threshold detection algorithm using Otsu's method. **(6)**

OR

18. (a) Explain Image degradation model with the help of a neat diagram. **(8)**

(b) Illustrate the split and merge algorithm for image segmentation using neat sketches. **(6)**

19. (a) Explain the purpose of morphological operations in digital image? Describe the opening and closing operations with examples. **(7)**

(b) Illustrate Hit or Miss Transformation. **(7)**

OR

20. (a) Explain the concept of the chain coding scheme with its applications. **(6)**

(b) Describe in detail any two boundary representation schemes and illustrate with examples. **(8)**

Teaching Plan

No	Contents	No. of Lecture Hours (36 hrs)
	Module-1 (Digital Image Fundamentals) (7 hours)	
1.1	Elements of Visual Perception, A Simple Image Formation Model	1
1.2	Spatial and Intensity Resolution, Image Interpolation, Classification of Digital Image.	1
1.3	Image Types, Image Storage Mechanisms.	1
1.4	Arithmetic and Logical Operations.	1
1.5	Geometric Spatial Transformations and Image Registration.	1
1.6	Image File Formats.	1

1.7	Colour Fundamentals and Colour Models.	1
Module-2 (Image Transforms) (8 hours)		
2.1	Basic concept of spatial domain and frequency domain.	1
2.2	Need of Image Transform, Basic properties of unitary transform.	1
2.3	Discrete Fourier transform, Proof DFT is Unitary.	1
2.4	4 order DFT Transform coefficients (Derivation).	1
2.5	Problems (4 order DFT).	1
2.6	Discrete Cosine Transform- 2D DCT.	1
2.7	4 order DCT Transform Coefficients(No derivation needed).	1
2.8	Hadamard Transform.	1
Module-3 (Image Enhancement in spatial and frequency domain) (8 hours)		
3.1	Point operations- Clipping and Thresholding, Digital Negative. Intensity Level Slicing.	1
3.2	Bit Extraction, Range Compression + (Work out problems).	1
3.3	Spatial Operations-Fundamentals of spatial convolution and correlation.	1
3.4	Spatial averaging and spatial Low pass filtering, Directional Smoothing.	1
3.5	Median Filtering, Unsharp masking and Crispening.	1
3.6	Basics of Filtering in Frequency Domain.	1
3.7	Smoothing Frequency Domain Filters : Ideal Low Pass Filter; Gaussian Low Pass Filter; Butterworth Low Pass Filter;	1
3.8	Sharpening Frequency Domain Filters: Ideal High Pass Filter; Gaussian High Pass Filter; Butterworth High Pass filter.	1
Module-4 (Image Restoration & Image Segmentation) (6 hours)		
4.1	Image degradation model, Noise models.	1
4.2	Mean Filters – Order Statistic filter – Adaptive filters.	1
4.3	Edge Detection, Gradient operators, Laplace operators and zero crossings.	1

4.4	Thresholding- Basic Global Thresholding, Optimum global thresholding using Otsu method.	1
4.5	Multiple thresholds, Variable thresholding, Multivariable thresholding.	1
4.6	Region-Based Approach to Segmentation.	1
Module-5 (Morphological Operations & Representation and Description) (7 hours)		
5.1	Structuring Element. Dilation and Erosion,	1
5.2	Morphological Opening, Closing.	1
5.3	Hit or Miss Transformation.	1
5.4	Boundary Following. Chain Codes, Polygonal Approximation.	1
5.5	Boundary Descriptors.	1
5.6	Regional Descriptors.	1
5.7	Relational Descriptors.	1

CST448	INTERNET OF THINGS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This course equips the learners with fundamental of the Internet of Things(IoT) and the IoT ecosystem. It covers the architecture of IoT, communication mechanisms, protocols, hardware, software, data analytics, and the cloud platforms for IoT. This course enables the students to design smart IoT applications for real world problems using Raspberry Pi.

Prerequisite: Basic knowledge in Data Communication, Computer Networks and Programming in Python

Course Outcomes: After the completion of the course the students will be able to

CO1	Outline the fundamentals of IoT and its underlying physical and logical architecture**(Cognitive Knowledge Level: Understand)**
CO2	Explain the hardware architectures for IoT **(Cognitive Knowledge Level : Understand)**
CO3	Outline the Network architectures for IoT**(Cognitive Knowledge Level : Understand)**
CO4	Implement data analytics on the IoT platforms **(Cognitive Knowledge Level : Apply)**
CO5	Appreciate the security considerations in IoT **(Cognitive Knowledge Level : Understand)**
CO6	Implement IoT applications using the available hardware and software. **(Cognitive Knowledge Level : Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓									✓
CO2	✓	✓	✓									✓
CO3	✓	✓	✓									✓
CO4	✓	✓	✓	✓	✓							✓

CO5	✓	✓	✓		✓							✓
CO6	✓	✓	✓	✓	✓	✓						✓

Abstract POs Defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and teamwork
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Blooms Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test 1 (Percentage)	**Test 2 (Percentage)**	
Remember	30	20	30
Understand	60	50	40
Apply	10	30	30
Analyze			

Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	**50**	**100**	**3 Hours**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus, and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have a maximum 2 subdivisions and carries 14 marks.

Syllabus

Module- 1 (IoT Architecture)

What is IoT, Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and IoT, IoT Challenges, IoT Network Architecture and Design, Drivers Behind New Network Architectures, Comparing IoT Architectures, A Simplified IoT Architecture, The Core IoT Functional Stack, IoT Data Management and Compute Stack.

Module- 2 (Engineering IoT Networks)

Smart Objects: The "Things" in IoT, Sensors, Actuators, and Smart Objects, Sensor Networks, Connecting Smart Objects, Communications Criteria, IoT Access Technologies

Module- 3 (IoT Network Layer)

IP as the IoT Network Layer, The Business Case for IP, The need for Optimization, Optimizing IP for IoT, Profiles and Compliances, Application Protocols for IoT, The Transport Layer, IoT Application Transport Methods

Module 4 (Data Analytics for IoT)

Data and Analytics for IoT, An Introduction to Data Analytics for IoT, Machine Learning, Big Data Analytics Tools and Technology, Edge Streaming Analytics, Network Analytics, Securing IoT, A Brief History of OT Security, Common Challenges in OT Security, Differences between IT and OT Security Practices and Systems, Formal Risk Analysis Structures: OCTAVE and FAIR.

Module 5 (Developing IoT Systems)

IoT Logical Design using Python, IoT Physical Devices and Endpoints - Raspberry Pi interfaces, Programming Raspberry Pi using Python, Other IoT devices, IoT Physical devices and Cloud offerings, Cloud Storage Models, WAMP - Autobahn for IoT, Django, Designing RESTful Web API, Cloud Web Services for IoT.

Textbooks

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1st Edition, Pearson Education (Cisco Press Indian Reprint)

2. Arshadeep Bahga, Vijay Madisetti, "Internet of Things: A hands-on approach", University Press, 2015 (First edition)

References

1. Rajkamal, "Internet of Things: Architecture and Design Principles", McGraw Hill (India) Private Limited

2. Dieter Uckelmann, Mark Harrison, Michahelles Florian (Ed.), Architecting the internet of things, Springer, 2011

3. Dr. Ovidiu Vermesan, Dr. Peter Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers, 2013

4. Simon Monk, "Programming Arduino: Getting Started with Sketches", McGraw Hill Publications

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Write a short note on the impact of IoT in the real world
2. Explain the challenges of IoT.
3. Compare OT and IT Technology.
4. Describe the elements of one M2M architecture of IoT

Course Outcome 2 (CO2):

1. Mention any four wireless technologies and its architectural characteristics
2. Comment things in IoT
3. Compare biosensors and biodegradable sensors used in IoT
4. Explain the term NBIoT(Narrow Band IoT)

Course Outcome 3 (CO3):

1. Discuss the need for optimization
2. Compare MQTT and COAP
3. Explain different schedule management and packet forwarding models of 6TiSCH

Course Outcome 4(CO4):

1. Compare Bigdata and edge analytics
2. Compare structured and unstructured data
3. Describe the components of FNF

Course Outcome 5(CO5):

1. What are the major challenges in IoT security?
2. Explain the impact of OT Network Characteristics onIoT Security.

Course Outcome 6(CO6):

1. Implement LDR interfacing with Raspberry Pi
2. Explain the development of a RESTful web API.

Model Question Paper

QP CODE: PAGES :3

Reg No: _______________

Name: _________________

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST448

Course Name: Internet of Things

Max.Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Explain the role of IoT in connected roadways,

2. Describe the functions of the various layers of simplified IoT Architecture Model.

3. Explain the communication protocols employed in Wireless Sensor Networks

4. What are the essential performance considerations of constrained-node networks?

5. Explain the parameters to be considered while choosing between IP adaptation / adoption for last mile communication.

6. With neat diagrams compare the IoT protocol stacks using 6LoWPAN and IP.

7. Differentiate the types of IoT data analytics results.

8. How can the insecure operational protocols be characterized?

9. Write a program to interface an LED and a switch with Raspberry Pi

10. List down the Raspberry Pi interfaces and explain. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Illustrate the impact of IoT in at least 2 domains of normal human life. **(9)**

(b) Describe the Application and Analytics sublayer of IoT Architecture **(6)**

OR

12. (a) Describe the Standardized IoT architectures. **(8)**

(b) Explain the functions of Access Network Sublayer of IoT Architecture **(6)**

13. (a) Describe the LoRaWAN technology as an IoT communication paradigm. **(10)**

(b) Describe various types of sensors. **(4)**

OR

14. (a) Define actuators. Describe the roles of actuators in IoT systems. **(6)**

(b) Explain the IEEE 802.15.4 standard for wireless communication. **(8)**

15. (a) Explain Message Queuing Telemetry Transport framework and message format. **(8)**

(b) Explain tunneling of legacy SCADA over IP Networks with a neat diagram. **(6)**

OR

16. (a) Explain SCADA Transport over LLNs with MAP-T. **(7)**

(b) Explain RPL encryption and authentication on constrained nodes. **(7)**

17. (a) Explain the Hadoop ecosystem with a neat diagram. **(7)**

(b) Explain the Flexible NetFlow Architecture. **(7)**

OR

18. (a) Explain the "The Purdue Model for Control Hierarchy" and OT network characteristics. **(8)**

(b) Explain any twp formal risk analysis structures **(6)**

19. (a) Explain the working of WAMP protocol. **(8)**

(b) Describehow AWS supports IoT development **(6)**

OR

20. (a) Demonstrate an example of Raspberry Pi applications for Industrial IoT. **(8)**

(b) Explain the Django Architecture **(6)**

TEACHING PLAN

No	Contents	No of Lecture Hrs (35 Hrs)
Module – 1 (IoT Architecture) (6 hrs) (TB-1, Chapter 1,2)		
1.1	What is IoT, Genesis of IoT, IoT and Digitization,	1
1.2	IoT Impact, Convergence of IT and IoT, IoT Challenges	1
1.3	IoT Network Architecture and Design	1
1.4	Drivers Behind New Network Architectures, Comparing IoT Architectures	1
1.5	A Simplified IoT Architecture,	1

1.6	The Core IoT Functional Stack, IoT Data Management and Compute Stack.	1
Module- 2 (Engineering IoT Networks) (7hrs)(TB-1, Chapter 3,4)		
2.1	Smart Objects: The “Things” in IoT,	1
2.2	Sensors, Actuators, and Smart Objects	1
2.3	Sensor Networks	1
2.4	Connecting Smart Objects	1
2.5	IoT Access Technologies –IEEE 802.15.4 (g/e), 1901.2a	1
2.6	IoT Access Technologies - 802.11ah, LoRaWAN	1
2.7	IoT Access Technologies – LoRaWAN, NBIoT, LTE	1
Module- 3 (IoT Network Layer) (7 hrs)(TB-1, Chapter 5,6)		
3.1	IP as the IoT Network Layer, The Business Case for IP	1
3.2	The need for Optimizing IP for IoT	1
3.3	Optimizing IP for IoT, Profiles, and Compliance	1
3.4	Application Protocols for IoT - CoAP	1
3.5	Application Protocols for IoT - MQTT	1
3.6	The Transport Layer, IoT Application Transport Methods	1
3.7	The Transport Layer, IoT Application Transport Methods	1
Module 4 (Data Analytics for IoT) (6hrs)(TB-1, Chapter 7,8)		
4.1	An Introduction to Data Analytics for IoT, Machine Learning	1
4.2	Big Data Analytics Tools and Technology	1
4.3	Edge Streaming Analytics, Network Analytics	1

4.4	A Brief History of OT Security, Common Challenges in OT Security	1
4.5	Differences between IT and OT Security Practices and Systems	1
4.6	Formal Risk Analysis Structures: OCTAVE and FAIR	1
Module 5 (Developing IoT Systems)(9 hrs) (TB-2, Chapter 6,7,8)		
5.1	IoT Logical Design using Python,	1
5.2	IoT Physical Devices and Endpoints	1
5.3	Raspberry Pi interfaces, Programming Raspberry Pi using Python	1
5.4	Other IoT devices	1
5.5	Cloud Storage Models	1
5.6	WAMP-Autobahn for IoT	1
5.7	Django	1
5.8	Designing RESTful Web API	1
5.9	Cloud Web Services for IoT.	1

CST458	SOFTWARE TESTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: This is a course in theoretical computer science that introduces the concepts and methods in software testing. It covers various techniques for test case design used to test software artifacts, including requirements, design, and code, the different techniques for test case design based on graphs, programming language syntaxes and symbolic execution using PEX tool. It enables the learners to follow a systematic software testing approaches while developing applications.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to:-

CO1	List a range of different software testing techniques and be able to apply specific unit testing method to the projects using Junit**.(Cognitive Knowledge Level: Understand)**
CO2	Illustrate using appropriate tools the mutation testing method for a given piece of code to identify hidden defects that can't be detected using other testing methods.**(Cognitive Knowledge Level: Apply)**
CO3	Explain graph coverage criteria in terms of control flow graph and data flow graph for a given program.**(Cognitive Knowledge Level: Understand**)
CO4	Demonstrate the importance of black-box approaches in terms of domain and functional testing.**(Cognitive Knowledge Level: Apply)**
CO5	Illustrate the use of PEX tool with symbolic execution.**(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO 9**	**PO10**	**PO11**	**PO12**
CO1	✓	✓	✓									✓
CO2	✓	✓	✓	✓	✓					✓		✓
CO3	✓	✓	✓							✓		✓
CO4	✓	✓	✓	✓								✓

CO5	✓	✓	✓	✓	✓					✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	**Continuous Assessment Tests**		**End Semester Examination Marks**
	Test 1 (Marks)	**Test 2 (Marks)**	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	**CIE Marks**	**ESE Marks**	**ESE Duration**
150	**50**	**100**	**3 hours**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of SeriesTests1& 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to Software Testing)

Some Popular Errors – Ariane 5, Therac 25, Intel Pentium Bug. What is Software testing? Why should it be tested? Software Quality, Role of Testing. Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking. Software Testing Terminologies - Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria. Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing, Functional testing, Stress testing, Performance testing, Usability testing and Regression testing. Testing Methods - Black Box testing, White Box testing, Grey Box testing.

Module - 2 (Unit Testing)

Concept of Unit testing. Static Unit testing. Dynamic Unit testing - Control Flow testing, Data Flow testing, Domain testing, Functional Program testing. Mutation testing - Mutation and Mutants, Mutation operators, Mutation score. Junit - Framework for Unit testing. Case Study - Mutation testing using Junit and Muclipse.

Module - 3 (Unit Testing - White Box Approaches)

Overview of Graph Coverage Criteria. Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage, Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage. Data Flow Criteria - du paths, du pairs. Subsumption Relationships among Graph Coverage Criteria. Graph Coverage for Source Code - Control flow graphs for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program – Statistics. Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph, Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root. Case Study - Graph Based testing using JUnit Framework.

Module - 4 (Unit Testing - Black Box Approaches)

Domain Testing / Input Space Partitioning - Partitions of a set. Input domain modelling - Interface-based approach, Functionality-based approach. Identifying values. Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage. TriTyp example. Functional Testing - Functional Testing Concepts of Howden. Functional testing - Important Steps. Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing. Case Study - Black Box testing approaches using JUnit.

Module - 5 (Grey Box Testing Approaches)

Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages. Techniques of Grey Box Testing - Matrix Testing, Regression Testing, Orthogonal Array Testing or OAT, Pattern Testing. An Introduction to PEX - Parameterized Unit Testing, The Testing Problem. Symbolic Execution – Example, Symbolic execution tree. PEX application Case Study – PEX.

Text Books

1. Paul Ammann and JeffOffutt , Introduction to Software Testing, Cambridge University Press
2. Kshirasagar Naik and Priyadarshi Tripathy, Software Testing And Quality Assurance: Theory And Practice, Wiley.

Reference Materials

1. King, James C, “Symbolic Execution and Program Testing”, Association for Computing Machinery, July 1976.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

Explain the following types of testing methods with examples.

(i) Balck-box testing.

(ii) White-box testing.

(iii) Grey-box testing.

Course Outcome 2 (CO2):

Define 12 mutants for the following method *power*() using effective mutation operators. Try to use each mutation operator at least once. Approximately, how many mutants do you think there would be, if all mutants for *power*() were created?

```
public static int power (int left, int right)
{
//**************************************
// Raises Left to the power of Right
// precondition : Right >= 0
// postcondition: Returns Left**Right
//**************************************
intrslt;
rslt = Left;
if (Right == 0)
{
rslt = 1;
}
else
{
for (int i = 2; i <= Right; i++)
rslt = rslt * Left;
}
return (rslt);
}
```

Course Outcome 3 (CO3):

Draw the control flow graph and data flow graph of given piece of code.

```
public static double ReturnAverage(int value[],int AS, int MIN, int MAX){
/*
Function: ReturnAverageComputes the averageof all those numbers in the input array in the positive range [MIN, MAX]. The maximum size of the array is AS. But, the array size could be smaller than AS in which case the end of input is represented by -999.
*/
int i, ti, tv, sum;
doubleav;
i = 0; ti = 0; tv = 0; sum = 0;
while (ti< AS && value[i] != -999) {
ti++;
if (value[i] >= MIN && value[i] <= MAX) {
tv++;
sum = sum + value[i];
}
i++;
}
if (tv> 0)
av = (double)sum/tv;
else
av = (double) -999;
return (av);
}
```

Course Outcome 4 (CO4):

Explain the following with examples.

1. Input domain modelling.
2. All Combinations Coverage (ACoC)
3. Each Choice Coverage (ECC)
4. Pair-wise Coverage
5. T-wise Coverage
6. Base Choice Coverage
7. Multiple Base Choices Coverage.

Course Outcome 5 (CO5):

Draw the symbolic execution tree for the following program code and explain the symbolic execution of testme (α1, α2).

```
int twice (int v) {
 return 2 ∗ v;
 }
 void testme (int x, int y ) {
 z = twice ( y);
 if ( z == x ){
 if ( x > y + 10)
ERROR;
 }
 }
int main() {
 x = sym input();
 y = sym input();
testme ( x , y);
```

```
return(0);
```

Model Question Paper

QP CODE: **PAGES: 3**

Reg No:______________ **Name :______________**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST458

Course Name: Software Testing

Max.Marks:100 **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

1. Explain the differences between Validation and Verification?
2. Explain the differences between Fault, Error, and Bug?
3. Define Ground string, Mutation score, and Mutants?
4. What are the functions of Test driver and Test stubs in dynamic unit testing?
5. Define Node coverage, Edge coverage and Prime path coverage in a control flow graph?
6. What are du paths and du pairs in a data flow graph?
7. Explain the two approaches in input domain modelling?
8. Explain the difference between Equivalence Class Partitioning and Boundary Value Analysis?
9. Briefly explain three techniques of Grey box testing?
10. Explain the concept of symbolic execution with the help of a toy example?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the following types of testing

(i) Black Box testing (ii) White Box testing (iii) GreyBox testing (iv) Unit testing (v) Integration testing (vi) System testing (vii) Acceptance testing **(14)**

OR

12. (a) Explain the following coverage criterias based on the code fragment given below? (i) Functional coverage (ii) Statement coverage (iii)Conditional coverage (iv)Branch coverage **(8)**

```
int foo (int x, int y){
    int z = 0;
    if ((x > 0) && (y > 0)){
    z = x;}
    return z;
    }
```

(b) Write positive and negative test cases for an ATM Machine? **(6)**

13. (a) Explain Dynamic unit test environment with a neat figure. **(8)**

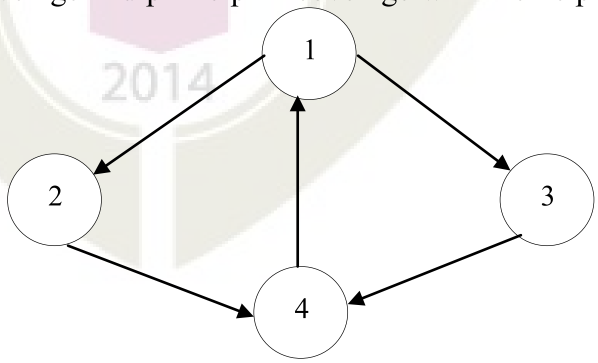
(b) Explain the major difference between control flow testing and data flow testing. **(6)**

OR

14. (a) Explain seven types of mutation operators with neat examples? **(14)**

15. (a) Explain touring, side trips and detours with a neat example **(7)**

(b) Explain simple path coverage and prime path coverage with the help of CFG given below? **(7)**



OR

16. (a) Draw CFG fragment for **(7)**

(i) Simple *if* (ii) Simple *while* loop (iii) Simple *for* loop

(b) Explain the following concepts with examples? **(7)**

(i)Call graph (ii) Inheritance graph (iii) Coupling du-pairs

17. (a) What are the four important steps in functional testing? **(7)**

(b) Briefly explain input domain modelling approaches? **(7)**

OR

18. (a) Consider the triangle classification program with a specification: **(6)**

The program reads floating values from the standard input. The three values *A*, *B*, and *C* are interpreted as representing the lengths of the sides of triangle. The program then prints a message to the standard output that states whether the triangle, if it can be formed, is scalene, isosceles, equilateral, orright angled. Determine the following for the above program:

(i) For the boundary condition $A + B > C$ case (scalene triangle), identify test cases to verify the boundary.

(ii) For the boundary condition $A = C$ case (isosceles triangle), identify testcases to verify the boundary.

(iii) For the boundary condition $A = B = C$ case (equilateral triangle), identify testcases to verify the boundary.

(b) Develop a decision table to generate test cases for this specification. **(8)**

19. (a) Explain the importance of grey box testing, its advantages and disadvantages? **(9)**

(b) Explain the concept of symbolic execution tree? **(5)**

OR

20. (a) Consider the code fragment given below: - **(7)**

```
1. POWER: PROCEDURE(X, Y);
2. Z ← 1;
```

```
3. J ← 1;
4. LAB:   IF Y ≥ J THEN
5. DO; Z← Z * X;
6. J ← J + 1;
7. GO TO LAB;  END;
8. RETURN (Z) ;
9. END;
```

a) Explain Symbolic execution of POWER (αl, α2).

(b) Explain Execution tree for POWER (αl, α2). **(7)**

TEACHING PLAN

No	**Contents**	**No of Lecture Hrs (35 hrs)**
	Module 1 (Introduction to Software Testing) -(7 Hours)	
1.1	Some Popular Errors– Ariane 5, Therac 25, Intel Pentium Bug.	1 Hour
1.2	What is Software testing? Why should it be tested? Software Quality, Role of Testing.	1 Hour
1.3	Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking.	1 Hour
1.4	Software Testing Terminologies- Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria.	1 Hour
1.5	Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing	1 Hour
1.6	Functional testing, Stress testing, Performance testing, Usability testing and Regression testing.	1 Hour
1.7	Testing Methods - Black Box testing, White Box testing, Grey Box testing.	1 Hour
	Module 2 (Unit testing)- (6 Hours)	
2.1	Concept of Unit testing, Static Unit Testing	1 Hour

2.2	Dynamic Unit testing - Control Flow testing, Data Flow testing, Domain testing, Functional Program testing.	1 Hour
2.3	Mutation testing - Mutation and Mutants, Mutation operators, Mutation score.	1 Hour
2.4	Junit - Framework for Unit testing.	1 Hour
2.5	Case Study - Mutation testing using Junit	1 Hour
2.6	Case Study - Mutation testing using Muclipse	1 Hour
	Module 3 (Unit Testing:- White Box Approaches)- (8 Hours)	
3.1	Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage	1 Hour
3.2	Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage.	1 Hour
3.3	Data Flow Criteria - du paths, du pairs	1 Hour
3.4	Subsumption Relationships among Graph Coverage Criteria	1 Hour
3.5	Graph Coverage for Source Code – Control Flow Graphs (CFG) for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program - Statistics	1 Hour
3.6	Graph Coverage for Design Elements – Structural graph coverage and data flow graph coverage for design elements	1 Hour
3.7	Case Study - Graph Based testing using JUnit Framework. (Lecture 1)	1 Hour
3.8	Case Study - Graph Based testing using JUnit Framework. (Lecture 2)	1 Hour
	Module 4 (Unit Testing:- Black Box Approaches) -(7 Hours)	
4.1	Domain Testing / Input Space Partitioning - Partitions of a set.	1 Hour
4.2	Input domain modelling - Interface-based approach, Functionality-based approach.	1 Hour

4.3	Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage.	1 Hour
4.4	Functional Testing - Functional Testing Concepts of Howden. Important Steps.	1 Hour
4.5	Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis	1 Hour
4.6	Decision Tables, Random Testing.	1 Hour
4.7	Case Study - Black Box testing approaches using JUnit.	1 Hour
Module 5 (Grey Box Testing Approaches)- (7 Hours)		
5.1	Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages.	1 Hour
5.2	Techniques of Grey Box Testing - Matrix Testing, Regression Testing, Orthogonal Array Testing or OAT, Pattern Testing.	1 Hour
5.3	An Introduction to Pex - Parameterized Unit Testing, The Testing Problem.	1 Hour
5.4	Symbolic Execution – Example, Symbolic execution tree.	1 Hour
5.5	Case Study – PEX (Lecture 1)	1 Hour
5.6	Case Study – PEX (Lecture 2)	1 Hour
5.7	Case Study – PEX (Lecture 3)	1 Hour

CST468	**BIOINFORMATICS**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**	**YEAR OF INTRODUCTION**
		PEC	**2**	**1**	**0**	**3**	**2019**

Preamble: This course helps the learners to understand the fundamental concepts in Molecular Biology, Genomics, Proteomics and Modelling. This course introduces bio macromolecules such as genes and proteins, different biological databases, and tools and algorithms for biological data processing, analysis and interpretation, and the elements of the systems approach to Molecular Biology. This course enables the learners to contribute towards drug discovery and computational analysis and modelling of biological process.

Prerequisite: Basic background in higher secondary biology

Course Outcomes: After the completion of the course, the student will be able to

CO 1	Describe the basic concepts of Bioinformatics with an emphasis on structure, function and synthesis of biomolecules (**Cognitive knowledge level : Understand**)
CO 2	Identify biological data formats and databases, retrieve bio-sequences, and align bio-sequences to identify similarity (**Cognitive knowledge level : Apply)**
CO 3	Employ similarity searching tools and algorithms to align sequences to highlight the similarity, and describe the structure of genes (**Cognitive knowledge level : Apply)**
CO 4	Demonstrate Protein Structure, visualize protein structure using tools, and explain how proteins interact (**Cognitive knowledge level : Apply)**
CO 5	Explain the fundamental aspects of Systems Biology, Computational Modeling and properties of models (**Cognitive knowledge level : Understand**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓	✓							✓

CO4	✓	✓	✓	✓								✓
CO5	✓	✓			✓							✓

PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	Test1 (%)	**Test2 (%)**	
Remember	**30**	**30**	**30**
Understand	**50**	**50**	**50**
Apply	**20**	**20**	**20**
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests (Average of Series Tests 1& 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module-1 (Introduction to bioinformatics)

Introduction to bioinformatics, Nature & Scope of Bioinformatics, DNA, RNA, and Protein: The Central Dogma, Messenger RNA, tRNA, rRNA, Genetic code, Gene Structure and Control, Transcription, translation

Module-2 (Introduction to bio sequences and analysis)

Introduction to Biological Databases, NCBI, Genbank, Bio sequence formats- FASTA, Sequence alignment- Global Alignment and Local Alignment, Dot Matrix Method, Dynamic Programming Method, Gap Penalties, Amino Acid Scoring Matrices - PAM and BLOSUM

Module-3 (Database Similarity Searching and genomics)

Database Similarity Searching, BLAST – Variants -BLASTN, BLASTP, BLASTX, Statistical Significance, Needleman and Wunsch and Smith–Waterman Method, Multiple Sequence Alignment, scoring function, Clustal, introduction to structure of prokaryotic and eukaryote gene

Module-4 (Proteomics)

Protein Structure, Ramachandran Plot, Hierarchies of Protein Structure, Determination of Protein three-dimensional structure, protein structure database-PDB, Protein structure visualization, introduction to Protein protein interaction, STRING database

Module-5 (Systems Biology)

Introduction to Systems Biology, Models and Modelling, Properties of models, Systems state and steady state, Variables, Parameters, and Constants in modelling, Purpose and Adequateness of Models, Advantages of Computational Modelling, Model Development, Network Versus Elements, Modularity, Robustness and Sensitivity, Data Integration

Text books

1. Zvelebil, Marketa J., and Jeremy O. Baum. *Understanding bioinformatics*. Garland Science, 2007.
2. Xiong, Jin. *Essential bioinformatics*. Cambridge University Press, 2006.
3. Klipp, E., Herwig, R., Kowald, A., Wierling, C., &Lehrach, H. *Systems biology in practice: concepts, implementation and application*. John Wiley & Sons. 2005

References

1. Baxevanis, Andreas D., Gary D. Bader, and David S. Wishart, eds. *Bioinformatics*. John Wiley & Sons, 2020.
2. Shaik, Noor Ahmad, et al. *Essentials of Bioinformatics, Volume I.* Springer, 2019

3. Selzer, Paul M., Richard J. Marhöfer, and Andreas Rohwer, *Applied bioinformatics. An introduction–Springer, Verlag,*, 2008.
4. S C Rastogi, N Mendiratta and PRastogi, *Bioinformatics: Methods and Applications* , PHI Learning Private Limited, New Delhi, 2015.
5. D E Krane and M L Raymer, *Fundamental Concepts of Bioinformatics*, Pearson Education, 2006.
6. Andreas D.Baxevanis, B F Francis Ouellette, *Bioinformatics - A Practical Guide to the Analysis of Genes and Proteins*, Third Edition, John Wiley & Sons INC. , U.K. 2006
7. Neil C Jones and Pavel A Pevzner, *An Introduction to Bioinformatics Algorithms*, MIT press, 2004.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare and contrast the DNA and RNA on the basis of structure and functions.
2. Demonstrate with the help of a flow diagram the generation of protein using the transcription and translation process.

Course Outcome 2 (CO2):

1. Download DNA sequence of human insulin form NCBI
2. Identify the following qualifiers for GenBank and give their definitions: [ACCN], [ALL], [AUTH], [ECNO], [FKEY], [GENE], [JOUR], [KYWD]
3. Construct a dot plot and find the sequence alignment between the following two sequences:
   Sequence1: GATTCTATCTAACTA, Sequence2: GTTCTATTCTAAC

Course Outcome 3 (CO3):

1. Apply Needleman-Wunsch Algorithm to perform sequence alignment for the following sequences: CGTGAATTCAT (sequence #1), GACTTAC (sequence #2)
2. Construct a BLAST procedure for sequence alignment(HSP) if a sequence and its corresponding database sequence are given. Assume the necessary data and demonstrate the procedure.

Course Outcome 4 (CO4):

1. Differentiate between the different protein molecular structure visualizations. Also mention the advantages and uses of each visualization technique.
2. Make use of an example and demonstrate the steps in protein comparison. Show how root mean square deviationis calculated while comparing two proteins.

Course Outcome 5 (CO5):

1. Explain how systems biology is used in data integration.
2. Explain the process of model development.

Model Question Paper

QP CODE:

Reg No: _______________

Name: __________________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST468

Course Name: Bioinformatics

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Differentiate DNA, Gene, genome and chromosome.
2. What are the functions of mRNA, tRNA and rRNA?
3. What do you mean by Gene expression?
4. Write difference between local and global alignment.
5. Write short note on Gap penalties and its usage in comparing Biological sequences.
6. List any three typesof BLAST and make short description on each.
7. What are the principle underlying the formation of Ramachandran plot?.
8. What are the experimental methods for determining protein structure?
9. What do you mean by steady state in a biological system.
10. Justify the statement - systems are modular. **(10x3=30)**

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) What is the central dogma of molecular biology? **(6)**

    (b) Explain the steps involved in the process of transcription. How is the primary transcript produced by a prokaryote different from that produced by a eukaryotic cell? **(8)**

OR

12. (a) Discuss translation process in protein synthesis. **(6)**

(b) Explain bio-molecules involved in central dogma, its structure and types. **(8)**

13. (a) Explain the importance of Primary and secondary databases in Bioinformatics **(6)**

(b) Illustrate the methods of pairwise sequence alignment. What is the use of assigning gap penalties in alignment? **(8)**

OR

14. (a) Illustrate sequence alignment. What are the applications of sequence alignment in Bioinformatics? **(7)**

(b) What is the use of scoring matrices? Differentiate between PAM and BLOSUM matrices and its usage in alignment. **(7)**

15. (a) Using Needleman and Wunsch dynamic programming method, construct the partial alignment score table for the following two sequences, using the scoring parameters: match score: +5, mismatch score: -1, gap penalty: -2. **(9)**
CCATGCU
GATTACA
Also write down the optimal global alignment between these sequences along with the optimal score.

(b) Interpret the blast result and statistical significance of the alignment by analyzing the results. **(5)**

OR

16. (a) Using Smith Waterman method construct the partial alignment scoring table and obtain the optimal local alignment of the following two sequences: **(9)**
ACGTATCGCGTATA
GATGCTCTCGGAJAA

(b) Illustrate multiple sequence alignment. **(5)**

17. (a) Discuss hierarchies of protein structure. **(6)**

(b) Explain how the protein structure is determined by using experimental techniques. **(8)**

OR

18. (a) Discuss protein interaction. How it contributes to the complexity of an organism? **(9)**

(b) Discuss on Protein Structure Database. **(5)**

19. (a) Discuss systems biology approach of understanding complex biological systems. **(6)**

(b) Explain on Variables, Parameters, and Constants in modeling biological systems. **(8)**

OR

20. (a) Explain on advantages of Computational Modeling of biological system. **(7)**

(b) What are the properties of models in biological system? **(7)**

TEACHING PLAN

No	Contents	No of Lecture (36 Hrs)
Module-1 (Introduction to bioinformatics)(8 hrs) Text 1 (Relevant topics from chapter 1.1, 1.2, 1.3)		
1.1	Introduction to bioinformatics	1
1.2	Nature & Scope of Bioinformatics	1
1.3	DNA, RNA, and Protein	1
1.4	The Central Dogma introduction	1
1.5	Messenger RNA, tRNA, rRNA,	1
1.6	Genetic code,	1
1.7	Gene Structure and Control	1
1.8	Transcription, Translation	1
Module-2 (Introduction to bio sequences and analysis) (7 hrs) Text 2 (Relevant topics from chapter 2, 3)		
2.1	Introduction to Biological Databases	1
2.2	NCBI Sequence retrieval	1
2.3	Genbank, Bio sequence formats- FASTA	1
2.4	Sequence alignment- Global Alignment and Local Alignment	1
2.5	Dot Matrix Method, Dynamic Programming Method	1

2.6	Gap Penalties	1
2.7	Amino Acid Scoring Matrices – PAM, BLOSUM	1
Module-3 (Database Similarity Searching and genomics) (7 hrs) Text 2 (Relevant topics from chapter 4 5 and 8)		
3.1	Database Similarity Searching, BLAST, Variants of BLAST - BLASTN, BLASTP, BLASTX	1
3.2	BLAST Analysis - Statistical Significance	1
3.3	Needleman and Wunsch Method	1
3.4	Smith–Waterman Method	1
3.5	Multiple Sequence Alignment, scoring function	1
3.6	Clustal tool	1
3.7	Gene Structure of prokaryotic, eukaryote	1

Module-4 (Proteomics) (7 hrs) Text 2 (Relevant topics from chapter 12, 13 and 19)		
4.1	Protein Structure, Ramachandran Plot	1
4.2	Hierarchies of Protein Structure	1
4.3	Determination of Protein three-dimensional structure	1
4.4	protein structure database-PDB	1
4.5	Protein structure visualization	1
4.6	Protein protein interaction	1
4.7	Protein protein interaction networks, STRING database	1
Module-5 (Systems Biology) (7 hrs) Text 3 (Relevant topics from Section 1.1-1.4)		
5.1	Introduction to Systems Biology, Properties of models	1
5.2	Systems state and steady state	1
5.3	Variables, Parameters, and Constants in modelling	1
5.4	Purpose and Adequateness of Models	1
5.5	Advantages of Computational Modelling ,Model Development (introduction only)	1
5.6	Network Versus Elements, Modularity,	1

5.7	Robustness and Sensitivity, Data Integration	1

CST478	COMPUTATIONAL LINGUISTICS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2019

Preamble: The course aims to teach the basics of Computational Linguistics to the students viewing language phenomena from a computational/statistical standpoint. This involves ideas about statstical and computational models and how these could be linked with various language processing tasks. The course helps the learner to appreciate the complexities involved in language processing tasks using a machine, in contrast with the ease with which human beings handle them. Some practical aspects are also discussed using the Python and NLTK framework to equip the student with the capability to design solutions to linguistic problems.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Explain the fundamental concepts of language processing **(Cognitive Knowledge Level: Understand)**
CO2	Demonstrate the concepts of probability, statistical inference and hidden Markov model. **(Cognitive Knowledge Level: Apply)**
CO3	Compare and summarize the various methods of word sense disambiguation, lexical acquisition and selectional preferences. (**Cognitive Knowledge Level: Apply**)
CO4	Make use of different Part-of-Speech Tagging methods for language modelling. **(Cognitive Knowledge Level: Apply)**
CO5	Examine Probabilistic Context Free Grammars and various probabilistic parsing methods **(Cognitive Knowledge Level: Apply)**
CO6	Develop simple systems for linguistic tasks using Python and NLTK. **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓											✓
CO2	✓	✓										✓
CO3	✓	✓	✓									✓
CO4	✓		✓		✓							✓
CO5	✓	✓	✓									✓
CO6	✓	✓	✓	✓	✓							✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test 1 (%)	Test 2 (%)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyze			
Evaluate			

Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3Hrs

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Continuous Assessment Tests(Average of SeriesTests1& 2)	**25 marks**
Continuous Assessment Assignment	**15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 full questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module- 1 (Preliminaries)

Introduction: Rationalist and Empiricist Approaches to Language-Questions that linguistics should answer-Noncategorical phenomena in language-Language and cognition as probabilistic phenomena

The Ambiguity of Language: Why natural language processing is difficult-Lexical resources-Word counts-Zipf's laws-Collocations-Concordances

Linguistic Essentials:

Parts of Speech and Morphology -Nouns and pronouns-Words that accompany nouns: Determiners and adjectives-Verbs-Other parts of speech-Phrase Structure-Phrase structure grammars **-**Semantics and Pragmatics-Corpus Based Work

Module -2 (Mathematical Essentials:)

Probability Theory-Probability spaces-Conditional probability and independence-Bayes' theorem-Random variables-Expectation and variance-Notation-Joint and conditional distributions-Standard distributions-Bayesian statistics

Statistical Inference: n-gram Models over Sparse Data-Bins: Forming Equivalence Classes-Reliability vs discrimination-n gram models

Markov Models-Hidden Markov Models-Why use HMMs?-General form of an HMM-Finding the probability of an observation-Finding the best state sequence

Module -3 (Word Sense Disambiguation)

Methodological Preliminaries- Supervised and unsupervised learning-Pseudowords-Upper and lower bounds on performance-Supervised Disambiguation-Bayesian classification-Dictionary based Disambiguation-Disambiguation based on sense definitions-Thesaurus based disambiguation

Lexical Acquisition-Evaluation Measures-Verb Subcategorization -Attachment Ambiguity-PP attachment- Selectional Preferences

Semantic Similarity: Vector space measures-Probabilistic measures

Module -4 (Grammar)

Part-of-Speech Tagging-The Information Sources in Tagging-Markov Model Taggers-Hidden Markov Model Taggers-Applying HMMs to POS tagging-The effect of initialization on HMM training-Transformation Based Learning of Tags

Probabilistic Context Free Grammars-Some Features of PCFGs-Questions for PCFGs -The Probability of a String -Using inside probabilities-Using outside probabilities-Finding the most likely parse for a sentence-parsing for disambiguation-parsing model versus language model

Module -5 (Language Processing with Python)

Introduction to NLTK, Text Wrangling and Text cleansing : Sentence Splitter, Tokenization, Stemming, Lemmatization, Stop word removal , Rare word Removal, Spell Correction. Part of Speech Tagging and NER. Parsing Structure in Text: Shallow versus deep parsing, different types of parsers and dependency parsing.

Text Books :

1. C.D. Manning and H. Schutze. Foundations of Statistical Natural Language Processing. MIT Press.
2. Steven Bird, |Ewan Klein, |Edward Loper, Natural Language Processing with Python and NLTK. O'reilly Pub.

References:

1. D. Jurafsky and J.H. Martin: Speech and Language Processing: Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, PHI.
2. James Allen: Natural Language Understanding. Pearson Pub.
3. Nitin Hardeniya, Jacob Perkins, Deepti Chopra, Nisheeth Joshi, ItiMathur: Natural Language Processing: Python and NLTK., **1**stEdition. Packt Publishing

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What do you understand by the term *collocations?* List their properties.
2. Define the term phrase structure grammar formally.

Course Outcome 2 (CO2):

1. State Bayes' theorem and explain briefly. Comment on it's usefulness in NLP.
2. How can n-grams be used to model natural language statistically?

Course Outcome 3 (CO3):

1. What is meant by attachment ambiguity? Show it using English sentences
2. What is meant by Word Sense Disambiguation (WSD)? Outline any one WSD algorithm

Course Outcome 4 (CO4):

1. How can HMM be used for Parts of speech tagging?
2. Outline an implementation procedure for HMM

Course Outcome 5 (CO5):

1. Show with an example how can probabilistic grammars be used to model human preferences in parsing.
2. Give the technique of Transformation-Based Learning of Tags

Course Outcome 6 (CO6):

1. Implement a python program for stop word removal in a simple paragraph.
2. Write a code to access a weather site and extract the forecast top temperature for your town or city today.

Model Question Paper

QP CODE:

Reg No: _______________

Name: __________________ **PAGES : 3**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST478

Course Name: Computational Linguistics

Max. Marks : 100 **Duration: 3 Hours**

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Define Zipf's law.
2. List the uses of a corpus in language processing?
3. What is a Hidden Markov Model?
4. State Bayes' theorem and explain briefly. Comment on its usefulness in NLP.
5. What is meant by supervised disambiguation? What are its prerequisites ?
6. Consider the sentence: "the children ate the cake with a spoon". Construct the parse tree for it and explain the attachment ambiguity.
7. Discuss the properties of Markov chain useful in POS tagging.
8. Explain the features of PCFG.
9. What is NLTK? How is it useful in text processing ?
10. Write a Python program to extract different date formats from a text document.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Write a note on the following terms with example: **(9)**
(i) Collocations (ii) Concordances (iii) Phrase structure grammars

(b) Differentiate stemming and lemmatization with examples. **(5)**

OR

12. (a) Write a note on all parts of speech tags of English language **(9)**

(b) What are the differences between Rationalist and Empiricist to Language approaches **(5)**

13. (a) What do you mean by a probability distribution? **(5)**
What are the approaches used in SNLP to estimate probability distribution of linguistic events?

(b) Give a formal definition of Hidden Markov Model (HMM) and state the relevant assumption while using HMM for language modeling **(9)**

OR

14. (a) Assume that a particular type of syntactic error detected by a system A occurs once in 1,00,000 sentences on an average. This system detects an error correctly with a probability 0.05. Suppose the system reports an error in a test sentence. What is the probability that this is true? **(5)**

(b) List some of the problems associated with sparse data in SNLP. **(9)**
Write a note on n-gram Models over Sparse Data

15. (a) What do you understand by Disambiguation based on sense definitions. **(9)**
Write and explain any one algorithm for this.

(b) With the help of Bayes' rule, explain the Bayesian disambiguation algorithm. **(5)**

OR

16. (a) Write a note on selectional preferences with an example **(5)**

(b) What is meant by attachment ambiguity? List different attachment issues. **(9)**

17. (a) Write a note on Transformation-Based Learning of tags. Give it's algorithm **(9)**

(b) How can HMM be used for parts of speech tagging **(5)**

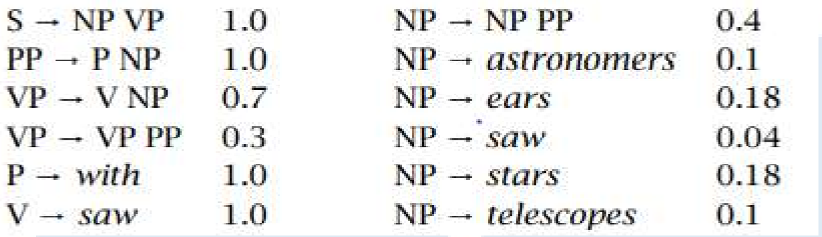
OR

18. (a) Write the formal definition of PCFG. **(5)**

Apply probabilistic parsing on the following sentence and find the correct parsing using the given grammar

Sentence: Astronomers saw stars with ears.

Probabilistic grammar:



S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → *astronomers*	0.1
VP → V NP	0.7	NP → *ears*	0.18
VP → VP PP	0.3	NP → *saw*	0.04
P → *with*	1.0	NP → *stars*	0.18
V → *saw*	1.0	NP → *telescopes*	0.1

(b) How do you find the probability of a string using inside and outside probabilities ? **(9)**

19. (a) Write a Python program for PoS tagging using the necessary Python packages. **(9)**

(b) Explain the process of Named Entity Recognition. **(5)**

List its uses and challenges involved.

OR

20. (a) Write a regular expression for removing punctuations, numbers and white spaces in a piece of text. **(9)**

(b) Write a Python program to count the number of sentences, words and line numbers in a given piece of text. Display each sentence along with that. **(5)**

TEACHING PLAN

No	Contents	No of Lecture Hrs (36 hrs)
	Module - 1 (Preliminaries) (9 hrs)	
1.1	Introduction: Rationalist and Empiricist Approaches to Language-Questions that linguistics should answer-	1
1.2	Non-categorical phenomena in language-Language and cognition as probabilistic phenomena	1
1.3	The Ambiguity of Language: Why natural language processing is difficult	1
1.4	Lexical resources-Word counts	1
1.5	Zipf's laws-Collocations-Concordances	1
1.6	Linguistic Essentials: Parts of Speech and Morphology -Nouns and pronouns	1
1.7	Words that accompany nouns: Determiners and adjectives-Verbs-Other parts of speech	1
1.8	Phrase Structure-Phrase structure grammars	1
1.9	Semantics and Pragmatics-Corpus Based Work	1
	Module – 2 (Mathematical Essentials) (7 hrs)	
2.1	Probability Theory-Probability spaces	1
2.2	Conditional probability and independence-Bayes' theorem	1
2.3	Random variables-Expectation and variance-Notation	1
2.4	Joint and conditional distributions-Standard distributions- Bayesian statistics	1
2.5	Statistical Inference: n-gram Models over Sparse Data-Bins: Forming Equivalence Classes	1
2.6	Markov Models-Hidden Markov Models: Why use HMMs?	1
2.7	General form of an HMM-Finding the probability of an observation-Finding the best state sequence	1
	Module – 3 (Word Sense Disambiguation) (7 hrs)	
3.1	Methodological Preliminaries-Supervised and unsupervised learning	1
3.2	Upper and lower bounds on performance-Supervised Disambiguation	1
3.3	Bayesian classification-Dictionary based Disambiguation-	1
3.4	Disambiguation based on sense definitions-Thesaurus based disambiguation	1
3.5	Lexical Acquisition-Evaluation Measures	1

3.6	Verb Subcategorization-Attachment Ambiguity, PP attachment-Selectional Preferences	1
3.7	Semantic Similarity:Vector space measures-Probabilistic measures	1

	Module – 4 (Grammar) (8 hrs)	
4.1	Part-of-Speech Tagging-The Information Sources in Tagging	1
4.2	Markov Model Taggers-Hidden Markov Model Taggers-	1
4.3	Applying HMMs to POS tagging-The effect of initialization on HMM training-	1
4.4	Transformation-Based Learning of Tags	1
4.5	Probabilistic Context Free Grammars-Some Features of PCFGs	1
4.6	Questions for PCFGs	1
4.7	The Probability of a String -Using inside probabilities Using outside probabilities	1
4.8	Finding the most likely parse for a sentence-parsing for disambiguation, parsing model *vs.* language model	1
	Module - 5 (Language Processing with Python) (5 hrs)	
5.1	Introduction to NLTK	1
5.2	Text Wrangling and Text cleansing : Sentence Splitter, Tokenization, Stemming,	1
5.3	Lemmatization, Stop word removal , Rare word Removal, Spell Correction.	1
5.4	Part of Speech Tagging and NER.	1
5.5	Parsing Structure in Text: Shallow versus deep parsing, types of parsers	1

SEMESTER VIII

MINOR

CSD482	MINI PROJECT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	0	3	4	2019

Preamble: The objective of this course is to apply the fundamental concepts of different courses learned in respective Minor Streams: Software Engineering, Machine Learning and Networking. This course helps the learners to get an exposure to the development of application software/hardware solutions/ software simulations in the field of Computer Science and Engineering. It enables the learners to understand the different steps to be followed such as literature review and problem identification, preparation of requirement specification &design document, testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

Prerequisite: A sound knowledge in courses studied in respective minor stream.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Identify technically and economically feasible problems **(Cognitive Knowledge Level: Apply)**
CO2	Identify and survey the relevant literature for getting exposed to related solutions. **(Cognitive Knowledge Level: Apply)**
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (**Cognitive Knowledge Level: Apply)**
CO4	Prepare technical report and deliver presentation **(Cognitive Knowledge Level: Apply)**
CO5	Apply engineering and management principles to achieve the goal of the project **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Mark Distribution

Total Marks	CIE Marks	ESE Marks
150	75	75

Continuous Internal Evaluation Pattern:

Attendance **10 marks**

Project Guide **15 marks**

Project Report **10 marks**

Evaluation by the Committee (will be evaluating the level of completion
and demonstration of functionality/specifications, presentation,
oral examination, work knowledge and involvement) : **40 marks**

Student Groups with 4 or 5 members should identify a topic of interest in consultation with a Faculty Advisor/Project Coordinator/Guide. Review the literature and gather information pertaining to the chosen topic. State the objectives and develop a methodology to achieve the objectives. Carryout the design/fabrication or develop codes/programs to achieve the objectives by strictly following steps specified in the teaching plan. Innovative design concepts, performance, scalability, reliability considerations, aesthetics/ergonomic, user experience and security aspects taken care of in the project shall be given due weight.

The progress of the mini project is evaluated based on a minimum of two reviews. The review committee may be constituted by a senior faculty member, Mini Project coordinator and project guide. The internal evaluation shall be made based on the progress/outcome of the project, reports and a viva-voce examination, conducted internally by a 3-member committee. A project report is required at the end of the semester. The project has to be demonstrated for its full design specifications.

End Semester Examination Pattern:

The marks will be distributed as

Presentation	: **30 marks**
Demo	: **20 marks**
Viva	: **25 marks.**
Total	: **75 marks**.

TEACHING PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc.
3. Create Requirements Specification
4. Create Design Document . This may include designs like,
    a. System Architecture Design
    b. Application Architecture Design
    c. GUI Design
    d. API Design
    e. Database Design
    f. Technology Stack
5. Deployment, Test Run & Get Results
6. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman18, Bold; Heading 2 – Times New Roman16, Bold; Heading 3 – Times New Roman14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.
- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figure title under the figure and table title above the table.

- **Suggestive order of documentation:**

  i. Top Cover
  ii. Title page
  iii. Certification page
  iv. Acknowledgement
  v. Abstract
  vi. Table of Contents
  vii. List of Figures and Tables
  viii. Chapters
  ix. Appendices, if any
  x. References/Bibliography

SEMESTER VIII

HONOURS

CSD496	MINI PROJECT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	0	3	2	2019

Preamble: The objective of this course is to apply the fundamental concepts of courses learned in respective Honors Streams: Security in Computing, Machine Learning and Formal Methods. This course helps the learners to get an exposure to the development of application software/hardware solutions/ software simulations in the field of Computer Science and Engineering. It enables the learners to understand the different steps to be followed such as literature review and problem identification, preparation of requirement specification &design document, testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

Prerequisite: A sound knowledge in courses studied in respective honor stream.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Identify technically and economically feasible problems **(Cognitive Knowledge Level: Apply)**
CO2	Identify and survey the relevant literature for getting exposed to related solutions. **(Cognitive Knowledge Level: Apply)**
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (**Cognitive Knowledge Level: Apply)**
CO4	Prepare technical report and deliver presentation **(Cognitive Knowledge Level: Apply)**
CO5	Apply engineering and management principles to achieve the goal of the project **(Cognitive Knowledge Level: Apply)**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓

Abstract POs defined by National Board of Accreditation			
PO#	**Broad PO**	**PO#**	**Broad PO**
PO1	Engineering Knowledge	**PO7**	Environment and Sustainability
PO2	Problem Analysis	**PO8**	Ethics
PO3	Design/Development of solutions	**PO9**	Individual and team work
PO4	Conduct investigations of complex problems	**PO10**	Communication
PO5	Modern tool usage	**PO11**	Project Management and Finance
PO6	The Engineer and Society	**PO12**	Lifelong learning

Assessment Pattern

Mark Distribution

Total Marks	CIE Marks	ESE Marks
150	**75**	**75**

Continuous Internal Evaluation Pattern:

Attendance	**10 marks**
Project Guide	**15 marks**
Project Report	**10 marks**
Evaluation by the Committee (will be evaluating the level of completion and demonstration of functionality/specifications, presentation, oral examination, work knowledge and involvement)	: **40 marks**

Student Groups with 4 or 5 members should identify a topic of interest in consultation with a Faculty Advisor/Project Coordinator/Guide. Review the literature and gather information pertaining to the chosen topic. State the objectives and develop a methodology to achieve the objectives. Carryout the design/fabrication or develop codes/programs to achieve the objectives by strictly following steps specified in the teaching plan. Innovative design concepts, performance, scalability, reliability considerations, aesthetics/ergonomic, user experience and security aspects taken care of in the project shall be given due weight.

The progress of the mini project is evaluated based on a minimum of two reviews. The review committee may be constituted by a senior faculty member, Mini Project coordinator and project guide. The internal evaluation shall be made based on the progress/outcome of the project, reports and a viva-voce examination, conducted internally by a 3-member committee. A project report is required at the end of the semester. The project has to be demonstrated for its full design specifications.

End Semester Examination Pattern:

The marks will be distributed as

Presentation	: **30 marks**
Demo	: **20 marks**
Viva	: **25 marks.**
Total	: **75 marks**.

TEACHING PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc.
3. Create Requirements Specification
4. Create Design Document . This may include designs like,
    a. System Architecture Design
    b. Application Architecture Design
    c. GUI Design
    d. API Design
    e. Database Design
    f. Technology Stack
5. Deployment, Test Run & Get Results
6. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman18, Bold; Heading 2 – Times New Roman16, Bold; Heading 3 – Times New Roman14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.

- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.
- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figuretitle under the figure and table title above the table.

- **Suggestive order of documentation:**

  i. Top Cover
  ii. Title page
  iii. Certification page
  iv. Acknowledgment
  v. Abstract
  vi. Table of Contents
  vii. List of Figures and Tables
  viii. Chapters
  ix. Appendices, if any
  x. References/Bibliography