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)$(\exists y)\,(M(y) \wedge \neg W(y))$ the conclusion $(x)(F(x) \rightarrow \neg S(x))$ 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

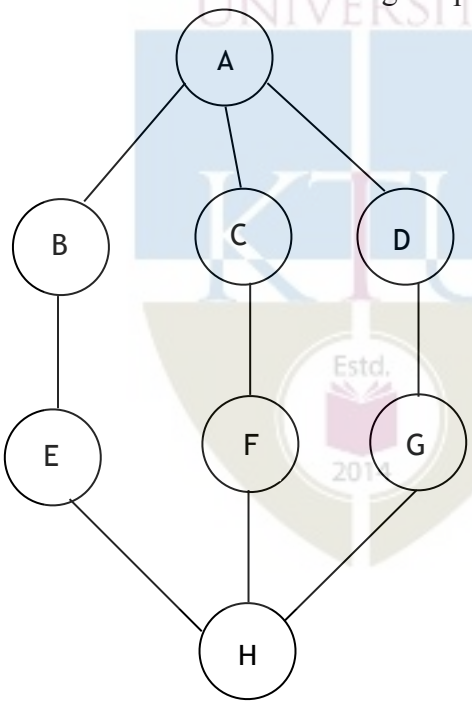
5. Write an algorithm to count the number of occurrences of a character in a linked list (each node contains only one character)

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.

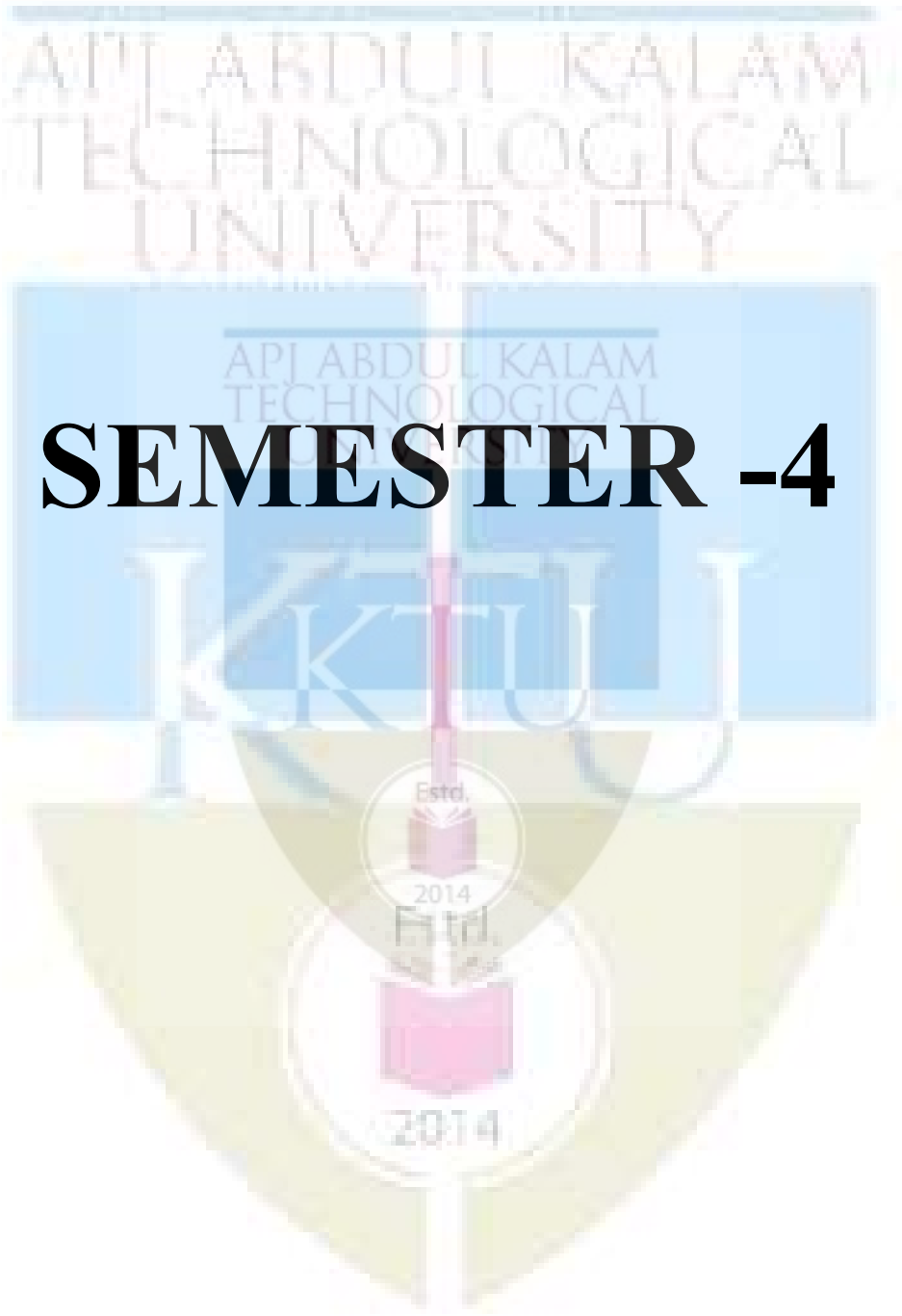
(8)

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):

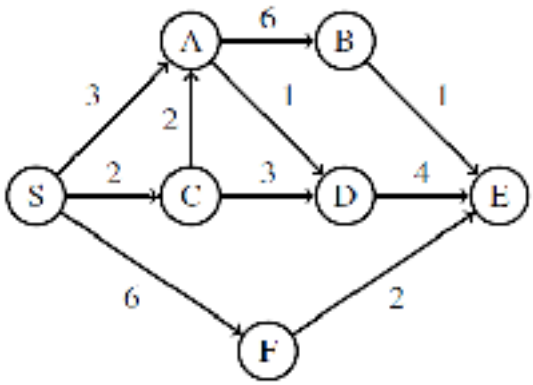
1. Differentiate a walk, path and circuit in a graph.
2. Is it possible to construct a graph with 12 vertices such that two of the vertices have degree 3 and the remaining vertices have degree 4? Justify
3. Prove that a simple graph with n vertices must be connected, if it has more than $\frac{(n-1)(n-2)}{2}$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

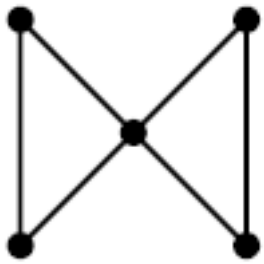
Course Name: GRAPH THEORY

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)

G

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)

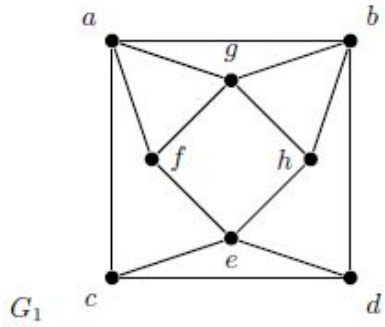
PART B

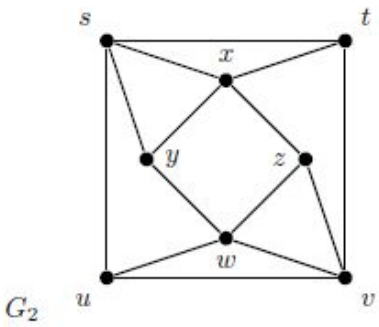
Answer any one Question from each module. Each question carries 14 Marks

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)

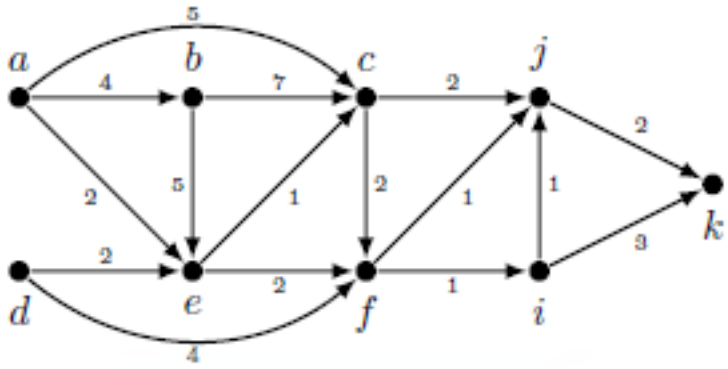
b) i. Let G be a graph that has exactly two connected components, both being Hamiltonian graphs. Find the minimum number of edges that one needs to add to G to obtain a Hamiltonian graph. (6)

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***.



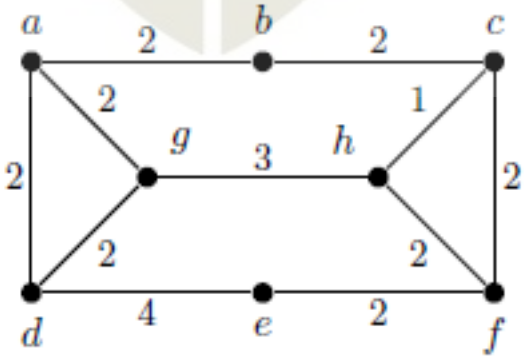
Find the shortest path between ***a*** and every other vertices in ***G*** using Dijkstra's shortest path algorithm.

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.

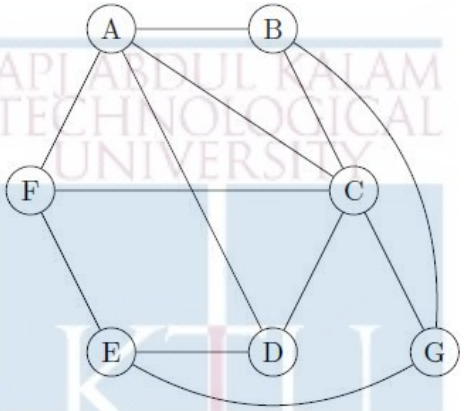
17 a) i. State and prove Euler's Theorem relating the number of faces, edges and vertices for a planar graph. (9)

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)

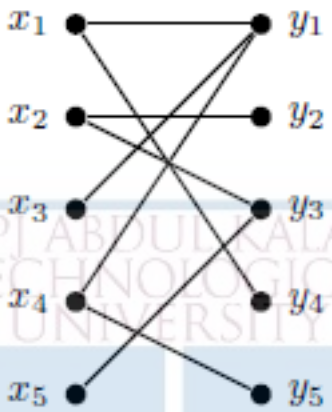
b) Give the definition of the chromatic polynomial $P_G(k)$. Directly from the definition, prove that the chromatic polynomials of W_n and C_n satisfy the identity $P_{Wn}(k) = k\, P_{Cn-1}\,(k-1)$. (7)

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?

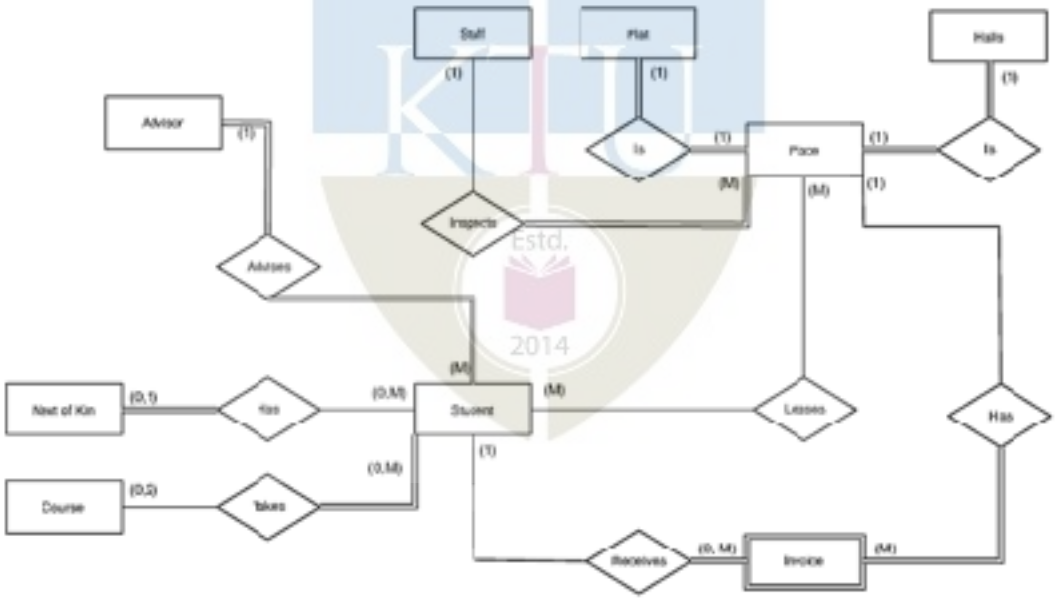
PART B

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)**
DEPARTMENT(DNO, DNAME, DLOCATION, DPHONE, MGRENO)
PROJECT(PNO, PNAME, PLOCATION, PCOST, CDNO) (14)

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(<u>employee-name</u>, street, city)
works(<u>employee-name, company-name</u>, salary)
company(<u>company-name</u>, city)
manages(<u>employee-name, manager-name</u>)

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.

Syllabus

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.

CSL204	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	MATHEMATICS FOR MACHINE LEARNING	CATEGORY	L	T	P	CREDIT
CST284		VAC	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**)

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	√	√	√	√		√						√

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 –Vector Spaces, Linear Independence, Basis and Rank. Linear Mappings – Matrix Representation of Linear Mappings, Basis Change, Image and Kernel.

Module 2

ANALYTIC GEOMETRY, MATRIX DECOMPOSITIONS: Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Orthonormal Basis, Orthogonal Complement, Orthogonal Projections – Projection into One Dimensional Subspaces, Projection onto General Subspaces, Gram-Schmidt Orthogonalization.

Determinant and Trace, Eigenvalues and Eigenvectors, Cholesky Decomposition, Eigen decomposition and Diagonalization, Singular Value Decomposition, Matrix Approximation.

Module 3

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 – Gradients in Deep Network, Automatic Differentiation.Higher Order Derivatives-Linearization and Multivariate TaylorSeries.

Module 4

PROBABILITY AND DISTRIBUTIONS : Construction of a Probability Space - Discrete and Continuous Probabilities, Sum Rule, Product Rule, and Bayes' Theorem. Summary Statistics and Independence – Gaussian Distribution - Conjugacy and the Exponential Family - Change of Variables/Inverse Transform.

Module 5

OPTIMIZATION : Optimization Using Gradient Descent - Gradient Descent With Momentum, Stochastic Gradient Descent. Constrained Optimization and Lagrange Multipliers - Convex Optimization - Linear Programming **-** Quadratic Programming.

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 bySpringer
3. Introduction to Applied Linear Algebra by Stephen Boyd and Lieven Vandenberghe, 2018 published by Cambridge UniversityPress
4. Convex Optimization by Stephen Boyd and Lieven Vandenberghe, 2004 published by Cambridge UniversityPress
5. Pattern Recognition and Machine Learning by Christopher M Bishop, 2006, published bySpringer
6. Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond by Bernhard Scholkopf and Smola, Alexander J Smola, 2002, published by MIT Press
7. Information Theory, Inference, and Learning Algorithms by David J. C MacKay, 2003 published by Cambridge UniversityPress
8. Machine Learning: A Probabilistic Perspective by Kevin P Murphy, 2012 published by MITPress.
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 solution sin$x$ of the following in homogeneous 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. Find the characteristic equation, eigenvalues, and eigenspaces corresponding to each eigenvalue of the following matrix

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

4. Diagonalize the following matrix, if possible

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}$$

5. 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 $P(J \square T)$
   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 followingtable:

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 winsif

i. A answers the firstquestion,

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 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 sufficient conditions under which a quadratic optimization problem be written as a linear least squaresproblem.
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 saddlepoints.
5. Consider the update equation for stochastic gradient descent. Write down the update when we use a mini-batch size ofone.
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 youranswer.

ii. Is (1 , -1) a local/global minimum? Justify youranswer.

7. Is the function $f(x, y) = 2x^2 + y^2 + 6xy - x + 3y - 7$ convex, concave, or neither? Justify youranswer.
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 simplexmethod.

$$max\ 5x_1 + 6x_2 + 9x_3 + 8x_4$$

subject to the constraints

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + x_4 &\leq 5 \\ x_1 + x_2 + 2x_3 + 3x_4 &\leq 3 \end{aligned}$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Model Question paper

	QP Code :		**Total Pages : 5**
Reg No.:_____		Name:_____	

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
IV SEMESTER B.TECH (HONOURS) DEGREE EXAMINATION, MONTH and YEAR

Course Code: CST 284

Course Name: MATHEMATICS FOR FOR MACHINE LEARNING

Max. Marks: 100 | Duration: 3 Hours

PART A

		Answer all questions, each carries3 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		Are the following sets of vectors linearly independent? Explain your answer. $x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$, $x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}$, $x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$	
3		Find the angle between the vectors $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.	
4		Find the eigen values of the following matrix in terms of k. Can you find an eigen vector corresponding to each of the eigen values? $\begin{bmatrix} 1 & k \\ 2 & 1 \end{bmatrix}$	
5		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).**	
6		Compute the Taylor polynomials T_n, $n = 0, \ldots, 5$ of $f(x) = \sin(x) + \cos(x)$ at $x_0 = 0$.	
7		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 $Y = X^2$.	
8		Show that if two events A and B are independent, then A and B' are independent.	
9		Explain the principle of the gradient descent algorithm.	
10		Briefly explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.	

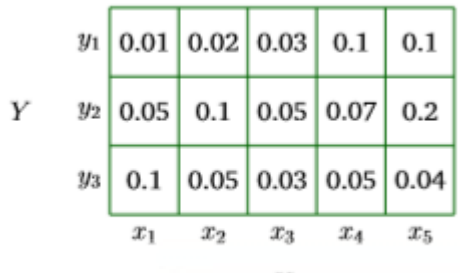
PART B

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

11	a)	i.Find all solutions to the system of linearequations $-4x + 5z = -2$ $-3x - 3y + 5z = 3$ $-x + 2y + 2z = -1$	(4)

		ii. Prove that all vectors orthogonal to $[2,-3,1]^T$ forms a subspace **W** of R^3. What is ***dim (W)*** and why?	(4)
	b)	A set of ***n*** linearly independent vectors in R^n forms a basis. Does the set of vectors **(2, 4,−3),(0, 1, 1),(0, 1,−1)** form a basis for R^3? Explain yourreasons.	(6)
		OR	
12	a)	Find all solutions in $x = \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} \in R^3$ of the equation system $Ax = 12x$, where $A = \begin{bmatrix} 6 & 4 & 3 \\ 6 & 0 & 9 \\ 0 & 8 & 0 \end{bmatrix}$ and $\sum_{i=1}^{3} x_i = 1$.	(7)
	b)	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***.	(7)
13	a)	Use the Gramm-Schmidt process to find an orthogonal basis for the column space of the following matrix. $\begin{bmatrix} 2 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	(7)
	b)	Find the SVD of the matrix. $\begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}$	(7)
		OR	
14	a)	i. Let ***L*** be the line through the origin in 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 ***L*** which is closest to the point **(7, 1)** and find the point on ***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 R^3 consisting of eigenvectors for the following matrix. $\begin{bmatrix} 1 & 0 & -2 \\ 0 & 5 & 0 \\ -2 & 0 & 4 \end{bmatrix}$	(8)
		ii. Finda3×3orthogonalmatrix***S***anda3×3diagonalmatrix***D*** such that $A = SDS^T$	

15	a)	Askierisonamountainwithequation$z=100-0.4x^2-0.3y^2$,where$z$ denotes height. 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 beginskiing. **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**.	(8)
	b)	Find the linear approximation to the function $f(x,y) = 2 - \sin(-x - 3y)$ at the point $(0, \pi)$, and then use your answer to estimate $f(0.001, \pi)$.	(6)
		OR	
16	a)	Let g be the function given by $$g(x,y) = \begin{cases} \dfrac{x^2y}{x^2+y^2} & \text{if } (x,y) \neq (0,0); \\ 0 & \text{if } (x,y) = (0,0). \end{cases}$$ i. Calculate the partial derivatives of g at **(0,0)**. ii. Show that g is not differentiable at **(0,0)**.	(8)
	b)	Find the second order Taylor series expansion for $f(x,y) = e^{-(x2+y2)} \cos(xy)$ about **(0 , 0)**.	(6)
17	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 fruitat 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. What is the probability that the mango was picked from bag 2?	(6)
	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: $p(x \mid \mu) = \mu^x (1-\mu)^{1-x}, \quad x \in \{0,1\}$ Choose a conjugate prior for the Bernoulli likelihood and compute the posterior distribution $p(\mu \mid x_1, \ldots, x_N)$.	(8)
		OR	
18	a)	Two dice are rolled. A = ‘sum of two dice equals 3’ B = ‘sum of two dice equals 7’ C = ‘at least one of the dice shows a 1’	(6)

		i. What is P(A|C)? ii. What is P(B|C)? iii. Are A and C independent? What about B and C?	
	b)	Consider the following bivariate distribution p(x,y) of two discrete random variables X and Y .  Compute: i. The marginal distributions p(x) and p(y). ii. The conditional distributions p(x|Y = y_1) and p(y|X = x_3).	(8)
19	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 $P = \begin{bmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}$, $q = \begin{bmatrix} -22.0 \\ -14.5 \\ 13.0 \end{bmatrix}$, and $r = 1$. Show that $x^* = (1, 1/2, -1)$ is optimal for the optimization problem $\min \quad \frac{1}{2}x^T Px + q^T x + r$ $\text{s.t.} \quad -1 \le x_i \le 1, \; i = 1, 2, 3.$	(6)
		OR	
20	a)	Derive the gradient descent training rule assuming that the target function is represented as $o_d = w_0 + w_1x_1 + \ldots + w_nx_n$. Define explicitly the cost/ error function E, assuming that a set of training examples D is provided, where each training example $d \; D$ 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)
		***	

Teaching Plan		
No	**Topic**	**No. of Lectures (49)**
	Module-I (LINEAR ALGEBRA)	**8**
1.1	Matrices, Solving Systems of Linear Equations	1
1.2	Vector Spaces	1
1.3	Linear Independence	1
1.4	Basis and Rank (Lecture – 1)	1
1.5	Basis and Rank (Lecture – 2)	1
1.6	Linear Mappings	1
1.7	Matrix Representation of Linear Mappings	1
1.8	Images and Kernel	1
	Module-II (ANALYTIC GEOMETRY, MATRIX DECOMPOSITIONS)	**11**
2.1	Norms, Inner Products	1
2.2	Lengths and Distances, Angles and Orthogonality	1
2.3	Orthonormal Basis, Orthogonal Complement	1
2.4	Orthogonal Projections – Projection into One Dimensional Subspaces	1
2.5	Projection onto General Subspaces.	1
2.6	Gram-Schmidt Orthogonalization	1
2.7	Determinant and Trace, Eigen values and Eigenvectors.	1
2.8	Cholesky Decomposition	1
2.9	Eigen decomposition and Diagonalization	1
2.10	Singular Value Decomposition	1
2.11	Matrix Approximation	1
	Module-III (VECTOR CALCULUS)	**9**
3.1	Differentiation of Univariate Functions, Partial Differentiation and Gradients	1
3.2	Gradients of Vector Valued Functions (Lecture 1)	1
3.3	Gradients of Vector Valued Functions (Lecture 2)	1

3.4	Gradients of Matrices	1
3.5	Useful Identities for Computing Gradients	1
3.6	Backpropagation and Automatic Differentiation – Gradients in deep Netwok	1
3.7	Automatic Differentiation	1
3.8	Higher Order Derivatives	1
3.9	Linearization and Multivariate Taylor Series	1
	Module-IV (PROBABILITY AND DISTRIBUTIONS)	**10**
4.1	Construction of a Probability Space	1
4.2	Discrete and Continuous Probabilities (Probability Density Function, Cumulative Distribution Function)	1
4.3	Sum Rule, Product Rule	1
4.4	Bayes' Theorem	1
4.5	Summary Statistics and Independence (Lecture 1)	1
4.6	Summary Statistics and Independence (Lecture 2)	1
4.7	Bernoulli, Binomial, Uniform (Discrete) Distributions	1
4.8	Uniform (Continuous), Poisson Distributions	1
4.9	Gaussian Distribution	1
4.10	Conjugacy and the Exponential Family (Beta – Bernoulli, Beta – Binomial Conjugacies)	1
	Module-V (OPTIMIZATION)	7
5.1	Optimization Using Gradient Descent.	1
5.2	Gradient Descent With Momentum, Stochastic Gradient Descent	1
5.3	Constrained Optimization and Lagrange Multipliers (Lecture 1)	1
5.4	Constrained Optimization and Lagrange Multipliers (Lecture 2)	1
5.5	Convex Optimization	1
5.6	Linear Programming	1
5.7	Quadratic Programming	1

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 $(3, \pm 5)$.

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 $\Phi(mn) = \Phi(m)\,\Phi(n)$.
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, $x \equiv 2 (\mathrm{mod}\ 3)$, $x \equiv 3 (\mathrm{mod}\ 5)$, $x \equiv 2 (\mathrm{mod}\ 7)$ (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 $(3, \pm 5)$. (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 $x^2 - 6y^2 = 1$. (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 CST294	COMPUTATIONAL FUNDAMENTALS FOR MACHINE LEARNING	CATEGORY	L	T	P	CREDIT
		VAC	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**)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	√	√	√	√								√
CO 2	√	√	√									√
CO 3	√	√	√	√								√
CO 4	√	√	√	√		√						√

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 –Vector Spaces, Linear Independence, Basis and Rank. Linear Mappings – Matrix Representation of Linear Mappings, Basis Change, Image and Kernel.

Module 2

ANALYTIC GEOMETRY, MATRIX DECOMPOSITIONS: Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Orthonormal Basis, Orthogonal Complement, Orthogonal Projections – Projection into One Dimensional Subspaces, Projection onto General Subspaces, Gram-Schmidt Orthogonalization.

Determinant and Trace, Eigenvalues and Eigenvectors, Cholesky Decomposition, Eigen decomposition and Diagonalization, Singular Value Decomposition, Matrix Approximation.

Module 3

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 – Gradients in Deep Network, Automatic Differentiation.Higher Order Derivatives-Linearization and Multivariate TaylorSeries.

Module 4

Probability and Distributions : Construction of a Probability Space - Discrete and Continuous Probabilities, Sum Rule, Product Rule, and Bayes' Theorem. Summary Statistics and Independence – Gaussian Distribution - Conjugacy and the Exponential Family - Change of Variables/Inverse Transform.

Module 5

Optimization : Optimization Using Gradient Descent - Gradient Descent With Momentum, Stochastic Gradient Descent. Constrained Optimization and Lagrange Multipliers - Convex Optimization - Linear Programming - Quadratic Programming.

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 bySpringer
3. Introduction to Applied Linear Algebra by Stephen Boyd and Lieven Vandenberghe, 2018 published by Cambridge UniversityPress
4. Convex Optimization by Stephen Boyd and Lieven Vandenberghe, 2004 published by Cambridge UniversityPress
5. Pattern Recognition and Machine Learning by Christopher M Bishop, 2006, published bySpringer
6. Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond by Bernhard Scholkopf and Smola, Alexander J Smola, 2002, published by MIT Press
7. Information Theory, Inference, and Learning Algorithms by David J. C MacKay, 2003 published by Cambridge UniversityPress
8. Machine Learning: A Probabilistic Perspective by Kevin P Murphy, 2012 published by MITPress.
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. Find the characteristic equation, eigenvalues, and eigenspaces corresponding to each eigenvalue of the following matrix

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

4. Diagonalize the following matrix, if possible

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}$$

5. 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 $P(J \square T)$
   - *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 winsif

   i. A answers the firstquestion,

   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 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 sufficient conditions under which a quadratic optimization problem be written as a linear least squaresproblem.
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 saddlepoints.
5. Consider the update equation for stochastic gradient descent. Write down the update when we use a mini-batch size ofone.
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 youranswer.

   ii. Is (1 , -1) a local/global minimum? Justify youranswer.
7. Is the function $f(x, y) = 2x^2 + y^2 + 6xy - x + 3y - 7$ convex, concave, or neither? Justify youranswer.
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 simplexmethod.

$$max\ 5x_1 + 6x_2 + 9x_3 + 8x_4$$

subject to the constraints

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + x_4 &\leq 5 \\ x_1 + x_2 + 2x_3 + 3x_4 &\leq 3 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

Model Question paper

	QP Code :		**Total Pages : 5**
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

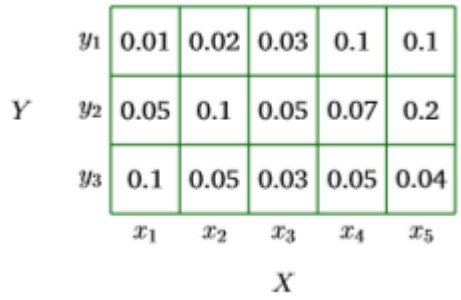
PART A

		Answer all questions, each carries3 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		Are the following sets of vectors linearly independent? Explain your answer. $x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$, $x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}$, $x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$	
3		Find the angle between the vectors $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.	
4		Find the eigen values of the following matrix in terms of k. Can you find an eigen vector corresponding to each of the eigen values? $\begin{bmatrix} 1 & k \\ 2 & 1 \end{bmatrix}$	
5		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).**	
6		Compute the Taylor polynomials T_n, $n = 0, \ldots, 5$ of $f(x) = \sin(x) + \cos(x)$ at $x_0 = 0$.	
7		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 $Y = X^2$.	
8		Show that if two events A and B are independent, then A and B' are independent.	
9		Explain the principle of the gradient descent algorithm.	
10		Briefly explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one	

		over the other.	
PART B			
Answer any one Question from each module. Each question carries 14 Marks			
11	a)	i.Find all solutions to the system of linearequations $-4x + 5z = -2$ $-3x - 3y + 5z = 3$ $-x + 2y + 2z = -1$	(4)
		ii. Prove that all vectors orthogonal to $[2,-3,1]^T$ forms a subspace **W** of R^3. What is ***dim (W)*** and why?	(4)
	b)	A set of ***n*** linearly independent vectors in R^n forms a basis. Does the set of vectors **(2, 4,−3),(0, 1, 1),(0, 1,−1)** form a basis for R^3? Explain yourreasons.	(6)
		OR	
12	a)	Find all solutions in $x = \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} \in R^3$ of the equation system $Ax = 12x$, where $A = \begin{bmatrix} 6 & 4 & 3 \\ 6 & 0 & 9 \\ 0 & 8 & 0 \end{bmatrix}$ and $\sum_{i=1}^{3} x_i = 1$.	(7)
	b)	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***.	(7)
13	a)	Use the Gramm-Schmidt process to find an orthogonal basis for the column space of the following matrix. $\begin{bmatrix} 2 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	(7)
	b)	Find the SVD of the matrix. $\begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix}$	(7)

		OR	
14	a)	i. Let ***L*** be the line through the origin in 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 ***L*** which is closest to the point **(7, 1)** and find the point on ***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 R^3 consisting of eigenvectors for the following matrix. $\begin{bmatrix} 1 & 0 & -2 \\ 0 & 5 & 0 \\ -2 & 0 & 4 \end{bmatrix}$	(8)
		ii. Find a 3×3 orthogonal matrix ***S*** and a 3×3 diagonal matrix ***D*** such that $A = SDS^T$	
15	a)	Ask ier is on a mountain with the equation $z = 100 - 0.4x^2 - 0.3y^2$, where z denotes height. 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**.	(8)
	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	
16	a)	Let ***g*** be the function given by $$g(x,y) = \begin{cases} \dfrac{x^2 y}{x^2 + y^2} & \text{if } (x,y) \neq (0,0); \\ 0 & \text{if } (x,y) = (0,0). \end{cases}$$	(8)

		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)
17	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 fruitat 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. What is the probability that the mango was picked from bag 2?	(6)
	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: $p(x \mid \mu) = \mu^x (1-\mu)^{1-x}, \quad x \in \{0,1\}$ Choose a conjugate prior for the Bernoulli likelihood and compute the posterior distribution $p(\mu \mid x_1, \ldots, x_N)$.	(8)
		OR	
18	a)	Two dice are rolled. A = 'sum of two dice equals 3' B = 'sum of two dice equals 7' C = 'at least one of the dice shows a 1' i. What is P(A|C)? ii. What is P(B|C)? iii. Are A and C independent? What about B and C?	(6)
	b)	Consider the following bivariate distribution p(x,y) of two discrete random	(8)

		variables X and Y .  Compute: i. The marginal distributions p(x) and p(y). ii. The conditional distributions $p(x\|Y = y_1)$ and $p(y\|X = x_3)$.	
19	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 $P = \begin{bmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}$, $q = \begin{bmatrix} -22.0 \\ -14.5 \\ 13.0 \end{bmatrix}$, and $r = 1$. Show that $x^* = (1, 1/2, -1)$ is optimal for the optimization problem $\min \quad \frac{1}{2}x^T Px + q^T x + r$ $\text{s.t.} \quad -1 \le x_i \le 1,\ i = 1, 2, 3.$	(6)
		OR	
20	a)	Derive the gradient descent training rule assuming that the target function is represented as $o_d = w_0 + w_1x_1 + \ldots + w_nx_n$. Define explicitly the cost/ error function E, assuming that a set of training examples D is provided, where each training example d 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)
		***	

Teaching Plan		
No	**Topic**	**No. of Lectures (49)**
	Module-I (LINEAR ALGEBRA)	**8**
1.1	Matrices, Solving Systems of Linear Equations	1
1.2	Vector Spaces	1
1.3	Linear Independence	1
1.4	Basis and Rank (Lecture – 1)	1
1.5	Basis and Rank (Lecture – 2)	1
1.6	Linear Mappings	1
1.7	Matrix Representation of Linear Mappings	1
1.8	Images and Kernel	1
	Module-II (ANALYTIC GEOMETRY, MATRIX DECOMPOSITIONS)	**11**
2.1	Norms, Inner Products	1
2.2	Lengths and Distances, Angles and Orthogonality	1
2.3	Orthonormal Basis, Orthogonal Complement	1
2.4	Orthogonal Projections – Projection into One Dimensional Subspaces	1
2.5	Projection onto General Subspaces.	1
2.6	Gram-Schmidt Orthogonalization	1
2.7	Determinant and Trace, Eigen values and Eigenvectors.	1
2.8	Cholesky Decomposition	1
2.9	Eigen decomposition and Diagonalization	1
2.10	Singular Value Decomposition	1
2.11	Matrix Approximation	1

	Module-III (VECTOR CALCULUS)	**9**
3.1	Differentiation of Univariate Functions, Partial Differentiation and Gradients	1
3.2	Gradients of Vector Valued Functions (Lecture 1)	1
3.3	Gradients of Vector Valued Functions (Lecture 2)	1
3.4	Gradients of Matrices	1
3.5	Useful Identities for Computing Gradients	1
3.6	Backpropagation and Automatic Differentiation – Gradients in deep Netwok	1
3.7	Automatic Differentiation	1
3.8	Higher Order Derivatives	1
3.9	Linearization and Multivariate Taylor Series	1
	Module-IV (PROBABILITY AND DISTRIBUTIONS)	**10**
4.1	Construction of a Probability Space	1
4.2	Discrete and Continuous Probabilities (Probability Density Function, Cumulative Distribution Function)	1
4.3	Sum Rule, Product Rule	1
4.4	Bayes' Theorem	1
4.5	Summary Statistics and Independence (Lecture 1)	1
4.6	Summary Statistics and Independence (Lecture 2)	1
4.7	Bernoulli, Binomial, Uniform (Discrete) Distributions	1
4.8	Uniform (Continuous), Poisson Distributions	1
4.9	Gaussian Distribution	1
4.10	Conjugacy and the Exponential Family (Beta – Bernoulli, Beta – Binomial Conjugacies)	1
	Module-V (OPTIMIZATION)	7
5.1	Optimization Using Gradient Descent.	1
5.2	Gradient Descent With Momentum, Stochastic Gradient Descent	1
5.3	Constrained Optimization and Lagrange Multipliers (Lecture 1)	1

5.4	Constrained Optimization and Lagrange Multipliers (Lecture 2)	1
5.5	Convex Optimization	1
5.6	Linear Programming	1
5.7	Quadratic Programming	1

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 $f : \mathbb{N} \to \mathbb{N}$, 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 $\beta_i : V_i \rightarrowtail L_i$ 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