<table>
<tr><td rowspan="2">CST 201</td><td rowspan="2">DATA STRUCTURES</td><td>CATEGORY</td><td>L</td><td>T</td><td>P</td><td>CREDIT</td><td>YEAR OF INTRODUCTION</td></tr>
<tr><td>PCC</td><td>3</td><td>1</td><td>0</td><td>4</td><td>2019</td></tr>
</table>

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

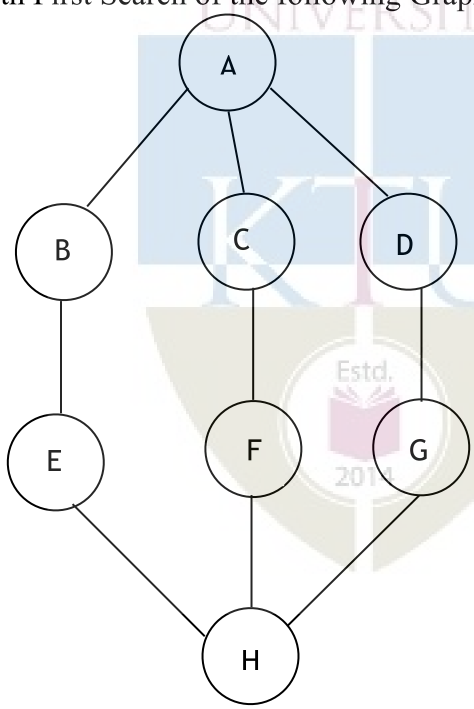
5. Write an algorithm to count the number of occurrences of a character in a linked list (each node contains only one character)

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