CODE	COURSE NAME	CATEGORY	L	T	P	CREDITS
EET443	DATA STRUCTURES	PEC	2	1	0	3

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 problems efficiently. This course introduces abstract concepts for data organization and manipulation using data structures such as stacks, queues, linked lists, binary trees and graphs for designing their own data structures to solve practical application problems.

Prerequisite: EST 102 Programming in C

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

CO 1	Analyze the time and space efficiency of the data structure(K3)
CO 2	Describe how arrays, records, linked structures, stacks and queues are used by algorithms (K1)
CO 3	Compare and contrast the benefits of dynamic and static data structures implementations(K3)
CO 4	Explain different memory management techniques and their significance (K3)
CO 5	Develop algorithms incorporating trees and graphs (K3)

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	3	2										
CO 2	3	2										
CO 3	3	2										
CO 4	3	2										
CO 5	3	2										
CO 6	3	2										

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember (K1)	10	10	20
Understand (K2)	25	25	50
Apply (K3)	15	15	30
Analyse (K4)	-	-	-
Evaluate (K5)	-	-	-
Create (K6)	-	-	-

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

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

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Derive the big O notation for f(n) = n2+2n+5 (K2,PO1)
2. What do you understand by complexity of an algorithm? Write worst case and best case
3. Find complexity of linear search.(K2,PO1)
4. Write an algorithm for matrix multiplication and calculate its time complexity. (K3,PO2)

Course Outcome 2 (CO2)

1. Write an algorithm/pseudo code to add a new element in a particular position of an array(K3,PO2)
2. Explain about the use and representation of header node in linked list (K1,PO1)
3. How a linked list can be used to represent the polynomial 5x4y6+24x3y4-17x2y3+15xy2+45.(K3,P02)
4. What is a circular queue? How it is different from normal queue? (K1,PO1)

Course Outcome 3(CO3):

1. Compare and contrast singly linked list and doubly linked list ((K2,PO1)
2. Write a program that implement stack (its operations) using i) Arrays ii) Linked list(Pointers) and compare performance(K3,PO2)
3. Compare array and linked list implementation of a general list.(K2,P02)
4. What are the disadvantages of representing a linear queue using array? How are they overcome (K1,PO1)

Course Outcome 4 (CO4):

1. Free memory blocks of size 60K, 25K, 12K, 20K, 35K, 45K and 40K are available in this order. Show the memory allocation for a sequence of job requests of size 22K, 10K, 42K, and 31K (in this order) in First Fit, Best Fit and Worst Fit allocation strategies (K3,PO2)
2. Explain how memory de-allocation is done in memory management (K1,PO1)
3. Compare various memory management techniques (K2,PO1)

Course Outcome 5 (CO5):

1. List the properties of a binary search tree. (K1,P01)
2. 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. .(K3,P02)
3. Develop an algorithm to add an element into a binary search tree (K3,P02)
4. Give any two representations of graph. Give algorithm for DFS. Demonstrate DFS using suitable example. (K2,P01)

Model Question Paper

QP CODE:

PAGES : 3

Reg No: _____

Name: _____

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

Course Code: EET443

Course Name: DATA STRUCTURES

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

PART A

Answer all Questions. Each question carries 3 Marks

1. Compare Structured Approach and Object Oriented Approach of Programming.

2. 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;
```

3. Write an Algorithm to reverse a string using Stack.
4. Explain the disadvantages of representing a Linear Queue using Array.
5. Write any three Applications of Linked List.
6. Explain DEQUEUE
7. Write a non recursive algorithm/ Pseudocode for pre-order traversal in a binary tree.

8. What is a binary search tree (BST)? Give an example of a BST with five nodes.
9. Give two different types of representation for graphs.
10. Compare Prim's and Kruskal's Algorithm

PART B

11.a) Explain space complexity and time complexity of an Algorithm. Write an Algorithm/pseudo code for linear search and mention the best case and worst case time complexity of Linear Search algorithm? (10)

b) Explain Modular Programming with Suitable Example (4)

OR

12.a) Explain System Lifecycle in detail. (10)

b) Explain an algorithm? How is its complexity analysed? (4)

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

b) Compare and Contrast a Circular Queue with a normal Queue (4)

OR

14.a) Write an Algorithm to evaluate Postfix operation. (8)

b) Convert the following infix expression into prefix expression (A-B/C) * (D*E-F) .Show the stack contents for each step. (6)

15.a) Write algorithms to perform the following operations on a doubly linked list.

(i) Insert a node with data 'y' after a node whose data is 'x'.

(ii) Delete a node whose data is 's' .

(iii) Insert a node with data 'a' as the 1st node of the list. (10)

b) Write an algorithm to count the number of occurrences of a character in a linked list (each node contains only one character). (4)

OR

16.a) Assume that a Stack is represented using Linked List. Write Algorithms for the following operations.

a) PUSH

b) POP (10)

b) Compare a Circular Linked List and a Doubly Linked List. (4)

17. a) Explain how memory de-allocation is done in memory management. (8)

b) Discuss the advantages and Disadvantages of First-fit,Best-fit and Worst-fit Allocation schemes. (6)

OR

18.a) Write an algorithm/Psudocode to perform the following operations on Binary . Search tree.

a) Insert an element k

b) Search for an element k (10)

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

19. a) Explain the various ways in which a graph can be represented bringing out the advantages and disadvantages of each representation (10)

b) Write an algorithm to perform bubble sort on a collection of 'n' numbers. (4)

OR

20.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)

Syllabus

Module 1

Basic Concepts of Data Structures

Introduction to programming methodologies – structured approach, object oriented approach, stepwise refinement techniques, Algorithms, Performance Analysis, Space Complexity, Time Complexity, Asymptotic Notation, Complexity Calculation of Simple Algorithms

Module 2

Arrays

Introduction to data structures: Stacks, Queues-Circular Queues, Priority Queues, Double Ended Queues, Evaluation of Expressions, Applications of stacks and queues

Module 3

Linked List

Singly Linked List-Operations on Linked List. Doubly Linked List, Circular Linked List, Stacks and Queues using Linked List, Polynomial representation using Linked List

Module 4

Memory Management and Trees

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

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

Module 5

Graphs

Graphs : Definitions, Representation of Graphs, Topological Sort, Depth First Search and Breadth First Search on Graphs, Shortest-path algorithms, Minimum spanning tree, Prim's and Kruskal's algorithms, Applications of graphs

Text Book

1. Fundamentals of Data structures in C, 2nd Edition, E.Horowitz, S.Sahni and Susan Anderson-Freed, University Press (India),2008.

Reference Books

1. Classic Data Structures, Samanta D., Prentice Hall India, 2/e,,2009.
2. Data Structures: A Pseudocode Approach with C, 2/e, Richard F. Gilberg, Behrouz A. Forouzan, Cengage Learning 2005.
3. Data Structures and Algorithms, Aho A. V., J. E. Hopcroft and J. D. Ullman Pearson Publication. 2nd Edition
5. Introduction to Data Structures with Applications, Tremblay J. P. and P. G. Sorenson,Tata McGraw Hill 1995
4. Advanced Data Structures, Peter Brass ,Cambridge University Press,2008
5. Theory and Problems of Data Structures, Lipschuts S., Schaum's Series 1996
6. 8A Structured Approach to Programming, . Hugges J. K. and J. I. Michtm, PHI. 1987

Course Contents and Lecture Schedule:

No	Topic	No. of Lectures
1	**Introduction**	**5**
1.1	Introduction to programming methodologies – structured approach, object oriented approach, stepwise refinement techniques\	**1**
1.2	Algorithms , Performance Analysis	**1**
1.3	Space Complexity, Time Complexity	**1**
1.4	Asymptotic Notation (Big O Notation)	**1**

1.5	Complexity Calculation of Simple Algorithms	**1**
2	**Arrays**	**7**
2.1	Stacks	**1**
2.2	Queues, Circular Queues	**1**
2.3	Priority Queues	**1**
2.4	Double Ended Queues	**1**
2.5	Conversion and Evaluation of Expressions	**1**
2.6	Applications of stacks and queues	**2**
3	**Linked List**	**8**
3.1	Singly Linked List	**1**
3.2	Doubly Linked List	**1**
3.3	Circular Linked List	**1**
3.4	Stacks using Linked List	**1**
3.5	Queues using Linked List	**1**
3.6	Polynomial representation using Linked List	**2**
4	**Memory Management and Trees**	**8**
4.1	Memory allocation and de-allocation	**1**
4.2	First-fit, Best-fit and Worst-fit allocation schemes	**2**
4.3	Binary Trees- Tree Operations	**1**
4.4	Binary Tree Representation, Tree Traversals	**2**
4.5	Binary Search Trees- Binary Search Tree Operations	**2**
5	**Graphs**	**7**
5.1	Graphs Definitions, Representation of Graphs	**1**
5.2	Topological sort, Depth First Search and Breadth First Search on Graphs,	**2**

5.3	Shortest-path algorithms,	**1**
5.4	Minimum spanning tree	**1**
5.5	Prim's and Kruskal's algorithms, Applications of graphs	**2**