U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

June 2025 Semester End Main Examinations

Programme: B.E. **Semester: III**

Branch: Artificial Intelligence and Machine Learning **Duration: 3 hrs.**

Course Code: 22AM3PCDST **Max Marks: 100**

Course: Data Structures

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT - I	*CO*	*PO*	Marks
1	a)	Illustrate the classification of data structures with examples.	CO1	*PO1*	**05**
	b)	Write a program and explain various dynamic memory allocation functions.	CO1	*PO1*	**08**
	c)	Write a C functions to demonstrate the following operations on a Singly Linked List (SLL) with header node: i. Insert an item at the rear end ii. Display the items in the SLL.	CO1	*PO1*	**07**
		OR			
2	a)	Differentiate static memory allocation and dynamic memory allocation.	CO1	*PO1*	**05**
	b)	Write a C program to implement a Singly Linked List (SLL) with a header node to perform the following basic operations: i. Insert at the front end ii. Search for a key in the list iii. Display the contents of the list	CO1	*PO1*	**10**
	c)	Explain the applications of Singly Linked List (SLL)	CO1	*PO1*	**05**
		UNIT - II			
3	a)	Explain the different operations that can be performed on stack using C functions.	CO1	*PO1*	**06**
	b)	Convert the following infix expressions to postfix expressions using stack: i. A $ B * C – D + E|F|(G+H) ii. (A+B) +C/D*E	CO2	*PO2*	**06**
	c)	Write C functions to demonstrate the following operations on a Circular Doubly Linked List (CDLL). i. Insert an item at the rear end ii. Delete an item at the front end	CO1	*PO1*	**08**
		OR			

	4	a)	Explain the advantages of linked list over arrays.	CO1	*PO1*	**04**
		b)	Illustrate the process of evaluation of the given postfix expression: 4 5 3 * + 7 - using a stack trace.	CO1	*PO2*	**06**
		c)	Write C function to demonstrate the following operations on a Doubly Linked List (DLL). i. Insert an item at the front end ii. Delete an item at rear end. iii. Display.	CO1	*PO1*	**10**
			UNIT - III			
	5	a)	Explain any two types of queues with suitable application.	CO2	*PO1*	**04**
		b)	Define recursion and write recursive function to generate Fibonacci series up to 'n'.	CO2	*PO1*	**06**
		c)	Demonstrate the working of circular queue of integers using array to perform the following operations: i. Insert ii. Delete iii. Display	CO2	*PO1*	**10**
			OR			
	6	a)	Write a program to implement a priority queue where elements with lower values have higher priority.	CO2	*PO1*	**10**
		b)	Describe how a deque works, using a music player as an example.	CO2	*PO1*	**05**
		c)	Write a recursive function to solve Tower of an Hanoi problem.	CO2	*PO1*	**05**
			UNIT - IV			
	7	a)	With a neat diagram explain Threaded binary tree.	CO3	*PO1*	**05**
		b)	Construct a Binary Search Tree (BST) for the sequence: 24, 60, 37, 8, 3, 91, 58, 20, 5, 62, 15, 50 and perform inorder and preorder traversals on it.	CO3	*PO3*	**10**
		c)	Write a C function to find the height of BST.	CO3	*PO1*	**05**
			OR			
	8	a)	Differentiate the following types of Binary Tree: i. Complete Binary Tree ii. Perfect Binary Tree iii. Balanced Binary Tree	CO3	*PO1*	**06**
		b)	Given the postorder and inorder traversals of a binary tree, construct and return the binary tree. Post-order = [4, 7, 5, 2, 8, 6, 3, 1] In-order = [4, 2, 7, 5, 1, 8, 6, 3]	CO3	*PO3*	**10**
		c)	Explain the applications of Binary Search Tree (BST)	CO3	*PO1*	**04**
			UNIT - V			
	9	a)	List the properties of Red-Black tree. Explain the process of insertion of items in Red-Black tree.	CO3	*PO1*	**10**
		b)	Construct an AVL tree by inserting the elements: 63, 9, 19, 27, 18, 108, 99, 81 in the given order.	CO3	*PO3*	**10**
			OR			

	10	a)	Illustrate any five splaying in splay tree with an example for each.	CO3	*PO1*	**10**
		c)	Construct a Red-Black tree by inserting the elements: 10, 20, 7, 17, 19, 27, 42, 82 in the same order by representing the process pictorially at every step.	CO3	*PO3*	**10**

****