

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING Industry Certified Value Added Course

on

Data Structures and Algorithm using JAVA 14.02.2025, 15.02.2025 & 19.02.2025-22.02.2025

Total hours: 45

Course Objectives:

To enable the students to

- Understand the basic concepts of recursion.
- o Understand the basic concepts of linear data structures
- Gain knowledge about linear data structures like stack, queue and its applications.
- o Be familiar with binary tree concepts and its applications.
- Gain knowledge about graph traversal methods and application of graphs.

UNIT-1- RECURSION

Introduction-Application of recursion-Recursion output practice-Print n to 1 using recursion-Print 1 to n using recursion-Tail recursion-Writing base cases in recursion-Natural numbers sum using recursion-Palindrome check using recursion-Sum of digits using recursion-Rope cutting problem-Generate subsets-Tower of Hanoi-Josephus problem-Printing all permutations.

UNIT-2- LINKED LIST

Introduction-Simple linked list implementation in java-Application of linked list-Traversing a linked list-Recursive traversal of singly linked list-Insert at begin of SLL-Insert at end of SLL-Insert at given position in SLL-Delete first node of SLL-Delete last node of SLL-Search in a SLL(Iterative & Recursive)-DLL-insert at beginning of DLL-insert at the end of DLL-insert at the end of DLL-Reverse a DLL-Delete head of DLL-Delete last node of DLL-Sorted insert in a SLL-Middle of a linked list-Nth node from end of linked list-Reverse a linked list iterative-Recursive reverse a linked list-Remove duplicate from a sorted SLL-Reverse a linked list in groups of size k-Detect loop-Detect loop using Floyd cycle detection-detect and remove loop in linked list-delete node with only pointer given to iterative-Segregate even and odd nodes-Intersection point of two linked list-Pairwise swap nodes of linked list-in java-clone a linked list with a random pointer in java-LRU cache design-merge two sorted linked list-palindrome linked list

UNIT-3- STACK, QUEUE

Stack data structure-Array implementation using stack-Linked list implementation using stack-Application of stack-Stack in java collection-Balanced paranthesis-Implement two stack in array-stock span problem-Previous greater problem-Next greater element-Largest rectangular area in histogram-largest rectangle with all 1's-Stack with get min() in O(1)-Infix prefix and postfix introduction-Infix to postfix(simple solution)-Infix to postfix(Efficient solution)-Evaluation of postfix-Infix to prefix(simple solution)-Infix to prefix(Efficient solution)- Introduction-Application of queue-Implementation of queue using array-Implementation of queue using linked list-Queue in java-Implementation of stack using queue-Reversing a queue-generate numbers with given digits

UNIT-4- TREE

Introduction to tree-Application of tree-Binary tree-Tree traversal-Implementation of inorder traversal-Implementation of preorder traversal-Implementation of postorder traversal-Height of binary tree-print

9

9

9

9

nodes at k distance-Level order traversal-Level order traversal line by line-Size of binary tree-Maximum in binary tree-Print left view of binary tree-Children some property-Check for balanced binary tree-Maximum with of binary tree-Convert binary tree to doubly linked list-Construct binary tree from inorder and preorder-Tree traversal in spiral form-Diameter of binary tree-LCA of binary tree -Burn a binary tree from a leaf-Count nodes in a complete binary tree-Iterative inorder traversal-Preorder traversal- Introduction-search in BST-insert in BST-deletion in BST-floor in BST-ceil in BST-find kth smallest in BST-check for BST-pair sum with given BST-vertical sum in BST-vertical traversal of a BST-top view of BST-Bottom view of BST

UNIT-5- GRAPH

9

Introduction-graph representation(adjacency matrix)- graph representation(adjacency list)-Adjacency list implementation-BFS-Application of BFS-DFS-Application of DFS-Shortest path in an unweighted graph-Detected cycle in undirected graph- Detected cycle in directed graph(part 1)-topological sorting- Detected cycle in directed graph(part 2)-Topological sorting(DFS based algorithm)-prim's algorithm/Minimum spanning tree-implementation of prim's algorithm-dijikstra's shortest path algorithm-implementation of dijikstra's algorithm

CO. No.	CO Statements	Knowledge Level		
CO1	Implement recursion in solving computational problems.	K3 - Apply		
CO2	Use the basic concepts of linear data structures to solve real-world problems.	K3 - Apply		
CO3	Apply stacks and queues in program design to handle data efficiently.	K3 - Apply		
CO4	Demonstrate the applications of tree data structures in various scenarios.	K3 - Apply		
CO5	Utilize different graph traversal methods to solve problems involving network structures.	K3 - Apply		

COURSE OUTCOMES (COs):

On successful completion of this course, the students will be able to:

CO.	POs											PSOs		
No.	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	Н	Μ	L	-	-	-	-	-	-	-	-	L	M	L
CO2	Н	М	L	-	-	-	-	-	-	-	-	L	Μ	L
CO3	Н	М	L	-	-	-	-	-	-	-	-	L	Μ	L
CO4	Н	М	L	-	-	-	-	-	-	-	-	L	Μ	L
CO5	Н	М	L	-	-	-	-	-	-	-	-	L	M	L