



Call Us : 0612 - 2321928, 9308654075

**RAYS Computer Education**

[A Unit of Rays Edutech Pvt. Ltd.]

Feel The Sensation of Your Destination

**COURSE TITLE**

**DATA STRUCTURE**

**DETAILED SYLLABUS**

SR.NO	NAME OF CHAPTERS & DETAILS	HOURS ALLOTTED
1	<b>USER DEFINED DATATYPE /STRUCTURE</b> <ul style="list-style-type: none"><li>➤ About structure</li><li>➤ Defining structure</li><li>➤ Accessing structure element</li><li>➤ Array of structure</li><li>➤ Use of pointer with structure</li></ul>	4 Hour
2	<b>DYNAMIC MEMORY ALLOCATION AND DEALLOCATION</b> <ul style="list-style-type: none"><li>➤ Use of sizeof operator</li><li>➤ Type casting</li><li>➤ Dynamic memory allocation by using inbuilt function malloc() and calloc()</li><li>➤ Differences between malloc() and calloc()</li><li>➤ Use of realloc()</li><li>➤ Deallocation by using inbuilt function free()</li></ul>	4 Hour

3

## LINKED LIST

- About Linked list
- Types of linked list
  - (i) Linear or single or one-way linked list
  - (ii) Double or two way linked list
- Linear or single or one way linked list
  - ❖ Implementation / Representation of single linked list
  - ❖ Adding new node to linked list
    - (a) Adding at beginning
    - (b) Adding at ending
    - (c) Adding before specified position
    - (d) Adding after specified position
  - ❖ Deleting a node from linked list
    - (a) Deleting beginning node
    - (b) Deleting end node
    - (c) Deleting specified position node
  - ❖ Searching a node in linked list
- Double or two-way linked list
  - ❖ Implementation / Representation of double linked list
  - ❖ Adding new node to linked list
    - (a) Adding at beginning
    - (b) Adding at ending
    - (c) Adding before specified position
    - (d) Adding after specified position
  - ❖ Deleting a node from linked list
    - (a) Deleting beginning node
    - (b) Deleting end node
    - (c) Deleting specified position node
  - ❖ Searching a node in linked list
- Some more linked list
  - ❖ Circular linked list
    - (a) Singly circular linked list
    - (b) Doubly circular linked list
  - ❖ Header linked list
    - (a) Singly header linked list
    - (b) Doubly header linked list

16 Hour

	<ul style="list-style-type: none"> <li>➤ Application of linked list <ul style="list-style-type: none"> <li>❖ Concatenation of two linked list</li> <li>❖ Polynomial <ul style="list-style-type: none"> <li>(a) Representation of polynomial</li> <li>(b) Addition of polynomial</li> <li>(c) Multiplication of two polynomial</li> </ul> </li> <li>❖ Sparse matrices <ul style="list-style-type: none"> <li>(a) Representation of sparse matrices <ul style="list-style-type: none"> <li>• By using array</li> <li>• By using linked list</li> </ul> </li> <li>(b) Addition of two sparse matrices</li> </ul> </li> </ul> </li> </ul>	
4	<p><b>STACK</b></p> <ul style="list-style-type: none"> <li>➤ Introduction to stack</li> <li>➤ Operation on stack</li> <li>➤ Stack implementation <ul style="list-style-type: none"> <li>❖ By using array</li> <li>❖ By using linked list</li> </ul> </li> <li>➤ Stack Application in computer field <ul style="list-style-type: none"> <li>❖ Representation two stacks</li> <li>❖ Arithmetical Expression</li> <li>❖ Different form of an expression <ul style="list-style-type: none"> <li>(a) Infix form</li> <li>(b) Prefix form or Polish form</li> <li>(c) Postfix form or Reverse Polish form</li> </ul> </li> <li>❖ Conversion one form to another form <ul style="list-style-type: none"> <li>(a) Conversion from infix to prefix and postfix form</li> <li>(b) Conversion from postfix to infix and prefix form</li> <li>(c) Conversion from prefix to infix and postfix form</li> </ul> </li> <li>❖ Algorithm to evaluate a postfix expression</li> </ul> </li> </ul>	6 Hour
5	<p><b>QUEUE</b></p> <ul style="list-style-type: none"> <li>➤ Introduction to Queue</li> <li>➤ Operation on Queue</li> <li>➤ Queue implementation</li> </ul>	6 Hour

	<ul style="list-style-type: none"> <li>❖ By using array</li> <li>❖ By using linked list</li> <li>➤ Types of Queue <ul style="list-style-type: none"> <li>(i) Circular Queue</li> <li>(ii) Dequeue</li> <li>(iii) Priority Queue</li> </ul> </li> </ul>	
6	<p><b>TREE</b></p> <ul style="list-style-type: none"> <li>➤ Introduction to tree</li> <li>➤ Important properties of tree</li> <li>➤ Some tree terminology</li> <li>➤ Types of tree <ul style="list-style-type: none"> <li>❖ Binary tree <ul style="list-style-type: none"> <li>(a) Strictly binary tree or two tree</li> <li>(b) Complete binary tree or full binary tree</li> </ul> </li> <li>❖ Binary search tree (BST)</li> <li>❖ Expression tree</li> <li>❖ AVL tree or Height Balance tree</li> <li>❖ Threaded binary tree</li> <li>❖ B – tree</li> </ul> </li> <li>➤ Binary Tree <ul style="list-style-type: none"> <li>❖ Representation of binary tree <ul style="list-style-type: none"> <li>(a) Array or Sequential representation</li> <li>(b) Linked list representation</li> </ul> </li> <li>❖ Traversal of binary tree <ul style="list-style-type: none"> <li>(a) Inorder traversal</li> <li>(b) Preorder traversal</li> <li>(c) Postorder traversal</li> <li>(d) Level-by-Level traversal</li> </ul> </li> <li>❖ Non-recursive traversal of binary tree <ul style="list-style-type: none"> <li>(a) Inorder traversal</li> <li>(b) Preorder traversal</li> <li>(c) Postorder traversal</li> </ul> </li> </ul> </li> <li>➤ Binary search tree (BST) <ul style="list-style-type: none"> <li>❖ Representation of BST <ul style="list-style-type: none"> <li>(a) Array or sequential representation</li> <li>(b) Linked list representation</li> </ul> </li> </ul> </li> </ul>	16 Hour

	<ul style="list-style-type: none"> <li>❖ Search in BST</li> <li>❖ Insertion into BST</li> <li>❖ Deletion from BST</li> <li>➤ Expression tree <ul style="list-style-type: none"> <li>❖ Representation of expression tree</li> <li>❖ Traversal of expression tree <ul style="list-style-type: none"> <li>(a) Inorder traversal</li> <li>(b) Preorder traversal</li> <li>(c) postorder traversal</li> </ul> </li> </ul> </li> <li>➤ AVL tree or Height balance tree <ul style="list-style-type: none"> <li>❖ Insertion of a node into an AVL tree</li> <li>❖ Deletion from an AVL tree</li> </ul> </li> <li>➤ B-tree <ul style="list-style-type: none"> <li>❖ Drawing of a B-tree</li> <li>❖ Deletion in B-tree</li> </ul> </li> </ul>	
7	<p><b>GRAPH</b></p> <ul style="list-style-type: none"> <li>➤ Introduction to Graph</li> <li>➤ Graph terminology</li> <li>➤ Types of Graph</li> <li>➤ Representation of Graph <ul style="list-style-type: none"> <li>(a) Adjacency matrix</li> <li>(b) Adjacency list</li> </ul> </li> <li>➤ Graph traversals <ul style="list-style-type: none"> <li>(a) Breadth first search (BFS)</li> <li>(b) Depth first search (DFS)</li> </ul> </li> <li>➤ Transitive closure <ul style="list-style-type: none"> <li>❖ Program to compute transitive closure</li> </ul> </li> <li>➤ Shortest path problem</li> <li>➤ About Minimal spanning tree (MST) <ul style="list-style-type: none"> <li>❖ Introduction to MST</li> <li>❖ Construction of MST <ul style="list-style-type: none"> <li>(a) Krushkal's method/algorithm</li> <li>(b) Prim's method/algorithm</li> </ul> </li> </ul> </li> </ul>	14 Hour

8	<p><b>SEARCHING AND SORTING</b></p> <ul style="list-style-type: none"> <li>➤ Searching <ul style="list-style-type: none"> <li>(a) Sequential searching</li> <li>(b) Binary searching</li> </ul> </li> <li>➤ Sorting <ul style="list-style-type: none"> <li>(a) Bubble sort      (b) Insertion sort      (c) Quick sort</li> <li>(d) Merge sort      (e) Heap sort      (f) selection sort</li> <li>(g) shell sort      (h) Radix sor</li> </ul> </li> </ul>	12 Hour
9	<b>TIME COMPLEXITY</b>	3 Hour

