

**Module 1****Chapter 1 : Introduction to Data Structure 1-1 to 1-27**

Syllabus : Introduction to Data Structures, Concept of ADT, Types of Data Structures-Linear and Nonlinear, Operations on Data Structures.

1.1	Data.....	1-1
1.1.1	Data Types.....	1-1
1.1.2	Abstract Data Types (ADT)	1-1
1.1.3	Data Object	1-3
1.2	Data Structures.....	1-3
1.2.1	Types of Data Structures.....	1-3
1.2.1(A)	Primitive and Non-Primitive	1-3
1.2.1(B)	Linear and Non-Linear.....	1-4
1.2.1(C)	Static and Dynamic.....	1-5
1.3	Relationship among Data Object, Data Type, Data Structure and Data Representation.....	1-5
1.3.1	Operations on Data Structure.....	1-5
1.4	Algorithm Analysis.....	1-6
1.4.1	Measuring the Running Time of a Program (Time Complexity).....	1-7
1.4.2	Measurement of Growth Rate (Asymptotic Growth Rate)	1-7
1.4.2(A)	Asymptotic Consideration.....	1-7
1.4.2(B)	Constant Factor in Complexity Measure.....	1-7
1.4.3	Notation O : (Pronounced as Big-Oh), ($O(n^2)$ is Pronounced as Big-Oh of n^2)	1-8
1.4.4	Best Case, Worst Case and the Average Case Behaviour.....	1-10
1.5	Introduction to Arrays	1-11
1.6	Representation and Analysis.....	1-11
1.7	One-Dimensional Arrays	1-12
1.8	Operations with Arrays	1-13
1.8.1	Deletion	1-14
1.8.2	Insertion.....	1-14
1.8.3	Search.....	1-16
1.8.4	Merging of Sorted Arrays.....	1-16
1.9	Two-Dimensional Arrays	1-18
1.9.1	Initializing Two-Dimensional Arrays.....	1-19
1.9.2	Address Calculation.....	1-19
1.10	Multi-Dimensional Arrays.....	1-21
1.11	Application of Arrays.....	1-21
1.11.1	Addition of Two 2-D Matrices	1-21
1.11.2	Transpose of Square Matrix	1-24

1.11.3	Finding whether a given Square Matrix is Symmetrical	1-25
1.11.4	Multiplication of Two Matrices $A_m \times n$ and $B_n \times p$	1-26

Module 2**Chapter 2 : Stack and Queues 2-1 to 2-63**

Syllabus : Introduction, ADT of Stack, Operations on Stack, Array Implementation of Stack, Applications of Stack – Well-formness of Parenthesis, Infix to Postfix Conversion and Postfix Evaluation, Recursion. Introduction, ADT of Queue, Operations on Queue, Array Implementation of Queue, Types of Queue-Circular Queue, Priority Queue, Introduction of Double Ended Queue, Applications of Queue.

2.1	Introduction	2-1
2.2	Operations on Stacks.....	2-1
2.3	Array Representation	2-1
2.3.1	'C' Functions for Primitive Operations on a Stack	2-2
2.3.2	Program Showing Stack Operations	2-2
2.3.3	Well-Formedness of Parenthesis	2-5
2.3.4	Operations on Stack Considering Overflow and Underflow.....	2-6
2.3.5	Stack as an ADT	2-6
2.4	Applications of Stack.....	2-7
2.4.1	Expression Representation	2-7
2.4.2	Evaluation of a Postfix Expression using a Stack.....	2-8
2.4.3	Conversion of an Expression from Infix to Postfix	2-12
2.5	Expression Conversion (A Fast Method).....	2-22
2.5.1	Infix to Postfix.....	2-22
2.5.2	Algorithm to Check Well-Formedness of Parenthesis	2-23
2.6	Introduction to Recursion	2-27
2.7	Converting a Recursive Function to an Equivalent C-Function.....	2-27
2.7.1	Finding Factorial of an Integer Number	2-27
2.7.2	Finding n^{th} Term of Fibonacci Sequence Recursive Definition	2-28
2.7.3	Finding GCD of given Numbers	2-28
2.7.4	Calculation of x^n using Recursion	2-29
2.7.5	Calculation of Sum of Digits	2-29
2.8	Examples of Recursion	2-29



3.4	Doubly Linked List	3-28	4.7	Binary Tree Traversal	4-7
3.4.1	Creation of a Doubly Linked List.....	3-29	4.7.1	Preorder Traversal (Recursive)	4-8
3.4.2	Deletion of a Node.....	3-31	4.7.1(A)	'C' Function for Preorder Traversal	4-8
3.5	Doubly Linked Circular List.....	3-36	4.7.2	Inorder Traversal (Recursive).....	4-9
3.6	Applications of Linked Lists	3-37	4.7.2(A)	'C' Function for Inorder Traversal.....	4-9
3.6.1	Polynomials as Linked Lists	3-38	4.7.3	Postorder Traversal (Recursive)	4-10
3.6.2	Addition of Two Polynomials	3-39	4.7.3(A)	'C' Function for Postorder Traversal.....	4-10
3.7	Linked Representation of a Stack.....	3-46	4.7.4	Non-Recursive Preorder Traversal.....	4-10
3.7.1	Functions for Stack Operations	3-47	4.7.4(A)	'C' Function for Non-Recursive Preorder of Tree Along with the ADT Stack	4-11
3.8	Linked Representation of a Queue	3-50	4.7.5	Non-Recursive Inorder Traversal	4-12
3.8.1	Comparison between Array Representation and the Linked Representation of a Queue	3-50	4.7.5(A)	'C' Function for Non-Recursive Inorder Traversal of a Binary Tree.....	4-12
3.8.2	Operations on Queue Implemented using Linked Structure.....	3-51	4.7.6	Non-Recursive Postorder Traversal	4-12
3.9	Queue using a Circular Linked List.....	3-55	4.7.6(A)	'C' Function for Non-Recursive Postorder Traversal.....	4-14
3.9.1	Implementation of a Priority Queue using a Linked List.....	3-58	4.7.7	Tree Traversal Examples	4-15

Module 4

Chapter 4 : Trees **4-1 to 4-102**

<p>Syllabus : Introduction, Tree Terminologies, Binary Tree, Binary Tree Representation, Types of Binary Tree, Binary Tree Traversals, Binary Search Tree, Operations on Binary Search Tree, Applications of Binary Tree-Expression Tree, Huffman Encoding, Search Trees-AVL, rotations in AVL Tree, operations on AVL Tree, Introduction of B Tree, B+ Tree.</p>
--

4.1	Basic Terminology	4-1	4.8.7	'C' Function for Finding Height of a Tree (Recursive).....	4-20
4.1.1	Introduction.....	4-1	4.8.8	'C' Function for Swapping of Left and Right Children of Every Node (Mirror)	4-20
4.1.2	Basic Terms	4-1	4.8.9	Finding Width of a Tree	4-20
4.2	Binary Tree.....	4-1	4.8.10	Function to List the DATA Fields of the Node of a Binary Tree T by Level. Within Levels Nodes are Listed Left to Right.....	4-21
4.3	Representation of a Binary Tree using an Array..	4-2	4.8.11	Non-Recursive Algorithm for Height of a Binary Tree.....	4-22
4.4	Linked Representation of a Binary Tree	4-3	4.8.11(A)	'C' Function for Height of a Tree (Non-Recursive).....	4-22
4.4.1	Program for Creation of a Sample Binary Tree	4-4	4.9	Creation of a Binary Tree from Traversal Sequence	4-23
4.4.2	'C' Function for Creation of a Binary Tree	4-4	4.9.1	Creation of Binary Tree from Preorder and Inorder Traversals	4-23
4.5	A General Tree.....	4-5	4.9.2	Creation of Tree from Postorder and Inorder Traversal	4-23
4.5.1	Node Declaration for a Tree	4-5			
4.6	Types of Binary Tree	4-6			
4.6.1	Full Binary Tree	4-6			
4.6.2	Complete Binary Tree.....	4-7			
4.6.3	Skewed Binary Tree	4-7			
4.6.4	Strictly Binary Tree	4-7			
4.6.5	Extended Binary Tree (2-Tree).....	4-7			



<p>4.9.3 Examples on Tree Creation from Traversal Sequence4-23</p> <p>4.10 Binary Search Tree (BST)4-27</p> <p>4.10.1 Definition4-27</p> <p>4.10.2 Operations on a Binary Search Tree4-27</p> <p>4.10.2(A) Initialize Operation.....4-27</p> <p>4.10.2(B) Find Operation.....4-27</p> <p>4.10.2(C) Make Empty Operation.....4-28</p> <p>4.10.2(D) Insert Operation.....4-28</p> <p>4.10.2(E) Example on Creation of a BST4-29</p> <p>4.10.2(F) Delete Operation4-30</p> <p>4.10.2(G) Create.....4-32</p> <p>4.10.2(H) Find Min.....4-32</p> <p>4.10.2(I) Find Max.....4-32</p> <p>4.10.3 Program for Various Operations on BST4-32</p> <p>4.11 AVL Trees4-36</p> <p>4.11.1 Height Balanced Tree.....4-37</p> <p>4.11.2 Balance Factor4-37</p> <p>4.11.3 Structure of a Node in AVL Tree.....4-38</p> <p>4.11.4 'C' Function for Finding the Balance Factor of a Node.....4-38</p> <p>4.11.5 Insertion of a Node into an AVL Tree4-38</p> <p>4.11.5(A) Rotate Left.....4-39</p> <p>4.11.5(B) Rotate Right4-40</p> <p>4.11.5(C) Single Rotation and Double Rotation4-41</p> <p>4.11.5(D) 'C' Function for Insertion of an Element into an AVL Tree4-64</p> <p>4.11.5(E) 'C' Function to Find Height of AVL Tree4-64</p> <p>4.11.5(F) 'C' Function to Rotate Right.....4-64</p> <p>4.11.5(G) 'C' Function to Rotate Left.....4-65</p> <p>4.11.5(H) 'C' Function for RR4-65</p> <p>4.11.5(I) 'C' Function for LL4-65</p> <p>4.11.5(J) 'C' Function for LR.....4-65</p> <p>4.11.5(K) 'C' Function for RL.....4-65</p> <p>4.12 Application of Trees.....4-65</p> <p>4.12.1 Expression Trees.....4-65</p> <p>4.12.2 Program on Expression Tree from Postfix Expression.....4-67</p> <p>4.12.3 Conversion of an Expression into Binary Tree4-68</p> <p>4.12.4 Construction of an Expression Tree from Infix Expression.....4-69</p> <p>4.13 Huffman Algorithm.....4-71</p> <p>4.13.1 Huffman Codes.....4-71</p>	<p>4.13.2 Representation of Binary Codes as a Binary Tree.....4-72</p> <p>4.13.3 Huffman's Algorithm4-73</p> <p>4.13.4 Program for Huffman Tree4-77</p> <p>4.14 B-Trees4-78</p> <p>4.14.1 Insertion of a Key into a B-tree4-79</p> <p>4.14.2 Deleting a Value from a B-tree4-85</p> <p>4.14.3 B-tree as an ADT.....4-89</p> <p>4.15 B+ Trees.....4-95</p> <p>4.16 Splay Tree.....4-97</p> <p>4.16.1 Bottom up Splaying4-97</p> <p>4.16.2 Top Down Splaying4-99</p> <p>4.17 Trie Indexing4-101</p> <p>4.17.1 Compact Trie.....4-102</p>
---	--

Module 5
Chapter 5 : Graphs
5-1 to 5-26

Syllabus : Introduction, Graph Terminologies, Representation of graph, Graph Traversals – Depth First Search (DFS) and Breadth First Search (BFS), Graph Application – Topological Sorting.

<p>5.1 Terminology and Representation5-1</p> <p>5.1.1 Definition5-1</p> <p>5.1.2 Undirected Graph5-1</p> <p>5.1.3 Directed Graph.....5-1</p> <p>5.1.4 A Complete Graph.....5-2</p> <p>5.1.5 Weighted Graph5-2</p> <p>5.1.6 Adjacent Nodes.....5-2</p> <p>5.1.7 Path.....5-2</p> <p>5.1.8 Cycle5-2</p> <p>5.1.9 Connected Graph5-2</p> <p>5.1.10 Subgraph.....5-3</p> <p>5.1.11 Component.....5-3</p> <p>5.1.12 Degree of a Vertex5-3</p> <p>5.1.13 Self Edges or Self Loops.....5-3</p> <p>5.1.14 Multigraph5-3</p> <p>5.1.15 Tree.....5-3</p> <p>5.1.16 Spanning Trees5-4</p> <p>5.1.17 Minimal Spanning Tree5-4</p> <p>5.2 Representation of Graphs5-4</p> <p>5.2.1 Adjacency Matrix.....5-4</p> <p>5.2.2 Adjacency List5-5</p> <p>5.2.3 Path Matrix5-11</p>
--



5.3	Traversal of Graphs.....	5-12
5.3.1	Depth First Search (DFS)	5-12
5.3.1(A)	Algorithm for Depth First Search (Recursive)	5-13
5.3.1(B)	Non-Recursive DFS Traversal.....	5-15
5.3.2	Breadth First Search(BFS)	5-17
5.3.2(A)	Algorithm for BFS	5-17
5.4	Topological Sorting.....	5-23
5.4.1	Program for Topological Sorting.....	5-23

Module 6

Chapter 6 : Sorting and Searching **6-1 to 6-49**

Syllabus : Linear Search, Binary Search, Hashing-Concept, Hash Functions, Collision resolution Techniques.

6.1	Searching	6-1
6.2	Sequential Search	6-1
6.2.1	Sequential Search on a Sorted Array	6-2
6.3	Binary Search.....	6-3
6.4	Sorting.....	6-8
6.4.1	Sort Stability	6-8
6.4.2	Sort Efficiency.....	6-8
6.4.3	Passes.....	6-9
6.5	Insertion Sort	6-9
6.5.1	Sorting an Array of Strings using Insertion Sort	6-11
6.5.2	Sorting an Array of Records on the given key using Insertion Sort	6-12
6.6	Bubble Sort.....	6-13
6.7	Selection Sort.....	6-15
6.8	Quick Sort.....	6-16
6.8.1	Picking a Pivot.....	6-16
6.8.2	Partitioning	6-16
6.8.3	Running Time of Quick Sort	6-25
6.8.3(A)	Worst-Case Analysis	6-25
6.8.3(B)	Best-Case Analysis	6-25

6.8.3(C)	Average-Case Analysis	6-25
6.8.4	Role of Pivot in Efficiency of Quick Sort	6-28
6.9	Two-Way Merge Sort	6-28
6.9.1	Merging	6-29
6.9.2	Analysis of Merge Sort	6-32
6.9.3	Non-Recursive Merge Sort.....	6-32
6.10	Comparison of Sorting Algorithms.....	6-33
6.11	Best-Case, Worst-Case and Average-Case Analysis of Sorting Algorithm... ..	6-34
6.12	External Vs Internal Sorting.....	6-34
6.13	Hash Tables	6-36
6.13.1	What is Hashing ?	6-36
6.13.2	Hash Table Data Structure.....	6-36
6.13.2(A)	Open Hashing Data Structure	6-37
6.13.2(B)	Closed Hashing Data Structure.....	6-37
6.13.3	Hashing Functions	6-37
6.13.3(A)	Characteristics of a Good Hash Function.....	6-37
6.13.3(B)	Division-Method	6-38
6.13.3(C)	Midsquare Methods.....	6-38
6.13.3(D)	Folding Method	6-38
6.13.3(E)	Digit Analysis.....	6-38
6.13.3(F)	Length Dependent Method.....	6-38
6.13.3(G)	Algebraic Coding.....	6-38
6.13.3(H)	Multiplicative Hashing	6-39
6.13.4	Collision Resolution Strategies (Synonym Resolution).....	6-39
6.13.4(A)	Separate Chaining	6-39
6.13.4(B)	Open Addressing	6-40
6.13.4(C)	Primary Clustering.....	6-49
• Lab Experiments		L-1 to L-33
• Appendix A : Solved University Question Papers of Dec. 2017, May 2018 and Dec. 2018.....		A-1 to A-2
• Appendix B : Solved University Question Papers of May 2019 and Dec. 2019		B-1 to B-2

