Course Title: **Data Structure and Algorithms** Credit: **3**
Course No.: **CSIT.121**
Nature of the Course: **Theory+Lab** Total hours: **48**
Level: **B.Sc.CSIT** Year: **First** Semester: **Second**

---

1. Course Description

Study of basic data structure vocabulary and the concept of an algorithm.

2. Course Objectives

- To introduce, fundamental data structures and problem solving paradigms
- To introduce time complexity analysis of problems.
- To study the representation, implementation & applications of data structures.
- To compare alternative implementations of data structures.
- To choose the appropriate data structure for modeling a given problem.

3. Specific Objectives and Contents

Specific Objectives	Contents
• Outline the classification of data type • Give typical examples of data type • Explain the relevance of data structures in programming.	**Unit 1. Introduction to data structures (3 Hrs.)** 1.1 Preliminary data type and Abstract data type 1.2 Data structure, Need and Types of Data Structure 1.3 Comparison between ADT and Data Structure 1.4 Review of Array, Structure and Pointer
• Define an algorithm • Explain an algorithm's performance • Describe algorithm analysis • Explain the notion used in algorithm analysis	**Unit 2: Algorithm analysis (2 Hrs.)** 2.1 Algorithm – definition, characteristics 2.2 Algorithm vs. Program 2.3 Space complexity, time complexity 2.3 Asymptotic notations (Big O, Omega Ω, Big Θ)
• Describe an array, its dimensionality and declaration • Explain the aim of sorting algorithm • Describe the types of sorting • Explain the classes of sorting algorithm • Choose appropriate searching strategy	**Unit 3: Array Data Structure (8 Hrs.)** 3.1 Introduction to Arrays - array representation 3.2 Advantages and Drawbacks of Arrays 3.2 sorting algorithms with efficiency: Bubble sort, Selection sort, Insertion sort, Merge sort, Quick Sort, Heap Sort, Radix sort, Bucket Sort, Concept of stable and unstable sorting 3.3 Searching Algorithms: Linear Search, Binary Search
• Describe a Linked List • Explain the operations and implementations of Lists. • Understand advantages and limitations of Different types of Linked List • Create and use balanced Trees	**Unit 4: Linked List (10 Hrs.)** 4.1 Introduction to Linked List Data Structure 4.2 Implementation of List – static & dynamic representation, 4.3 Singly Linked List, Circular Linked List, Doubly Linked List, Doubly circular Linked List 4.4 Operations on List: Insertion, Deletion, Searching, Merging

	4.5 Applications of Linked List – polynomial manipulation 4.6 Generalized linked list – concept & representation
• Describe the stack data structure • Identify two basic modes of implementing a stack • Outline the applications of stacks in computing	**Unit 5: Stacks (7 Hrs.)** 5.1 Introduction: Definition, Stack as ADT 5.2 Operations on stack 5.3 Implementation of Stack: Using Arrays and using Linked List 5.4 Application - infix to postfix & prefix, postfix evaluation, bracket matching, recursion 5.5 Concept of Multiple stacks
• Describe a queue data structure • Outline the different applications of queues in computing • Explain the operations on a queue • Understand the different type of queue implementation	**Unit 6: Queues (6 Hrs.)** 6.1 Introduction: Definition, Queue as ADT 6.2 Operations on Queue 6.3 Implementation of Queue: Using Arrays and using Linked List 6.4 Applications- Printing, Scheduling etc 6.4 Circular queue, Dequeue, Priority Queues 6.5 Concept of Multiple Queues
• Give a basic definition of a binary tree and BST • Perform different tree operations • Evaluate arithmetic expressions by means of tree traversals. • Explain the implementation of AVL search trees.	**Unit 7: Trees (6 Hrs.)** 7.1 Concept & Terminologies 7.2 Binary tree, Binary Search Tree 7.3 Implementation of Trees: Static and Dynamic 7.4 Operations on BST – create. Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes 7.5 Balanced Trees: AVL trees and Rotations, Red Black Trees 7.6 Applications: Expression tree
• Describe the graph theory its applications • Understand different representations of graph • Explain graph traversal. • Implement MST and shortest Path Algorithm	**Unit 8: Graph (6 Hrs.)** 8.1 Concept & terminologies 8.2 Graph Representation 8.3 Traversals – BFS & DFS 8.4 Minimum Spanning Trees: Kruskals Algorithm 8.5 Shortest Path Algorithms: Dijkstra Algorithm

6. Recommended Books:

1. Horowitz Sahani, **Fundamentals of Data Structures,** Galgotia Publication
2. Data Structure Using C & C++, Langsam Yedidyah, Augenstein Moshe J., Tennenbaum Aaron M., PHI
3. ISRD Group, **Data Structures using C,** Tata McGraw Hill
4. Nitin Upadhyay, SK, **The Design and Analysis of Algorithm**, Kataria & Sons

Course Title: Data Structure and Algorithms LAB **Credit: 1**
Course No.: CSIT.121
Nature of the Course: LAB
Level: B.Sc. CSIT Year: First Semester: Second

Laboratory Work Guidelines: Students will have to complete the assigned practical work throughout the semester and Practical examination will be conducted at the end of academic semester. The practical exam will be graded on the basis of the following marking scheme:

In-Semester Evaluation (Lab Book or Journal)	25 %
Final Exam Written	50 %
Final Exam Oral	25 %

Following are the guideline for the lab work:

1. There should be a lab book for the practical work related to the subject
2. The lab book will contain details of all practical's to be conducted in the lab
3. Students should read the lab book before coming to the lab
4. Every practical should have:
    a. Title
    b. Objectives
    c. Description
    d. Examples
    e. Self Activities
        i. Objective questions
        ii. Sample programs to be typed and executed
    f. Task list to be decided by the lab in-charge.
    g. Outputs to be verified by the lab in-charge.
5. Each practical should be conducted in the following manner:
    a. Explanation by lab in-charge – 10 minutes
    b. Self activities by students
    c. Lab in-charge will allocate tasks to each student (selection from a list / modify given task / specify new task)

d. At the end of the slot, the lab in-charge has to verify the outputs and give a remark (Complete / Incomplete / Needs Improvement)

Assignment List for Lab Work

All the students will have to complete the following set of programming using the "C" Programming language. Lab in-charge may assign additional assignment depending upon the time available.

1. Sorting Algorithms – Bubble sort, Insertion, selection, quick sort and merge.
2. Static/Dynamic stack implementation, infix to postfix, infix to prefix and evaluation of Postfix.
3. Static and Dynamic Queue Implementation.
4. Singly Linked List, Doubly Linked List and Circular Linked List.
5. Polynomial addition (Using Linked list).
6. Binary Tree Traversal: Create, add, delete, and display nodes.
7. Graph: in degree, out degree, DFS, BFS.
8. Shortest path Dijkstra algorithm.
9. Adjacency matrix to adjacency list conversion.

Recommended Books

5. Horowitz Sahani, **Fundamentals of Data Structures,** Galgotia Publication
6. ISRD Group, **Data Structures using C,** Tata McGraw Hill
7. Ashok Kamthane, **Introduction to Data Structures using C**
8. Bandopadhyay & Dey, **Data Structures using C,** Pearson
9. Nitin Upadhyay, SK, **The Design and Analysis of Algorithm**, Kataria & Sons