GEETHANJALI INSTITUTE OF SCIENCE & TECHNOLOGY

(AUTONOMOUS)
Accredited by NAAC with 'A' Grade, Accredited by NBA (B.Tech.,-ECE, EEE & Mech : 2024-2027
(Approved by AICTE & and UGC, New Delhi & Affiliated to JNTUA, Anantapur.)
3rd Mile, Nellore - Bombay Highway, Gangavaram (V), Kovur (M), S.P.S.R. Nellore Dt, Andhra Pradesh - India. 524 137
website: gist.edu.in / e-mail: geethanjali@gist.edu.in.

TRAINING , PLACEMENTS & CAREER GUIDANCE CELL

I-II CICTT Data Structures Curriculum for CSE & Allied Branches AY:2025-26

Week	Unit	MODULE	Core Topics	Lab Exercises (Guided Practice)	Practice Problems in LMS(Self-Assessment)	Weekly Assessment
1	Revision	Array Manipulation	One & Two-Dimensional Arrays: Array declaration, initialization, and operations on 1D and 2D arrays (matrices).	1. Find min/max in a 1D array. 2. Find missing number, 3. Find unique element, 4. Sum of matrix diagonals.	1. Sum of positive numbers of an array 2. Delete Duplicate Elements from an Array. 3. Matrix addition and multiplication.	Test 1 (50 min)
2	Revision	String Manipulation	Strings: String I/O, array of strings, string manipulation functions (STL).	1. Check for palindrome string, 2. Count vowels/consonants, 3. Print ASCII value, 4.Case conversion, 5.Replace vowels,	1. String operations (copy, concat, length,substring) using library functions. 2. String operations without library functions.	Test 2 (50 min)
3	1	Linear Data Structures Searching	Introduction to Linear Data Structures: Definition and importance of linear data structures, Abstract data types (ADTs) and their implementation, Overview of time and space complexity analysis for linear data structures. Searching Techniques: Linear & Binary Search	1. Reverse a 1D array. 2. C Program to implement the Linear 3. C Program to implement the Binary Search	1. Find 2nd smallest/largest in array. 2. Find first repeating character. 3. Cumulative Sum of Array	Test 3 (50 min)
4	1	Linear Data Structures Sorting	Sorting Techniques: Bubble sort, Selection sort, Insertion Sort	1. C Program to implement Bubble Sort, 2. C Program to implement Selection 3. C Program to implement Insertion Sort	1. Sum of digits 2. Reverse a number 3. GCD/LCM	Test 4 (50 min)
5	Revision	Functions And Recursion	Introduction to function, call by value and call by reference, introduction to Recursion.	1. Check for Palindrome Using function 2. Check for Armstrong number Using function 3. Factorial unsing recursion 4. Print first 'n' Fibonacci terms using recursion	1. Strong number using function 3. Decimal to Binary using recursion 4. Binary to Decimal using function	Test 5 (50 min)
6	2	LinkedList	introduction to Singly linked lists,representation and operations,Comparing arrays and linked lists.	1.Implement a singly linked list and perform insertion, deletion and traversal operations. 2. Develop a program to reverse a linked list iteratively and recursively.		Test 6 (50 min)

7	2	LinkedList	Applications of SLL linked lists.	1. Create a program to detect and remove duplicates from a linked list. 2. Implement a linked list to represent polynomials and perform addition.		Test 7 (50 min)
8	2	LinkedList	Introduction to doubly linked lists and circular linked lists, representation and operations, Applications of Double linked lists.	1. Implement a doubly linked list and perform various operations to understand its properties and applications. 2. Implement a circular linked list and perform insertion, deletion, and traversal.		Test 8 (50 min)
9	3	Stack	Stacks: Introduction to stacks: properties and operations, implementing stacks using arrays and linked lists, Applications of stacks in expression evaluation, backtracking, reversing list etc.	1. Implement a stack using arrays and linked lists. 2. Write a program to evaluate a postfix expression using a stack. 3. Implement a program to check for balanced parentheses using a stack.		Test 9 (50 min)
10	4	Queue	Queues: Introduction to queues: properties and operations, implementing queues using arrays and linked lists, Applications of queues in breadth-first search, scheduling, etc. Deques: Introduction to deques (double-ended queues), Operations on deques and their applications	1. Implement a queue using arrays and linked lists. 2. Develop a program to simulate a simple printer queue system. 3. Solve problems involving circular queues.		Test 10 (50 min)
11	3, 4	Stack and Queue	Applications of Stack and Queue	1. Use a stack to evaluate an infix expression and convert it to postfix. 2. Create a program to determine whether a given string is a palindrome or not. 3. Implement a stack or queue to perform comparison and check for symmetry.		Test 11 (50 min)
12	5	Tree	Introduction to Trees, Binary Tree- Insertion, Deletion & Traversal, Binary Search Tree – Insertion, Deletion & Traversal	1. Implementing a Binary tree using Linked List 2.Traversing of Binary tree(In-order,pre-order,post-order) 3. Implementing a BST using Linked List. 4. Traversing of BST		
13	5	Graph	Introduction to Graphs, Graph Traversals – BFS,DFS.	1. Implementing BFS 2.Implementing DFS		Test 12 (50 min)
14	5	Hashing	introduction to hashing and hash functions, Collision resolution techniques: chaining and open addressing, Hash tables: basic implementation and operations, Applications of hashing in unique identifier generation, caching, etc.	1. Implement a hash table with collision resolution techniques. 2. Write a program to implement a simple cache using hashing.		Test 13 (50 min)

Strategy for Student Performance Assessment and Segregation:

The goal is to create a continuous and holistic assessment model that allows for the accurate segregation of students into performance batches.

1. Assessment Components & Weightage:

Use a combination of all graded activities to calculate a final performance score for each student.

Compon		**Description**
Weekly		14 tests of 50 minutes each, focusing on
Lab		Graded completion of the weekly guided
Practice		Completion of the "Problems for
Total		-

2. Defining Performance Batches:

At the end of the semester, use the final weighted score to segregate students.

These thresholds can be reviewed and adjusted after a mid-term assessment

Batch	**Score Range**
Excellent	**75% - 100%**
Good	**60% - 75%**
Average	**45% - 60%**
Poor	**< 45%**