SILVER OAK UNIVERSITY

College of Technology
Bachelor of Technology
Information Technology
Course Name: Data Structures
Course Code:1010043222
Semester: 3rd

Prerequisite:

Basic of C Programming

Objective:

1. Data structure will continue to be a critical subject in the realm of technology. Organizing and structuring data will remain essential for implementing efficient algorithms and creating software. Effective problem-solving will necessitate the use of suitable data structures in program development.
2. Understanding data structures will remain essential, facilitating comprehension of programming languages. Practicing and assimilating data structure techniques will continue to be vital for programming. Practical exercises throughout the course of study will reinforce knowledge of "C" language and data structures, aiding students in developing the capability to select specific data structures.

Teaching Scheme:

Teaching Scheme				
L	T	P	Contact Hours	Credit
4	0	4	8	6

Contents:

Unit	Topics	Teaching Hours	Weightage %
1	**INTRODUCTION TO DATA STRUCTURE:** Data Management concepts, Data types – primitive and non- primitive, Performance Analysis and Measurement (Time and space analysis of algorithms-Average, best and worst case analysis), Types of Data Structures- Linear & Non Linear Data Structures.	07	10

2	**LINEAR DATA STRUCTURE:** **Array:** Representation of arrays, Applications of arrays, sparse matrix and its representation **Stack:** Stack-Definitions & Concepts, Operations On Stacks, Applications of Stacks, Polish Expression, Reverse Polish Expression And Their Compilation, Recursion, Tower of Hanoi **Queue:** Representation Of Queue, Operations On Queue, Circular Queue, Priority Queue, Array representation of Priority Queue, Double Ended Queue, Applications of Queue **Linked List:** Singly Linked List, Doubly Linked list, Circular linked list, Linked implementation of Stack, Linked implementation of Queue, Applications of linked List.	15	30
3	**NONLINEAR DATA STRUCTURE :** **Tree:** Definitions and Concepts, Representation of binary tree, Binary tree traversal (Inorder, postorder, preorder), Threaded binary tree, Binary search trees, Conversion of General Trees To Binary Trees, Heap, Applications Of Trees- Some balanced tree mechanism, eg. AVL trees, 2-3 trees, Height Balanced, Weight Balance **Graph:** Matrix Representation Of Graphs, Elementary Graph operations,(Breadth First Search, Depth First Search, Spanning Trees, Shortest path, Minimal Spanning Tree)	15	30
4	**HASHING AND FILE STRUCTURES :** **Hashing:** The symbol table, Hashing Functions, Collision- Resolution Techniques, and Applications of Hashing. **File Structure:** Concepts of fields, records and files, Sequential, Indexed and Relative/Random File Organization, Indexing structure for index files, hashing for direct files, Multi-Key file organization and access methods.	10	15
5	**SORTING & SEARCHING:** **Sorting:** Bubble Sort, Selection Sort, Quick Sort, Merge Sort **Searching:** Sequential Search and Binary Search	08	15

Course Outcomes:

Sr. No.	CO Statement	Unit
CO-1	Classify various data structures, storage structures and common operations on them.	1
CO-2	Compose various linear data structures with their representation and perform different operations on them.	2
CO-3	Create various nonlinear data structures with their representation and perform different operations on them.	3

CO-4	Apply various searching and sorting techniques on data sets.	5
CO-5	Evaluate problems using an appropriate data structure to achieve optimal performance and compare its performance with other possible data structures.	4

Teaching & Learning Methodology:

The various methods or tools follows by the faculties to teach the above subject are:

1. The course includes a laboratory, where students have an opportunity to build an appreciation for the concepts being taught in lectures.
2. Problem - based learning

List of Experiments: Total Hours: 56

Sr. No.	Practical Name
1	Introduction to pointers. Call by Value and Call by reference.
2	Introduction to Dynamic Memory Allocation. DMA functions malloc(), calloc(), free() etc.
3	Implement a program for stack that performs following operations using array. PUSH (b) POP (c) PEEP (d) CHANGE (e) DISPLAY
4	Implement a program to convert infix notation to postfix notation using stack.
5	Let stack_ptr be a pointer to a stack of integers and an item be an integer variable. Write functions like Push, Pop, Initialize, Empty, and Full for doing the following tasks. [You may declare additional variable in your functions in needed] (a) Return the top element of the stack and leave the top element unchanged. If the stack is empty, return INT_MAX. (b) Return the third element from the top of the stack, provided that the stack contains at least three integers. If not, return INT_MAX. Leave the stack unchanged. (c) Returns the bottom element of stack (or INT_MAX if stack empty), and leave the stack unchanged. (d) Delete all occurrences of x from the stack, leaving the other elements of the stack in the same order.
6	Write a program to implement QUEUE using arrays that performs following operations (a) INSERT (b) DELETE (c) DISPLAY
7	Write a program to implement Circular Queue using arrays that performs following operations. (a) INSERT (b) DELETE (c) DISPLAY
8	Use the functions developed to write other functions that will (a) Empty one stack onto the top of another stack (b) Move all the items from a queue onto a stack. (c) Start with a queue and an empty stack, and use the stack to reverse the order of all the items in the queue.
9	Write a menu driven program to implement following operations on the singly linked list.

	(a) Insert a node at the front of the linked list. (b) Insert a node at the end of the linked list. (c) Insert a node such that the linked list is in ascending order. (according to info. Field) (d) Delete the first node of the linked list. (e) Delete a node before specified position. (f) Delete a node after specified position.
10	Write a program to implement stack using linked list.
11	Write a program to implement Queue using a linked list.
12	Write a program to implement following operations on the doubly linked list. (a) Insert a node at the front of the linked list. (b) Insert a node at the end of the linked list. (c) Delete the last node of the linked list. (d) Delete a node before specified position.
13	Write a program to implement following operations on the circular linked list. (a) Insert a node at the end of the linked list. (b) Insert a node before the specified position. (c) Delete the first node of the linked list. (d) Delete a node after specified position.
14	Write a program which creates a binary search tree.
15	Implement recursive and non-recursive tree traversing methods inorder, preorder and post- order traversal.
16	Write a program to implement Bubble Sort, Merge Sort, Quick Sort.
17	Write a program to implement Binary Search.

Major Equipment:

C Compiler

Books Recommended:

1. Jean-Paul Tremblay & Paul G. Sorenson, "An Introduction to Data Structures with Applications", Tata McGraw Hill.
2. Aaron M. Tenenbaum,"Data Structures using C & C++", Prenctice-Hall International.
3. Horowitz, Sahni, "Fundamentals of Computer Algorithms", Galgotia Pub.
4. Sartaj Sahani "Fundamentals of Data Structures in C++" ,Computer Science Press.
5. Gilberg & Forouzan, "Data Structures: A Pseudo-code approach with C", Thomson Learning.

List of Open Source Software/learning website:

1. http:/silveroakuni.ac.in/video-lecture
2. https:/www.youtube.com/watch?v=RBSGKlAvoiM
3. https:/www.youtube.com/watch?v=zWg7U0OEAoE

CO-PO-Matrix

Co. No.	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO-1	2	3	1	3	3				2		1	2	1	1
CO-2	3	1	3	3	3				1			2	1	1
CO-3	3	2	3	2	3				1		1	2	1	1
CO-4	2	3	2	2	2							1	1	1
CO-5	2	2	1	2	2							2	1	1