SARVAJANIK UNIVERSITY
INCLUSIVE | INTEGRATED | INNOVATIVE

SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and Technology
Bachelor of Technology

Year: B. Tech II (Semester III)

Subject Name: Data Structures and Applications
Subject Code: BTCO13301
Type of course: Professional Core Course
Prerequisite: Basic Programming Concepts

Rationale: This course is of prime importance in Computer science and engineering. Understanding of various data structures is essential for efficient problem solving. This course introduces static and dynamic data structures and their applications.

Teaching and Examination Scheme:

TEACHING SCHEME				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
3	1	2	5	60	25	15	30	20	150

CA1: Continuous Assessment (assignments/projects/open book tests/closed book tests) CA2: Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems TEE: Term End Examination TEP: Term End Practical Exam (Performance and viva on practical skills learned in course) CA3: Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

Content:

Sr. No.	Content	Total Hrs
1	**Module:1** : Introduction to Algorithms and Analysis : Overview and importance of algorithms and data structures. Abstract Data Types, Basic concepts of data Structures, Types of Data structures –Data structure operations, Fundamentals of algorithm analysis, Space and time complexity of an algorithm, Types of asymptotic notations and orders of growth, Algorithm efficiency – best case, worst case, average case.	6
2	**Module:2 :** Linear Data Structures : Stack - Applications of stack: Expression Evaluation - Conversion of Infix to postfix and prefix expression, Recursion and Tower of Hanoi. Queue - Types of Queue: Simple Queue, Circular Queue, Double Ended Queue (deQueue), Priority Queue using Arrays, Applications List - Singly linked lists – Doubly linked lists - Circular linked lists, Applications -Polynomial Manipulation	10

3	**Module:3** : Sorting and Search Techniques : Searching - Linear Search and binary search, Sorting - Insertion sort - Selection sort – Bubble sort – Counting Sort - Quick sort- Merge sort , Analysis, Applications, Heaps - Heap sort, Applications -Priority Queue using Heaps	7
4	**Module:4** : Non-linear Data Structures – Trees : Tree - Terminology, Binary Tree – Terminology and Properties, Tree Traversals, Expression Trees – Binary Search Trees – operations in BST – insertion, deletion, finding min and max, Finding the kth minimum element in a BST, AVL trees – Terminology - basic operations(rotation, insertion and deletion)	8
5	**Module:5** : Non-linear Data Structures – Graphs : Graph – basic definition and Terminology – Representation of Graph – Graph Traversal: Breadth First Search (BFS), Depth First Search (DFS) - Minimum Spanning Tree: Prim's, Kruskal's- Single Source Shortest Path: Dijkstra's Algorithm.	8
6	**Module:6** : Hashing : Hash functions, open hashing-separate chaining, closed hashing - linear probing, quadratic probing, double hashing, random probing, rehashing, extendible hashing, Applications – DictionaryTelephone directory	6

Suggested Specification table with Marks (Theory): (For B.Tech only)

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
15	25	20	0	0	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

Reference Books:

Sr no	Title of book /article	Author(s)	Publisher and details like ISBN	Year of publication	Publication Edition
1	An Introduction to Data Structures with Applications	Jean-Paul Tremblay and Paul G. Sorenson	Tata McGraw Hill		

2	Data Structures using C & C++	Langsam and Augenstein and Tenenbaum	PHI		
3	Fundamentals of Data Structures in C++	Sartaj Sahni			

Course Outcomes (CO):

Sr. No.	CO statement	Marks % weightage
CO-1	Differentiate primitive and non-primitive structures	20%
CO-2	Design and apply appropriate data structures for solving computing problems.	20%
CO-3	Apply sorting and searching algorithms to the small and large data sets	20%
CO-4	Ability to describe stack,queue and linked list operation.	20%
CO-5	Ability to have knowledge of tree and graphs concepts.	20%

List of Open learning website:

1. https://www.programiz.com/dsa
2. https://www.tutorialspoint.com/data_structures_algorithms/index.htm
3. https://www.javatpoint.com/data-structure-tutorial

List of Open Source Software:

1. Visual Studio Code
2. Eclipse
3. Code::Blocks
4. Geany
5. CodeLite

List of Experiments:

SARVAJANIK UNIVERSITY
INCLUSIVE | INTEGRATED | INNOVATIVE

SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and Technology
Bachelor of Technology

Sr. No	Practical
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. (a) PUSH (b) POP (c) PEEP (d) CHANGE (e) DISPLAY
4.	Implement a program to convert infix notation to postfix notation using stack.
5.	Write a program to implement QUEUE using arrays that performs following operations (a) INSERT (b) DELETE (c) DISPLAY
6.	Write a program to implement Circular Queue using arrays that performs the following operations. (a) INSERT (b) DELETE (c) DISPLAY
7.	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 linked list is in ascending order.(according to info. Field) (d) Delete a first node of the linked list. (e) Delete a node before specified position. (f) Delete a node after specified position.
8.	Write a program to implement stack using linked list.
9.	Write a program to implement queue using linked list.
10.	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 a last node of the linked list. (d) Delete a node before specified position.
11.	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 specified position. (c) Delete a first node of the linked list. (d) Delete a node after specified position.
12.	Write a program which create binary search tree.

13	Implement recursive and non-recursive tree traversing methods inorder, preorder and postorder traversal.
14	Write a program to implement Queue Sort
15	Write a program to implement Merge Sort
16	Write a program to implement Bubble Sort
17	Write a program to implement Binary Search