



## B.C.A. (Honours) & B.C.A. (Honours with Research)

(Semester - 1 and Semester - 2)

Saurashtra University

To be effective from June – 2023

### CS-08: DATA STRUCTURE USING C LANGUAGE

#### Objectives:

- To provide the knowledge of basic data structures and their implementations.
- To understand importance of data structures in context of writing efficient programs.
- To develop skills to apply appropriate data structures in problem solving

#### Prerequisites:

- Computer Programming Knowledge
- Fundamental knowledge of C Programming

Sr. No.	Topic	Detail
1	Algorithm Analysis	<ul style="list-style-type: none"><li>• The analysis of algorithm.</li><li>• Time and space complexities.</li><li>• Asymptotic notation.</li><li>• Classes of algorithm.</li><li>• Big-Oh Notation</li><li>• Big-Omega Notation</li></ul>
	File Handling	<ul style="list-style-type: none"><li>• Concept of data files</li><li>• File handling</li><li>• Use of file handling functions fopen, fclose, fprintf, fscanf, getw, putw, fseek, ftell, rewind, freopen, remove, rename, feof, ferror</li><li>• I/O operations</li><li>• Command line arguments</li></ul>
2	Sorting and Searching	<ul style="list-style-type: none"><li>• Bubble sorting</li><li>• Insertion sorting</li><li>• Quick sorting</li><li>• Bucket sorting</li><li>• Merge sorting</li><li>• Selection sorting</li><li>• Shell sorting</li><li>• Basic searching technique: Index searching, Sequential searching, Binary searching</li></ul>
3	Introduction To data Structure	Primitive and simple structures Linear and nonlinear structures file organization.
	Elementary Data Structure	<ul style="list-style-type: none"><li>• Stack<ul style="list-style-type: none"><li>○ Definition</li><li>○ Operations on stack</li></ul></li></ul>



## B.C.A. (Honours) & B.C.A. (Honours with Research)

(Semester - 1 and Semester - 2)

Saurashtra University

To be effective from June – 2023

		<ul style="list-style-type: none"><li>○ Implementation of stacks using arrays<ul style="list-style-type: none"><li>■ Function to insert an element into the stack</li><li>■ Function to delete an element from the stack</li><li>■ Function to display the items</li></ul></li><li>○ Recursion and stacks</li><li>○ Evaluation of expressions using stacks<ul style="list-style-type: none"><li>■ Postfix expressions</li><li>■ Prefix expression</li></ul></li><li>● Queue<ul style="list-style-type: none"><li>○ Introduction</li><li>○ Array implementation of queues</li><li>○ Function to insert an element into the queue</li><li>○ Function to delete an element from the queue</li></ul></li><li>● Circular queue<ul style="list-style-type: none"><li>○ Function to insert an element into the queue</li><li>○ Function for deletion from circular queue</li><li>○ Circular queue with array implementation</li></ul></li><li>● Deques</li><li>● Priority queues</li></ul>
4	<b>Linked List &amp; Implementation</b>	<ul style="list-style-type: none"><li>● Applications of the linked lists</li><li>● Types of Linked Lists<ul style="list-style-type: none"><li>○ Singly Linked List</li><li>○ Doubly linked list</li><li>○ Header Linked List</li><li>○ Circular Linked List</li></ul></li><li>● Implementation using Singly Linked List, Doubly Linked List and Circular Singly Linked List<ul style="list-style-type: none"><li>○ Insertion of a node at the beginning</li><li>○ Insertion of a node at the end</li><li>○ Insertion of a node after a specified node</li><li>○ Traversing the entire linked list</li><li>○ Deletion of a node from linked list</li><li>○ Updating of a specific node</li></ul></li><li>● Implementation of merging of two Singly Linked List</li><li>● Implementation of reversing of Singly Linked List</li></ul>
5	<b>Tree</b>	<ul style="list-style-type: none"><li>● Objectives</li><li>● Properties of a tree</li><li>● Binary trees<ul style="list-style-type: none"><li>○ Properties of binary trees</li><li>○ Implementation</li><li>○ Traversals of a binary tree</li></ul></li></ul>



## B.C.A. (Honours) & B.C.A. (Honours with Research)

(Semester - 1 and Semester - 2)

Saurashtra University

To be effective from June – 2023

		<ul style="list-style-type: none"><li>▪ In order traversal</li><li>▪ Post order traversal</li><li>▪ Preorder traversal</li><li>• Binary search trees (bst)<ul style="list-style-type: none"><li>○ Insertion in bst</li><li>○ Deletion of a node</li><li>○ Search for a key in bst</li></ul></li><li>• Height balanced tree</li><li>• B-tree Algorithm<ul style="list-style-type: none"><li>○ Insertion, Deletion</li></ul></li></ul>
	<b>Graph</b>	<ul style="list-style-type: none"><li>• Adjacency matrix and adjacency lists</li><li>• Graph traversal<ul style="list-style-type: none"><li>○ Depth First Search (DFS)</li><li>○ Implementation</li><li>○ Breadth First Search (BFS)</li><li>○ Implementation</li></ul></li><li>• Shortest path problem</li><li>• Minimal spanning tree</li></ul>

Seminar - 5 Lectures

Expert Talk - 5 Lectures

Test - 5 Lectures

**Total Lectures 60 + 15 = 75**

### Reference Books:

1. Data Structure through C/C++ Author : Tennaunbuam.
2. Let us C Author : Kanitkar.
3. Pointer in C Author : Kanitkar.
4. Data and File Structure Author : Trembley & Sorrenson.

### Course Outcome:

- Able to Understand basic data structures and their implementations.
- Able to Understand importance of data structures in context of writing efficient programs.
- Able to Develop skills to apply appropriate data structures in problem solving
- Able to Explore tree and graph data structure

### Additional Topics to be taught during the semester – 2 (Not to be asked in examination):

- Case studies of data structure