

1. Basics of C Programming

- Review C programming fundamentals (if not already familiar).
 - **Variables and Data Types**
 - **Control Structures** (if, else, switch, loops)
 - **Functions and Recursion**
 - **Pointers and Memory Allocation**
 - **Structures and Unions**

2. Introduction to Data Structures

- Understand what data structures are and their importance.
- Learn about abstract data types (ADTs).

3. Arrays

- Declaration, Initialization, and Access.
- Multi-dimensional Arrays.
- Common operations (insertion, deletion, searching, sorting).

4. Strings

- Declaration and Initialization.
- String manipulation functions (`strlen`, `strcpy`, `strcat`, `strcmp`, etc.).

5. Linked Lists

- **Singly Linked List:**
 - Definition and implementation.
 - Operations: insertion, deletion, traversal, searching.
- **Doubly Linked List:**
 - Definition and implementation.
 - Operations: insertion, deletion, traversal.
- **Circular Linked List:**
 - Definition and implementation.
 - Operations: insertion, deletion, traversal.

6. Stacks

- Definition and applications.
- Implementation using arrays and linked lists.
- Operations: push, pop, peek, isEmpty.

7. Queues

- Definition and applications.
- Implementation using arrays and linked lists.
- Operations: enqueue, dequeue, peek, isEmpty.
- **Circular Queue** and **Deque** (Double-ended Queue).

8. Trees

- **Binary Trees:**
 - Definition and properties.
 - Implementation and traversal (in-order, pre-order, post-order).
- **Binary Search Trees (BST):**
 - Definition and properties.
 - Operations: insertion, deletion, searching.
- **Balanced Trees** (like AVL Trees, Red-Black Trees) - basic understanding.
- **Heaps:**
 - Definition and types (min-heap, max-heap).
 - Implementation and operations (insert, delete).

9. Graphs

- Definition and types (directed, undirected).
- Representation (adjacency matrix, adjacency list).
- Traversal algorithms (BFS, DFS).
- Basic algorithms (Dijkstra's, Prim's, Kruskal's).

10. Hashing

- Concept of hashing.
- Hash functions.
- Collision resolution techniques (chaining, open addressing).

11. Advanced Data Structures (Optional)

- **Tries**
- **Segment Trees**
- **Fenwick Trees (Binary Indexed Trees)**

12. Algorithms

- **Sorting Algorithms:**
 - Bubble Sort, Selection Sort, Insertion Sort.
 - Quick Sort, Merge Sort, Heap Sort.

- **Searching Algorithms:**
 - Linear Search, Binary Search.
- **Dynamic Programming** - basic concepts and common problems.
- **Greedy Algorithms** - basic concepts and common problems.

13. Practice and Projects

- Solve problems on platforms like LeetCode, HackerRank, CodeChef.
- Implement data structures from scratch.
- Work on small projects to reinforce concepts (e.g., a text editor, a simple game, etc.).