DATA STRUCTURES

TECHSQUARE SOLUTIONS

Module 1: Introduction to Data Structures
- Definition and Classification of Data Structures
- Need and Importance of Data Structures
- Abstract Data Type (ADT)
- Types of Data Structures: Linear vs Non-Linear, Static vs Dynamic, Homogeneous vs Heterogeneous
- Applications of Data Structures in Real Life

Module 2: Arrays
- Definition and Basics
- One-dimensional and Multi-dimensional Arrays
- Operations on Arrays:Insertion, Deletion, Traversal, Searching, and Sorting
- Sparse Matrix and its Representation
- Applications of Arrays
- Strings and various operations on Strings

Module 3: Linked Lists
- Introduction to Linked Lists
- Types of Linked Lists: Singly Linked List, Doubly Linked List, Circular Linked List
- Operations on Linked Lists: Insertion, Deletion, Traversal, and Searching
- Applications of Linked Lists
- Comparison of Arrays and Linked Lists

Module 4: Stacks
- Definition and Concept of Stack
- Stack Operations (Push, Pop, Peek)
- Stack Implementation: Arrays / Linked Lists
- Applications of Stack
- Expression Evaluation (Infix to Postfix, Prefix Conversion)
- Backtracking (Maze Problem, N-Queens Problem)
- Function Calls and Recursion

Module 5: Queues
- Definition and Concept of Queue
- Types of Queues: Simple Queue, Circular Queue, Double-Ended Queue (Deque), Priority Queue
- Queue Implementation: Arrays / Linked Lists
- Applications of Queues:
- Scheduling in Operating Systems
- Printer Queue Management

Module 6: Recursion
- Concept of Recursion
- Difference Between Recursion and Iteration
- Direct and Indirect Recursion
- Tail and Non-Tail Recursion
- Factorial Calculation / Fibonacci Series
- Tower of Hanoi / Recursive Tree and Graph Traversal

Module 7: Searching Algorithms
- Linear Search, Binary Search, Jump Search
- Interpolation Search, Exponential Search
- Applications and Complexity Analysis

Module 8: Sorting Algorithms
- Bubble Sort, Selection Sort, Insertion Sort
- Merge Sort, Quick Sort, Heap Sort
- Counting Sort, Radix Sort, and Bucket Sort
- Comparison of Sorting Algorithms

Module 9: Trees
- Introduction to Trees
- Binary Trees: Types (Full, Complete, Balanced, Degenerate)
- Traversals (Preorder, Inorder, Postorder)
- Operations (Insertion, Deletion, Search)
- Binary Search Tree (BST)
- AVL Trees (Self-Balancing BST)
- B-Trees and B+ Trees
- Applications of Trees: Expression Trees, Huffman Coding, Decision Trees

Module 10: Graphs
- Introduction to Graphs
- Directed and Undirected Graphs / Weighted and Unweighted Graphs
- Graph Representation: Adjacency Matrix and Adjacency List
- Graph Traversal Algorithms:
- BFS and DFS
- Shortest Path Algorithms: Dijkstra's Algorithm, Floyd-Warshall Algorithm and Bellman-Ford Algorithm
- Applications of Graphs: Social Networks, Google Maps (Path Finding) and Computer Networks

Module 11: Hashing
- Concept of Hashing / Hash Functions
- Collision Resolution Techniques:
- Open Addressing (Linear Probing, Quadratic Probing, Double Hashing) and

- Chaining (Separate Chaining)
- Applications of Hashing: Symbol Tables and Database Indexing

Module 12: Heaps and Priority Queues
- Introduction to Heaps
- Types of Heaps: Min Heap andMax Heap
- Heap Operations: Insertion, Deletion, Heapify
- Priority Queue Implementation Using Heaps
- Heap Sort Algorithm
- Applications of Heaps: Job Scheduling and Memory Management

Module 13: Advanced Data Structures
- Trie (Prefix Tree) and its Applications
- Segment Tree (Range Queries)
- Fenwick Tree (Binary Indexed Tree)
- Disjoint Set (Union-Find Algorithm)
- Red-Black Trees
- Splay Trees

Module 14: Complexity Analysis and Optimization
- Time Complexity and Space Complexity
- Big-O, Big-Theta, and Big-Omega Notations
- Best, Worst, and Average Case Analysis
- Amortized Analysis
- Optimization Techniques for Efficient Data Structures

Module 15: Real-World Applications and Case Studies
- Data Structures in Database Management Systems (DBMS)
- Data Structures in Artificial Intelligence and Machine Learning
- Data Structures in Web Development
- Case Studies of Large-Scale Systems (Google Search, Amazon Recommendations)

Choice of Programming Language

1. C (Best for Fundamentals & Interviews)
- C is **closest to memory management**, making it great for understanding pointers, dynamic memory allocation, and linked lists.
- Used in **competitive programming** and **technical interviews**.
- Helps build a **strong foundation** in algorithmic thinking.

Best For:
- Beginners who want to understand how data structures work in detail at a low level.
- Students preparing for **interviews in core software companies**.

2. C++ (Best for Competitive Programming & Industry Use)
- Has **STL (Standard Template Library)**, which makes implementing data structures easier.
- Faster execution compared to Java/Python.
- Used in **competitive programming** (Codeforces, Leetcode, etc.).

Best For:
- Students interested in **competitive programming**.
- Those preparing for **coding interviews** in companies like Google, Amazon, etc.

3. Java (Best for Job-Oriented Learning)
- Provides **built-in data structures** like ArrayList, HashMap, TreeMap, etc.
- **Widely used in software development** and backend applications.
- Object-oriented approach helps in understanding **real-world applications** of data structures.

Best For:
- Students who aim for **software development jobs**.
- Those preparing for **Java-based interviews** (Spring, Hibernate, etc.).

4. Python (Best for Simplicity & Machine Learning)
- **Easy syntax**, great for beginners.
- Has built-in data structures like lists, sets, dictionaries.
- Used in **Data Science, AI, and Web Development**.

Best For:
- Students interested in **AI, ML, and Data Science**.
- Those who **find C/C++ too difficult**.

5. JavaScript (For Web Developers)
- Data structures are used in **frontend (React, Angular)** and **backend (Node.js)**.
- Great for **full-stack development**.

Best For:
- Students who want to work in **Web Development**.
- Those interested in **frontend/backend programming**.