

DSA Training Course

Beginner Level

Module 1: Introduction to Programming and Computers

- **Topics:**
 - What is Programming?
 - How Computers Work: Hardware, Software, and Operating Systems
 - Introduction to Programming Languages (what are they, examples)
 - Setting up a Development Environment (Installing an IDE, Introduction to Text Editors)
 - Understanding Compilers and Interpreters
- **Mini-Project:**
 - "Hello World" Program: A simple program to print text to the screen.

Module 2: Basic Programming Concepts

- **Topics:**
 - Variables and Data Types (int, float, string, boolean)
 - Operators (arithmetic, logical, relational)

- Input and Output (taking user input and displaying output)
 - Control Flow (if, else, and else-if statements)
 - Loops (for, while loops)
- **Mini-Project:**
 - Simple Calculator: Build a basic calculator that takes user input and performs operations.

Module 3: Introduction to Algorithms

- **Topics:**
 - What is an Algorithm?
 - Importance of Algorithms in Problem Solving
 - Basic Algorithm Steps (Input, Process, Output)
 - Pseudocode and Flowcharts
 - Time Complexity (Big O notation – an introduction)
- **Mini-Project:**
 - "Guess the Number" Game: An interactive game where the program selects a random number, and the user has to guess it.

Module 4: Arrays and Lists

- **Topics:**
 - What are Arrays?
 - One-dimensional Arrays (definition, initialization, access)
 - Multi-dimensional Arrays (introduction and usage)
 - Array Operations (searching, sorting, inserting, deleting)

- Introduction to Lists and Linked Lists
 - **Mini-Project:**
 - Array Manipulation Program: Create a program that accepts a list of numbers and performs operations like adding, removing, and sorting.
-

Intermediate Level

Module 5: Searching and Sorting Algorithms

- **Topics:**
 - Linear Search and Binary Search (concepts and implementation)
 - Bubble Sort, Selection Sort, and Insertion Sort (concepts, implementation, and time complexity)
 - Merge Sort and Quicksort (introduction and basic implementation)
- **Mini-Project:**
 - Sorting Visualization: Create a program that visually demonstrates different sorting algorithms on a set of random numbers.

Module 6: Stacks and Queues

- **Topics:**
 - What is a Stack? (definition, LIFO principle, implementation using arrays and linked lists)
 - What is a Queue? (definition, FIFO principle, implementation)
 - Applications of Stacks and Queues (e.g., Undo operation, Task Scheduling)
- **Mini-Project:**

- Implement a Stack-based Expression Evaluator: Evaluate mathematical expressions like "3 + 5 * 2".

Module 7: Recursion

- **Topics:**

- What is Recursion?
- Recursive Function Structure (base case, recursive case)
- Common Recursion Examples (factorial, Fibonacci series)
- Recursion vs Iteration

- **Mini-Project:**

- Recursive Number Printing: Print numbers from 1 to N using recursion.

Module 8: Linked Lists

- **Topics:**

- What is a Linked List? (definition, structure, and advantages over arrays)
- Types of Linked Lists (Singly, Doubly, and Circular Linked Lists)
- Operations on Linked Lists (insertion, deletion, traversal)
- Reverse a Linked List

- **Mini-Project:**

- Linked List Implementation: Create a program that manages a list of students, adding and removing student names.

Advanced Level

Module 9: Trees and Graphs

- **Topics:**
 - What is a Tree? (basic terminology, binary trees)
 - Tree Traversals (In-order, Pre-order, Post-order)
 - Binary Search Tree (definition, operations)
 - What is a Graph? (types: directed, undirected, weighted, unweighted)
 - Graph Traversal Algorithms (Breadth-First Search, Depth-First Search)
- **Mini-Project:**
 - Binary Search Tree Implementation: Create and implement a binary search tree and perform basic operations like search, insert, and delete.

Module 10: Dynamic Programming

- **Topics:**
 - Introduction to Dynamic Programming (concept, problems that can be solved using DP)
 - Basic Problems (Fibonacci sequence, Knapsack problem)
 - Memoization vs Tabulation
- **Mini-Project:**
 - Fibonacci Series using Dynamic Programming: Write an efficient Fibonacci series program using memoization.

Module 11: Greedy Algorithms

- **Topics:**
 - What is a Greedy Algorithm? (concept and approach)
 - Greedy Algorithm Problems (e.g., Coin Change, Fractional Knapsack)

- Analyzing Greedy Algorithms (optimal solutions and limitations)
- **Mini-Project:**
 - Coin Change Problem: Implement a program to solve the coin change problem using a greedy algorithm.

Module 12: Graph Algorithms

- **Topics:**
 - Shortest Path Algorithms (Dijkstra's Algorithm, Bellman-Ford)
 - Minimum Spanning Tree (Prim's Algorithm, Kruskal's Algorithm)
 - Topological Sorting (for Directed Acyclic Graphs)
- **Mini-Project:**
 - Implement Dijkstra's Algorithm: Solve the shortest path problem in a weighted graph.