Java In Data Structures and Algorithm

Course Duration: 40 Hours **Course Code: JDSA101**

1. Course Overview

The **JDSA101 – Java in Data Structures and Algorithm** course is designed to provide learners with a strong foundation in **data structures, algorithms, and problem-solving techniques** using **Java programming language**. This course emphasizes practical learning with real-world coding exercises to help students build **efficient, optimized, and scalable solutions** for technical and competitive programming challenges.

By the end of the course, participants will gain proficiency in designing algorithms, analyzing complexity, and implementing data structures with Java.

2. Learning Objectives

- Understand the fundamentals of **data structures and algorithms**.
- Implement common **data structures** (arrays, linked lists, stacks, queues, trees, graphs) in Java.
- Learn **algorithmic techniques** such as recursion, divide and conquer, dynamic programming, and greedy algorithms.
- Analyze **time and space complexity** using Big-O notation.
- Solve **real-world and competitive programming problems** efficiently.
- Prepare for **coding interviews** and technical assessments.

Phone: +91-999-911-1686 **Mail Us:** info@ssdntech.com

3. Target Audience

- Students and beginners learning **Java and algorithms**.
- Software developers preparing for **technical interviews**.
- Competitive programming enthusiasts.
- Professionals looking to strengthen **problem-solving skills**.

4. Pre-Requisites

- Basic knowledge of **Java programming**.
- Understanding of **OOP concepts** (Classes, Objects, Inheritance, Polymorphism).
- Familiarity with **loops, conditionals, and arrays**.

5. Course Modules

Module 1: Introduction to DSA with Java

- Importance of Data Structures & Algorithms
- Introduction to Big-O notation (Time & Space Complexity)
- Problem-solving approaches

Module 2: Arrays and Strings

- One-dimensional & multi-dimensional arrays
- String operations and algorithms
- Searching and sorting in arrays

Module 3: Linked Lists

- Singly linked list implementation
- Doubly and circular linked lists

- Applications of linked lists

Module 4: Stacks and Queues

- Stack implementation using arrays and linked lists
- Queue, Circular Queue, and Priority Queue
- Deque and applications

Module 5: Recursion and Backtracking

- Basics of recursion
- Backtracking algorithms (Maze, N-Queens, Sudoku solver)

Module 6: Trees

- Binary Trees and Binary Search Trees (BST)
- Tree traversals (Inorder, Preorder, Postorder)
- Heaps and Heap Sort
- Introduction to Trie

Module 7: Graphs

- Graph representation (Adjacency Matrix, List)
- Breadth-First Search (BFS) and Depth-First Search (DFS)
- Shortest path algorithms (Dijkstra, Floyd-Warshall)
- Minimum Spanning Tree (Kruskal, Prim)

Module 8: Advanced Algorithms

- Divide and Conquer (Merge Sort, Quick Sort)
- Greedy Algorithms (Activity Selection, Huffman Coding)
- Dynamic Programming (Knapsack, Fibonacci, Matrix Chain Multiplication)

Module 9: Interview Problem Solving

- Common coding interview problems
- LeetCode, HackerRank, and Codeforces style challenges
- Best practices for solving DSA problems