**Trainer Name:** Er. Manish Singh [M.Tech(CSE), B.Tech(CSE)]
**Trainer Experience:** 12+ Years of IT Industry

# Course Overview:

**Current C Standard for Implementation:** C23
**Training Mode:** Hands-on Offline classroom program
**Training Duration:** Regular 45 Days (2 Hours Per Day)
**Training Fees:** ₹3000
**Training Deliverables:**

- Daily Practical Assignment
- Per Day Lab Sessions
- Interview Oriented Programs
- One Complete Real-Time Working Project
- Course Completion Certificate from Naomika Computer Consultancy

## About Data Structure:

Data Structure is a method of organizing, storing and managing the data in computer memory so that it can be accessed and modified efficiently. It defines the format for arranging the data, physically storing it and defines the operations that can be performed on the data such as insertion, deletion, search, and updating. There are different types of data structures that come with different types of operations. We choose appropriate data structure to solve our problem in less time and space.

## Objective of Learning Data Structure:

The main objective of learning data structure is to acquire skills to select more appropriate data structure for a given problem so that we can develop a software that solves the problem in less time and uses memory space efficiently. Data structure enhances coding skills and improves the ability to write clean and robust code. Data structure lays down the core data management concepts that help us to understand more advanced data management algorithm and foster us to understand new technology.

## Why are Data Structures So Important?

The Data structures and algorithms are a core component of technical interviews at many tech companies, and proficiency in this area is crucial for getting IT jobs.

# Course Contents:

## Module 1: Introduction to Data Structures

- What is Data Structure?
- Types of Data Structures
  - Linear Data Structures
  - Non-Linear Data Structures
- Abstract Data Types(ADTs)
- What is an Algorithm?
- The Analysis of Algorithms
- Goal of the Analysis of Algorithms
- Running Time Analysis
- How to Compare Algorithms
- What is Rate of Growth?
- Types of Analysis
- Asymptotic Notation
- Big-O Notation (Upper Bounding Function)
- Omega-Q Notation (Lower Bounding Function)
- Theta-Θ Notation (Order Function)

## Module 2: Arrays

- Definition
- Single and Multi-Dimensional Arrays
- Common Operations on Arrays
- Representation of Arrays
  - Row Major Order
  - Column Major Order
- Applications of Arrays
- Sparse Matrices and Their Representations
- Arrays Problem and Solution

## Module 3: Linked Lists

- What is Linked Lists?
- Linked Lists ADTs
- Why Linked Lists?
- Implementation
  - Array Implementation
  - Pointer Implementation
- Singly Linked Lists
- Doubly Linked Lists
- Circular Linked Lists
- Linked Lists Problem and Solution

## Module 4: Stacks

- What is a Stack?
- How Stacks are used
- Stack ADT
- Applications
- Implementation
  - Array Implementation
  - Linked List Implementation
- Comparison of Implementations

- Stacks: Problems & Solutions

## Module 5: Queues

- What is a Queue?
- How are Queues Used?
- Queue ADT
- Exceptions
- Applications
- Implementation
  - Array Implementation
  - Linked List Implementation
- Queues: Problems & Solutions

## Module 6: Trees

- What is a Tree?
- Basic Terminology
- Binary Trees
- Representation
  - Array Representation
  - Linked List Representation
- Types of Binary Trees
- Properties of Binary Trees
- Operations on Binary Trees
- Binary Tree Traversals
- Binary Search Trees (BSTs)
- Balanced Binary Search Trees
- AVL(Adelson-Velskii and Landis) Trees
- B Tree
- Threaded Binary Trees

## Module 7: Priority Queues and Heaps

- What is a Priority Queue?
- Priority Queue ADT
- Priority Queue Applications
- Priority Queue Implementations
- Heaps and Binary Heaps
- Binary Heaps
- Heapsort
- Priority Queues [Heaps]: Problems & Solutions

## Module 8: Graphs

- Introduction
- Applications of Graphs
- Graph Representation
  - Adjacency Matrix
  - Adjacency List
- Graph Traversals
  - Depth First Search
  - Breadth First Search
- Shortest Path Algorithms
- Minimal Spanning Tree
  - Prim's Algorithm

    - Kruskal's Algorithm
- Graph Algorithms: Problems & Solutions

## Module 9: Sorting

- What is Sorting?
- Why is Sorting Necessary?
- Classification of Sorting Algorithms
- Bubble Sort
- Selection Sort
- Insertion Sort
- Shell Sort
- Merge Sort
- Heap Sort
- Quick Sort
- Radix Sort
- Sorting: Problems & Solutions

## Module 10: Searching

- What is Searching?
- Why do we need Searching?
- Types of Searching
- Unordered Linear Search
- Sorted/Ordered Linear Search
- Binary Search
- Symbol Tables and Hashing
- String Searching Algorithms
- Searching: Problems & Solutions