LEARN 2 EARN LABS

Programming Fundamentals

(Learn the concept of C, C++ & Data Structure)

Curriculum Brochure

Programming Fundamentals

TRAINING

Programming fundamentals is a foundational course that introduces students to the basics of programming languages such as C and C++. It also includes an introduction to data structures, which are essential for solving complex problems efficiently.

The scope of a programming fundamentals course is vast, as it covers the fundamental concepts of programming that are essential for students pursuing a career in computer science or related fields. The course starts with an introduction to computer programming, where students learn the basic concepts of programming languages and how to write simple programs using C and C++.

The course then progresses to more advanced topics such as control structures, functions, arrays, pointers, and dynamic memory allocation. These topics are essential for writing complex programs that can solve real-world problems. Students also learn how to design and implement algorithms using programming languages, which are essential for solving complex problems efficiently.

The course also includes an introduction to data structures such as stacks, queues, linked lists, trees, and graphs. Data structures are essential for storing and manipulating data efficiently, and they are used extensively in software engineering and computer science. Students learn how to design, implement, and analyze the efficiency of data structures using programming languages.

By the end of the course, students should be able to write simple to moderately complex programs using C and C++. They should also be familiar with basic data structures and algorithms, which are essential for solving problems efficiently. The scope of the course is not limited to just writing code, but also includes debugging, testing, and analyzing the efficiency of the code.

In conclusion, the scope of a programming fundamentals course that includes C, C++, and data structures is vast and essential for students pursuing a career in computer science or related fields. It covers the fundamental concepts of programming languages, data structures, and algorithms, which are essential for solving complex problems efficiently. The course provides students with a strong foundation that they can build upon in their future coursework and careers.

Programming Fundamentals

This program is designed to train students with the skills to write programs by understanding the concepts of C & C++. With Data Structure, they would be capable in defining & implementing algorithm to solve complex problems.

Practical Based Sessions

Training program available in four months & six months duration

Dummy Projects

To build your hands-on expertise & portfolio

Resume Building Assistance

To create an attractive resume for your candidature

Interview Preparation

So you can present your skills in a better way

Mentoring & Job Assistance

To help you in getting good career or placements

Who can join

Any graduate or post graduate student from B.tech or M.tech (any specialization), BCA or MCA, B.Sc. or M.Sc. (CS / IT) can join this Programming Fundamentals training program .

Anyone who wants to learn programming and understand the data structure concepts can join this Programming Fundamentals training program.

Training Mode

Online Live Classes are available

- 4x more effective way of learning
- Hands-on experience with projects & assignments
- Virtual class with real interaction with trainer
- Monitoring support & troubleshooting issues
- Masterclass from industry experts & leaders
- Live class recordings for revision purpose

Programming Fundamentals Training in Agra

Learn2Earn Labs

F-4, First Floor, Anna Ikon Complex, In Front of Deviram Food Circle, Sikandra-Bodla Road, Sikandra, Agra, Uttar Pradesh – 282007

Call : +91-9548868337

Training Modules

Objectives :

The training objective of a programming fundamentals course that includes C, C++, and data structures is to provide students with a strong foundation in computer programming and problem-solving. The course aims to introduce students to the fundamental concepts of programming languages and data structures, and to develop their skills in designing and implementing efficient algorithms.

Course 1 : C Programming

Module 1 : Basic (30 Hours/Days)

Introduction to C Language, Simple C Program, Program execution phases, Backslash character constants, Character set, Constants, Number systems, Format specifiers, Identifiers, Keywords, Variables, Data Types, Arithmetic operators, Increment and decrement operators, Relational operators, Logical operators, bitwise operators, assignment operators, conditional operator, size of operator, comma operator, Type casting operator, data types, input output library functions, control statements (if-else, switch-case, for, while, do-while, nested loop, jump control statements, break, continue, goto, exit, return), functions, return statement, recursion, library functions, local and global variables, storage classes, Pointers, Pointer to Pointer, Pointer Arithmetic, De-reference & increment pointer, pointer to function, Array, types of Arrays, accessing array elements, Pointer to array, Array & functions, Malloc, calloc, realloc, free, core dump, memory leak.

Module 2 : Advance (30 Hours/Days)

Introduction to String, gets() & puts() functions, String handling functions, Structure, accessing structure members, size of structures, reading and displaying structure variables, array to structure, nested structures, self-referential structures, structure and functions, Union, enum keyword, typedef keyword, introduction to file handling, buffer and streams, working with text files and binary files, file operations using std library and system calls, file management I/O functions, random access files.

Course 2 : C++ Programming

Module 1 : Core Programming (30 Hours/Days)

Introduction to C++, First C++ Program, How C++ differs from C, Variables Declaration, Function overloading, Optional Parameters, Reference Variables, Operator overloading, Basics of Console Input and Output, Constant Pointers, Dynamic Memory Allocation, Operators - arithmetic, assignment, logical, bitwise, Conditions like if / else / switch, Arrays / multi-dimensional arrays, Loops - for / while / do-while, Functions, overloading functions, passing variables to functions, Structures, References, Pointers, C++ programs & Practices.

Module 2 : C++ Advance Programming (30 Hours/Days)

Overview of OOPs Principles, Introduction to classes & objects, Creation & destruction of objects, Data Members, Member Functions, this Pointer, Constructor & Destructor, Static class member, Friend class and functions, Namespace, Introduction to inheritance and benefits, Access Specifier, Base and Derived class Constructors, Types of Inheritance, Down casting and up casting, Function overriding, Virtual functions, Destructor overriding, Polymorphism, Pure virtual functions, Virtual Base Class, C++ Class Hierarchy, File Stream, Text File Handling, Binary File Handling, Error handling during file operations, Overloading << and >> operators, Introduction to Exception, Benefits of Exception handling, Try and catch block, Throw statement, Pre-defined exceptions in C++, Writing custom Exception class, Stack Unwinding, C++ Templates, Function Templates, Class Templates.

Course 3 : Data Structure & Algorithms

Module 1 : Introduction

What is an algorithm, Data Structure and Types, Asymptotic Notations, Master Theorem, Divide and Conquer Algorithm.

Module 2 Data Structure

Stack, Queue, Types of Queues, Circular Queue, Priority Queue, Deque, Linked List, Linked List Operations, Types of Linked List, Hash Table

Heap Data Structure, Fibonacci Heap, Decrease Key and Delete node from Fibonacci Heap

Module 3 Tree Based Data Structure

Tree Data Structure, Tree Traversal, Binary Tree, Full Binary Tree, Perfect Binary Tree, Complete Binary Tree, Balanced Binary Tree, Binary Search Tree, AVL Tree

B Tree, Insertion into B-tree, Deletion from B-tree, B+ Tree, Insertion on a B+ Tree, Deletion from a B+ Tree, Red Black Tree, Insertion in Red Black Tree, Deletion from Red Black Tree

Module 4 Graph Based DSA

Graph Data Structure, Spanning Tree, Strongly Connected Components, Adjacency Matrix, Adjacency List, DFS Algorithm, Breadth-first Search, Bellman Ford's Algorithm

Module 5 Sorting & Searching Algorithms

Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, Counting Sort, Radix Sort, Bucket Sort, Heap Sort, Shell Sort, Linear Search, Binary Search

Module 6 Greedy Algorithms

Greedy Algorithm, Ford-Fulkerson Algorithm, Dijkstra's Algorithm, Kruskal's Algorithm, Prim's Algorithm, Huffman Code

Module 7 Dynamic Programming

Dynamic Programming, Floyd Warshall Algorithm, Longest Common Subsequence

Project Work

After the completion of all the courses during this programming fundamentals training program, some projects would be assigned to you to improve your practical knowledge and project experiences.

Below are the tentative details about the projects on which you can work under the guidance of respective faculty member.

Mini Projects (C / C++ programming)

Calculator: Create a simple calculator that can perform basic mathematical operations like addition, subtraction, multiplication, and division.

Tic Tac Toe Game: Create a simple Tic Tac Toe game that allows two players to play against each other.

Contact Management System: Create a simple contact management system that allows users to add, edit, and delete contacts.

Quiz Game: Create a quiz game that asks multiple-choice questions and keeps track of the player's score.

File Compression Tool: Write a program that can compress a file to reduce its size and then decompress it to restore its original content.

Encryption and Decryption Tool: Create a program that can encrypt and decrypt messages using various encryption algorithms like Caesar Cipher, Vigenere Cipher, and RSA.

Bank Management System: Create a program that can manage customer accounts, perform transactions like deposit, withdrawal, and transfer, and generate account statements.

Library Management System: Create a program that can manage books, issue and return books, and generate reports on overdue books and borrowed books.

Image Processing Tool: Write a program that can perform basic image processing tasks like resizing, cropping, filtering, and adding effects.

Mini Projects (Data Structure & Algorithms)

Sorting Algorithms: Implement various sorting algorithms like Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, and Quick Sort and compare their performance.

Binary Search Tree: Implement a Binary Search Tree and perform operations like insertion, deletion, and search.

Hash Table: Implement a Hash Table and handle collisions using techniques like chaining and open addressing.

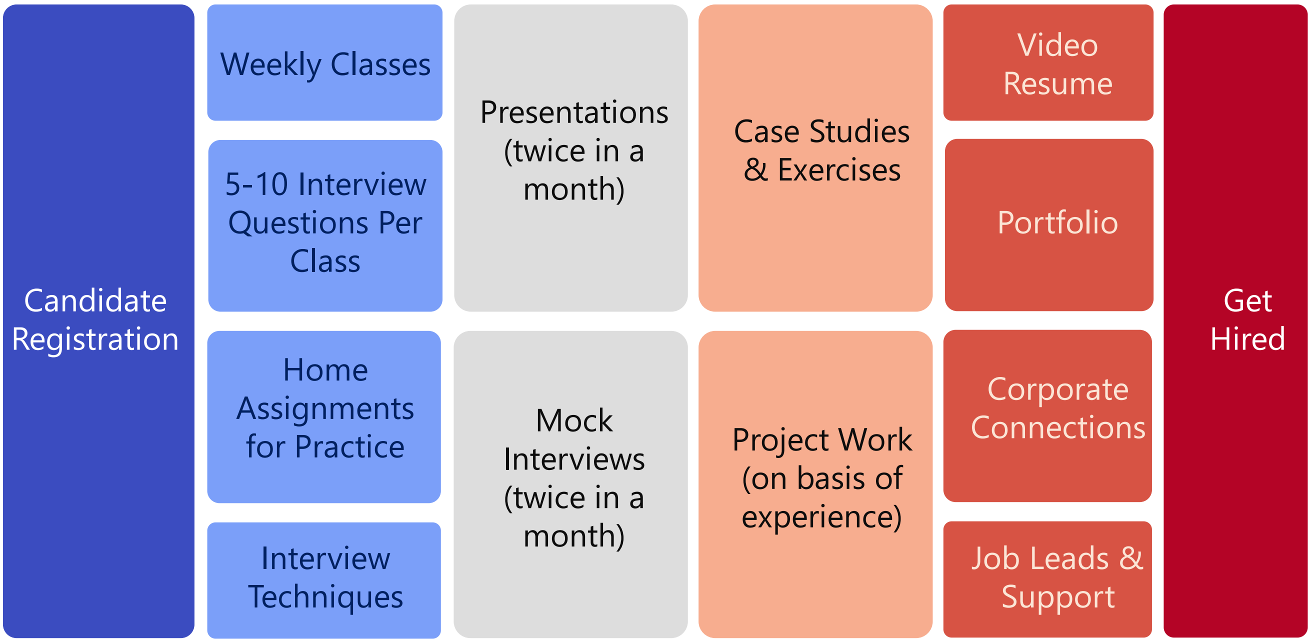
Graph Algorithms: Implement various graph algorithms like Depth-First Search, Breadth-First Search, Dijkstra's Algorithm, and Kruskal's Algorithm.

Text Editor: Implement a basic text editor that allows users to create, edit, and save documents and performs operations like searching, replacing, and formatting.

We follow a dynamic & vibrant
Job Assistance Execution Path

In present scenario providing job surety is easy, only we need to focus on outcome based learning and practical work enhancement. We know that every candidate is enough capable to understand the concepts and implement those concepts for improving his/her practical knowledge and experience.

Hence we are following a dynamic & vibrant Job Assistance Execution Path while conducting our job guarantee training programs and job assistance training programs. We are proud to say that we prepare candidates who can perform better throughout their professional journey and will always remains unbeatable.



Candidate Registration

Weekly Classes

5-10 Interview Questions Per Class

Home Assignments for Practice

Interview Techniques

Presentations (twice in a month)

Mock Interviews (twice in a month)

Case Studies & Exercises

Project Work (on basis of experience)

Video Resume

Portfolio

Corporate Connections

Job Leads & Support

Get Hired

We also offers
Variety of Job Oriented Training Programs

Java Full Stack
Duration : 4 months | 6 Months

Full Stack Web Development
Duration : 4 months | 6 Months

8 LPA+
Job Guarantee

Full Stack Software Engineer
Duration : 1 Year | 600 Hours

Cloud Computing & DevOps
Duration : 4 months | 6 Months

Digital Marketing
Duration : 6 months | 1 Year

Data Science & ML
Duration : 4 months | 6 Months

Learn2Earn Labs
A training unit of
Ninepages Techsolutions Private Limited, Agra
Email : query@learntoearnlabs.com
Contact No : +91-9837-705-705