

*(Building Futures Through Digital Knowledge and Innovation)*

## C/C++ Programming (03 Months)

### Syllabus

#### 01: Introduction to Programming & Basics of C

- ✓ What is programming? Types of programming languages
- ✓ Evolution of C language and its use cases
- ✓ Structure of a C program
- ✓ Compilation and execution process
- ✓ Setting up the environment (GCC, Turbo C++, Visual Studio Code)
- ✓ Basic syntax and coding style
- ✓ Writing and running your first C program

#### 02: Data Types, Variables, and Operators

- ✓ Keywords and identifiers
- ✓ Data types in C (int, char, float, double, void)
- ✓ Constants and variables
- ✓ Type modifiers (signed, unsigned, long, short)
- ✓ Typecasting and data conversion
- ✓ Operators:
  - Arithmetic
  - Relational
  - Logical
  - Bitwise
  - Assignment
  - Increment/Decrement
  - Conditional (Ternary)
- ✓ Operator precedence and associativity

#### 03: Control Structures

- Decision making with:
  - if, if-else, nested if-else
  - switch-case
- Looping constructs:
  - for, while, do-while
- break, continue, goto
- Nested loops and conditional nesting

#### 04: Functions in C

- ✓ Function declaration and definition
- ✓ Function calling and return types
- ✓ Parameter passing methods (call by value, call by reference)
- ✓ Scope of variables: local, global, static
- ✓ Recursion: basics and examples
- ✓ Storage classes in C

#### 05: Arrays and Strings

- ✓ Introduction to arrays
- ✓ One-dimensional arrays
- ✓ Two-dimensional and multidimensional arrays
- ✓ Array manipulation techniques
- ✓ Character arrays and strings
- ✓ String handling functions (strlen, strcpy, strcmp, strcat, etc.)
- ✓ Input/output of strings
- ✓ String pointer manipulation

#### 06: Pointers in C

- ✓ Introduction to pointers and memory addressing
- ✓ Pointer arithmetic
- ✓ Pointers and arrays
- ✓ Pointers and functions
- ✓ Pointers to pointers
- ✓ Void pointers
- ✓ Dynamic memory allocation:
  - malloc, calloc, realloc, free
- ✓ Memory leak and memory management best practices

#### 07: Structures and Unions

- ✓ Defining and using structures
- ✓ Array of structures
- ✓ Nested structures
- ✓ Pointers to structures
- ✓ Passing structures to functions
- ✓ Self-referential structures
- ✓ Unions: definition and use cases
- ✓ Difference between structures and unions
- ✓ Enumerations and typedef

#### 08: File Handling in C

- ✓ File operations: create, read, write, append
- ✓ File I/O functions: fopen, fclose, fread, fwrite, fprintf, fscanf, fgets, fputs
- ✓ File types: text and binary
- ✓ File pointers and buffers
- ✓ Command line arguments

## 09: Introduction to C++

- ✓ Need and features of C++
- ✓ Differences between C and C++
- ✓ Setting up C++ compiler/environment
- ✓ Syntax and structure of a C++ program
- ✓ Namespace and iostream

## 10: Object-Oriented Programming in C++

- ✓ Introduction to OOP concepts: Class, Object, Encapsulation, Inheritance, Polymorphism, Abstraction
- ✓ Defining classes and objects
- ✓ Member functions, access specifiers
- ✓ Constructors and destructors (default, parameterized, copy)
- ✓ this pointer and object memory layout
- ✓ Static members and functions
- ✓ Friend functions and classes

## 11: Inheritance in C++

- ✓ Types of inheritance:
  - Single
  - Multiple
  - Multilevel
  - Hierarchical
  - Hybrid
- ✓ Constructor invocation in inheritance
- ✓ Function overriding
- ✓ Virtual functions and run-time polymorphism
- ✓ Abstract classes and pure virtual functions

## 12: Operator Overloading and Templates

- Operator overloading: arithmetic, relational, I/O operators
- Function overloading
- Templates:
  - Function templates
  - Class templates
  - Generic programming with templates

## 13: Exception Handling in C++

- ✓ Introduction to exceptions
- ✓ try, catch, throw blocks
- ✓ Catching all exceptions
- ✓ Nested try-catch
- ✓ Custom exception classes
- ✓ Best practices for error handling

## 14: Standard Template Library (STL)

- Overview of STL
- Containers:
  - Vectors
  - Lists
  - Queues
  - Stacks
  - Maps
  - Sets
- Iterators and algorithms
- STL function objects
- Lambda expressions in C++

## 15: File Handling in C++

- ✓ C++ stream classes: ifstream, ofstream, fstream
- ✓ Reading and writing to files
- ✓ File modes and manipulation
- ✓ File pointers and random access files
- ✓ Binary file I/O

## 16: Advanced C++ Concepts

- ✓ Pointers to objects
- ✓ Dynamic memory in C++ (new, delete)
- ✓ Deep vs shallow copy
- ✓ Operator new and delete overloading
- ✓ Smart pointers (C++11 onwards)
- ✓ Move semantics and rvalue references (C++11)
- ✓ Lambda expressions and auto keyword

## 17: Data Structures with C/C++ (Overview)

- Linked Lists (Singly, Doubly, Circular)
- Stacks and Queues (Linear and Circular)
- Trees (Binary, BST, AVL)
- Graphs (Adjacency list/matrix, DFS/BFS)
- Hashing techniques
- Sorting and searching algorithms (bubble, insertion, quick, binary search)

### **18: Mini Projects and Capstone Project**

- C/C++ based mini-projects (Banking system, Student management, etc.)
- File-based inventory/project systems
- Console games using arrays/pointers (Tic Tac Toe, Snake, etc.)
- Final capstone project: Full-featured software using OOP, file handling, and data structures
- Git & GitHub version control integration

### **Course Requirements**

- No prior programming knowledge required (for beginners)
- Basic knowledge of operating systems and file handling
- For advanced modules, basic problem-solving and logic-building experience is recommended