

C PROGRAMMING SYLLABUS

Module 1: Introduction to C Programming

- **History and Features of C**
- **Structure of a C Program**
- **Compilation and Execution Process**
- **Basic Syntax and Semantics**
- **Learning Outcomes:** Understand the fundamental structure and compilation process of C.

Module 2: Data Types and Operators

- **Data Types:** int, char, float, double
- **Constants and Variables**
- **Operators:** Arithmetic, Logical, Relational, Bitwise
- **Type Casting and Conversions**
- **Learning Outcomes:** Utilize operators and manage variables/data types.

Module 3: Control Structures

- **Conditional Statements:** if, if-else, nested if
- **Switch-Case Statement**
- **Loops:** for, while, do-while
- **Break and Continue**
- **Learning Outcomes:** Implement decision making and iteration.

Module 4: Functions

- **Function Declaration and Definition**
- **Call by Value and Reference**
- **Recursion**
- **Scope and Lifetime of Variables**
- **Learning Outcomes:** Organize code using reusable functions.

Module 5: Arrays and Strings

- **1D and 2D Arrays**
- **String Handling Functions** (strlen, strcpy, strcmp, etc.)
- **Character Arrays vs Strings**
- **Learning Outcomes:** Manage data collections with arrays and strings.

Module 6: Pointers

- **Pointer Declaration and Initialization**
- **Pointer Arithmetic**
- **Pointers and Functions**
- **Pointers and Arrays**
- **Learning Outcomes:** Use pointers for efficient memory access.

Module 7: Structures and Unions

- **Structure Definition and Usage**
- **Nested Structures**
- **Arrays of Structures**
- **Unions and Differences with Structures**
- **Learning Outcomes:** Bundle multiple data items using structures.

Module 8: File Handling

- **File Operations:** fopen, fclose, fread, fwrite, fprintf, fscanf
- **File Modes and Types**
- **Random Access in Files**
- **Learning Outcomes:** Store and retrieve data using files.

Libraries & Tools:

- **Standard Library:** stdio.h, stdlib.h, string.h
- **IDEs:** Turbo C, Code::Blocks, Dev-C++

Real-World Applications:

- Embedded Systems
 - Operating System Kernels
 - Compilers
 - Network Drivers
-

C++ PROGRAMMING SYLLABUS

Module 1: Introduction to C++

- **C++ vs C**
- **Basic Syntax and Structure**
- **Input/Output Streams**
- **Learning Outcomes:** Understand OOP basics and syntax.

Module 2: Object-Oriented Programming (OOP)

- **Classes and Objects**
- **Encapsulation and Abstraction**
- **Constructors and Destructors**
- **Friend Functions**
- **Learning Outcomes:** Apply OOP principles for modular programming.

Module 3: Inheritance

- **Types of Inheritance: Single, Multiple, Multilevel, Hierarchical**
- **Constructor Overriding**
- **Learning Outcomes:** Reuse code and extend class behavior.

Module 4: Polymorphism

- **Function Overloading**
- **Operator Overloading**
- **Virtual Functions and Runtime Polymorphism**
- **Learning Outcomes:** Implement polymorphic behavior in programs.

Module 5: Templates and Exception Handling

- **Function and Class Templates**
- **Try-Catch Blocks**
- **Standard Exceptions**
- **Learning Outcomes:** Create generic and robust code.

Module 6: File Handling in C++

- **Streams:** ifstream, ofstream, fstream
- **File Modes and Manipulators**
- **Reading/Writing Objects**
- **Learning Outcomes:** Perform operations on files using streams.

Module 7: Standard Template Library (STL)

- **Vectors, Lists, Sets, Maps**
- **Algorithms and Iterators**
- **Learning Outcomes:** Use STL for efficient data management.

Libraries & Tools:

- **Header Files:** <iostream>, <fstream>, <vector>
- **IDEs:** Code::Blocks, Visual Studio, Dev-C++

Real-World Applications:

- **Game Development**
 - **Simulation and Modeling Tools**
 - **GUI-based Applications**
-

JAVA PROGRAMMING SYLLABUS

Module 1: Introduction to Java

- **Java Features & Architecture (JVM, JDK, JRE)**
- **Data Types, Variables, and Operators**
- **Input/Output Using Scanner and BufferedReader**
- **Learning Outcomes:** Set up Java environment and write basic programs.

Module 2: Control Statements & Arrays

- **If-Else, Switch**
- **Loops: for, while, do-while**
- **1D and 2D Arrays**
- **Learning Outcomes:** Implement decision-making and iteration.

Module 3: Object-Oriented Programming in Java

- **Classes, Objects, and Methods**
- **Constructors and Method Overloading**
- **Inheritance and Method Overriding**
- **Abstract Classes and Interfaces**
- **Learning Outcomes:** Apply OOP principles effectively.

Module 4: Exception Handling & Multithreading

- **Try-Catch-Finally**
- **Throws and Throw**
- **Thread Class and Runnable Interface**
- **Learning Outcomes:** Handle errors and run parallel threads.

Module 5: Collections and Generics

- **List, Set, Map Interfaces**
- **ArrayList, HashMap, HashSet**
- **Generic Classes and Methods**
- **Learning Outcomes:** Manage dynamic data collections.

Module 6: File I/O and Streams

- **FileReader, FileWriter, BufferedReader/Writer**
- **Serialization and Deserialization**
- **Learning Outcomes:** Read and write data using Java I/O.

Module 7: GUI and Event Handling (AWT/Swing)

- **Basic Components: Frame, Button, TextField**
- **Event Listeners and Adapters**
- **Learning Outcomes:** Create simple GUI-based applications.

Libraries & Tools:

- Java SE, Eclipse IDE, IntelliJ IDEA
- Java Packages: `java.util`, `java.io`, `javax.swing`

Real-World Applications:

- Android Development
 - Enterprise Software
 - Banking Applications
-

DATA STRUCTURES AND ALGORITHMS (DSA) SYLLABUS

Module 1: Introduction to DSA

- **Role and Importance of DSA**
- **Complexity Analysis: Time and Space**
- **Big O, Omega, and Theta Notation**
- **Learning Outcomes:** Analyze algorithm efficiency.

Module 2: Linear Data Structures

- **Arrays and Operations**
- **Linked Lists: Singly, Doubly, Circular**
- **Stacks and Queues (Implementation using Arrays and Linked Lists)**
- **Learning Outcomes:** Implement and apply linear structures.

Module 3: Non-linear Data Structures

- **Trees: Binary Trees, BST, AVL Trees**
- **Graphs: Representation, BFS, DFS**
- **Heaps and Priority Queues**
- **Learning Outcomes:** Work with hierarchical and networked data.

Module 4: Searching and Sorting Algorithms

- **Linear and Binary Search**
- **Bubble, Insertion, Selection, Merge, Quick Sort**
- **Learning Outcomes:** Efficiently search and sort data.

Module 5: Hashing and Hash Tables

- **Hash Functions**
- **Collision Resolution Techniques**
- **Applications of Hashing**
- **Learning Outcomes:** Implement fast data lookup.

Module 6: Greedy and Dynamic Programming

- **Greedy Algorithms:** Activity Selection, Huffman Coding
- **Dynamic Programming:** Knapsack, LCS
- **Learning Outcomes:** Solve optimization problems efficiently.

Module 7: Backtracking and Divide & Conquer

- **Backtracking:** N-Queens, Sudoku Solver
- **Divide & Conquer:** Merge Sort, Binary Search
- **Learning Outcomes:** Design recursive and strategic algorithms.

Libraries & Tools:

- C/C++ STL, Java Collections
- IDEs: Code::Blocks, IntelliJ IDEA, Eclipse

Real-World Applications:

- Route Planning (Maps)
- Database Indexing
- Scheduling Algorithms
- Search Engines