

Module 1: Introduction to C

- History and features of C language
- Structure of a C program
- Compilers and IDEs (Turbo C, GCC, Code::Blocks)
- Input/output functions (printf, scanf)
- Understanding header files and libraries

Module 2: Basics of C Programming

- Data types, variables, constants
- Operators (arithmetic, relational, logical, bitwise)
- Control structures (if, if-else, switch)
- Loops (for, while, do-while)
- Break and continue statements

Module 3: Functions & Modular Programming

- Defining and calling functions
- Function arguments (call by value, call by reference)
- Recursion basics
- Scope of variables (local, global, static)

Module 4: Arrays & Strings

- One-dimensional arrays
- Multi-dimensional arrays (matrices)
- String handling functions (strlen, strcpy, strcmp, etc.)
- Practical applications (sorting, searching)

Module 5: Pointers

- Introduction to pointers
- Pointer arithmetic
- Pointers and arrays
- Pointers and functions
- Dynamic memory allocation (malloc, calloc, free)

Module 6: Structures & Unions

- Defining and using structures
- Nested structures
- Arrays of structures
- Unions and differences from structures
- Practical applications (student records, employee database)

Module 7: File Handling

- Opening and closing files
- Reading and writing text files
- Binary file operations
- File pointers and error handling
- Practical applications (report generation, data storage)

Module 8: Advanced Topics & Projects

- Command-line arguments
- Preprocessor directives (#define, #include, macros)
- Bitwise operations and applications
- Building small projects (calculator, library management, student database)
- Final project: Complete application using C concepts

Learning Outcomes

By the end of this course, students will:

- Understand C syntax and program structure
- Apply control structures, functions, arrays, and pointers effectively
- Manage data using structures, unions, and files
- Build small to medium-sized applications in C
- Gain confidence for advanced programming (C++, Java, Python)