

Course Aims:

This course is design to provide non-C programmers with the essential skills and knowledge necessary to allow them to become competent in developing basic ANSI-compliant C++ console programs.

Course Outline:

The History of C++

The Structure of the C++ Program

The Microsoft Visual C++ IDE:

- Starting a new project
- Compiling a C++ Program

Variables & Keywords:

- Variable Names; Data Types; Declarations
- Assignment Operator; Constants
- Initialisation of Variables
- cin & cout
- Arithmetic Operators; Expression Evaluation
- Increment & Decrement Operators
- Bitwise Logical Operators
- Compound Assignment Operators
- Type Conversion; Comma Operator
- Order of Precedence

Composite Data Types:

- typedef
- Enumeration types
- Theory of Arrays; Range Checking; Initialisation
- Vectors
- Strings
- Structures

Flow Control of a Program

- Types of Flow Control Statements
- Logic States; Relational Operators
- Boolean Logical Operators; if-else Statement
- Nested if Statements
- Conditional Operator; Compound Statements
- Switch Statement; While loop; Do-while Loop;
- For Loop; For Loop Variations
- Break & Continue Statements

Functions

- Header & source files
- The Definition of a Function; Function Parameters
- Non-Returning Functions; Void Functions

- The Return Statement
- Functions Returning Non-integers
- Passing Arrays to Functions
- Inline functions

Scoping

- Storage Class Types; Automatic Variables
- Register, Static & External Variables
- Initialisation Rules for the Storage Classes

Pointers

- The & and * Operators
- Pointers & Function Arguments
- Pointer Arithmetic; Pointers & Arrays
- Pointer Initialisation; Arrays of Pointers
- Pointer to Pointers; Command Line Arguments
- Pointers to Functions
- Pointers to Structures

Dynamic Memory

- Memory Leakage
- new() & delete()

The C++ Preprocessor

- The Pre-processor Operation
- #include; Defined Constants
- #ifdef & #ifndef; #pragma

Classes & Objects Overview

- Class Implementation
- Constructors & Destructors
- Using objects
- Inheritance
- Polymorphism
- Virtual functions
- Abstract classes

Target Audience:

Systems and applications programmers who will be developing systems in C++. Anyone who wants a practical understanding of C++ will benefit from this course. It is suitable for hardware and software engineers who want to expand their knowledge in a powerful all-purpose language, technical managers who want to manage C++ programming projects.

Assumed Knowledge:

Participants should have a good knowledge of a programming techniques and at least one programming language such as COBOL, BASIC, FORTRAN or the like.