C Programming	*5 Days*

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 ANSI-compliant C programs.

Course Outline:

The History of C Programming
The Structure of the C Program:
- Compiling a C Program

Variables & Constants:
- Variable Names; Data Types; Declarations
- Assignment Operator; Constants
- Initialisation of Variables; Enumeration Types
- printf( ); getchar( )

Expressions & Operators
- Arithmetic Operators; Expression Evaluation
- Increment & Decrement Operators
- Bitwise Logical Operators
- Compound Assignment Operators
- Type Conversion; Comma Operator
- Order of Precedence

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
- exit( ) Statement; goto Statement

Arrays
- Theory of Arrays; Range Checking; Initialisation
- Unsized Array Initialisation

The Standard I/O Library
- getchar( ) & putchar( ); gets( ) & puts( )
- printf( ); scanf( )

Functions
- The Definition of a Function; Function Parameters
- Non-Returning Functions; Void Functions
- The Return Statement
- Functions Returning Non-integers
- Passing Arrays to Functions; Recursive 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
- Arrays as Function Arguments
- Pointer Initialisation; Arrays of Pointers
- Pointer to Pointers; Command Line Arguments
- Pointers to Functions

File Handling
- The File Pointer; fopen( )
- stdin, stdout & stderr File Pointers; fclose( )
- fgetc( ); fputc( ); ungetc( ); fgets( ); fputs( )
- fprintf( ); fscanf( ); fread( ) & fwrite( )
- fseek( ); ftell( ); rewind( ); fflush( ); File Status

Structures & Unions
- Members of a Structure Variable
- Pointers to Structures
- Structure Initialisation & Assignment
- Structures & Functions; Nested Structures
- Bit Fields; Unions; sizeof( ); typedef;

Dynamic Memory
- Memory Leakage
- malloc() & free()

The C Preprocessor
- The Pre-processor Operation
- #include; Defined Constants
- Defined Macros; Macro Side Effects
- Deleting a Definition; Conditional Compilation
- #ifdef & #ifndef; #line

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. This course is a prerequisite for the C++ programming course.