C Programming

Course 6064 – 32 Hours

Overview

C is one of the most widely-used languages for systems software and workstation application programming, largely due to its power and flexibility. This course will provide you a highly effective, structured approach to learning the C language.

Programming skills will be enhanced as delegates will be able to use the powerful features of C to best effect and avoid errors that lead to faulty code or code that cannot be maintained. QA's programming course is outstanding because of its emphasis on writing style, pitfalls to avoid and techniques to use that make the code clear, concise and maintainable.

In addition to the lecture material there are graded practical sessions that cover each of the major areas of C. There are also optional exercises for further study after the course. Delegates may take away worked solutions, together with some small sample demonstration programmers.

On Completion, Delegates will be able to

- Use the major elements of the C language
- Write programs using the strengths of the C language. For example, pointers
- Write and use the data structuring features of the language, which can result in better program design
- Work with the C run-time library: a major source of programmer productivity
- Spot and remedy common programming errors in C
- Write in a good C programming style

Who Should Attend

- Experienced programmers wishing to learn the C language.

Prerequisites

- Delegates must have professional programming skills and a good working knowledge of a block-structured language, such as Basic/Visual Basic, Pascal/Modula2, Fortran, Algol or PL/1, and be familiar with a programming environment.

Course Contents

An Overview of C

- History and evolution of C
- Key characteristics of C

Writing a Simple Program

- Program structure
- Data and code statements
- C software development life cycle

Data Types

- Scalar types
- Variables and constants
- Storage considerations
- Initialising variables

Operators and Expressions

- Standard arithmetic operators
- Increment, decrement, assignment and relational operators
- Automatic and programmer-controlled type conversion

Program Looping

- Boolean expressions
- While, Do and For loops
- Looping style considerations

Decision Making

- If, Else and Switch statements
- Other statements affecting flow of control
- Decision-making style considerations

Functions and Program Structure

- Inter-function communication
- Function prototypes, calls and definitions
- Scope and storage classes

Structured Data Types

- Arrays, structures and unions
- Nested data structures

Pointers

- The concept of indirection
- Pointers and address arithmetic
- Pointers and functions

Pointers and Data Structures

- Pointers and arrays
- Pointers and structures
- Complex data structures

Preprocessor

- Tokens and macros
- Include files
- Conditional compilation

Input and Output

- Using run-time routines
- Character and formatted I/O
- File I/O

Further Data Types

- Bit manipulation
- User-defined types

Working with Larger Programs

- C and modular programming
- C's standard library considerations