'C' PROGRAMMING

This hands-on course provides programmers with an introduction to the ANSI 'C' programming language. The course covers the syntax of the language together with practical application and is designed to run on a Windows, UNIX or Linux platform. Good programming practice is encouraged throughout the course.

Audience

The course is intended for those delegates with little programming experience or programmers experienced in other languages who need to be able to understand and write 'C' application programs.

Prerequisites

Delegates should, ideally, be familiar with using another procedural programming language such as Basic, Pascal or Cobol, or have an aptitude for programming. The course is not suitable for delegates who have not programmed before, have no concept of program design or are from a non-technical background.

Duration

Five days. The course uses a combination of lectures and hands-on practical exercises to enable the delegate to gain experience in developing 'C' programs.

Course objectives

On completion of this course, the delegate will be able to:

- use basic 'C' language expressions and statements
- code, compile, test and debug 'C' programs
- use program flow-control constructs
- write and use 'C' functions
- use some of the 'C' libraries and standard header files
- use storage classes
- understand and use arrays, pointers and structures
- write file handling code

Course contents

Introduction

History; Overview of the 'C' language; Conventions.

Structure of a 'C' program

Program structure; Layout; Introducing the printf() function; Statements; Comments; Naming conventions; Keywords; Compiling a 'C' program.

Data types and operators

Data types; Declaring and initialising variables; Storage Classes; Symbolic constants; Simple I/O; Operators; Expressions.

Flow control

Keywords; Decision-making; program flow control; break; continue; goto; switch.

Functions

Function prototypes and parameters; *call by value*; Return values; Passing arguments to functions; Functions in separate source files; Library functions.

The 'C' preprocessor

Preprocessor keyword statements; #define; #undef; Conditional compilation; #if; #endif; #include; Macros.

Arrays

Declaring; Initialising; Single-dimension arrays; Multi-dimension arrays; Character strings; Passing arrays to functions.

'C' PROGRAMMING

Course contents continued

Pointers

Declaring; Initialising; Pointer operations; Pointer arithmetic; Pointers and arrays; Passing pointers to functions; Returning pointers from functions; Passing arguments to main().

Structures

Structure declaration; Templates; Initialising; Array of structures; Member access.

File Handling

Standard Input and Output; The FILE pointer; Opening and closing files; Handling data streams; Library functions; File positioning functions; Formatting output.