

Complete C Programming

A 5 day **Hands on** training course



Description

A hands on introduction to programming in the ANSI C programming language. The course initially moves at a fast pace in order to spend as much time as possible on the subject of pointers - the area which cause the most bugs in C programs.



Key outcomes

By the end of the course delegates will be able to:

- ✓ Write ANSI C programs.
- ✓ Use the C libraries.
- ✓ Debug C programs.
- ✓ Examine existing code and determine its function.



Training Approach

This structured course uses Instructor Led Training to provide the best possible learning experience. Small class sizes ensure students benefit from our engaging and interactive style of teaching with delegates encouraged to ask questions throughout the course. Quizzes follow each major section allowing checking of learning. Hands on sessions are used throughout to allow delegates to consolidate their new skills.



Details

Who will benefit?

Programmers wishing to learn C.
Programmers wishing to learn C++ or Java.

Prerequisites

None, although experience in another high-level language would be useful.

Duration: 5 days

Overall rating:



Generic Training



Generic training compliments product specific courses covering the complete picture of all relevant devices including the protocols "on the wire".

"Friendly environment with expert teaching that teaches the why before the how."
G.C. Fasthosts

Small Class Sizes



We limit our maximum class size to 8 delegates; often we have less than this. This ensures optimal interactivity between delegates and instructor.

"Excellent course. The small class size was a great benefit..."
M.B. IBM

Hands On Training



The majority of our courses use hands on sessions to reinforce the theory.

"Not many courses have practice added to it. Normally just the theoretical stuff is covered."
J.W. Vodafone

Our Courseware



We write our own courses; courseware does not just consist of slides and our slides are diagrams not bullet point text.

"Comprehensive materials that made the course easy to follow and will be used as a reference point."
V.B. Rockwell Collins

Customise Your Course



Please contact us if you would like a course to be customised to meet your specific requirements. Have the course your way.

"I was very impressed by the combination of practical and theory. Very informative. Friendly approachable environment, lots of hands on."
S.R. Qinetiq

Complete C Programming

Course Content

Getting started

The compilation process, comments, main(), statement blocks, printf().

C data types and operators

char, int, float and double, qualifiers, arithmetic and assignment operators, precedence, associativity.

Basic I/O

C libraries, stdin and stdout, getchar(), putchar(), printf() formatting.

Flow control

if else, dangling elses, else if, while and for loops. Switch statements, the null statement, break, continue and gotos.

Functions

Function calls, arguments and return types, function declarations (prototypes), function definitions, scope of variables.

The preprocessor

Preprocessor actions, macros, #include. Libraries and their relationship with header files. Conditional compilation.

More data types and operators

Logical, bitwise and other operators, type conversion, casting, typedefs and access modifiers.

Arrays

Declaring and handling arrays, common gotchas, multidimensional arrays.

Pointers

What are pointers? Why they are so important, declaring and using pointers. The three uses of the *, pointer example - scanf, pointers as arguments.

More pointer

Golden rules of pointers and arrays, pointers to arrays, pointer arithmetic, arrays of pointers, multiple indirection.

Character/string manipulation

Arrays of characters, string definition, working with strings, String library.

Program arguments

argc and argv, example uses, char *argv[] versus char ** argv.

Program structure and storage classes

Globals (externals), multi source programs, the look of a C program.

Structures

Declaration, the . and -> operators, unions and bitfields.

Library functions

File handling, fopen and fclose, reading from and writing to files, fseek().calloc() and malloc().

