

C Programming Fundamentals

Duration: 3 Days

Description

The C programming language can be used to develop both system and application software on any platform (Linux/Unix, Windows, Mac, embedded systems and mainframe). The popularity of C is explained by its flexibility and efficiency, its simple way to address low-level functions, and the fact that programs can easily be transferred between platforms. Also, lots of highly efficient and often even open-source software libraries have been written in C and can thus be easily called from other C programs.

This course will cover the syntax and structure of the ISO/ANSI definition of the C language (more specifically C99 and C11). The course also covers the C preprocessor, debugging tools, and build tools like configure and make.

On successful completion of this course you will be able to:

- understand and use the standard C libraries for input/output, string manipulation and memory management.
- describe the basics of ANSI C, its data types and pointers
- write C applications

Audience

Application programmers wanting to write or maintain C programs.

Prerequisites

Knowledge of programming structures and algorithmic concepts is required. Knowledge of another programming language would also be an advantage

Outline

Basic syntax • program structure • elementary data types • operators • expressions • strings.

Comparison operators and flow control (while/for/do/if/switch).

Arrays • pointers • structures • union • bit fields • enum • bitwise operators.

Using pointers to modify arguments • strings and common string functions • arrays of pointers
Functions and prototyping • identifier scope and storage class • header files • recursivity.
The C preprocessor • working with macros • conditional compilation.
Getting and displaying text • working with files.
Using standard input and standard output.
Linked lists • function pointers • "hook" functions.
Dynamic memory allocation.

Prerequisites

Knowledge of programming structures and algorithmic concepts is required.