

Seminar Abstract

Programming in C

© 1987–1996, 2005, 2009, 2018 Rex Jaeschke. All rights reserved.

Course Overview:

This course teaches students how to program in Standard C. The time is spent two thirds on lectures and one third on writing and debugging lab problems. To reinforce the theory, lab sessions are run immediately after the lectures to which they apply. The course is not hardware or operating system-specific.

Course Length: 30 hours (typically over 5 days)

Goals:

Provided students meet the prerequisites, at the end of the course they should have a good understanding of the following:

- Have a working knowledge of the language's statements and operators.
- Have a working knowledge of the purpose and syntax of the preprocessor.
- Be conversant about the standard runtime library and its headers.
- Have been exposed to issues regarding the importance of programming style.
- Be aware of the strengths and weaknesses of C and what it will take to master and exploit it.
- Be able to “hit the ground running” with regard to working on existing code and in writing new code.

Who Should Attend:

Programmers and technical managers who are seriously interested in, or are about to begin, programming in the C language; or who wish to evaluate the suitability of C for projects and/or programming personnel.

Prerequisites:

A strong working background in at least one language, such as any assembly language, Ada, BASIC, Pascal, Fortran, PHP, Java™, JavaScript™, or C#. If COBOL is the only language a student knows, the transition will likely require extra work since COBOL is a different kind of language from the others listed. However, the transition can be made.

If a student's only or primary language is a database query language, a simple scripting language, or HTML, they will almost certainly have considerable difficulty keeping up and are advised to seek a language more conducive to learning a first programming language, such as Visual BASIC™, C#, or Java.

Attendees are expected to be conversant with the following:

- Communicating between procedures by passing arguments and/or by returning a value.
- The terms *bit* and *byte*.
- Binary and either octal or hexadecimal number systems and arithmetic.
- Looping and testing language constructs.
- Single- and multi-dimensional arrays.
- Global variables.
- Creating and using sequential files.
- Formatted I/O.
- Using a text editor.
- The purpose of a compiler and linker/link editor.

Students interested in C from a portability perspective should have some knowledge of different hardware platforms and operating environments.

Materials:

Programming in C – This manuscript was written specifically for teaching. It serves as a useful reference once the course has been completed.

Detailed Topics:

The main topics covered are:

- Identifiers and constants
- Types and type conversion
- Looping, branching, and testing
- Storage classes
- Operators and expressions
- Functions, argument passing, and value returning
- Single and multi-dimensional arrays
- String manipulation
- The preprocessor directives
- Data pointers
- Dynamic memory allocation
- Command-line processing
- Basic I/O
- Structures, unions, and bit fields
- Implementation of linked lists