

Programming in C++ (for Non-C Programmers)

© 1998, 2001, 2005, 2009 Rex Jaeschke. All rights reserved.

COURSE OVERVIEW:

This intensive course teaches students how to program in Standard C++. A significant lab component is included. The course is not hardware or operating system-specific.

COURSE LENGTH:

The course is 10 days, typically run in two sessions with a break in-between.

GOALS:

Provided students meet the prerequisites, at the end of the course, they should:

- Have a working knowledge of the language's statements and operators.
- Be conversant about the standard runtime library.
- 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 design and implement simple classes.
- Have a basic understanding of encapsulation, inheritance, and polymorphism.
- Understand templates.
- Understand exception handling.

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 of the following languages is expected: any assembler, any systems language (such as PL/I, PL/M, or Bliss), Ada, Algol, compiled BASIC, DIBOL or DBL, FORTRAN, or Pascal. 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 any of the following: Interpreted BASIC, database query language, 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.

MATERIALS:

- *Programming in C++* — This textbook was written specifically for teaching Standard C++. It contains all of the main features added during the standardization of the language. This book serves as a useful reference once the course has been completed.

DETAILED TOPICS:

The main topics covered are:

1. Procedural Programming in C++:
 - The Basics
 - Looping, Testing, and Branching
 - Arrays
 - Functions
 - Storage Classes
 - Pointers and Addresses
 - Dynamic Memory Allocation
 - Structures, Bit-Fields, and Unions
2. Using the Standard Class Libraries:
 - String handling
 - I/O
3. Object-Oriented Programming in C++:
 - Introduction to Classes
 - Constructors and Destructors
 - Operator Overloading

- Inheritance
- Exception
- Handling
- Function and class templates