

Programming in Java™

© 1996–1999, 2005, 2009, 2019 Rex Jaeschke. All rights reserved.

Course Overview:

This course covers the Java language and its use in producing stand-alone programs called *applications*. This course or its equivalent is a prerequisite for the companion course *Advanced Programming in Java*. Lab time is included. The course is not hardware or operating system-specific.

This is not an introduction to programming course! Rather, it's an introduction to programming in Java for existing programmers.

Course Length: 18, 24, OR 30 hours (typically over 3, 4, or 5 days)

The length varies depending on the programming background of the participants, as follows:

# Days	Language Background
3	Fluent in C++
4	Fluent in some other Object-Oriented Programming language
4	Fluent in C
5	Fluent in a procedural language

Goals:

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

- The goals of Java
- The core data types, operators, and statements
- The basic principles of object-oriented programming, such as data hiding, encapsulation, inheritance, and polymorphism.
- Error handling via exceptions

- What's available in the Java Class Library
- For C/C++ programmers: How Java is/is not like C and C++; The advantages of Java over C and C++; How to survive and thrive without an overt pointer type.

Who Should Attend:

Programmers and technical managers who are seriously interested in evaluating, and possibly using, Java for any purpose. Also, engineers and scientists currently using a procedural language, who want to reap many of the benefits of C/C++ without paying the significantly high price that programming in those languages extracts.

Prerequisites:

Knowledge of at least one high-level language is assumed.

- Those claiming fluency in C++ are expected to have a good working knowledge of the following topics: all the statements and operators, pointers and references, dynamic memory usage via `new` and `delete`, function inlining, function overloading, basic class design, operator overloading, single inheritance, virtual functions, and exception handling.
- Those claiming fluency in some other OO-language are expected to have a good working knowledge of the following topics: basic class design, single inheritance, and exception handling.
- Those claiming fluency in C are expected to have a good working knowledge of the following topics: all the statements and operators, pointers, dynamic memory usage via `malloc` and `free`, argument passing and return value handling, arrays, string handling, declaring and defining functions, and all aspects of structures.
- Those claiming fluency in some procedural language are expected to have a good working knowledge of the following topics: variables, arrays, looping, operator precedence, type conversion, string processing, I/O, passing arguments to, and returning values from, a procedure, number system theory, bit manipulation, data representation, and group items (called records or structures in some languages).

Someone with a working knowledge of C# should be able to learn Java on their own quite quickly.

Materials:

- *Programming in Java*— 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:

- Basic Language Elements
- Looping and Testing
- Methods
- References, Strings, and Arrays
- Classes
- Inheritance
- Exception handling
- Input and Output
- Packages

- Interfaces

Detailed Topics Covered in the Advanced Java Seminar:

- Threads
- Serialization
- Sockets
- Cloning Objects
- Documentation Comments
- Java Archives