



Capstone Courseware, LLC

33 Boylston Street  
Jamaica Plain, MA 02130

877-227-2477  
capstonecourseware.com

## 102. Introduction to Java Programming

### Version 6.0

Java is a popular and powerful language. Although comparatively simple in its language structure, there are a number of subtleties that can trip up less experienced programmers. It is based on C, and the terse nature of C may be unfamiliar to students with some previous experience programming in languages such as COBOL or Visual Basic who do not have experience with C or C++. And, of course, Java is object-oriented.

The five-day timeline of this course explicitly targets less experienced programmers, providing them with a thorough step-by-step introduction to Java programming. It lays a firm foundation for further study of Java. There are a large number of example programs and many labs. The course software also includes an optional overlay of workspace and project files to support use of the Eclipse IDE in the classroom. (This requires that the instructor be experienced in use of Eclipse and able to walk students through basic tasks in the IDE.)

An important thrust of this course is to teach programming from an object-oriented perspective. It is often difficult for programmers trained originally in a procedural language to start "thinking in objects." This course introduces object-oriented concepts very early, and Java is developed in a way that leverages its object orientation. Most of the course emphasizes simple classes without inheritance. The last chapter introduces inheritance and polymorphism, along with interfaces and collections.

This revision of the course targets the 6.0 version of the Java language and Core API; but it is equally applicable to Java 5 and groups looking for Java training who know they'll be using Java 5 are encouraged to use this course. For training within the Java 1.4 environment, please see version 1.4.1 of this course; to read more about different versions of Java and for help deciding on which version of this course to use, see "Java Versions and Terminology Demystified".)

### Prerequisites

- Good general problem solving skills. Some previous experience programming in a procedural language is essential.



## Learning Objectives

- Learn the basic principles of object-oriented programming
- Learn the essentials of the Java programming language
- Acquire the skills needed to design, code and debug computer programs in the Java language

**Timeline: 5 days.**

## IDE Support: Eclipse Europa

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse Europa. Students can code and build all exercises from within the IDE. Most exercises can be tested from within the IDE as well, though some must be tested from the command line. See also our orientation to Using Capstone's Eclipse Overlays, and please be advised that this is an optional feature; it is not a separate version of the course, and the course itself does not contain explicit Eclipse-specific lab instructions.





## **Chapter 1. What is Java?**

- Object Orientation
- Java Language
- Java Virtual Machine
- Java Libraries
- World Wide Web and Java
- Java Platforms
- Java as a First Programming Language

## **Chapter 2. First Java Programs**

- Hello, World
- Program Structure
- Output in Java
- Variables and Expressions
- Calculations Using Java

## **Chapter 3. Introduction to Objects**

- Object Models
- Classes and Objects
- State and Behavior
- Input in Java
- InputWrapper Class
- Packages

## **Chapter 4. Data Types and Operators**

- Strong Typing
- Integer Data Types
- Floating Point
- Conversions Between Types
- Arithmetic Operators
- Doing Math in Java
- Precedence
- Errors in Integer Arithmetic

## **Chapter 5. Booleans and Enumerations**

- Boolean Variables
- Logical and Relational Operators
- If Tests





- Compound Statements
- Switch Statement
- Enumerated Types

### **Chapter 6. Loops and Program Flow**

- While Loops
- Quitting Infinite Loops
- Curly Braces and Indenting
- Sentinels and Counters
- For Loops
- Loops and If Tests Together
- Nested If Statements

### **Chapter 7. Objects and Classes**

- Structured Data
- Classes
- References
- Instantiating an Object
- Assignment of Objects
- Initialization
- Garbage Collection
- Methods
- Public vs. Private
- Encapsulation
- Constructors
- Formatted Output

### **Chapter 8. Characters and Strings**

- Char Data Type
- Character Codes
- ASCII and Unicode
- String Class
- String Input and Output
- String Methods

### **Chapter 9. Modular Programming**

- Monolithic Programs
- Static Variables and Methods
- Functional Modularity





Object Modularity  
Top-Down and Bottom-Up Development  
Pass-By-Value and Pass-By-Reference  
Nested Classes

### **Chapter 10. Arrays**

One Dimensional Arrays  
Subscripts  
Initializing Arrays  
Arrays and the For-Each Loop  
Copy and Assignment  
Arrays of Objects  
Multidimensional Arrays  
Searching

### **Chapter 11. Bit Operations**

Bitwise Operators  
Truth Tables  
Hexadecimal Notation  
Shifting and Masking

### **Chapter 12. Exception Handling and More Flow Control**

Exceptions  
Errors in Integer Arithmetic  
Floating Point Operations  
I/O Exceptions vs. Runtime Exceptions  
Named Exceptions  
Exception Methods  
Break  
Continue  
Do

### **Chapter 13. Advanced Java Features**

Reusable Software Components  
Abstraction  
Inheritance  
Inheritance Hierarchies  
Polymorphism  
Abstract Classes





Interfaces  
Collections  
Iterators  
Auto-Boxing

## Appendix A. Learning Resources

### System Requirements

**Hardware Requirements (Minimum)**

500 MHz, 128 meg RAM, 500 meg disk space.

**Hardware Requirements (Recommended)**

1.5 GHz, 512 meg RAM, 1 gig disk space.

**Operating System**

Tested on Windows XP Professional. Course software should be viable on all systems which support a Java 6 Developer's Kit.

**Network and Security**

Limited privileges required -- please see our standard security requirements at <http://capcourse.com/Guides/Security.html>.

**Software Requirements**

All free downloadable tools.

