



Capstone Courseware, LLC

33 Boylston Street
Jamaica Plain, MA 02130

877-227-2477
capstonecourseware.com

103. Java Programming

Version 1.4.3

This course teaches programming in the Java language -- the Java 2 Standard or J2SE platform. It is intended for programmers with experience in languages other than Java, but who may or may not have any previous Java experience. It focuses on procedural-coding skills first, and then offers meticulous, in-depth coverage of object-oriented concepts and how to apply them to Java software design and development. The latter part of the course moves from these basic skills into key parts of the J2SE Core API, including collections, exception handling, and object serialization. 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.)

This revision of the course focuses on the 1.4.2 SDK and language, but highlights missing features and areas that are improved in the 5.0 JDK and language. It includes two code examples which will of course not build in the 1.4.2 environment, but work in Java-5 or Java-6 environments and offer examples of emerging Java-5 coding practices. (For training entirely within the latest Java environment, see the most recent version 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".)

Students come to Java from a wide range of backgrounds, and this course is designed to be as flexible as possible over the upper end of that range. Specifically:

- Experienced C and C++ programmers will find this course a very good fit and if anything will find that they complete it in a little less than the full five-day timeline.
- Those with experience in languages less like Java, such as Visual Basic, ASP and other Web-scripting languages, and other pseudo-object-oriented languages may need more time in the early going, and this course covers its introductory topics in good depth and offers many optional and "challenge" labs to support this.
- Less experienced programmers or those coming from non-structured languages -- such as COBOL, PL/1, or 4GL tools -- will probably not cover the whole course in a week, and may want to pursue an abbreviated version at a slower pace. This too is quite feasible, but this audience may also want



to consider our Course 102, "Introduction to Java Programming," for a more relaxed pace through the early material.

Prerequisites

- No prior Java experience is required, but students must be experienced programmers in another third-generation (high-level) language. See the overview for suggestions about pace and scope for different backgrounds.





Learning Objectives

- Chiefly, learn to program effectively in the Java language.
- Understand the Java software architecture, and the design decisions which make Java software portable, efficient, secure and robust.
- Learn how to configure a simple Java development environment.
- Know the grammar, data types and flow control constructs of the Java language for simple procedural programming.
- Understand Java as a purely object-oriented language, and implement software as systems of classes.
- Implement and use inheritance and polymorphism, including interfaces and abstract classes.
- Design appropriate exception handling into Java methods.
- Understand the structure of streams in Java, and learn how to use streams to manage file I/O.
- Learn how to use Java Serialization to internalize and externalize potentially complex graphs of objects.

Timeline: 5 days.





Chapter 1. The Java Environment

- Overview of Architecture
- Forms for Java Software
- Three Platforms
- Java Virtual Machine
- The Core API
- Java Runtime Environment
- Java SDK
- Java Class Path
- Portability and Efficiency

Chapter 2. Language Fundamentals

- Source File Format
- Application Classes
- Code Grammar and Expressions
- Identifiers
- Literals
- Operators
- Expressions
- Calling Methods

Chapter 3. Data Types

- Primitive Types
- Type Conversion
- Numeric Types
- Characters and Booleans
- Java 1.5: Enumerations
- Object References
- Comparing and Assigning References
- Strings
- Arrays

Chapter 4. Flow Control

- The main Method
- Calling and Returning from Methods
- Conditional Constructs
- Looping Constructs
- Java 1.5: the For-Each Loop
- Processing Arrays





Recursion

Chapter 5. Object-Oriented Software

- Complex Systems
- Abstraction
- Classes and Objects
- Responsibilities and Collaborators
- UML
- Relationships
- Visibility

Chapter 6. Classes and Objects

- Java Classes
- Constructors and Garbage Collection
- Naming Conventions and JavaBeans
- Packages and Imports
- Relationships Between Classes
- Using this
- Visibility
- Overloading Methods
- JARs

Chapter 7. Inheritance and Polymorphism in Java

- Extending Classes
- Using Derived Classes
- Type Identification
- Compile-Time and Run-Time Type
- Polymorphism
- Overriding Methods
- Superclass Reference

Chapter 8. Using Classes Effectively

- Class Loading
- Static Members
- Statics and Non-Statics
- Static Initializers
- Prohibiting Inheritance
- Costs of Object Creation
- Strings and StringBuffer





Controlling Object Creation

Chapter 9. Interfaces and Abstract Classes

- Separating Interface and Implementation
- UML Interfaces and Realization
- Defining Interfaces
- Implementing and Extending Interfaces
- Abstract Classes

Chapter 10. Collections

- Dynamic Collections
- Collections vs. Arrays
- The Collections API
- Abstraction: The Collection Interface
- Vector, LinkedList, ArrayList
- Reading Elements and Downcasting
- Collecting Primitive Values
- Algorithmic Programming
- Iterators
- Maps
- Sorted Collections
- Java 1.5: Generics
- Java 1.5: Auto-Boxing
- Java 1.5: Type-Safe Collections
- Java 1.5: Variable Argument Lists
- Java 1.5: Formatted Output

Chapter 11. Exception Handling

- Reporting and Trapping Errors
- Exception Handling
- Throwing Exceptions
- Declaring Exceptions per Method
- Catching Exceptions
- The finally Block
- Catch-and-Release
- Chaining Exceptions

Chapter 12. Inner Classes

- Passing Behavior





- Named Inner Classes
- Outer Object Reference
- Static Inner Classes
- Anonymous Inner Classes

Chapter 13. The Java Streams Model

- Delegation-Based Stream Model
- InputStream and OutputStream
- Media-Based Streams
- Filtering Streams
- Readers and Writers

Chapter 14. Working with Files

- File Class
- Modeling Files and Directories
- File Streams
- Random-Access Files

Chapter 15. Advanced Stream Techniques

- Buffering
- Data Streams
- Push-Back Parsing
- Byte-Array Streams and String Readers and Writers

Chapter 16. Java Serialization

- The Challenge of Object Serialization
- Serialization API
- Serializable Interface
- ObjectInputStream and ObjectOutputStream
- The Serialization Engine
- Transient Fields
- readObject and writeObject
- Externalizable Interface

Appendix A. Learning Resources

System Requirements

Hardware Requirements (Minimum)

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





103. Java Programming

Outline

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 J2SE 1.4 SDK.

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.

