



Capstone Courseware, LLC

33 Boylston Street
Jamaica Plain, MA 02130

877-227-2477
capstonecourseware.com

104. Intermediate 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 students with previous Java experience or training, who already know the fundamentals of the Java architecture and basic procedural programming. This course provides 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 key parts of the J2SE Core API, including collections, exception-handling, logging, streams, 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 5.0 and offer examples of emerging Java-5.0 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".)

Prerequisites

- Students must be able to write, compile, test, and debug simple Java programs, using structured programming techniques, strong data types, and flow-control constructs such as conditionals and loops. Course 102 is ideal preparation for this course.



Learning Objectives

- Chiefly, learn to program effectively in the Java language.
- 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.
- Use the standard logging API to write diagnostic information at runtime.
- 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. Review of Java Fundamentals

- The Java Architecture
- Forms for Java Software
- Three Platforms
- The Java Language
- Numeric Types
- Characters and Booleans
- Java 5.0: Enumerations
- Object References
- Strings and Arrays
- Conditional Constructs
- Looping Constructs
- Java 5.0: the For-Each Loop

Chapter 2. Object-Oriented Software

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

Chapter 3. 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 4. Inheritance and Polymorphism in Java

- Extending Classes
- Using Derived Classes
- Type Identification
- Compile-Time and Run-Time Type





- Polymorphism
- Overriding Methods
- Superclass Reference

Chapter 5. 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 6. Interfaces and Abstract Classes

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

Chapter 7. 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 8. 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 9. Inner Classes

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

Chapter 10. The Java Streams Model

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

Chapter 11. Working with Files

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

Chapter 12. Advanced Stream Techniques

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

Chapter 13. 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.

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.

