

## **Java Programming Training for Experienced Programmers**

(5 Days)

This Java training course is intended for students with experience in a procedural or object-oriented language. It covers most Java syntax elements, including recent additions such as Generics and AutoBoxing. In addition to the fundamental library classes that would be needed in any Java program, it also covers several other important APIs, including Collections and Streams.

The class introduces the concept of objects as one of the first topics, in order that the later concepts are covered with an OO perspective. The material is based on Java 10.

If you are new to object-oriented programming, you should take our Introduction to Java for New Programmers class instead.

### **What You'll Learn**

- Learn how Java works.
- Understand the "write once, run anywhere" concept.
- Understand and learn how to create basic Java objects.
- Learn how to implement flow-control concepts in Java.
- Understand Java's package concept and create packages of Java classes.
- Understand Java arrays and write code to create and use arrays.
- Learn how to write Java classes using inheritance.
- Learn how to create and use interfaces.
- Understand and use the concept of polymorphism in an application
- Understand how Java's exception-handling mechanism works and learn how to apply exception-handling to Java applications.
- Understand and use the Java Logging API.
- Understand and use inner classes.
- Understand Generics and use them with the Collections API.
- Learn about Java's Streams concept, and how to apply to files using text and binary data.
- Create and run JUnit tests for Java classes.

### **Prerequisites**

Object-oriented Programming Experience in a language such as C++ or C#.

## Course Outline

### 1. Java Introduction

- Conventions in These Notes
- The Java Environment - Overview
- Writing a Java Program
- Obtaining The Java Environment
- Setting up your Java Environment
- Creating a Class That Can Run as a Program
  - The main() Method
- Useful Stuff Necessary to go Further
  - System.out.println()
- Using the Java Documentation

### 2. Java Basics

- Basic Java Syntax
  - General Syntax Rules
  - Java Statements
  - Blocks of Code
  - Comments
  - Variables
- Data
  - Primitive Data Types
  - Object Data Types
  - Literal Values
  - Constants and the final keyword
- Mathematics in Java
  - Expressions
  - Operator Precedence
  - Multiple Assignments
  - Order of Evaluation
  - Bitwise Operators
  - Compound Operators
  - Expressions that Mix Data Types: Typecasting
- Creating and Using Methods
  - Creating Methods
  - Variable Scope

### 3. Java Objects

- Objects
  - Object-Oriented Languages
  - Object-Oriented Programs
  - Encapsulation
  - Creating and Using an Instance of an Object
  - References
  - Defining a Class
  - Java Beans
  - Constructors

- Method Overloading
- The this Keyword
- static Elements
- Garbage Collection
- Java Packages
- Dealing with Keyboard Input
- String, StringBuffer, and StringBuilder
- Creating Documentation Comments and Using javadoc
- Javadoc Comments

#### **4. Comparisons And Flow Control Structures**

- Controlling Program Flow
  - Boolean-Valued Expressions
  - Complex boolean Expressions
  - Simple Branching
  - Two Mutually Exclusive Branches
  - Nestedif... elseStatements - Comparing a Number of MutuallyExclusive Options
  - Comparing a Number of Mutually ExclusiveOptions - The switchStatement
  - Comparing Objects
  - Conditional Expression
  - while and do. . .while Loops
  - for Loops
- Additional Loop Control: break and continue
  - Breaking Out of a Loop
  - Continuing a Loop
- Classpath, Code Libraries, and Jar files
  - Using CLASSPATH
  - Creating a jar File (a Library)

#### **5. Arrays**

- Arrays
  - Defining and Declaring Arrays
  - Instantiating Arrays
  - Initializing Arrays
  - Working With Arrays
  - Array Variables
  - Copying Arrays
  - Arrays of Objects
- Enhanced for Loops - the For-Each Loop
- Multi-Dimensional Arrays
  - Multidimensional Arrays in Memory
  - Example - Printing a Picture
- Typecasting with Arrays of Primitives

## 6. Inheritance

- Inheritance

  - Payroll with Inheritance

  - Derived Class Objects

- Polymorphism

  - Inheritance and References

  - Dynamic Method Invocation

- Creating a Derived Class

  - Inheritance and Access

  - Inheritance and Constructors - the super Keyword

  - Derived Class Methods That Override Base Class Methods

  - Inheritance and Default Base Class Constructors

  - The Instantiation Process at Runtime

- Example - Factoring Person Out of Employee and Dependent

- Typecasting with Object References

  - Typecasting, Polymorphism, and Dynamic Method Invocation

  - More on Overriding

  - Object Typecasting Example

  - Checking an Object's Type: Using instanceof

  - Typecasting with Arrays of Objects

- Other Inheritance-Related Keywords

  - abstract

  - final

- Methods Inherited from Object

## 7. Interfaces

- Interfaces

- Creating an Interface Definition

- Implementing Interfaces

  - Implementing Interfaces - Example

- Reference Variables and Interfaces

  - Calling an Interface Method

- Interfaces and Inheritance

- Some Uses for Interfaces

  - Interfaces and Event-Handling

  - Interfaces and "Pluggable Components"

## 8. Exceptions

- Exceptions

  - Handling Exceptions

  - Exception Objects

- Attempting Risky Code - try and catch

- Guaranteeing Execution of Code - the finally Block

- Letting an Exception be Thrown to the Method Caller

- Throwing an Exception

- Exceptions and Inheritance

  - Exception Class Constructors and Methods

- Creating and Using Your Own Exception Classes

- Rethrowing Exceptions
- Initializer Blocks
  - Static Initializer Blocks
  - Assertions

## **9. Generics and Collections**

- Fundamental Collections: Sets, Lists, and Maps
- Iterators
- Creating Collectible Classes
  - hashCode and equals
  - Comparable and Comparators
- Generics
  - Basic Generics Syntax
  - Bounded Types and Wildcards
- Advanced Topics
  - Type Erasure
  - instanceof Tests
  - Multiple-Bounded Types

## **10. Inner Classes**

- Inner Classes, aka Nested Classes
  - Inner Class Syntax
  - Instantiating an Inner Class Instance from Within the Enclosing Class
  - Inner Classes Referenced from Outside the Enclosing Class
  - Working with Inner Classes

## **12. Streams**

- Introducing Streams
  - Input Stream Classes
  - Output Stream Classes
  - Using System.in
- Files and Directories
  - Filename Filters
  - Creating a Directory Listing
- Files and Streams
  - Writing to a File
  - Creating a File Copying Program
- Working with Binary Data
  - Primitives
  - Object Serialization
- Properties and Properties Files

## **13. Unit Testing**

- Software Testing Concepts
- Unit Testing
- Using JUnit
  - Creating a Test Case