



Java 7 SE Programming for OO Experienced Developers

Code: TT2100-J7

5 days

Course Overview

This comprehensive, hands-on introduction to Java is geared for developers who have prior working knowledge of object-oriented (OO) programming languages such as C++.

Throughout the course, you will learn the best practices for writing great OO programs in Java 7 using sound development techniques, new improved features for better performance, and new capabilities for addressing rapid application development. You will participate in exercises, and you will work through a case study that covers the entire spectrum, from use cases to OO design to implemented classes. You can use the case study during and after the course as a reference and a tool for reviewing and practicing what you learned in class.

Who Needs To Attend

Experienced programmers and developers who have prior OO exposure and background (C++, etc)

Course Details

Topics Include

- Fundamentals of the Java language as well as its importance, uses, strengths, and weaknesses
- How Java relates to OO programming and the object model
- Use Java multi-threading and exception-handling features
- Use classes, inheritance, and polymorphism
- Use collections, generics, autoboxing, and enumerations, including new Java 7 features and capabilities
- Work with the logging API and framework that is part of the Java platform
- Use the Java Database Connectivity (JDBC) API for database access
- Use Java for networking and communication applications
- Work with annotations
- Work with the classes in the concurrent package
- Options for GUI applications in Java
- Take advantage of the Java tooling that is available with the programming environment being used in the class

Specific Java 7 features that are covered in the course include:

- Automatic closure of resources
- Handling multiple exceptions in a single catch block
- The diamond operator
- Support for multi-core platforms
- The Java NIO package

Course Outline

1. Java: A First Look

- Using the JDK
 - Setting Up Environment
 - The Development Process
 - Class Files
 - Java Applications
- Writing a Simple Class
 - Classes in Java
 - Class Modifiers and Types
 - Variables and Primitives
 - Creating Objects
 - The main Method
 - Using the Dot Operator
 - Writing Output
- The Java Platform
 - Defining Java
 - Java SE Development Kit (JDK)
 - Executing Programs
 - Lifecycle of a Java Program
 - Java Virtual Machine (JVM)
 - Java is Dynamic: The Runtime Process
 - Garbage Collection
 - Documentation and Code Reuse
 - JavaDoc Provides Documentation Delivery
 - In-Line Comments are Translated into HTML Rendering

2. OO Concepts In Java

- OO Programming
 - The OO Way
 - Real-World Objects
 - Classes and Objects
 - Object Behavior
 - Methods and Messages
- Inheritance, Abstraction, and Polymorphism
 - Encapsulation
 - Inheritance
 - Method Overriding
 - Aggregation
 - Type Abstraction: Grouping as Supertype
 - Polymorphism

3. Getting Started with Java

- Adding Methods to the Class
 - Instance Methods
 - Passing Parameters into Methods
 - Overloaded Methods
 - Constructors
- Language Statements
 - Operators
 - Comparison and Logical Operators
 - Using Comparison and Logical Operators
 - Looping: The for Statement
 - Looping: The while Statement
 - Looping: The do Statement
 - Continue and Break Statements
 - The switch Statement
- Using Strings
 - Strings
 - StringBuffer
 - StringTokenizer
 - Scanner
 - Formatter
- Specializing in a Subclass
 - Extending a Class
 - The extends Keyword
 - Casting
 - Overriding Superclass Methods
 - Method Overriding Diagram
 - Calling Superclass Methods from Subclass
 - The Object Class
 - The equals Method
 - Constructors

4. Essential Java Programming

- Fields and Variables
 - Fields vs. Variables
 - Instance vs. Local Variables: Usage Differences
 - Data Types
 - Default Values
 - Block Scoping Rules
 - Using this
 - Final and Static Fields
 - Static Variable Diagram
- Using Arrays
 - Arrays
 - Accessing the Array
 - Multidimensional Arrays
 - Copying Arrays
- Static Methods and Fields
 - Static Fields

- Simple Example of Static Fields
- Static Methods
- Java Packages and Visibility
 - The Problem
 - Packages
 - Class Location of Packages
 - The Package Keyword
 - Importing Classes
 - Executing Programs
 - Accessibility/Visibility
 - Java Naming Conventions
 - Packages Diagram

5. Advanced Java Programming

- Inheritance and Polymorphism
 - Polymorphism
 - Polymorphism: The Subclasses
 - Derived Classes as the Superclass
 - Casting to the Derived Class
 - Using instanceof for Downcasting
 - Upcasting vs. Downcasting
 - Calling Superclass Methods from Subclass
 - The final Keyword
- Interfaces and Abstract Classes
 - Separating Capability from Implementation
 - Abstract Classes
 - Shape as an Abstract Class
 - Polymorphism with Abstract Classes
 - Interfaces
 - Implementing an Interface
 - Extending Interfaces
 - Polymorphism with Interfaces
 - Type Che

Prerequisites

- OO development experience (such as working with C++)
- OO analysis and design experience similar to the content covered in Object-Oriented Analysis and Design with UML