

## Java 8 Programming for OO Experienced Developers

### Course Summary

#### Description

Java 8 Programming for OO Developers is a five-day, comprehensive hands-on introduction to Java training course geared for developers who have prior working knowledge of object-oriented programming languages such as C++. Throughout the course students learn the best practices for writing great object-oriented programs in Java 8, using sound development techniques, new improved features for better performance, and new capabilities for addressing rapid application development.

All Java courses include a separate, unique Java & OO Development Case Study, a documented, task-driven project that covers the entire application development spectrum from use cases to object-oriented design to implemented classes. This project supplements the course, and can be used during and after the course as a reference and a tool for reviewing and practicing newly learned skills.

#### Objectives

At the end of this course, students will be able to:

- Understand not only the fundamentals of the Java language, but also its importance, uses, strengths and weaknesses
- Understand the basics of the Java language and how it relates to OO programming and the Object Model
- Learn to use Java multi-threading and exception handling features
- Understand and use classes, inheritance and polymorphism
- Understand and use collections, generics, autoboxing, and enumerations including new Java features and capabilities
- Work with the logging API and framework that is part of the Java platform
- Use the JDBC API for database access
- Use Java for networking and communication applications
- Work with annotations
- Understand and work with the classes in the concurrent package
- Outline the options for GUI applications in Java
- Take advantage of the Java tooling that is available with the programming environment being used in the class

#### Topics

- The Java Platform
- Using the JDK
- Writing a Simple Class
- Adding Methods to the Class
- Language Statements
- Specializing in a Subclass
- Fields and Variables
- Using Arrays
- Java Packages and Visibility
- Inheritance and Polymorphism
- Interfaces and Abstract Classes
- Exceptions
- Utility Classes
- Enumerations and Static Imports
- The Date/Time API
- Introduction to Generics
- Collections
- Introduction to Lambda Expressions
- Streams
- Collectors
- Multithreading
- Concurrent Java
- File System Access
- Introduction to Annotations
- Java Data Access JDBC API

## **Java 8 Programming for OO Experienced Developers**

### **Course Summary (cont'd)**

#### **Audience**

Students who attend Java 8 Programming for OO Experienced Developers will leave this course armed with the required skills to develop solid object-oriented applications written in Java, using sound coding techniques and best coding practices. Geared for developers with prior OO development experience in languages such as C++ or SmallTalk, this course will teach students everything they need to become productive in essential Java programming.

#### **Prerequisites**

This is an introductory- level Java programming course, designed for experienced programmers with prior hands-on Object Oriented development experience in languages such as C++, C# or SmallTalk.

#### **Duration**

Five days

## Java 8 Programming for OO Experienced Developers

### Course Outline

- I. The Java Platform**
  - A. Java Platforms
  - B. Lifecycle of a Java Program
  - C. Responsibilities of JVM
  - D. Documentation and Code Reuse
- II. Using the JDK**
  - A. Setting Up Environment
  - B. Locating Class Files
  - C. Compiling Package Classes
  - D. Source and Class Files
  - E. Java Applications
- III. Writing a Simple Class**
  - A. Classes in Java
  - B. Class Modifiers and Types
  - C. Class Instance Variables
  - D. Primitives vs. Object References
  - E. Creating Objects
- IV. Adding Methods to the Class**
  - A. Passing Parameters Into Methods
  - B. Returning a Value From a Method
  - C. Overloaded Methods
  - D. Constructors
  - E. Optimizing Constructor Usage
- V. Language Statements**
  - A. Operators
  - B. Comparison and Logical Operators
  - C. Looping
  - D. Continue and Break Statements
  - E. The switch Statement
  - F. The for-each() Loop
  - G. Lesson: Using Strings
  - H. Strings
  - I. String Methods
  - J. String Equality
  - K. StringBuffer
  - L. StringBuilder
- VI. Specializing in a Subclass**
  - A. Extending a Class
  - B. Casting
  - C. The Object Class
  - D. Default Constructor
  - E. Implicit Constructor Chaining
- VII. Fields and Variables**
  - A. Instance vs. Local Variables: Usage Differences
  - B. Data Types
  - C. Default Values
  - D. Block Scoping Rules
  - E. Final and Static Fields
  - F. Static Methods
- VIII. Using Arrays**
  - A. Arrays
  - B. Accessing the Array
  - C. Multidimensional Arrays
  - D. Copying Arrays
  - E. Variable Arguments
- IX. Java Packages and Visibility**
  - A. Class Location of Packages
  - B. The Package Keyword
  - C. Importing Classes
  - D. Executing Programs
  - E. Java Naming Conventions
  - F. Session: Advanced Java Programming
- X. Inheritance and Polymorphism**
  - A. Polymorphism: The Subclasses
  - B. Upcasting vs. Downcasting
  - C. Calling Superclass Methods From Subclass
  - D. The final Keyword
- XI. Interfaces and Abstract Classes**
  - A. Separating Capability from Implementation
  - B. Abstract Classes
  - C. Implementing an Interface
  - D. Abstract Classes vs. Interfaces
- XII. Exceptions**
  - A. Exception Architecture
  - B. Handling Multiple Exceptions
  - C. Automatic Closure of Resources
  - D. Creating Your Own Exceptions
  - E. Throwing Exceptions
  - F. Checked vs. Unchecked Exceptions
- XIII. Utility Classes**
  - A. Wrapper Classes
  - B. The Number Class
  - C. Random Numbers
  - D. Autoboxing/Unboxing
  - E. The Date Class

## Java 8 Programming for OO Experienced Developers

### Course Outline (cont'd)

- XIV. Enumerations and Static Imports**
  - A. Enumeration Syntax
  - B. When You Should Use Enumerations
  - C. Using Static Imports
  - D. When You Should Use Static Imports
  - E. Lesson: The Date/Time API
  - F. The Date Class
  - G. Introduce the new Date/Time API
  - H. LocalDate, LocalDateTime, etc.
  - I. Formatting Dates
  - J. Working with time zones
  - K. Manipulate date/time values
  - L. Session: Collections and Generics
- XV. Introduction to Generics**
  - A. Generics and Subtyping
  - B. Bounded Wildcards
  - C. Generic Methods
  - D. Legacy Calls To Generics
  - E. When Generics Should Be Used
- XVI. Collections**
  - A. Characterizing Collections
  - B. Collection Interface Hierarchy
  - C. Iterators
  - D. The Set Interface
  - E. The List Interface
  - F. Queue Interface
  - G. Map Interfaces
  - H. Using the Right Collection
  - I. Collections and Multithreading
- XVII. Introduction to Lambda Expressions**
  - A. Functional vs OO Programming
  - B. Anonymous Inner-classes
  - C. Lambda Expression Syntax
  - D. Functional Interfaces
  - E. Method references
  - F. Constructor references
- XVIII. Streams**
  - A. Processing Collections of data
  - B. The Stream interface
  - C. Reduction and Parallelism
  - D. Filtering collection data
  - E. Sorting Collection data
  - F. Map collection data
  - G. Find elements in Stream
  - H. Numeric Streams
  - I. Create infinite Streams
  - J. Sources for using Streams
- XIX. Collectors**
  - A. Creating Collections from a Stream
  - B. Group elements in the Stream
  - C. Multi-level grouping of elements
  - D. Partitioning Streams
- XX. Multithreading**
  - A. Principles of Multithreading
  - B. Creating a Threaded Class
  - C. Basic Features of the Thread Class
  - D. Thread Scheduling
  - E. Thread Synchronization
- XXI. Concurrent Java**
  - A. Concurrent Locks are Explicit and Flexible
  - B. Executor Interfaces Provide Thread Management
  - C. Challenges for Concurrent Use of Collections
  - D. Concurrent Collections
  - E. Atomic Variables Avoid Synchronization
- XXII. File System Access**
  - A. The File Class
  - B. File Utility Methods
  - C. Lesson: Java I/O
  - D. The Java I/O Mechanism
  - E. Subclasses Accessing Real Data
  - F. Filter Classes
  - G. New File IO - NIO
  - H. NIO Overview
- XXIII. Introduction to Annotations**
  - A. Annotations Overview
  - B. Working with Java Annotations
- XXIV. Java Data Access JDBC API**
  - A. Connecting to the Database
  - B. Statement and PreparedStatement
  - C. ResultSet
  - D. Executing Inserts, Updates, and Deletes
  - E. Controlling Transactions and Concurrency
  - F. Connecting to the Database
  - G. Statement and PreparedStatement
  - H. ResultSet
  - I. Executing Inserts, Updates, and Deletes
  - J. Controlling Transactions and Concurrency