Java 8 and OO Essentials for COBOL, Mainframe, and non-OO Developers

Course Summary

Description

Java 8 and OO Essentials is a five-day, comprehensive hands-on Java training course geared for developers who have little or no prior working knowledge of object-oriented programming languages (such as those working on (C, COBOL, 4GL, etc.). 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. Special emphasis is placed on object oriented concepts and best practices.

Objectives

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

- Understand what OO programming is and what the advantages of OO are in today's world
- Work with objects, classes, and OO implementations
- Understand the basic concepts of OO such as encapsulation, inheritance, polymorphism, and abstraction
- Understand not only the fundamentals of the Java language, but also it’s 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 exception handling and logging features
- Understand and use classes, inheritance and polymorphism
- Understand and use collections, generics, autoboxing, and enumerations
- Use the JDBC API for database access
- Use Java for networking and communication applications
- Work with annotations
- Take advantage of the Java tooling that is available with the programming environment being used in the class

Topics

- Java 8: A First Look
- OO Concepts
- Getting Started with Java™
- Essential Java™ Programming
- Advanced Java™ Programming
- Developer’s Toolbox
- Java ™ 8 Lambda and Streams
- Java ™ Application Development

Audience

This is an introductory-level course, designed for experienced developers who wish to get up and running with Java, or who need to reinforce sound object oriented Java coding practices.

Prerequisites

Before taking this course, should have practical skills equivalent to or should have received training in the following topic(s) as a pre-requisite: Object-Oriented Analysis & Design Using UML 2.x

Duration

Five days
Java 8 and OO Essentials for COBOL, Mainframe, and non-OO Developers

Course Outline

I. Java 8: A First Look
   A. The Java™ Platform
      1. Java Platforms
      2. Lifecycle of a Java Program
      3. Responsibilities of JVM
      4. Documentation and Code Reuse
   B. Using the JDK
      1. Setting Up Environment
      2. Locating Class Files
      3. Compiling Package Classes
      4. Source and Class Files
      5. Java Applications
   C. The Eclipse Paradigm
      1. Workbench and Workspace
      2. Views
      3. Editors
      4. Perspectives
      5. Projects
   D. Using Strings
      1. Strings
      2. String Methods
      3. String Equality
      4. StringBuffer
      5. StringBuilder
      6. Java 8 - StringJoiner

II. OO Concepts
   A. Object-Oriented Programming
      1. Real-World Objects
      2. Classes and Objects
      3. Object Behavior
      4. Methods and Messages
   B. Inheritance, Abstraction, and Polymorphism
      1. Encapsulation
      2. Inheritance
      3. Method Overriding
      4. Polymorphism

III. Getting Started with Java™
   A. Writing a Simple Class
      1. Classes in Java™
      2. Class Modifiers and Types
      3. Class Instance Variables
      4. Primitives vs. Object References
      5. Creating Objects
   B. Adding Methods to the Class
      1. Passing Parameters Into Methods
      2. Returning a Value From a Method
      3. Overloaded Methods
      4. Constructors
      5. Optimizing Constructor Usage
   C. Language Statements
       1. Operators
       2. Comparison and Logical Operators
       3. Looping
       4. Continue and Break Statements
       5. The switch Statement

IV. Essential Java™ Programming
   A. Fields and Variables
      1. Instance vs. Local Variables: Usage Differences
      2. Data Types
      3. Default Values
      4. Block Scoping Rules
      5. Final and Static Fields
      6. Static Methods
   B. Using Arrays
      1. Arrays
      2. Accessing the Array
      3. Multidimensional Arrays
      4. Copying Arrays
      5. Variable Method Arguments
   C. Java™ Packages and Visibility
      1. Class Location of Packages
      2. The Package Keyword
      3. Importing Classes
      4. Executing Programs
      5. Java Naming Conventions

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically.

PT0642_JAVA8ANDOOESSENTIALSFORCOBOL.DOC
Java 8 and OO Essentials for COBOL, Mainframe, and non-OO Developers

Course Outline (cont’d)

V. Advanced Java™ Programming
   A. Inheritance and Polymorphism
      1. Polymorphism: The Subclasses
      2. Upcasting vs. Downcasting
      3. Calling Superclass Methods From Subclass
      4. The final Keyword
   B. Interfaces and Abstract Classes
      1. Separating Capability from Implementation
      2. Abstract Classes
      3. Implementing an Interface
      4. Java 8 - Implementing methods in an interface
      5. Abstract Classes vs. Interfaces
   C. Exceptions
      1. Exception Architecture
      2. Handling Multiple Exceptions
      3. Automatic Closure of Resources
      4. Creating Your Own Exceptions
      5. Throwing Exceptions
      6. Checked vs. Unchecked Exceptions

VI. Developer’s Toolbox
   A. Utility Classes
      1. Wrapper Classes
      2. The Number Class
      3. Random Numbers
      4. Autoboxing/Unboxing
      5. The Date Class
      6. Java 8 – Date/Time API
      7. Java 8 – The Optional class
   B. Generics
      1. Introduction to generics
      2. Generic classes and methods
      3. Bounded Wildcards
      4. Generic Methods
      5. Legacy Calls To Generics
      6. When Generics Should Be Used
   C. Collections
      1. Characterizing Collections
      2. Collection Interface Hierarchy
      3. Iterators
      4. The Set Interface
      5. The List Interface
      6. Queue Interface
      7. Map Interfaces
      8. Using the Right Collection
      9. Collections and Multithreading

VII. Java™ 8 Lambda and Streams
   A. Introduction to Lambda Expressions
      1. Functional Interfaces
      2. Lambda Expression syntax
   B. Streams
      1. Introduce the Stream API
      2. Streams and Concurrency
   C. Collections, Lamda and Streams
      1. Using Streams in Collections
      2. Filtering and sorting Collection data
      3. Sorting Collection data

VIII. Java™ Application Development
   A. Introduction to Annotations
      1. Annotations Overview
      2. Working with Java Annotations
   B. JDBC™
      1. Connecting to the Database
      2. Statement and PreparedStatement
      3. ResultSet
      4. Executing Inserts, Updates, and Deletes
      5. Controlling Transactions and Concurrency

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically.