ProTech *"Charting the Course ...*

protechtraining.com

... to Your Success!"

Gentle Java and OO Development for COBOL, Mainframe and non-OO Developers
Course Summary

Description

Gentle Java and OO Development is geared to audiences with no background in Object-Oriented programming or Java. It uses a steady pace and numerous hands-on labs to provide an easy entry into the world of Java and OO Programming.

Java is one of the most important programming languages used today, and its correct usage is a critical part of many systems being written now. The basics of the language are relatively easy, but the challenge lies in learning how to use it well. This is especially true regarding the Object-Oriented nature of Java, which for many developers is a new way of approaching system design and construction.

This course covers the core OO and Java concepts that developers need to use the Java programming language to create well designed Java programs. It focuses on key OO capabilities and how to use them in Java. This includes material on creating well designed Java classes, using encapsulation, building more complex functionality with composition, and using inheritance to share functionality and create specialized types.

This course is suitable for environments using any recent version of Java. The material includes coverage of all current Java capabilities that are within the scope of the material, including new Java 8 capabilities such as the new Date/Time support. It also stresses the use of good coding practices for all the examples and labs. All labs are doable in any of the supported Java environments. This course is available for most IDEs, such as Eclipse and IBM RAD.

This course draws on our extensive experience to provide a solid understanding of the concepts and practices needed to write good object oriented programs in Java. Be prepared to work hard and learn a great deal! This course can be followed by our Intermediate Java course. The course can be held on-site and customized to fit your needs.

All labs can be done with the Eclipse IDE or a simple editor, and the lab instructions include detailed directions on both environments.

Objectives

After taking this course, students will be able to:

- Understand Java's importance, uses, strengths, and weaknesses
- Understand Java language basics
- Write, compile, and run Java programs
- Understand the Object Model and Object Oriented Programming
- Understand and use classes, inheritance, polymorphism
- Create well designed classes and use them in your Java programs
- Use composition and delegation to create objects from other objects
- Understand & use packages to organize code
- Be familiar with more advanced concepts such as inheritance and JDBC
- Learn good Java coding style
- Create well-structured Java programs
- Compile and execute programs with the Java JDK and with an Integrated Development Environment (IDE) of your choice
- Be familiar with other new and advanced Java features

Gentle Java and OO Development for COBOL, Mainframe and non-OO Developers

Course Summary (cont'd)

Topics

- A First Look
- Java Basics
- Class and Object Basics
- More on Classes and Objects
- Flow of Control
- Strings and Arrays
- Packages
- Composition and Inheritance
- Interfaces Overview
- Exceptions
- JDBC
- Java Collections and Generics
- Additional Java Features

Audience

Gentle Java and OO Development is geared to audiences with no background in Object-Oriented programming or Java.

Prerequisites

There are no prerequisites for this course.

Duration

Four or five days

Gentle Java and OO Development for COBOL, Mainframe and non-OO Developers
Course Outline

I. A First Look
- A. A Simple Java Class
- B. Java's "Hello World" Program

Lab:
- **Hello World: A Simple Application**

II. Java Basics
- A. Language and Platform Features
- B. Program Life Cycle
- C. The Java SE Development Kit (JDK)

Lab:
- **Working with the Development Environment**

III. Class and Object Basics
- A. The Object Model and Object-Oriented Programming
- B. Classes, References, and Instantiation
- C. Adding Data to a Class Definition
- D. Adding Methods (Behavior)

Labs:
- **Exploring Types and Object Instances**
- **Writing and Using a Class Definition with Fields and Methods**

IV. More on Classes and Objects
- A. Accessing data, the "this" variable
- B. Encapsulation and Access Control, public and private Access
- C. Constructors and Initialization
- D. static Members of a Class
- E. Scopes, Blocks, References to Objects

Labs:
- **Encapsulation / Access Protection**
- **Writing and Using Constructors**
- **(Optional) Static Members**
- **Using the Debugger**

V. Flow of Control
- A. Branching: if, if-else, switch
- B. Iteration: while, do-while, for, break, continue

Lab:
- **Flow of Control / Data Validation**

VI. Strings and Arrays
- A. String, StringBuffer, StringBuilder
- B. Arrays, Primitive Arrays, Arrays of Reference Types
- C. varargs

Lab:
- **Using Strings and Arrays**

VII. Packages
- A. Package Overview - Using Packages to Organize Code
- B. import statements
- C. Creating Packages, package Statement, Required Directory Structure
- D. Finding Classes, Packages and Classpath

Lab:
- **Using Packages to Organize Code**

VIII. Composition and Inheritance
- A. Using Composition to Deal With Complexity
- B. Composition/HAS-A, Delegation
- C. Using Inheritance and Polymorphism to share commonality
- D. IS-A, extends, Inheriting Features, Overriding Methods, Using Polymorphism
- E. Class Object
- F. Abstract Classes

Labs:
- **(Optional) Working with Composition**
- **Using Inheritance to Specialize Classes**

IX. Interfaces Overview
- A. Using Interfaces to Define Types
- B. Interfaces and Abstract Classes

Gentle Java and OO Development for COBOL, Mainframe and non-OO Developers

Course Outline (cont'd)

X. Exceptions
- A. Exceptions and the Exception Hierarchy
- B. try and catch
- C. Handling Exceptions
- D. Program Flow with Exceptions
- E. finally

Lab:
- **Throwing and Handling Exceptions**

XI. JDBC
- A. JDBC basics
- B. JDBC Architecture
- C. Using JDBC drivers & DriverManager
- D. Class Connection and connecting to a database
- E. Class Statement and executing SQL statements
- F. Other statement types

Lab:
- **Connecting and Querying using JDBC**

XII. Java Collections and Generics
- A. The Collections Framework and its API
- B. Collections and Java Generics
- C. Collection, Set, List, Iterator
- D. Autoboxing
- E. for-each Loop

Lab:
- **Using Lists and Generics**

XIII. Additional Java Features
- A. Java 8 Date/Time support
- B. Annotations
- C. Other Java Features