



Basic Java 8 and OO Programming Essentials for Developers New to OO

Coming from C, COBOL or other Non-OO Language? Gain the Skills Needed to Start Writing Solid Object-Oriented Applications in Java

www.triveratech.com

Course Snapshot

- **Course: TT2120-J8: Basic Java 8 and OO Programming for Developers New to OO**
- **Duration:** 5 days
- **Audience & Skill-Level :** Basic-level topics for experienced developers new to Object Oriented development (coming from languages such as C, COBOL, 4GL, etc.). This course is not for non-developers.
- **Focus:** This course is for the Java 8 LTS / Long-Term Support edition. This course is also offered for Java 11 (LTS) or the most current editions (14+, etc.). Please inquire for details.
- **Hands-on Learning:** This course is approximately **50% hands-on**, combining expert lecture, real-world demonstrations and group discussions with machine-based practical programming labs and exercises. Student machines are required.
- **Delivery Options:** This course is available for **onsite private classroom presentation, live online virtual presentation**, or can be presented in a **flexible blended learning format** for combined onsite and remote attendees. Please also ask about our **Self-Paced / Video / QuickSkills or Mini-Camp Flex Hours / Short Course** options.
- **Public Schedule:** This course has active dates on our live-online open enrollment **Public Schedule**.
- **Customizable:** This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please inquire for details.

Overview

Basic Java 8 and OO Programming Essentials for Developers New to OO is a five-day, 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**.

This course introduces new features in **Java 9** and **Java 10**, including the Java Modular System and Local Variable Type Inference. Developers leaving this course will be able to participate in projects that are still on **Java 8**, while they are also ready to move onto projects using **Java 10**

Learning Objectives

This “skills-centric” course is about **50% hands-on lab and 50% lecture**, designed to train attendees in core OO coding and Java development skills, coupling the most current, effective techniques with the soundest industry practices. Our engaging instructors and mentors are highly experienced practitioners who bring years of current “on-the-job” experience into every classroom.

Working in a hands-on learning environment, guided by our expert team, attendees will learn 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 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 exception handling
- Understand and use classes, inheritance and polymorphism
- Understand and use collections, generics, autoboxing, and enumerations
- Become familiar with the concept of functional programming using Lambda Expressions
- Process large amounts of data using the Stream API introduced in Java 8
- Discover the new Date/Time API
- Use the JDBC API for database access
- Work with annotations

- Take advantage of the Java tooling that is available with the programming environment being used in the class
- Java 8 Features: Lambda Expressions, Method and Constructor references, The Streams API, Collectors, The Optional class

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate. We offer additional Java and JEE programming, design, testing and application security courses which may be blended with this course for a track that best suits your development objectives. Our team will collaborate with you to understand your needs and will target the course to focus on your specific learning objectives and goals.

Audience & Pre-Requisites

This is a **basic-level** programming course designed for attendees with prior development experience in another language, such as COBOL, 4GL, Mainframe or other non-object oriented languages. This course is not geared for non-developers.

Related & Alternative Java Essentials Courses: This course is for experienced developers with little to no prior OO development experience, such as C or COBOL programmers. We offer other courses with similar topics, geared for different experience levels or background:

- TT2000-J8 Getting Started with Programming, OO, and Java 8 Basics for Non-Developers
- TT2120-J8 Basic Java 8 and OO Programming Essentials for Developers New to OO (such as C, COBOL, 4GL)
- TT2104-J8 Fast Track to Java 8 Programming for OO Developers (such as C+, C#, etc.)
- TT2100-J8 Introduction to Java 8 Programming for OO Developers (such as C+, C#, etc.)

Take After: Our core Java training courses provide students with a solid foundation for continued learning based on role, goals, or their areas of specialty. Our object oriented, Java developer learning paths offer a wide variety of follow-on courses such as:

- Continued Java & JEE training: Intermediate to Advanced Java, JEE Essentials, Java for Web, Spring / Spring Boot, Microservices / Web Services / REST, RxJava, Tuning, Patterns & More
- TDD / Test Driven Development, JUnit / Unit Testing, Agile development training
- Secure Java Coding / Java Security and secure application development training
- Mobile developer / Android training
- Please contact us for recommended next steps tailored to your longer-term education, project, role or development objectives.

Enhanced Learning Services: Please also ask about our **Pre-Training Class OnRamp & Prep / Primer offerings, Skills Gap Assessment Services, Case Studies, Knowledge Check Quizzes, Skills Immersion Programs & Camps, Collaborative Mentoring Services and Extended Learning Support & Post Training services.**

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We will work with you to tune this course and level of coverage to target the skills you need most. Course agenda, topics and labs are subject to adjust during live delivery in response to student skill levels, interests and participation.

Session: Java: A First Look

Lesson: The Java Platform

- Java Platforms
- Lifecycle of a Java Program
- Responsibilities of JVM
- Documentation and Code Reuse

Lesson: Using the JDK

- Setting Up Environment
- Locating Class Files
- Compiling Package Classes
- Source and Class Files
- Java Applications
- [Lab: Exploring MemoryViewer](#)

- [Lab: Exploring ColorPicker](#)

Lesson: The Eclipse Paradigm

- Workbench and Workspace
- Views
- Editors
- Perspectives
- Projects
- [Tutorial: Working with Eclipse](#)

Lesson: Writing a Simple Class

- Classes in Java
- Class Modifiers and Types
- Class Instance Variables
- Primitives vs. Object References

- Creating Objects
- [Lab: Create a Simple Class](#)

Session: OO Concepts

Lesson: Object-Oriented Programming

- Real-World Objects
- Classes and Objects
- Object Behavior
- Methods and Messages
- [Lab: Define and use a New Java class](#)

Lesson: Inheritance, Abstraction, and Polymorphism

- Encapsulation
- Inheritance
- Method Overriding
- Polymorphism
- [Lab: Define and use Another Java Class](#)

Session: Getting Started with Java**Lesson: Adding Methods to the Class**

- Passing Parameters into Methods
- Returning a Value from a Method
- Overloaded Methods
- Constructors
- Optimizing Constructor Usage
- [Lab: Create a Class with Methods](#)

Lesson: Language Statements

- Operators
- Comparison and Logical Operators
- Looping
- Continue and Break Statements
- The switch Statement
- The for-each() Loop
- [Lab: Looping](#)
- [Lab: Language Statements](#)

Lesson: Using Strings

- Strings
- String Methods
- String Equality
- StringBuffer
- StringBuilder
- [Lab: Fun with Strings](#)
- [Lab: Using StringBuffers and StringBuilders](#)

Lesson: Specializing in a Subclass

- Extending a Class
- Casting
- The Object Class
- Default Constructor
- Implicit Constructor Chaining
- [Lab: Creating Subclasses](#)
- [Lab: Defining the Student Subclass](#)

Session: Essential Java Programming**Lesson: Fields and Variables**

- Instance vs. Local Variables: Usage Differences
- Data Types
- Default Values

- Block Scoping Rules
- Final and Static Fields
- Static Methods
- [Lab: Field Test](#)

Lesson: Using Arrays

- Arrays
- Accessing the Array
- Multidimensional Arrays
- Copying Arrays
- Variable Arguments
- [Lab: Creating an Array](#)
- [Lab: Defining the Student Array](#)

Lesson: Java Packages and Visibility

- Class Location of Packages
- The Package Keyword
- Importing Classes
- Executing Programs
- Java Naming Conventions

Session: Advanced Java Programming**Lesson: Inheritance and Polymorphism**

- Polymorphism: The Subclasses
- Upcasting vs. Downcasting
- Calling Superclass Methods from Subclass
- The final Keyword
- [Lab: Salaries - Polymorphism](#)

Lesson: Interfaces and Abstract Classes

- Separating Capability from Implementation
- Abstract Classes
- Implementing an Interface
- Abstract Classes vs. Interfaces
- [Lab: Mailable - Interfaces](#)

Lesson: Exceptions

- Exception Architecture
- Handling Multiple Exceptions
- Automatic Closure of Resources
- Creating Your Own Exceptions
- Throwing Exceptions
- Checked vs. Unchecked Exceptions
- [Lab: Exceptions](#)

Session: Java Developer's Toolbox**Lesson: Utility Classes**

- Wrapper Classes
- The Number Class
- Random Numbers
- Autoboxing/Unboxing

- The Date Class
- [Lab: Using Primitive Wrappers](#)

Lesson: Enumerations and Static Imports

- Enumeration Syntax
- When You Should Use Enumerations
- Using Static Imports
- When You Should Use Static Imports
- [Lab: Enumerations](#)

Lesson: The new Date/Time API

- Introduce the new Date/Time API
- LocalDate, LocalDateTime, etc.
- Formatting Dates
- Working with time zones
- Manipulate date/time values
- [Lab: Agenda](#)

Lesson: Formatting Strings

- StringJoiner
- String.format
- System.out.printf
- The Formatter class
- Using the formatting syntax

Session: Collections and Generics**Lesson: Introduction to Generics**

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls to Generics
- When Generics Should Be Used
- [Lab: DynamicArray](#)
- [Lab: Adding Generics to Dynamic Array](#)

Lesson: Collections

- Characterizing Collections
- Collection Interface Hierarchy
- Iterators
- The Set Interface
- The List Interface
- Queue Interface
- Map Interfaces
- Using the Right Collection
- Collections and Multithreading
- [Lab: Using Hashtable and HashMap](#)
- [Lab: Collections Poker](#)
- [Lab: Writing a Collection](#)

**Session: Lambda Expressions;
Collections and Streams****Lesson: Introduction to Lambda
Expressions**

- Functional vs OO Programming
- Anonymous Inner-classes
- Lambda Expression Syntax
- Functional Interfaces
- Method references
- Constructor references

Lesson: Java 8 Collection Updates

- Introduce the ConcurrentHashMap
- Lambda expressions and Collections
- [Exercise: Functional Collections](#)

Lesson: Streams

- Processing Collections of data
- The Stream interface

- Reduction and Parallelism
- Filtering collection data
- Sorting Collection data
- Map collection data
- Find elements in Stream
- Numeric Streams
- Create infinite Streams
- Sources for using Streams
- [Lab: Working with Streams](#)

Lesson: Collectors

- Creating Collections from a Stream
- Group elements in the Stream
- Multi-level grouping of elements
- Partitioning Streams
- [Lab: Collecting](#)

**Session: Java Application
Development****Lesson: Introduction to Annotations**

- Annotations Overview
- Working with Java Annotations
- [Lab: Annotations](#)
- [Lab: Using Annotations](#)

Lesson: Java Data Access JDBC API

- Connecting to the Database
- Statement and PreparedStatement
- ResultSet
- Executing Inserts, Updates, and Deletes
- Controlling Transactions and Concurrency
- [Tutorial: Setup the Derby Database](#)
- [Lab: Reading Table Data](#)
- [Lab: Using JdbcRowSet](#)
- [Lab: Executing within a Transaction](#)

Course Materials: Each student will receive a **Student Guide** with course notes, code samples, software tutorials, step-by-step written lab instructions, diagrams and related reference materials and links (as applicable). Students will also receive the project files (or code, if applicable) and solutions required for the hands-on work.

Hands-on Setup Made Simple! Our dedicated tech team will work with you to **ensure your student machines and learning environment is setup, tested and ready to go** well in advance of the course delivery date, ensuring a smooth start to class and seamless hands-on experience for your students. We offer several flexible student machine setup options including **guided manual set up** for simple installation directly on student machines, or **cloud based / remote hosted lab solutions** where students can log in to a complete separate lab environment minus any installations, or we can supply **complete turn-key, pre-loaded equipment** to bring ready-to-go student machines to your students or in-person facility. Please inquire for details.

For More Information

For more information about our dedicated training services, collaborative mentoring services, courseware licensing options, courseware development services, public course schedule, training management services, partner and reseller programs, or to see our complete list of course offerings and special offers please visit us at www.triveratech.com, email Info@triveratech.com or call us toll free at **844-475-4559**. Our pricing and services are always satisfaction guaranteed.

TRIVERA TECHNOLOGIES • Collaborative IT Training, Coaching & Skills Development Solutions
www.triveratech.com • toll free +1-844-475-4559 • Info@triveratech.com • Twitter TriveraTech

ONSITE, ONLINE & BLENDED TRAINING SOLUTIONS | PUBLIC / OPEN ENROLLMENT COURSES | COURSEWARE LICENSING & DEVELOPMENT
MENTORING | ASSESSMENTS | LEARNING PLAN DEVELOPMENT | SKILLS IMMERSION PROGRAMS / RESKILLING / NEW HIRE / BOOT CAMPS
PARTNER & RESELLER PROGRAMS | CORPORATE TRAINING MANAGEMENT | VENDOR MANAGEMENT SERVICES

Trivera Technologies is a Woman-Owned Small-Business Firm

