QUICKSTART

Over 35 Years Of Technology Training

Document Generated: 12/18/2025

Learning Style: Virtual Classroom

Technology: Java

Difficulty: Beginner

Course Duration: 5 Days

Introduction to Java 8 Programming for Developers New to OO Programming (such as C, Mainframe, 4GL) (TT2120-J8)

About this course:

This training course of Java 8 and OO Programming Essentials is a hands-on, five days course designed for engineers who have practically zero earlier information on object-oriented programming dialects, (for example, those working on (C, 4GL, COBOL, and so forth.) Throughout the course, understudies gain proficiency with the prescribed procedures for writing object-oriented projects in Java 8, using sound advancement methods, new improved highlights for better execution, and new capacities for addressing quick application advancement. Exclusive accentuation is put on object-oriented ideas and best practices.

Note: Learners with earlier Object-Oriented introduction and foundation, (for example, C++, C#, Smalltalk, and so forth.) ought to consider the Programming of TT2100-J9 Java 8 for Object-Oriented (OO) Experienced Developers as another option.

The normal compensation of a Java Developer is $97,000 every year.

Course Objective:

- Comprehend about OO programming and what the benefits of OO are in this current time
- Work with classes, objects, and OO usage
- Work with the Java 9 particular framework (Project Jigsaw)
- Take benefit of the Java tooling that is accessible with the programming condition being utilized in the class.
- Comprehend the basics of the Java language, and its significance uses, weakness, and strength.
- Comprehend the essential ideas of OO, for example, polymorphism, inheritance, encapsulation, and abstraction.
- Comprehend the rudiments of the Java language and how it identifies with OO programming and the Object Model
- Comprehend and use generics, collections, autoboxing, and enumerations
- The Modular framework (Project Jigsaw)
- Figure out how to utilize Java exception handling highlights
- Comprehend and use inheritance, classes, and polymorphism
- Procedure a lot of information using the Stream API and Lambda expressions.
- Utilize the JDBC API for access to the database.
- Private techniques in interfaces
- Understand and use the Stream API
- Discover the new Date/Time API
- Use the JDBC API for database access
- The new Date/Time API
- The Optional class
- Lambda Expressions
- Method and Constructor references
- Work with annotations
- The Streams API
- Collectors

Audience:

This course is an entry-level programming course of Java, intended for experienced engineers who wish to find a good pace with Java, or who need to strengthen Java coding practices.

Prerequisite:

Participants ought to have earlier handy programming involvement with another language.

Course Outline:

Module 1: 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

- Exercise: Exploring ColorPicker and MemoryViewer

Lesson: The Eclipse Paradigm

- Workbench and Workspace
- Views
- Editors
- Perspectives
- Projects
- Tutorial: Working with Eclipse Neon

Lesson: Writing a Simple Class

- Classes in Java
- Class Modifiers and Types
- Class Instance Variables
- Primitives vs. Object References
- Creating Objects
- Exercise: Create a Simple Class

Module 2: OO Concepts

Lesson: Object-Oriented Programming

- Real-World Objects
- Classes and Objects
- Object Behavior
- Methods and Messages

Lesson: Inheritance, Abstraction, and Polymorphism

- Encapsulation
- Inheritance
- Method Overriding

- Polymorphism

Module 3: 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
- Exercise: 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
- Exercise: Looping

Lesson: Using Strings

- Strings
- String Methods
- String Equality
- StringBuffer
- StringBuilder
- Exercise: Fun with Strings
- Exercise: Using StringBuffers and StringBuilders

Lesson: Specializing in a Subclass

- Extending a Class
- Casting
- The Object Class
- Default Constructor
- Implicit Constructor Chaining
- Exercise: Creating Subclasses

Module 4: 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
- Exercise: Field Test

Lesson: Using Arrays

- Arrays
- Accessing the Array
- Multidimensional Arrays
- Copying Arrays
- Variable Arguments
- Exercise: Creating an Array

Lesson: Java Packages and Visibility

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

Module 5: Advanced Java Programming

Lesson: Inheritance and Polymorphism

- Polymorphism: The Subclasses
- Upcasting vs. Downcasting
- Calling Superclass Methods From Subclass
- The final Keyword
- Exercise: Salaries - Polymorphism

Lesson: Interfaces and Abstract Classes

- Separating Capability from Implementation
- Abstract Classes
- Implementing an Interface
- Abstract Classes vs. Interfaces
- Exercise: Mailable - Interfaces

Lesson: Exceptions

- Exception Architecture
- Handling Multiple Exceptions
- Automatic Closure of Resources
- Creating Your Own Exceptions
- Throwing Exceptions
- Checked vs. Unchecked Exceptions

- Exercise: Exceptions

Module ?6: Java Developer's Toolbox

Lesson: Utility Classes

- Wrapper Classes
- The Number Class
- Random Numbers
- Autoboxing/Unboxing
- The Date Class
- Exercise: Using Primitive Wrappers

Lesson: Enumerations and Static Imports

- Enumeration Syntax
- When You Should Use Enumerations
- Using Static Imports
- When You Should Use Static Imports
- Exercise: 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
- Exercise: Agenda

Module 7: Collections and Generics

Lesson: Introduction to Generics

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods
- Legacy Calls To Generics
- When Generics Should Be Used
- Exercise: ShoppingCart

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
- Exercise: Using Hashtable and HashMap
- Exercise: Collections Poker
- Exercise: Writing a Collection

Module 8: Java Lambda Expressions and Streams

Lesson: Introduction to Lambda Expressions

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

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
- Exercise: Working with Streams

Lesson: Collectors

- Creating Collections from a Stream
- Group elements in the Stream
- Multi-level grouping of elements
- Partitioning Streams
- Exercise: Collecting

Module 9: Java Application Development

Lesson: Introduction to Annotations

- Annotations Overview
- Working with Java Annotations
- Exercise: Annotations

- Exercise: 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
- Exercise: Reading Table Data
- Exercise: Using JdbcRowSet
- Exercise: Executing within a Transaction