

## **Java 9/10 Programming for OO Experienced Developers (C#, C++, etc.) (TT2100-J9/10)**

**Modality:** Virtual Classroom

**Duration:** 5 Days

**SATV Value:**

**CLC:**

**NATU:**

**SUBSCRIPTION:** No

### **About this course:**

***Java 9 Programming for Object-Oriented Experienced 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# or C++. Throughout the course students learn the best practices for writing great object-oriented programs in Java 9, using sound development techniques, new improved features for better performance and new capabilities for addressing rapid application development. In addition to the normal exercises that are liberally sprinkled throughout the course, there is a case study that covers the entire spectrum from use cases to object-oriented design to implemented classes. This case study supplements the course and can be used during and after the course as a reference and a tool for reviewing and practicing what was learned in class.

The average salary for a Java 9 Programmer is **\$117,227** per year.

### **Course Objective:**

Working within in an engaging, hands-on learning environment, guided by our expert team, attendees will learn 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 exception handling features
- Work with the Java 9 modular system (Project Jigsaw)
- Understand and use classes, inheritance, and polymorphism
- Understand and use collections, generics, autoboxing, and enumerations
- Process large amount of data using Lambda expressions and the Stream API
- Use the JDBC API for database access
- Take advantage of the Java tooling that is available with the programming environment being used in the class
- Become familiar with the APIs used to access data and other resources
- Understand the concepts of Object-to-Relational mapping frameworks using the Java

Persistence API (JPA)

- Get an introduction to dependency injection using CDI

Specific Java 9 features that are covered in the course include:

- The Modular system (Project Jigsaw)
- Private methods in interfaces

## **Audience:**

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

## **Prerequisite:**

Attendees should have a working knowledge of developing OO software applications.

## **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

#### **Lesson: The Eclipse Paradigm**

- Workbench and Workspace
- Views
- Editors
- Perspectives
- Projects

### **Module 2: Getting Started with Java**

#### **Lesson: Writing a Simple Class**

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

### **Lesson: Adding Methods to the Class**

- Passing Parameters into Methods
- Returning a Value from a Method
- Overloaded Methods
- Constructors
- Optimizing Constructor Usage

## **Module 3: Essential Java Programming**

### **Lesson: Language Statements**

- Operators
- Comparison and Logical Operators
- Looping
- Continue and Break Statements
- The switch Statement
- The for-each() Loop

### **Lesson: Using Strings**

- Strings
- String Methods
- String Equality
- StringBuffer
- StringBuilder

### **Lesson: Specializing in a Subclass**

- Extending a Class
- Casting
- The Object Class
- Default Constructor
- Implicit Constructor Chaining

### **Lesson: Fields and Variables**

- Instance vs. Local Variables: Usage Differences
- Data Types
- Default Values
- Block Scoping Rules
- Final and Static Fields

- Static Methods

## **Lesson: Using Arrays**

- Arrays
- Accessing the Array
- Multidimensional Arrays
- Copying Arrays
- Variable Arguments

## **Lesson: Java Packages and Visibility**

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

## **Module 4: Object Oriented Development**

### **Lesson: Inheritance and Polymorphism**

- Polymorphism: The Subclasses
- Upcasting vs. Downcasting
- Calling Superclass Methods from Subclass
- The final Keyword

### **Lesson: Interfaces and Abstract Classes**

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

## **Module 5: Exception Handling**

### **Lesson: Introduction to exception handling**

- Exception Architecture
- Throwing Exceptions
- Checked vs. Unchecked Exceptions

### **Lesson: Exceptions**

- Creating Your Own Exceptions
- Handling Multiple Exceptions
- Automatic Closure of Resources

## Module 6: Java Developer's Toolbox

### Lesson: Utility Classes

- Wrapper Classes
- Autoboxing/Unboxing
- Working with Dates
- Enumeration Syntax
- Using Static Imports

### Lesson: Formatting Strings

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

## Module 7: Advanced Java Programming

### Lesson: Introduction to Generics

- Generics and Subtyping
- Bounded Wildcards
- Generic Methods

### Lesson: Lambda Expressions and Functional Interfaces

- Functional vs OO Programming
- Lambda Expression Syntax
- Functional Interfaces
- Type Inference
- Method references

## Module 8: Working with Collections

### Lesson: The Collection API

- Characterizing Collections
- Collection Interface Hierarchy
- The Set, List and Queue Interfaces
- Map Interfaces

### Lesson: Using Collections

- Collection Sorting
- Comparators
- Using the Right Collection

- Lambda expressions in Collections

## **Module 9: Stream API**

### **Lesson: Streams**

- Processing Collections of data
- Filtering and sorting collection data
- Find elements in Stream
- Numeric Streams
- Sources for using Streams

### **Lesson: Collectors**

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

## **Module 10: The Java Module system (Jigsaw)**

### **Lesson: Introduction to the Module System**

- Introduce Project Jigsaw
- Classpath and Encapsulation
- The JDK internal APIs
- Java 9 Platform modules
- Defining application modules
- Define module dependencies
- Implicit dependencies
- Implied Readability
- Exporting packages

## **Module 11: Accessing Resources**

### **Lesson: Java Data Access JDBC API**

- Connecting to the Database
- Statement and PreparedStatement
- ResultSet
- Executing Inserts, Updates, and Deletes
- Controlling Transactions and Concurrency

### **Lesson: Introduction to Annotations**

- Annotations Overview
- Working with Java Annotations

## **Lesson: Overview of JPA**

- Introduce the Java Persistence API (JPA)
- Benefits of Using an ORM framework
- Hibernate and JPA

## **Lesson: Introduction to CDI**

- Context Dependency Injection (CDI)
- The @Inject Annotation
- The @Default Annotation
- The @Alternative Annotation
- The @Named Annotation