

## Introduction to Java 9/10 Programming for Developers New to OO Programming (such as C, Mainframe, 4GL) (TT2120-J9/10)

Modality: Virtual Classroom

Duration: 5 Days

SATV Value:

CLC:

NATU:

SUBSCRIPTION: No

### About this course:

**Java 9 and OO Programming Essentials** 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 9, 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**.

*Note: Students with prior Object-Oriented exposure and background (such as C#, C++, Smalltalk, etc.) should consider the TT2100-J9 Java 9 Programming for Object Oriented (OO) Experienced Developers as an alternative.*

The average salary of a Java Developer is **\$90,992** per year.

### Course Objective:

- 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
- Work with the Java 9 modular system (Project Jigsaw)
- Learn to use Java exception handling features
- 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
- 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 prior practical programming experience in 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

#### **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: 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 4: 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 5: 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 6: 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 7: Java Developer's Toolbox**

### **Lesson: Utility Classes**

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

### **Lesson: Java Date/Time**

- The Date and Calendar classes
- Introduce the new Date/Time API
- LocalDate, LocalDateTime, etc.
- Formatting Dates
- Working with time zones
- Manipulate date/time values

### **Lesson: Formatting Strings**

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

## **Module 8: 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 9: 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 10: 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 11: 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 12: 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 (optional)**

- Annotations Overview
- Working with Java Annotations