

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

Modality: Virtual Classroom

Duration: 5 Days

SATV Value:

CLC:

NATU:

SUBSCRIPTION: No

### About this course:

**Java 8 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-J8 Java 8 Programming for Object Oriented (OO) Experienced Developers as an alternative.*

The average salary of a Java Developer is **\$97,000** 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 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
- Understand and use Lambda Expressions
- Understand and use the Stream API
- 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
- The new Date/Time API
- Lambda Expressions
- Method and Constructor references
- The Streams API
- Collectors
- The Optional class

## **Audience:**

This is an entry-level Java course, intended for experienced developers who are looking to become functional with Java, or who need to strengthen their Java coding practices with the best practices within Java.

## **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
  
- 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

