

# **Advanced Java 8 Programming (TT3100-J8)**

**Modality: Virtual Classroom**

**Duration: 4 Days**

## **About this Course:**

This intermediate-level 4 days training program is specifically designed for experienced Java Developers striving to master the art of programming in Java 8. This course provides a comprehensive review of advanced programming techniques and APIs. This course nurtures the programming skills of developers and is best for professionals striving to pursue a career as a Java Developer. On average, a professional Java Developer earns \$90,992 annually.

This course covers the key concepts and functionalities of Advanced Java and sheds light on the new features and performance tools in Java 8. Professionals get to learn the art of harnessing and applying these tools to augment the performance of web applications and programs. This course provides professionals with the practical experience of working in a Java environment and help them build familiarity with innovative coding techniques and programming practices.

## **Course Objectives:**

The core objective of this course is to help professionals develop a better understanding and sound knowledge of the following key concepts:

- Functional Programming Features, Applications, and Functionalities
- Java Performance Enhancement with Functional Interfaces & Lambda Expressions
- Working with Collectors and Collections using Stream Constructs
- Java 8 Collection API Methods and Applications
- Working with Default Methods in Java 8
- JavaScript Nashorn Engine Features and Functionalities
- Java Virtual Machines Application and Features
- Working with Dependency Injections in JEE and Java
- Java Bean Validation Applications and WebSockets Applications

## **Audience:**

This course is specifically tailored for the following group of professionals and interested candidates:

- Experienced Java 8 Developers
- Intermediate- Level Web Developers
- IT Professionals and Experts

## **Prerequisites:**

Professionals planning to enroll in the Advanced Java 8 Programming (TT3100-J8) Course must have practical experience of working with Java 8 Applications and familiarity with Java 8 Program

Development.

## **Course Outline:**

### **Module 1: Introduction to Lambda Expressions**

#### **Lesson: Functional Programming**

- Functional vs OO Programming
- Anonymous Inner-classes
- Utility Methods
- Lambda Expressions

#### **Lesson: Lambda Expressions and Functional Interface**

- Lambda Expression Syntax
- Functional Interfaces
- Type Inference in Java 8
- Method references

### **Module 2: Streams**

#### **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

#### **Lesson: Collectors**

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

### **Module 3: Additional Java 8 Enhancements**

#### **Lesson: Evolving Interfaces**

- Interfaces in Java 8
- Default methods
- Static methods
- Multiple Inheritance?

## **Lesson: Optional**

- Introduce Optional
- Implement Optional attributes
- Lambda expressions and Optional

## **Lesson: Java 8 Collection Updates**

- Introduce the ConcurrentHashMap
- Lambda expressions and Collections

## **Lesson: Java 8 Concurrency Updates**

- The common thread pool
- Atomic variables
- LongAdder and LongAccumulator
- CompletableFuture
- Non-blocking asynchronous tasks

## **Lesson: Nashorn JavaScript Engine**

- Working with JavaScript and Java
- Accessing Nashorn
- Executing JavaScript from Java
- Executing Java from JavaScript
- Implementing a Java Interface
- Alternatives to Nashorn

## **Module 4: Best Practices and Performance**

### **Lesson: Java Virtual Machine**

- Architecture of the JVM
- Application Execution

## **Module 5: Effective Java**

### **Lesson: Objects, Declarations, and Initializations**

- Abstraction and Responsibilities
- Low Coupling

- Programming principles
- Inheritance

## **Lesson: Exceptions**

- Proper use of Exceptions
- Managing state in exceptional situations
- Checked vs. Unchecked Exceptions

## **Module 6: Java 8 Performance**

### **Lesson: Performance and Java 8**

- Performance Across Java Versions
- Tiered Compilation
- Nashorn versus Java 7's JavaScript Engine
- Lambda Expressions versus Inner Classes
- Impact of Streams versus Collections
- Date/Time Classes
- I/O Stream Recommendations
- Encoding/Decoding
- Strings and String Interning

## **Module 7: Contexts and Dependency Injection (CDI)**

### **Lesson: Introduction to CDI**

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

### **Lesson: Using CDI**

- Qualifiers
- @PostConstruct and @PreDestroy
- The @Produces Annotation

## **Module 8: Java API for WebSocket**

### **Lesson: Introduction to WebSocket**

- Java API for WebSocket Overview
- Using WebSocket in Java EE

- Endpoint Instances

## **Lesson: Implementing WebSocket Endpoint**

- Annotated Endpoints
- Receiving messages
- Send Response to Client(s)
- JavaScript to Setup a WebSocket Connection

## **Module 9: Java Bean Validation (JSR 349)**

### **Lesson: Introduction to Bean Validation**

- Bean Validation
- Define Constraints on Object Models
- Core Validation Annotations
- Validate Objects and Object Graphs

### **Lesson: Bean Validation**

- Validate Parameters and Return Values
- Develop Custom Constraints

## **Module 10: Additional Topics (Optional)**

### **Lesson: Java I/O**

- The Java I/O Mechanism
- Subclasses Accessing Real Data
- Filter Classes
- New File IO - NIO
- NIO Overview

### **Lesson: Packaging Applications**

- JAR Files
- Versioning
- Sealing Packages
- Signing and Verification of JARs
- Enhancements for JAR processing

### **Lesson: Internationalization**

- Fundamentals of Localization
- Localizing Strings
- Localizing Numbers

- Localizing Dates