Document Generated: 12/15/2025

Learning Style: Virtual Classroom

Technology: Java

Difficulty: Intermediate

Course Duration: 2 Days

Transitioning to Java 8 (TT2108)

About this course:

The two-day course of Transitioning to Java 8 is a hands-on quick track course designed for developers that have a previous working understanding of the earlier Java versions. All through the course, applicants get the proficiency with great procedures for exploiting the new system of Java Module and also different new designs on this helpful update to the programming language of Java.

The normal pay of a Java Developer is $90,992 every year.

Course Objective:

Learners who attempt Transitioning to Java 8 will leave this course equipped with the essential abilities to develop workable code that scales efficiently into multi-core conditions. Planned for professionals presently working in Java 7 or 6 who requires to move their applications or aptitudes to Java 8, this course will instruct understudies all that they have to effectively master and execute the most recent benefits and features of Java 8 and become a dominant developer of Java 8.

Performing within in a hands-on learning and engaging environment that is directed by our professional team, participants will learn to:

- What does the mean of practical programming and what it carries to the table?
- How functional interfaces and Lambda expressions can greatly develop other aspects of Java.
- Explore Collection API’s new features.
- To utilize the new Stream builds to work with Collectors and Collections.
- Instructions to work with default procedures and pass procedures as arguments.
- The most effective method to utilize the new features to help concurrency in multi-code frameworks
- Function with the new Time/Date API and other new highlights.

Specific features of Java 8 that are covered in the course contain:

- Anonymous functions
- Functional programming
- Streams and Collectors
- Lambda expressions and default methods

- String joiner and repeating annotations
- New Date/Time API

Targeted Audience:

This Java developer course is an intermediate- level course, planned for experienced programmers of Java.

Prerequisite:

Participants ought to have working information on creating Java applications.

Course Outline:

Module 1: Introduction to Java 8

Lesson: Overview of Java 8 Features

- Overview of Java 8 features and enhancements
- Anonymous functions, Streams

Lesson: Evolving Interfaces

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

Exercise: Interfaces

Module 2: 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

Exercise: Using Lambda

Module 3: Collection Updates

Lesson: Java 8 Collection Updates

- Introduce the ConcurrentHashMap
- Lambda expressions and Collections

Exercise: Functional Collections

Module 4: 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

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 5: Additional Java 8 Enhancements

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

Lesson: Optional

- Introduce Optional

- Implement Optional attributes
- Lambda expressions and Optional

Exercise: Optional

Lesson: Other new Java 8 features

- StringJoiner
- Repeating and Type Annotations
- Parallel Array Sorting
- Improved type inference
- Method parameter reflection
- Updated command line tooling

Exercise: Reflection

Exercise: Annotation Processing

Lesson: Java 8 Concurrency Updates

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

Exercise: CompletableFuture

Lesson: Nashorn JavaScript Engine (Optional)

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

Exercise: Using JavaScript