

Document Generated: 07/13/2024

Learning Style: Virtual Classroom

Provider: Java

Difficulty: Intermediate

Course Duration: 2 Days

## Transitioning to Java 9 (TT2109)



## About this course:

The two-day course of Transitioning to Java 9 is a hands-on quick track course intended for engineers who have earlier working information on earlier Java versions. All through the course, understudies gain proficiency with the accepted procedures for exploiting the new system of Java Module and also other new structures on this great update to the programming language of Java.

The normal pay of a Java Developer is \$90,992 annually.

## Course Objective:

Applicants who attempt Transitioning to Java 9 will leave this course equipped with the necessary abilities to create a supportable code that scales effectively into multi-core situations. Intended for designers presently working in Java 8 who need to move their applications or aptitudes to Java 9, this course will instruct understudies all that they have to effectively ace and execute the most recent advantages and features of Java 9 and become a powerful developer of Java 9.

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

1. Comprehend the essentials of the Java language as well as its uses, importance, weaknesses, and strengths.
2. Create applications of modular in Java.
3. Move current Java applications to the platform of Java 9.
4. Use the tooling that is given in Java 9 to monitor, migrate, and optimize applications
5. Utilize the new tools of JShell to rapidly test the constructs of java.
6. Improve executions already utilizing the Stream API of Java 8 by using the approaches new in Java 9.
7. Understand how the String class implementation has been refreshed to diminish the impression of memory
8. Utilize the improvements made to the Concurrency API, functioning with CompletableFuture occasion of Thread pools.

Java 9 specific features that are discussed in the course comprise:

- The Java Module System (project Jigsaw)
- Updates to Collection and Stream API
- Updated try-with-resources
- JShell
- Performance enhancements in Java 9

## **Targeted Audience:**

This is a Java SE (JSE) developer course for intermediate level, intended for experienced developers of Java 8.

## **Prerequisite:**

Participants must have a working understanding of developing the applications of basic Java 8.

## **Course Outline:**

### **Module 1: Java 9**

#### **Lesson: Introduction to Java 9**

- Introduction to Java 9
- Java Community Process (JCP)
- JDK Enhancement Proposal (JEP)

#### **Lesson: New in Java 9**

- Introduce some of the 'smaller' Java 9 topics
- Java versioning
- The JDK/JRE file structure
- Deprecation
- The jdepscan tool
- Multi-Release JAR Files
- HTML 5 compliant JavaDoc

#### **Lesson: Milling Project Coin**

- Changes made to the language since Java 6
- Multi-catch
- Using effectively final variables in try-with-resources
- Suppressed Exceptions
- Binary literals
- Reserved underscore (Java 9)
- Type inference in anonymous classes (Java 9)
- @SafeVargs (updates in Java 9)
- Default and static methods in interfaces (Java 8)
- Private methods in interfaces (Java 9)

### **Lesson: Collection and Stream Updates**

- Factory methods for Immutable Collection types
- takeWhile and dropWhile methods
- The Stream Iterate and ofNullable methods

## **Module 2: The Java Module system (Jigsaw)**

### **Lesson: Why JigSaw?**

- Problems with Classpath
- Encapsulation and the public access modifier
- Application memory footprint
- Java 8's compact profiles
- Using internal JDK APIs

### **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

### **Lesson: The Module Descriptor**

- Define module requirements
- Explain qualified exports
- Open modules for reflection
- Use ServiceLoader
- This provides and uses keywords

### **Lesson: Working with Modules**

- Being backwards compatible
- The ModulePath and ClassPath
- Unnamed Modules
- Automatic Modules
- The JLink tool

## **Module ?3: JShell**

### **Lesson: JShell**

- Introduction to JShell
- Running Expressions in JShell
- Importing packages
- Defining methods and types
- Using the JShell editor
- Save and loading state

## **Module 4: More Java 9**

### **Lesson: Concurrency Updates in Java 9**

- Brief overview of Concurrency in Java
- Overview of CompletableFuture
- Subclassing the CompletableFuture
- The default Executor
- New Factory methods
- Dealing with time-outs

### **Lesson: Other New Java 9 Features**

- Enhancements on the Optional class
- Improvements made in the Process API
- The Stack-Walking API
- The HTTP2 Client
- The Multi-Resolution API
- The Reactive Flow API

### **Lesson: Performance Optimizations**

- Performance in Java 9
- Compact Strings
- String deduplication
- Ahead-Of-Time Compilation
- Hotspot Diagnostic commands
- The G1 Garbage collector
- Variable and Method Handles

### **Lesson: Memory Management**

- Understand memory management in Java
- Discuss the various garbage collectors
- The Garbage-First (G1) Garbage Collector

## Credly Badge:



### **Display your Completion Badge And Get The Recognition You Deserve.**

Add a completion and readiness badge to your LinkedIn profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

[Find Out More](#) or [See List Of Badges](#)