

## Course duration

- 2 days

## Course Benefits

- Understand the new Java release cycle and Long Term Support (LTS) releases
- Be familiar with significant deprecated and removed features, and how to work around them
- Create and use Java modules, understanding module descriptors, modular JARs, exports and dependencies, and the modulepath
- Understand the structure and behavior of the modular JDK, how it supports modular applications as well as legacy classpath-based code, and the implications of strong encapsulation on each
- Migrate classpath-based applications to Java 11, understanding the stages of migration and options available
- Recognize the issues with 3rd party libraries in a migration effort, and how to work with them on the modulepath and classpath
- Use local-variable type inference with var, including lambda parameters
- Gain a practical working knowledge of the JShell REPL tool, including working with code snippets and variables, configuration, and using external libraries
- Use the HTTP Client to access HTTP resources from Java, as well as understand the other options available
- Work with various HTTP request and response types, using both synchronous and asynchronous techniques
- Outline the new factory methods in Java Collections and understand native immutable collections
- Describe the motivation for multi-release JAR files (MR-JARs), understand their structure, and how to create them
- Understand the runtime behavior of MR-JARs in both legacy and modern JVMs, and principles and strategies for working with them effectively
- Outline the principles of Reactive Programming and how it differs from traditional synchronous invocation models
- Describe Reactive Streams and the role of the Flow API that defines the Java platform's support for them
- Understand the characteristics and benefits of custom runtime images
- Use jdeps to analyze application dependencies, and create custom runtimes with jlink, for both modular and classpath-based applications
- Explore some of the more important additional features and APIs, including new features for interfaces, the Process API, new JDK tools and command line options

## Available Delivery Methods

## Public Class

Public expert-led online training from the convenience of your home, office or anywhere with an internet connection. Guaranteed to run .

## Private Class

Private classes are delivered for groups at your offices or a location of your choice.

## Course Outline

1. Introduction to Modules
  1. Motivation and Overview
  2. Types of Modules
  3. Modular JDK
  4. Our Approach
2. Working with Java 9 Modules
  1. Defining and Using Modules
  2. Services
  3. Compatibility and Migration
  4. Conclusion
3. Type Inference
  1. Local-Variable Type Inference
  2. Brief Overview of Lambdas
  3. Local-Variable Syntax for Lambdas
4. JShell
  1. Introduction to JShell
  2. Working with Code
  3. Using Libraries (Modules, Jars, etc.)
5. Http Client
  1. Overview
  2. API
  3. Usage and Features
6. Other New Features
  1. Collection Factory Methods
  2. Multi-Release JARs
  3. Reactive Programming
  4. Miscellaneous
  5. Upcoming Features in Java 12+
7. Custom Runtime Images
  1. Application-Specific Runtimes
  2. Benefits
  3. Creating Runtime Images with jlink

## Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

### Class Prerequisites

Experience in the following *is required* for this Java class:

- Working knowledge of Java programming, including use of inheritance, interfaces, and exceptions