

**New Features in Java 11**

**Course Number:** JAV-414  
**Duration:** 2 days

**Overview**

Accelebrate's New Features in Java training course walks attendees through the important new features through Java 11 LTS and provides an overview of the new features in Java 12+ that are also likely to appear in Java 17 (the next LTS release). Students learn how to leverage the new Java Platform Module System (JPMS), which presents a fundamental shift in how applications are organized and interconnected to the libraries they use. Participants learn how to use modules and explore migration from non-modular applications.

**Prerequisites**

All Java training students be comfortable with writing general Java code at an intermediate level.

**Materials**

Attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

* A recent version of Windows, macOS, or Linux with at least 8 GB RAM
* JDK 11 or later
* Eclipse or IntelliJ IDEA
* Other free software - please contact us if you have purchased this class

**Objectives**

* 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

**Outline**

* Introduction
* Java State of the Union
  + New Release Cycle
  + What's In
  + What's Out
  + The Move to Java Modules
* Introduction to Modules
  + Motivation and Overview
  + Types of Modules
  + Modular JDK
  + Our Approach
* Working with Java 9 Modules
  + Defining and Using Modules
  + Services
  + Compatibility and Migration
  + Conclusion
* Type Inference
  + Local-Variable Type Inference
  + Brief Overview of Lambdas
  + Local-Variable Syntax for Lambdas
* JShell
  + Introduction to JShell
  + Working with Code
  + Using Libraries (Modules, Jars, etc.)
* Http Client
  + Overview
  + API
  + Usage and Features
  + - Session 6: Other New Features
  + Collection Factory Methods
  + Multi-Release JARs
  + Reactive Programming
  + Miscellaneous
  + Upcoming Features in Java 12+
* Custom Runtime Images
  + Application-Specific Runtimes
  + Benefits
  + Creating Runtime Images with jlink
* Conclusion