

Learning Modern Java

Introduction

Java, introduced in 1996, has seen constant evolution with new features and better performance. Initially, releases were infrequent, allowing developers to specialize in existing features, often missing out on later ones. As a result, catching up with new features across multiple Java versions can be challenging for busy developers.

This one-day course will provide students with an updated understanding of using the newer features of Java found in Java 8 through Java 17. The goal of this course is to provide students with an overview of the basic features introduced by version as well as hands-on experience working with these features. As such, the modules will include an introduction to the concepts, demonstrations of the features at work and then an opportunity for students to put the concepts into practice. Students will leave this course with the ability to pragmatically utilize the new features found in the recent Java releases.

Course objectives

- Be able to use key features of Java 8, including Lambdas, Functional Interfaces, Collection enhancements using Lambdas, Java Streams API, and Java Interface enhancements such as default and static methods.
- Be able to use Java 9 features, such as Modules for project file management, Java Shell, and Reactive Streams
- Be able to use Java 10 and 11 features, such as Local Variable type inference (the var type), running Java code without requiring a 'javac' compile, and additional String methods
- Be able to use the Java 17 features such as Switch syntax improvements, sealed classes, Records, and TextBlocks.
- Explore the features recently released in Java 21, such as Virtual Threads, Record patterns, pattern matching for switch and more

Target Audience

- Java developers wanting to update their Java skills to leverage newer features of Java
- Architects wanting to understand newer Java features and how they may fit into a solution

Prerequisites

- Experience with basic Java concepts, such as Java Collections, Interfaces, etc
- Experience with IDEs, such as IntelliJ, Eclipse, or Visual Studio Code

Course Modules

Overview of features by release version

- Java 8 enhancements
- Java 9 & 10 enhancements
- Java 11 enhancements
- Java 17 enhancements
- The new release model and LTS
- Garbage collectors

Java 8: Lambdas and Functional Interfaces

- Functional interfaces
- Writing Lambda functions
- Invoking Lambda functions
- Collection enhancements that use Lambdas
- Default and Static methods on interfaces and Method references
- **Lab:** Implement search, sort, and formatting using functional interface and Lambda expressions

Java 8: Java Streams API and other additions

- Method references
- Optional
- Java Streams API
- Chaining and composing Lambdas
- Java Time enhancements
- **Lab:** Enhancing prior lab to use streams API and chaining.

Java 9-11: Reactive Streams, HttpClient and More

- Reactive Streams
- Additional String methods
- Local variable type inference
- Java Shell
- HTTP Client
- Modules
- **Lab:** Handling remote requests using HttpClient and Asynchronous requests

Java 12-17: Records, Text Blocks and more

- Java Records
- Textblocks
- Switch syntax improvement
- Sealed classes
- **Lab:** Utilizing Records, TextBlocks and Switch enhancements for clean code

Java 18-21: Catching up to today

- Simple Web Server
- Record Patterns
- Pattern matching for Switch
- Virtual threads
- Scoped Values (preview)
- String templates (preview)