Intermediate Java Programming | Next-Level Java Skills - TT2211

Explore Modular System, Jigsaw, Concurrency, JShell, JDBC, Performance Optimization, CDI, JPA, Project Lombok & More

Duration: 3 Days
Skill Level: Intermediate
Available Format: Instructor-Led Online ; On Public Schedule

Intermediate Java Programming is hands-on fast-track course geared for experienced developers who have prior working of basic Java who want to take advantage of the newest features of Java 11 that can help them improve performance and functionality of their Java applications.

What You'll Learn

Overview

Intermediate Java Programming is hands-on fast-track course geared for experienced developers who have prior working of basic Java who want to take advantage of the newest features of Java 11 that can help them improve performance and functionality of their Java applications. Students will explore and learn how to leverage Modules, scale applications into multi-core environments, improve performance, and become a more effective Java developer.

Objectives

This 'skills-centric' course is about 50% hands-on, designed to train attendees in advanced development skills, coupling the most current, effective techniques with the soundest industry practices. Students will leave this course armed with the required skills to improve their Java applications using sound coding techniques and best practices.

Working in a hands-on learning environment, guided by our expert team, attendees will learn to:

- Develop modular applications in Java
- Explore the Module service loader
- Utilize the tooling that is provided in Java 11 to migrate, monitor and optimize applications
- Use the new JShell tool to quickly test java constructs
- Develop multi-threaded applications
- Work with the CompletableFuture instances introduced in Java 8
- Use JDBC to read, write and update records in a relational database
- Use the HTTP Client API introduced in Java 11
- Explore the Dependency Injection (CDI) and Persistence (JPA) API
- Apply the Introspection and Reflection APIs
- Understand the importance of Reference Objects
- Utilize Project Lombok

Audience

This in an intermediate-level Java development course geared for students experienced with Java 8 or higher programming essentials who wish to quickly get up and running with advanced Java skills.

This course does not cover Java programming fundamentals.

Pre-Requisites

This in an intermediate-level Java development course geared for students experienced with Java 8 or higher programming essentials who wish to quickly get up and running with advanced Java skills. This course does not cover Java programming fundamentals.

TT2100	Core Java Programming Developer's Workshop
TT2120	Basic Java Programming for Developers New to OO (C, COBOL, etc.)
TT2104	Fast Track to Core Java Programming for OO Experienced Developers
TT2000	Getting Started with Programming, OO & Java Basics for Non-Developers

Agenda

Session: The Java Module system (Jigsaw)

Lesson: Why Jigsaw?

- Problems with Classpath
- Encapsulation and the public access modifier
- Application memory footprint
- Java 8 compact profile
- 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
- Lab: Defining Modules

Lesson: The Module Descriptor

- Define module requirements
- Explain qualified exports
- Open modules for reflection
- Use ServiceLoader
- The provides and uses keywords
- Lab: Modules and the ServiceLoader
- Lab: Using Reflection on modules

Lesson: Working with Modules

- Being backwards compatible
- The ModulePath and ClassPath
- Unnamed Modules
- Automatic Modules
- The JLink tool
- Lab: Migrating to modules

Session: JShell

Lesson: JShell

- Introduction to JShell

- Running Expressions in JShell
- Importing packages
- Defining methods and types
- Using the JShell editor
- Save and loading state
- Lab: Working With JShell

Session: Accessing Resources

Lesson: Java Data Access JDBC API

- Connecting to a database using JDBC
- Executing a statement against a database that returns a ResultSet
- Setting up and working with PreparedStatements
- Extracting multiple rows of data from a ResultSet
- Inserting, updating and deleting rows in a table
- Tutorial: Simple Maven Setup with Eclipse
- Lab: Intro to JDBC

Lesson: Introduction to Annotations

- Discussing how annotations work in Java
- Understanding what is required to work with Java's annotations
- Using annotations
- Other technologies that are using annotations

Lesson: Introduction to CDI

- Understand the value of CDI
- Explore dependency injection (DI)
- Understand alternatives
- Understand annotation processing
- Use and configure CDI
- Lab: Introduction to CDI
- Lab: Adding CDI Qualifiers (optional)

Lesson: Overview of JPA

- Discuss Object to Relational (O/R) Mapping (ORM)
- Explore the Java Persistence API (JPA)
- Explain the ORM framework configuration
- Map a 'simple' entity to a database table
- Examine how to read, write and search for entities

- Lab: Introduction to JPA

Session: Multithreading and Concurrency

Lesson: Introduction to Multithreading and Concurrency

- Principles of Multithreading
- The Thread class and Runnable interface
- Explore thread synchronization
- Introduce the Java Concurrency API
- Lab: MultiThreading
- Lab: Futures

Lesson: Concurrent Java

- Thread management using Executors
- The common thread-pool
- Submitting and controlling asynchronous tasks
- Explore the locking API
- Lab: Working with Concurrent Java
- Lab: CompletableFuture

Lesson: Non-blocking asynchronous tasks

- The CompletableFuture
- Define non-blocking processes
- Exception handling in multithreaded processes
- The Fork-Join framework
- Lab: ForkJoin
- Lab: CompletionStage (optional)

Session: HTTP Client API

Lesson: The HTTP Client API

- Making HTTP (Hypertext Transfer Protocol) requests
- Explain Incubator Modules
- HTTP2 Client API
- Introduce WebSockets
- Communicate with WebSocket endpoints

- Lab: HTTP Clients

Session: More Java

Lesson: Other New Java Features

- Enhancements on the Optional class
- Improvements made in the Process API
- The Stack-Walking API
- The HTTP2 Client
- The Multi-Resolution API
- Lab: Working with Native processes

Lesson: Performance Optimizations

- Ahead-Of-Time Compilation
- Hotspot Diagnostic commands
- Variable and Method Handles
- Lab: JIT Compiler

Lesson: Memory Management

- Understand memory management in Java
- Discuss the various garbage collectors
- The Garbage-First (G1) Garbage Collector
- The No-Op and ZGS Garbage Collectors

Session: Reflection and References

Lesson: Reference Objects

- List the kinds of object references available in Java
- Introduce Weak, Soft and PhantomReference
- Explain the ReferenceQueue
- Lab: Reference Objects

Bonus Topics: Time Permitting

Lesson: Introspection and Reflection

- Reflection classes
- Introspection
- Dynamic invocation of methods
- Using annotations
- Type annotations

- Receiver parameter
- Lab: Introspection and Reflection

Lesson: Project Lombok

- Introduce the Lombok Project
- Configure the Lombok Annotation processor
- Introduce some of the commonly used Lombok annotations
- Lab: Project Lombok

Student Materials & Lab Environment

All course software (limited versions, for course use only), digital courseware files or course notes, labs / data sets and solutions (as applicable) are provided for you in our "easy access / no install required" high-speed remote lab environment. Our tech team works with every student to ensure everyone is set up with working access and ready to go prior to every course start date, ensuring a smooth delivery and great hands-on experience. Please ask for details.

For More Information

Please contact us or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.