

Advanced Java: Concurrency, JUnit & Spring Boot

Learn to apply Java concurrency, memory management, JUnit and Mockito testing, and Spring Boot to build robust applications.

26

MODULES

115

TOPICS

COURSE OVERVIEW

This course begins with a refresher on core Java syntax and object-oriented concepts, then compares Java with C++ before covering memory management and garbage collection tuning. It moves into threads and concurrency, the Java Collections Framework, generics, and development tools, alongside JVM internals, code review practices and performance profiling. It then addresses unit testing and mocking with JUnit and Mockito, including parameterized tests, lifecycle annotations and argument matchers. It closes with Spring Boot, covering project setup, dependency injection, REST APIs, data persistence, security, and asynchronous and event-driven techniques.

WHO THIS COURSE IS FOR

Java Developers

Backend Developers

QA Engineers

Software Engineers

Spring Boot Developers

LEARNING OBJECTIVES

- Explain core Java syntax and object-oriented concepts
- Compare Java and C++ regarding memory management and data structures
- Analyse Java memory management and garbage collection tuning options
- Apply thread creation, synchronization and concurrency best practices
- Implement Java Collections such as List, Set, Map and Queue
- Apply generics and stream-based filtering to collection operations
- Develop unit tests and mocks using JUnit and Mockito
- Configure Spring Boot projects with dependency injection and REST APIs

26 Modules, 115 Topics

01

Basics of Java Programming (Brief Refresher)

TOPICS

- Overview of the Java Ecosystem
- Basic Syntax and Data Types
- **Control Structures:** If-Else, Loops, Switch
- Functions and Methods in Java
- Classes, Objects, and Constructors
- Inheritance and Polymorphism
- Encapsulation and Abstraction
- Interfaces and Abstract Classes

02

Comparison Between C++ and Java

TOPICS

- Key Differences Between C++ and Java
- **Memory Management:** Manual vs. Automatic
- **Object-Oriented Programming:** Similarities and Differences
- Comparing Data Structures in C++ and Java

03

Memory Management and Garbage Collection

TOPICS

- Basics of Java Memory Management
- **Java's Memory Model:** Heap and Stack
- Components of the JVM Memory
- Garbage Collection in Java
- Introduction to Garbage Collection
- **Types of Garbage Collectors:** Serial, Parallel, CMS, G1
- Garbage Collection Algorithms
- How Garbage Collection Works
- Generational Garbage Collection
- Tuning Garbage Collection
- Understanding Garbage Collection Metrics
- JVM Flags for GC Tuning
- Analyzing Garbage Collection Logs

04

Threads and Concurrency in Java

TOPICS

- Basics of Java Threads
- What is a Thread?
- **Creating Threads:** Thread Class vs. Runnable Interface
- Thread Lifecycle and Management
- **Thread States:** New, Runnable, Blocked, Waiting, Timed Waiting, Terminated
- Starting and Joining Threads
- Understanding Multithreading
- Creating and Managing Threads
- Synchronization and Locks
- Concurrency Best Practices

05

Java Collections Framework

TOPICS

- Overview and Importance of Collections
- **Core Interfaces:** List, Set, Map, Queue
- **Implementations:** ArrayList, LinkedList
- When to Use Each Implementation
- List Iteration Techniques
- **Implementations:** HashSet, LinkedHashSet, TreeSet
- Characteristics of Different Set Implementations
- Choosing the Right Set for Your Use Case
- **Implementations:** HashMap, LinkedHashMap, TreeMap
- Keys and Values in Maps
- Iteration Over Maps
- **Implementations:** LinkedList, PriorityQueue, ArrayDeque
- Differences Between FIFO and Priority-Based Queues
- Use Cases for Different Queue Types

06

Common Collection Operations

TOPICS

Sorting and Searching within Collections · Filtering and Transforming with Streams · Using Comparator and Comparable Interfaces

07

Generics in Collections

TOPICS

Understanding Generics in Java · Benefits of Using Generics with Collections · Common Generic Patterns and Restrictions

08

Tools for Java Development

TOPICS

- **Integrated Development Environments (IDEs):** IntelliJ, Eclipse, NetBeans, VSCode
- **Build Tools:** Maven and Gradle
- **Version Control Systems:** Git
- **Other Useful Tools** (e.g., Static Analysis Tools, Profiling Tools)

09

JVM Internals

TOPICS

Java Virtual Machine (JVM) Basics · Class Loading and Bytecode · Just-In-Time (JIT) Compilation · JVM Flags and Tuning

10

Code Review and Best Practices

TOPICS

Importance of Code Reviews · Conducting Effective Code Reviews · Common Code Review Mistakes to Avoid · Tools for Code Review and Collaboration

11

Introduction to JUnit

TOPICS

Overview of Unit Testing · Setting Up JUnit in a Java Project

12

Writing Unit Tests with JUnit

TOPICS

- **Creating Test Classes and Test Methods**
- **Using Annotations:** @Test, @BeforeEach, @AfterEach
- **Assertions in JUnit:** assertEquals, assertTrue, etc.

13

Testing Techniques with JUnit

TOPICS

- **Parameterized Tests**
- **Exception Testing with** @Test(expected = Exception.class)
- **Nested Tests with** @Nested

14

Advanced JUnit Concepts

TOPICS

- Lifecycle Management with `@BeforeAll` and `@AfterAll`
- Using Test Suites to Group Tests
- Conditional Test Execution with `@EnabledIf` and Similar Annotations

15

Introduction to Mockito

TOPICS

What is Mockito? · Importance of Mocking in Unit Testing · Setting Up Mockito in a Java Project

16

Mocking with Mockito

TOPICS

Creating Mocks with `Mockito.mock()` · Mocking Method Calls with `Mockito.when()` · Using `Mockito.verify()` to Verify Interactions

17

Mockito Advanced Techniques

TOPICS

- Argument Matchers with `Mockito.any()`, `Mockito.eq()`, etc.
- Mocking Static Methods with `Mockito.mockStatic()`
- Spying on Real Objects with `Mockito.spy()`

18

Performance Tuning and Profiling in Java

TOPICS

Identifying Performance Bottlenecks · Profiling Tools and Techniques · Basic Performance Tuning Strategies · JVM Flags for Performance Optimization

19

Introduction to Spring Boot

TOPICS

What is Spring Boot? · History and Evolution of Spring Framework · Benefits of Using Spring Boot

20

Setting Up a Spring Boot Project

TOPICS

Creating a Spring Boot Project · Maven and Gradle for Spring Boot Projects · Project Structure and Configuration

21

Core Spring Boot Components

TOPICS

- Main Application Class and @SpringBootApplication
- Auto-Configuration and Spring Boot Starters
- Application Properties and Configuration

22

Dependency Injection and Beans

TOPICS

- Basics of Dependency Injection
- Defining Beans with @Bean and Component Scanning
- **Scope of Beans:** Singleton, Prototype, etc.

23

Spring Boot with RESTful APIs

TOPICS

- Creating REST Controllers with @RestController
- **Handling HTTP Requests:** @GetMapping, @PostMapping, etc.
- Data Serialization and Deserialization with Jackson

24

Data Persistence in Spring Boot

TOPICS

- Introduction to Spring Data JPA
- Configuring Data Sources and Entity Relationships
- Working with Repositories and Custom Queries

25

Security in Spring Boot

TOPICS

- Introduction to Spring Security
- Configuring Basic Security Settings
- Authentication and Authorization with Spring Security

Advanced Spring Boot Topics

TOPICS

- Asynchronous Processing with @Async
- Building Event-Driven Applications with Spring Boot
- Using Spring Boot with Microservices Architectures

WHAT IS INCLUDED

Everything You Need to Succeed

- Koenig course material for every module
- Delivery by certified instructors and subject matter experts
- Hands-on exercises throughout the course
- Course completion certificate from Koenig

ABOUT KOENIG

Who We Are

Established in 1993, Koenig Solutions is a globally recognised training provider with 35+ technology partnerships, 5,000+ training programmes and a network of certified instructors and subject matter experts, supporting individuals and enterprise teams worldwide.

FLEXIBLE DELIVERY

Learn the Way That Suits You

[Classroom Training](#)[Live Online Classes \(ILO\)](#)[Fly-Me-A-Trainer \(FMAT\)](#)[Flexi \(Self-Paced\)](#)[1-on-1 Training](#)

Ready to Get Started?

Talk to a learning advisor about schedules, group training and customised delivery.

[Visit koenig-solutions.com](https://koenig-solutions.com)[✉ info@koenig-solutions.com](mailto:info@koenig-solutions.com)