Advanced Java Development with Spring Boot, Microservices, and Projects

Duration: 15 days
Prerequisites: Working Knowledge of Java Programming

Day 1 – Core Java Intermediate Essentials

- JVM architecture essentials (runtime areas, JIT, garbage collection basics)
- Intermediate data types & operations (wrapper classes, type casting, operator nuances)
- Control flow optimization (enhanced switch, labeled loops)
- Method overloading, varargs, recursion best practices
- Arrays & Strings (StringBuilder, StringBuffer, common performance tips)

Lab:

- Utility library for string & array manipulation
- Build recursive & iterative algorithm implementations

Day 2 – OOP Intermediate Concepts

- Encapsulation & constructors (chaining, best practices)
- Inheritance, method overriding, dynamic binding
- Interfaces (default, static methods, multiple inheritance)
- Nested classes & enums with logic

Lab:

- Role-based access system using inheritance/interfaces
- Enum-based workflow automation

Day 3 – Exception Handling Mastery

- Exception hierarchy recap (Throwable, Exception, Error)
- Checked vs unchecked exceptions in API design
- Advanced try-with-resources (AutoCloseable)
- Multi-catch & suppressed exceptions
- Custom exception hierarchies for domain-specific logic
- Exception chaining & wrapping
- Logging best practices with SLF4J / Logback
- Using CompletableFuture.exceptionally() for async handling

Lab:

- File I/O with custom exception handling & logging
- Async API call simulation with exception propagation

Day 4 – Collections Framework Deep Dive

- Collection hierarchy & interfaces in depth
- Choosing correct collection type for performance
- Advanced List (CopyOnWriteArrayList, Vector)
- Advanced Set (EnumSet, ConcurrentSkipListSet)
- Map internals (HashMap vs ConcurrentHashMap)
- Sorting with comparator chains
- Collections utility methods (unmodifiable, synchronized, disjoint)

Lab:

- Data deduplication & sorting for customer dataset
- Build a thread-safe shared cache using ConcurrentHashMap

Day 5 – Advanced Collections & Functional Programming

- Concurrent queues (PriorityBlockingQueue, LinkedBlockingDeque)
- NavigableMap & NavigableSet usage
- Deep dive into Streams API (flatMap, peek, parallel streams)
- Grouping & partitioning collectors
- Custom collectors for aggregation
- Functional interfaces in real-world scenarios
- Advanced Optional usage (filter, map, flatMap)

Lab:

- Report generator using custom collectors
- Null-safe product search using Optional & Streams

Day 6 – Multithreading Fundamentals

- Thread lifecycle & scheduling algorithms
- Thread groups & daemon threads
- Synchronization blocks & synchronized methods
- Volatile keyword & visibility problems
- ThreadLocal for request-scoped data

Lab:

- Multi-threaded simulation of ticket booking system
- ThreadLocal-based request tracking in concurrent environment

---

Day 7 – Advanced Multithreading & Concurrency

- Lock API & ReentrantLock
- ReadWriteLock for performance optimization
- Deadlock detection & prevention strategies
- Thread-safe collections vs sync wrappers
- Executor framework & tuning

- CompletableFuture async pipelines with timeouts
- ForkJoinPool for parallel computation

Lab:
- Parallel data processing with ForkJoinPool
- Deadlock detection & resolution simulation

Day 8 – Git & Build Tools (Maven / Gradle)

- Advanced Git branching (Git Flow, Trunk Based)
- Git hooks for automation
- Maven plugin system & custom goals
- Gradle multi-module project setup
- Dependency scope management
- Managing transitive dependency conflicts

Lab:
- Build multi-module Maven project and resolve dependency conflict
- Automate version bumping using Git hooks

Day 9 – Spring Boot Fundamentals

- Core Spring concepts (Beans, Context, Profiles)
- Dependency injection patterns
- Conditional beans & profiles
- Configuration properties mapping & validation
- REST controllers with custom media types
- Spring Boot DevTools & hot reload

Lab:
- REST API with dynamic configuration switching via profiles
- Externalized configuration using YAML & validation annotations

---

Day 10 – Spring Data JPA

- Entity mapping: inheritance strategies, embeddables
- One-to-one, one-to-many, many-to-many mapping
- Cascade types & fetch strategies
- Query derivation & JPQL
- PagingAndSortingRepository usage
- Criteria API for dynamic queries
- Locking strategies

Lab:
- Inventory Management API with paging & sorting

- Complex search filter using Criteria API

Day 11 – Microservices with RabbitMQ

- Event-driven architecture principles
- Message queues vs topics
- RabbitMQ exchange types (direct, topic, fanout)
- Publisher confirms & message acknowledgments
- Dead-letter queues & retry patterns
- Microservice communication design
- Spring Boot integration with RabbitMQ

Lab:

- Order placement microservice with async confirmation via RabbitMQ
- Dead-letter queue handling for failed messages

Day 12 – REST API Security, Async, CI/CD & Performance Tuning

- REST API design best practices
- JWT authentication & refresh tokens
- Role-based access control
- Asynchronous API calls with WebClient
- Scheduling jobs with Spring
- Debugging techniques with IDE & remote debugging
- JVM monitoring & profiling tools
- CI/CD with GitHub Actions / Jenkins
- API performance optimization techniques

Lab:

- Secure async API with JWT authentication
- Performance profiling & tuning of a Spring Boot API

Project Work (Day 13–15)

Project 1 – Employee Task Management System

Focus: Spring Boot, JPA, RabbitMQ, Async, JWT, CI/CD

- CRUD APIs for employees & tasks
- Role-based authentication & JWT security
- Async task notifications via RabbitMQ
- CI/CD pipeline for deployment

Project 2 – Online Course Enrollment Platform

Focus: Collections, JPA, Exception Handling, Performance Tuning

- Course & student registration APIs
- Advanced search & filtering using Streams
- Exception handling for enrollment conflicts
- Optimize queries with pagination

Project 3 – Inventory & Order Management Microservice

Focus: Microservices, RabbitMQ, Asynchronous Processing

- Inventory microservice with stock management
- Order microservice communicating via RabbitMQ
- Async restocking alerts
- API gateway & service registry

Project 4 – Real-time Event Booking System

Focus: Multithreading, Async API calls, Debugging, Performance

- Event & booking management APIs
- Async ticket confirmation emails
- Thread pool for bulk booking processing
- Debug & optimize slow endpoints