Java Programming: Beginner to Expert Course

OA25JP01

Course Description

Master Java programming in this all-in-one course, designed to take you from beginner to expert. Covering core Java, object-oriented programming, web development with Spring, and advanced topics like multithreading and microservices, this course is perfect for Java 17+ (LTS) and compatible with tools used in 2025. The program bundles four levels of learning: Beginner, Intermediate, Advanced, and Expert with Projects.

What You'll Learn

- Master Java syntax and object-oriented programming
- Build robust applications with Java and Spring
- Work with databases, APIs, and modern frameworks
- Apply Java to real-world projects, including web apps and microservices
- Deploy Java applications and optimize performance

Course Requirements

- No prior programming knowledge required
- A computer with Java Development Kit (JDK 17+), IntelliJ IDEA, and Maven installed

Who This Course Is For

- Beginners starting their programming journey
- Intermediate developers looking to master Java
- Professionals aiming to build enterprise-grade applications with Java

Syllabus (35 Sessions)

1. **Java Fundamentals**
   - Installing JDK and IntelliJ IDEA
   - Introduction to Java: Syntax, variables, and data types
   - Running Java programs (main method, compilation)
   - Common IDE shortcuts and debugging
2. **Basic Programming Concepts**
   - Working with primitive types (int, double, boolean)
   - Operators: Arithmetic, relational, logical
   - Input/output with Scanner
   - Writing simple Java programs

3. **Control Flow**
    - Conditional statements: if, else if, switch
    - Loops: for, while, do-while
    - Break, continue, and labeled statements
    - Handling user input with conditionals
4. **Methods in Java**
    - Defining and calling methods
    - Parameters, return types, and overloading
    - Method scope and visibility
    - Recursion basics
5. **Object-Oriented Programming (OOP) Basics**
    - Classes, objects, and constructors
    - Instance variables and methods
    - Encapsulation with getters and setters
    - The this keyword
6. **OOP Advanced Concepts**
    - Inheritance and polymorphism
    - Abstract classes and interfaces
    - Method overriding and super
    - Packages and access modifiers
7. **Java Collections Framework**
    - Introduction to List, Set, and Map
    - Using ArrayList, HashSet, and HashMap
    - Iterating collections with for-each and Iterator
    - Sorting and searching collections
8. **Exception Handling**
    - Try-catch blocks and multiple catch
    - Throwing exceptions with throw
    - Custom exceptions
    - The finally block and try-with-resources
9. **Practice Session**
    - Build a simple inventory management program
    - Congratulations! You’ve built a Java foundation.
10. **Java Strings and Text Processing**
    - Working with String and StringBuilder
    - String methods: substring, replace, split
    - Regular expressions with Pattern and Matcher
    - Parsing and formatting text
11. **File I/O**
    - Reading and writing files with File and Files
    - Handling CSV files with BufferedReader
    - Serialization and deserialization
    - Path handling with java.nio.file
12. **Java Generics**
    - Introduction to generics

- Generic classes and methods
- Bounded types and wildcards
- Using generics with collections

13. **Java Streams API**
    - Introduction to functional programming
    - Stream operations: filter, map, reduce
    - Working with Optional
    - Parallel streams for performance
14. **Database Connectivity with JDBC**
    - Setting up JDBC and PostgreSQL
    - Performing CRUD operations
    - Prepared statements and batch processing
    - Handling database exceptions
15. **Introduction to Spring Boot**
    - Setting up a Spring Boot project with Maven
    - Creating RESTful APIs
    - Dependency injection with Spring
    - Configuring Spring Boot applications
16. **Practice Session**
    - Build a simple REST API with Spring Boot
    - Congratulations! You’re an intermediate Java developer.
17. **Advanced OOP**
    - Design patterns: Singleton, Factory, Observer
    - Composition vs. inheritance
    - SOLID principles
    - Building reusable components
18. **Multithreading**
    - Introduction to threads and Runnable
    - Thread synchronization with synchronized
    - Using ExecutorService and thread pools
    - Handling concurrency issues
19. **Spring Data JPA**
    - Setting up Spring Data JPA
    - Creating entities and repositories
    - Performing complex queries
    - Managing relationships (one-to-many, many-to-many)
20. **Spring Security**
    - Implementing JWT authentication
    - User authentication and authorization
    - Securing REST endpoints
    - Role-based access control
21. **Working with APIs**
    - Consuming REST APIs with RestTemplate
    - Parsing JSON with Jackson
    - Handling API errors

  - Building a client for external APIs

22. **Java Modules and JPMS**
    - Introduction to Java Platform Module System
    - Creating and using modules
    - Module descriptors (module-info.java)
    - Benefits of modular applications
23. **Testing in Java**
    - Unit testing with JUnit 5
    - Mocking with Mockito
    - Integration testing for Spring Boot
    - Test-driven development (TDD) basics
24. **Practice Session**
    - Build a task management API with Spring Boot
    - Congratulations! You’ve mastered Java programming.
25. **Project 1: Task Manager API**
    - Build a REST API with Spring Boot
    - Implement CRUD for tasks
    - Add JWT authentication
    - Test with Postman
26. **Project 2: Inventory System**
    - Create a console-based inventory system
    - Use collections and file I/O
    - Implement search and sort functionality
    - Handle exceptions gracefully
27. **Project 3: E-Commerce Backend**
    - Build an e-commerce API with Spring Boot
    - Manage products and orders with JPA
    - Integrate Stripe for payments
    - Document with Swagger
28. **Project 4: Blog API**
    - Create a blog API with Spring Boot
    - Manage posts and users with JPA
    - Add authentication with Spring Security
    - Deploy to Heroku
29. **Project 5: Multithreaded File Processor**
    - Build a file processing app
    - Use multithreading for parallel processing
    - Handle large CSV files
    - Optimize performance
30. **Project 6: Real-Time Chat Backend**
    - Build a chat API with Spring WebSocket
    - Implement user presence and messaging
    - Store messages in PostgreSQL
    - Secure with JWT
31. **Microservices with Spring Boot**

  - Introduction to microservices
  - Creating microservices with Spring Boot
  - Inter-service communication with REST
  - Using Spring Cloud for discovery

32. **Deploying Java Applications**
    - Deploying Spring Boot to Heroku
    - Containerizing with Docker
    - Setting up CI/CD with GitHub Actions
    - Configuring production environments
33. **Project 7: Microservices-Based App**
    - Build a microservices-based e-commerce app
    - Implement product and order services
    - Use Spring Cloud Gateway
    - Deploy with Docker
34. **Advanced Java Features**
    - Records and sealed classes
    - Pattern matching with switch
    - Virtual threads (Project Loom)
    - Performance optimization with JMH
35. **Final Exam and Wrap-Up**
    - Comprehensive exam on Java concepts
    - Review of key skills and best practices
    - Guidance on Java career paths
    - Certificate of completion