

Java Backend Engineering Course

2-Month Program Outline

Objective

Transform learners into **Java Backend Engineers** capable of:

- Writing clean, scalable Java code
- Building REST APIs using Spring Boot
- Working with databases (JPA/Hibernate)
- Developing production-ready backend systems

Program Duration

8 Weeks (2 Months)

Structure

- **2 Days** → Learning
- **2 Days** → Hands-on POC (Proof of Concept)
- **1 Day** → Demo + Knowledge Sharing

PHASE 1: Java Fundamentals (Week 1-2)

Topics

- Java basics (syntax, variables, data types)
- Control statements (if/else, loops)
- Methods & functions
- OOP concepts:
 - o Classes & objects
 - o Inheritance
 - o Polymorphism
 - o Encapsulation
- Collections Framework (List, Set, Map)

Tools

- Java (JDK 17+)
- IntelliJ IDEA / Eclipse

POC Ideas

1. CLI Calculator
2. Student Management System (OOP-based)
3. File handling utility

Outcome: Strong Java foundation and understanding of object-oriented programming.

PHASE 2: Advanced Java (Week 3)

Topics

- Exception handling
- Multithreading basics
- Streams API & Lambda expressions
- Functional programming concepts
- Java I/O & serialization

POC Ideas

1. Multi-threaded file processor
2. Data filtering using Streams
3. Log analyzer

Outcome: Write efficient, modern Java code and handle real-world data processing.

PHASE 3: Spring Boot Fundamentals (Week 4-5)

Topics

- Intro to Spring Boot
- Project structure & dependencies
- REST API development
- Controllers, Services, Repositories
- Dependency Injection (IoC)

Tools

- Spring Boot
- Maven / Gradle
- Postman

POC Ideas

1. Simple REST API (CRUD)
2. Notes Management API
3. User Service API

Outcome: Build backend APIs using Spring Boot and understand layered architecture.

PHASE 4: Database Integration (Week 6)

Topics

- SQL basics
- JPA & Hibernate
- Entity relationships (OneToMany,ManyToOne)
- CRUD operations
- Query optimization

Tools

- MySQL / PostgreSQL
- JPA / Hibernate

POC Ideas

1. Blog API (users + posts)
2. Inventory Management System

Outcome: Build database-driven applications and manage relational data effectively.

PHASE 5: Advanced Backend Concepts (Week 7)

Topics

- Spring Security (basic auth + JWT)
- Validation & exception handling
- Pagination & filtering
- Logging (SLF4J / Logback)
- API documentation (Swagger)

POC Ideas

1. Secure API with login/signup
2. Role-based access system
3. Protected endpoints

Outcome: Build secure and scalable APIs.

PHASE 6: Deployment + DevOps (Week 8)

Topics

- Application packaging (JAR)
- Docker basics
- Environment configuration
- Deployment strategies

Tools

- Docker
- AWS / Railway / Render

POC Ideas

1. Dockerize Spring Boot app
2. Deploy API to cloud

Outcome: Deploy production-ready backend services.

FINAL PROJECT (End of Course)

Choose ONE of the following options to build end-to-end:

1. **Task Management System:** Users + tasks, Authentication, REST APIs
2. **E-commerce Backend:** Products, Orders, Users, Payment simulation
3. **Blog Platform Backend:** Posts + comments, User roles
4. **Enterprise API System:** Multi-module architecture, Secure endpoints

Recommended Tech Stack

- **Backend:** Java, Spring Boot
- **Database:** MySQL / PostgreSQL
- **ORM:** Hibernate / JPA
- **DevOps:** Docker, AWS / Railway

High-Value POC Categorization

- **Beginner:** CLI apps, File processors
- **Intermediate:** REST APIs, CRUD systems
- **Advanced:** Secure backend systems, Multi-module applications

Weekly Execution Plan

Day	Activity
Monday	Learning
Tuesday	Learning
Wednesday	Build
Thursday	Build
Friday	Demo / Knowledge Sharing

Team Setup & Role Rotation

- Java Developer
- Backend API Developer
- Database Engineer
- DevOps

Evaluation Metrics

- ✓ Code quality
- ✓ API performance
- ✓ Security implementation
- ✓ Deployment success

Outcome After 2 Months

- Build Java backend systems
- Create REST APIs using Spring Boot
- Work with databases using JPA
- Deploy production-ready applications

How This Completes Your Full Stack

From your previous roadmaps:

- **Python** → AI backend
- **Java** → Enterprise backend
- **Frontend** → UI

Final Result: Full Stack + AI + Enterprise Engineer