

Java Full Stack Web Developer

Duration: 20 days

Tools: Java, JSP, MySQL, Spring Boot, Bootstrap, JavaScript

Prerequisites: Knowledge of Web Development and Any Programming Language

Day 1: Java Setup & Core Programming Basics

Topics:

- Installing JDK and IntelliJ/Eclipse
- Java Program Structure
- Data Types, Type Conversion
- Operators: Arithmetic, Relational, Logical
- Control Statements: if, switch, while, do-while, for
- Debugging & Best Practices

Labs:

- Lab 1: Create a simple calculator
 - Lab 2: Grade calculator using switch-case and loops
-

Day 2: Java OOPs - Classes and Objects

Topics:

- Classes & Objects
- Constructors (default, parameterized)
- this keyword
- Access Modifiers
- Static Keyword
- Method Overloading

Labs:

- Lab 1: Bank Account simulation
 - Lab 2: Shopping Cart with static count
-

Day 3: Inheritance, Abstraction, Polymorphism

Topics:

- Inheritance Types
- Method Overriding
- final, super, abstract class
- Interfaces vs Abstract Classes
- Dynamic Polymorphism

Labs:

- Lab 1: Employee management using inheritance
 - Lab 2: Shape calculation using interfaces
-

Day 4: Collections Framework

Topics:

- Collection vs Collections
- List, Set, Map Interfaces
- ArrayList, HashSet, LinkedList, TreeSet
- HashMap, TreeMap, LinkedHashMap
- Comparator vs Comparable

Labs:

- Lab 1: Student record system using List & Map
 - Lab 2: Sort employees by salary using Comparator
-

Day 5: Exception Handling & File Handling

Topics:

- Exception Types
- try, catch, finally, throw, throws
- Custom Exception
- FileReader, FileWriter, BufferedReader
- Serialization & Deserialization

Labs:

- Lab 1: Custom exception for AgeValidator
 - Lab 2: Read and write product data to file
-

Day 6: Java 8+ Features

Topics:

- Functional Interfaces
- Lambda Expressions
- Method References
 - Stream API (filter, map, reduce, collect)
 - Optional Class, forEach

Labs:

- Lab 1: Employee analytics using Stream API
 - Lab 2: Filtering orders using Lambda expressions
-

Day 7: Frontend – HTML, CSS, JavaScript Essentials

Topics:

- HTML5 Semantics & Forms
- CSS Styling, Selectors, Box Model
- JavaScript Basics: variables, arrays, objects
- DOM manipulation
- Event Handling

Labs:

- Lab 1: Login & Registration Form with JS validation
 - Lab 2: Form input to dynamic table (JS DOM)
-

Day 8: Responsive UI with Bootstrap

Topics:

- Bootstrap Setup
- Grid System, Buttons, Forms, Cards, Modals
- Icons & Utilities
- Responsive Design Principles

Labs:

- Lab 1: Build a responsive product catalog page
 - Lab 2: Styled registration form with modal confirmation
-

Day 9: Introduction to MySQL & Database Design

Topics:

- ER Diagram and Schema Design
- MySQL installation and Workbench
- Data Types, Keys, Normalization
- SQL DDL, DML, DCL, TCL
- Joins, Subqueries, Indexing

Labs:

- Lab 1: Design schema for E-Commerce app
 - Lab 2: Complex joins between orders & products
-

Day 10: JDBC – Java + MySQL Integration

Topics:

- JDBC Architecture & Setup
- Connection, Statement, ResultSet

- PreparedStatement & SQL Injection Prevention
- CRUD Operations
- Connection Pooling Overview

Labs:

- Lab 1: Connect Java to MySQL DB and run queries
 - Lab 2: Student management CRUD app using JDBC
-

Day 11: Servlet API and Web Development Basics

Topics:

- Servlet Lifecycle: init(), service(), destroy()
- Deployment Descriptor (web.xml)
- Request & Response objects
- Session & Cookie Management
- RequestDispatcher & Response Redirect

Labs:

- Lab 1: Servlet-based login/logout using session
 - Lab 2: Feedback form using doPost()
-

Day 12: JSP, JSTL, and EL (Expression Language)

Topics:

- JSP Directives, Scriptlets
- Expression Language (EL)
- JSTL Tags (Core, Format)
- JSP vs Servlet
- MVC Pattern with JSP

Labs:

- Lab 1: Create JSP + JSTL dynamic dashboard
 - Lab 2: Online quiz with session management
-

Day 13: Maven & Layered Architecture

Topics:

- Maven Lifecycle & Dependencies
- POM.xml structure
- Multi-Module Project Structure
- Packages: Model, DAO, Service, Controller
- Build & Run WAR files

Labs:

- Lab 1: Convert JDBC app to Maven project
 - Lab 2: Create layered architecture with DAOs
-

Day 14: Spring Boot – Getting Started**Topics:**

- Spring Boot Architecture
- Spring Initializr
- Application.properties/yml
- RESTful APIs: @RestController, @RequestMapping
- Dependency Injection (@Autowired)

Labs:

- Lab 1: Build REST API for products
 - Lab 2: Setup custom port & welcome endpoint
-

Day 15: Spring Boot + JPA + MySQL**Topics:**

- Entity, Repository, Service Layer
- Spring Data JPA Overview
- CRUD Operations
- H2 vs MySQL Integration
- Relationships (OneToMany,ManyToOne)

Labs:

- Lab 1: Build Employee + Department CRUD with JPA
 - Lab 2: Implement OneToMany order-product relationship
-

Day 16: DTOs, Validation & Exception Handling**Topics:**

- DTO vs Entity
- ModelMapper
- Bean Validation: @NotNull, @Size, @Email
- Global Exception Handling with @ControllerAdvice
- Custom Error Response DTOs

Labs:

- Lab 1: Validate API for user registration
- Lab 2: Build DTO & Service for clean separation

Day 17: Spring Boot Security

Topics:

- Intro to Spring Security
- Authentication & Authorization
- BCrypt Password Encoder
- Custom Login Endpoint
- Role-based Access Control (RBAC)

Labs:

- Lab 1: Secure CRUD API with roles (USER/ADMIN)
 - Lab 2: Implement token-based protection (JWT optional)
-

Day 18: Git, Docker & CI/CD Overview

Topics:

- Git Workflow: init, clone, commit, push, pull
- Branching & Merging
- Dockerfile for Spring Boot
- Docker Compose: MySQL + App
- CI/CD Intro (GitHub Actions/Heroku)

Labs:

- Lab 1: Dockerize spring boot + MySQL app
 - Lab 2: Push to GitHub, deploy to Railway/Render
-

Day 19: Capstone Project – Inventory App (Build Day 1)

Topics:

- Project planning and module breakdown
- Setup: Maven + Spring Boot + MySQL
- REST APIs for Admin, Products, Inventory
- Security Layer Integration

Labs:

- Lab 1: Implement backend APIs & DB structure
 - Lab 2: Add validations, DTOs, and security
-

Day 20: Capstone Project – Inventory App (Build Day 2 & Deploy)Topics:

- Integrate frontend (HTML/CSS/Bootstrap/JS)
- Final testing and bug fixing

- Deploy using Docker

Labs:

- Lab 1: Connect UI to REST APIs
- Lab 2: Build, dockerize, and deploy