

## 15 Days (Introduction to Java)

### Objective:

- Understand the basics of Java programming and build simple applications.

### Syllabus:

1. **Day 1-3:** Introduction to Java
  - a. Java overview and setup (JDK, IDEs like IntelliJ IDEA, Eclipse).
  - b. Writing and executing basic Java programs.
2. **Day 4-6:** Java Fundamentals
  - a. Data types, variables, and operators.
  - b. Control flow statements (if-else, loops).
3. **Day 7-9:** Object-Oriented Programming (OOP) Basics
  - a. Classes, objects, and methods.
  - b. Constructors and access modifiers.
4. **Day 10-12:** Collections and Arrays
  - a. Arrays, ArrayLists, and basic collections framework.
  - b. Iterators and loops for collections.
5. **Day 13-15:** Hands-On Practice
  - a. Simple console-based projects (e.g., calculator, student management system).

## 30 Days (Beginner-Level Java Development)

### Objective:

- Build a strong foundation in Java with OOP principles and basic application development.

### Syllabus:

1. **Week 1:** Core Java
  - a. Advanced OOP concepts: inheritance, polymorphism, encapsulation, and abstraction.
  - b. Static and final keywords, nested and inner classes.
2. **Week 2:** Exception Handling and I/O
  - a. Error and exception handling (try-catch, custom exceptions).
  - b. File handling using Java I/O (FileReader, FileWriter).
3. **Week 3:** Multithreading and Concurrency
  - a. Threads, runnable interface, and thread lifecycle.

- b. Synchronization and thread-safe programming.
- 4. **Week 4:** Basic Project
  - a. Develop a console-based project using all learned concepts (e.g., Library Management System).

## 45 Days (Intermediate-Level Java Development)

### *Objective:*

- Learn intermediate Java features and introduce database and GUI programming.

### *Syllabus:*

1. **Week 1-2:** Advanced Java Concepts
  - a. Generics and lambda expressions.
  - b. Streams API and functional programming.
2. **Week 3:** Database Programming
  - a. Introduction to JDBC.
  - b. Connecting Java applications with databases (MySQL, PostgreSQL).
3. **Week 4:** GUI Programming
  - a. JavaFX or Swing basics.
  - b. Building simple graphical user interfaces.
4. **Week 5:** Intermediate Project
  - a. Build a database-connected application (e.g., Inventory Management System).

## 60 Days (Advanced Java Development)

### *Objective:*

- Master Java development with frameworks and enterprise features.

### *Syllabus:*

1. **Week 1-2:** Advanced Frameworks
  - a. Introduction to Java frameworks (Spring, Hibernate).
  - b. Dependency Injection (DI) and Spring Boot basics.
2. **Week 3-4:** Web Development with Java
  - a. Basics of servlets and JSP.

- b. Building RESTful web services using Spring Boot.
- 3. **Week 5:** Application Security
  - a. Authentication and authorization with Spring Security.
  - b. Secure application development best practices.
- 4. **Week 6:** Advanced Project
  - a. Develop a web application using Spring Boot and Hibernate (e.g., Employee Management System).

## 90 Days (Comprehensive Java Development)

### *Objective:*

- Develop scalable web applications and gain industry-ready skills.

### *Syllabus:*

1. **Week 1-4:** Advanced Java Frameworks
  - a. Deep dive into Spring Boot features.
  - b. Advanced Hibernate (caching, relationships, and transactions).
2. **Week 5-6:** Microservices
  - a. Introduction to microservices architecture.
  - b. Building microservices with Spring Cloud.
3. **Week 7:** Testing and Debugging
  - a. Unit testing with JUnit and Mockito.
  - b. Debugging techniques and performance tuning.
4. **Week 8-9:** Full-Stack Integration
  - a. Connecting Angular/React frontend with Java backend.
  - b. Deployment on cloud platforms (AWS, Azure).

## 180 Days (Expert-Level Java Development)

### *Objective:*

- Achieve expertise in enterprise-grade Java development, focusing on scalability and performance.

### *Syllabus:*

1. **Month 1-3:** Advanced Java Features
  - a. Streams and parallel programming.
  - b. Advanced topics in Spring Boot (Batch processing, AOP).

- c. Integration with third-party APIs and services.
- 2. **Month 4-5:** Enterprise Development
  - a. Building large-scale enterprise applications.
  - b. Implementing GraphQL and advanced RESTful APIs.
  - c. Kafka and message queues.
- 3. **Month 6:** Capstone Project and Industry Readiness
  - a. Develop an enterprise-grade application (e.g., E-commerce platform).
  - b. Resume building, portfolio creation, and interview preparation.