**Course Title:**

**Core Java 21 , Spring 6 and Spring Boot 3.x - 5 Days Hands-on Training**

**Day 1: Core Java Fundamentals**

**Prerequisites:**

* **Installed Java Development Kit (JDK)**
* **IDE (Eclipse/IntelliJ IDEA) or text editor (VSCode, Notepad++)**
* **Understanding of basic programming concepts (optional)**

**Morning Session (9:00 AM – 12:30 PM)**

**1. Java Overview**

* **Duration: 45 mins**
* **Topics:**
  + **History of Java**
  + **Features of Java (Platform Independence, Object-Oriented, etc.)**
  + **JDK, JRE, and JVM: Differences and Use Cases**
  + **Installation & Environmental Setup**
* **Lab Requirements:**
  + **Set up Java environment on local machines.**
  + **Verify the setup by running a simple Java program (e.g., "Hello, World!").**

**2. Java Basics**

* **Duration: 1.5 hours**
* **Topics:**
  + **Variables, Data Types, and Operators**
  + **Control Statements (if, else, switch, loops: for, while, do-while)**
* **Hands-On:**
  + **Write a program demonstrating variables, data types, and operators.**
  + **Create a program using control statements for decision making.**

**3. Java Object-Oriented Programming (OOP)**

* **Duration: 1 hour**
* **Topics:**
  + **Introduction to OOP (Encapsulation, Abstraction, Inheritance, Polymorphism)**
  + **Classes and Objects: Definition, Methods, and Constructor**
* **Hands-On:**
  + **Create a class, instantiate objects, and define methods.**

**Afternoon Session (1:30 PM – 5:00 PM)**

**4. Inheritance and Polymorphism**

* **Duration: 1 hour**
* **Topics:**
  + **Inheritance: extends keyword, Single and Multilevel Inheritance**
  + **Polymorphism: Overloading and Overriding**
* **Hands-On:**
  + **Create a base class and derived classes demonstrating inheritance.**
  + **Implement method overloading and overriding in a class hierarchy.**

**5. Exception Handling**

* **Duration: 1 hour**
* **Topics:**
  + **Types of Exceptions: Checked and Unchecked**
  + **try, catch, finally, throw, throws**
* **Hands-On:**
  + **Create programs demonstrating exception handling with custom exceptions.**

**Case Study:**

* **Duration: 1 hour**
* **Description:**
  + **Develop a mini console-based application (e.g., Banking system, Library management) using the OOP concepts, inheritance, polymorphism, and exception handling.**

**Day 2: Advanced Java Concepts and Java Libraries**

**Prerequisites:**

* **IDE (Eclipse/IntelliJ IDEA) or text editor**
* **Database server (MySQL, PostgreSQL, etc.) installed**
* **Tomcat or any web server installed for servlet execution**

**Morning Session (9:00 AM – 12:30 PM)**

**1. Java Libraries**

* **Duration: 1 hour**
* **Topics:**
  + **Core Classes (String, wrapper classes, Date, Time, Instant, time zones)**
  + **Serialization**
  + **File Handling**
* **Hands-On:**
  + **Write a program to manipulate strings.**
  + **Implement file handling (reading/writing files) and serialization.**

**2. Java Collections Framework**

* **Duration: 1 hour**
* **Topics:**
  + **Collection Interfaces: Set, List, and Map**
  + **Implementations: ArrayList, HashSet, HashMap**
* **Hands-On:**
  + **Create programs that utilize collections such as List, Set, and Map.**

**3. Java Generics and Thread Basics**

* **Duration: 1 hour**
* **Topics:**
  + **Java Generics: Type Safety, Bounded Types**
  + **Threads: Multithreading, Runnable Interface, Thread Class**
* **Hands-On:**
  + **Write a multithreaded program demonstrating threads and synchronization.**

**Afternoon Session (1:30 PM – 5:00 PM)**

**4. Streams and Lambda Expressions**

* **Duration: 45 mins**
* **Topics:**
  + **Introduction to Streams**
  + **Lambda Functions in Java**
* **Hands-On:**
  + **Use Streams and Lambda expressions to filter and process data from a collection.**

**5. Java Frameworks**

* **Duration: 45 mins**
* **Topics:**
  + **JDBC: Connecting Java with a Database**
  + **Web Programming: Servlets and JSP Overview**
* **Hands-On:**
  + **Write a program to connect to a database using JDBC.**
  + **Demonstrate a simple servlet in a Tomcat server.**

**6. Unit Testing with JUnit and Mockito**

* **Duration: 1 hour**
* **Topics:**
  + **JUnit Basics: Writing Unit Tests**
  + **Mockito: Mocking Objects for Testing**
  + **Testing Exceptions**
* **Hands-On:**
  + **Write unit tests using JUnit and mock dependencies with Mockito.**
  + **Demonstrate testing for exceptions in a program.**

**7. Java Build Systems (Maven)**

* **Duration: 1 hour**
* **Topics:**
  + **Maven Basics: POM.xml, Dependency Management**
  + **Maven Build Life Cycle**
* **Hands-On:**
  + **Set up a Maven project and manage dependencies.**

**Case Study:**

* **Duration: 1 hour**
* **Description:**
  + **Develop a small web application (e.g., Student Management System) using JDBC, servlets, JSP, and test it using JUnit and Mockito.**

**Lab Requirements:**

* **Java IDE installed (Eclipse/IntelliJ IDEA)**
* **MySQL/PostgreSQL or any RDBMS for database-related hands-on sessions**
* **Web server (Apache Tomcat) for Servlet/JSP hands-on**
* **Maven installed for dependency management**

**End of Day 2: Wrap-up and Q&A**

**Day 3**

**Core Spring 6 In-Depth Course Content (4 hours)**

**Prerequisites:**

* Understanding of Java basics and object-oriented programming
* Experience with Spring Framework basics (Spring Core, Dependency Injection)
* IDE (IntelliJ IDEA/Eclipse) with Spring Boot plugin
* Maven or Gradle installed for project setup

**Course Schedule (4 hours)**

**1. Overview of Spring 6 and New Features**

* **Duration:** 2 Hours
* **Topics:**
  + Introduction to Spring 6
  + Key Changes and Enhancements in Spring 6
  + JDK 17+ Requirements
  + New Features in Spring Core and Spring Boot 3 (Spring Boot 3 compatibility)
* **Hands-On:**
  + Set up a new Spring 6 project.
  + Explore basic project structure and run a Bean Factory and Application Context.

**2. Advanced Spring Core Concepts**

* **Duration:** 2 Hours
* **Topics:**
  + Advanced Dependency Injection techniques (Constructor Injection, Setter Injection, Field Injection)
  + Component Scanning and Stereotype Annotations (@Component, @Repository, @Service, @Controller)
  + Scopes in Spring (@Scope, Singleton vs Prototype)
  + Custom Bean Lifecycle Callbacks (@PostConstruct, @PreDestroy)
  + Using @Configuration and @Bean in-depth for bean creation
* **Hands-On:**
  + Create a project demonstrating advanced DI techniques, including Constructor Injection and custom bean lifecycle management.
  + Implement custom scopes and showcase bean lifecycle with @PostConstruct and @PreDestroy.

**Spring Boot 3.x Course Content for 2.5 Days**

**Prerequisites:**

* Basic knowledge of Java and OOP concepts
* Understanding of Spring Framework basics (Spring Core, Dependency Injection, MVC)
* Familiarity with web development concepts (HTML, HTTP, REST)
* IDE (Eclipse/IntelliJ IDEA) with Maven/Gradle support
* Basic knowledge of databases (SQL queries, transactions)
* Maven or Gradle installed for project setup

**Lab Requirements:**

* Java Development Kit (JDK 17+) installed
* IDE (IntelliJ IDEA, Eclipse) with Spring Boot plugins
* Maven or Gradle installed
* MySQL/PostgreSQL database installed for data-related exercises
* Postman for testing APIs
* Docker (optional) for containerizing applications
* GraalVM installed for Native Image support (optional)

**Course Schedule (2.5 Days)**

**Day 3: Introduction and Core Concepts**

**Afternoon Session (1:30 PM – 05:00 PM)**

**1. Introduction to Spring Boot 3.x**

* **Duration:** 45 mins
* **Topics:**
  + Overview of Spring Boot and its evolution
  + Key features of Spring Boot 3.x
  + Understanding Spring Boot’s auto-configuration and embedded server support (Tomcat, Jetty)
  + Spring Boot 3.x and JDK 17 compatibility
* **Hands-On:**
  + Set up a simple Spring Boot 3.x project using Spring Initializr.
  + Run and configure your first Spring Boot application.
  + Explore application.properties and application.yml files.

**2. Spring Boot Auto-Configuration and Starter Dependencies**

* **Duration:** 45 mins
* **Topics:**
  + Understanding Spring Boot Starter POMs (dependencies)
  + How auto-configuration works under the hood
  + Customizing auto-configuration (exclude/include beans)
* **Hands-On:**
  + Add different starter dependencies and configure a custom auto-configuration example.
  + Customize an application using properties and environment profiles.

**3. Spring Boot Application Structure and Command Line Runner**

* **Duration:** 1 hour
* **Topics:**
  + Spring Boot project structure and packaging (JAR/WAR)
  + @SpringBootApplication annotation explained
  + Understanding Command Line Runner and Application Runner
* **Hands-On:**
  + Create a Command Line Runner to run custom code during application startup.
  + Explore the structure of a Spring Boot JAR application.

**Day 4**

**Morning Session (9:00 PM - 12:30 PM)**

**4. RESTful Web Services with Spring Boot**

* **Duration:** 2.5 hours
* **Topics:**
  + Building RESTful APIs using @RestController and @RequestMapping
  + Handling HTTP methods (GET, POST, PUT, DELETE)
  + Validating request data with @Valid and @Validated
* **Hands-On:**
  + Create a simple CRUD API (e.g., for managing employees or products) using Spring Boot REST controllers.
  + Implement request validation with @Valid annotations.

**Advanced Spring Boot Concepts**

**Afternoon Session (1:30 PM - 05:00 PM)**

**5. Spring Boot Data Access with Spring Data JPA**

* **Duration:** 1.5 hours
* **Topics:**
  + Introduction to Spring Data JPA
  + Setting up a database connection (MySQL/PostgreSQL)
  + @Entity, @Repository, and JpaRepository explained
  + Query methods and JPQL
  + Pagination and Sorting in repositories
* **Hands-On:**
  + Create a CRUD repository using Spring Data JPA to manage entities in a database.
  + Demonstrate pagination and sorting using repository methods.
  + Configure database properties in application.yml.

**6. Spring Boot Security (Spring Security 6)**

* **Duration:** 1.5 hours
* **Topics:**
  + Introduction to Spring Security 6 and key updates in Spring Boot 3.x
  + Securing REST APIs using Spring Security with JWT (JSON Web Tokens)
  + Configuring authentication and authorization with role-based access control
  + Custom login/logout pages
* **Hands-On:**
  + Secure the previously developed CRUD API using Spring Security with JWT-based authentication.
  + Create roles for admin and user, and implement role-based access control.

**Day 5**

**Morning Session (9:00 AM - 12:30 PM)**

**7. Spring Boot Actuator**

* **Duration:** 45 mins
* **Topics:**
  + Introduction to Spring Boot Actuator for monitoring and metrics
  + Exposing health, info, and custom metrics
  + Configuring security for Actuator endpoints
* **Hands-On:**
  + Enable Actuator in the project and expose health, metrics, and custom endpoints.
  + Secure Actuator endpoints using Spring Security.

**8. Spring Boot Testing with JUnit and Mockito**

* **Duration:** 1 hour
* **Topics:**
  + Unit testing in Spring Boot with JUnit 5
  + Mocking dependencies using Mockito
  + Testing Spring Boot controllers and services
  + Writing integration tests with @SpringBootTest
* **Hands-On:**
  + Write unit tests for services and controllers using JUnit and Mockito.
  + Test RESTful APIs using MockMvc.
  + Perform an integration test using @SpringBootTest.

**9. Spring Boot DevTools and Profiles**

* **Duration:** 45 mins
* **Topics:**
  + Introduction to Spring Boot DevTools for rapid application development
  + Configuring and using environment-specific profiles (dev, prod)
  + Hot reloading with DevTools
* **Hands-On:**
  + Enable DevTools in a Spring Boot project for live reloading.
  + Create environment-specific configurations using Spring profiles (dev, prod).

**Afternoon Session (1:30 PM - 5:00 PM)**

**10. Case Study: Building a Full-Stack Spring Boot Application**

* **Duration:** 1.5 hours
* **Description:**
  + **Scenario:** Build a full-stack employee management system with the following features:
    - RESTful API for managing employee data (CRUD operations)
    - Secure the API with JWT-based authentication and role-based access control
    - Persist data using Spring Data JPA connected to a MySQL/PostgreSQL database
    - Expose health and metrics using Spring Boot Actuator
    - Write unit and integration tests to ensure code quality
    - Dockerize the application for deployment using Docker Compose
* **Hands-On:**
  + Set up the project using Spring Boot 3.x.
  + Implement the RESTful API with Spring Data JPA and secure it with Spring Security JWT.
  + Write unit and integration tests for the project.
  + Monitor the application with Actuator endpoints.

**Wrap-up and Q&A**

* **Duration:** 30 mins
* Review key concepts learned over the 2.5 days.
* Answer participant questions and discuss further learning paths (Spring Cloud, Spring Security advanced topics, etc.).