Spring Framework

1. Spring Core

- **Introduction to Spring**: Understand what Spring is, its advantages, and how it simplifies Java enterprise development.
- **Inversion of Control (IoC) and Dependency Injection (DI)**: Learn the core principles of IoC and DI, which are the foundation of Spring.
- **Spring Bean Lifecycle**: Study bean scopes, lifecycle methods, and bean configuration.
- **Spring Expression Language (SpEL)**: Get a basic understanding of SpEL for use in configuring beans.

2. Spring AOP (Aspect-Oriented Programming)

- **AOP Basics**: Learn what AOP is and how it handles cross-cutting concerns (e.g., logging, security).
- **AspectJ Annotations**: Get hands-on with AspectJ annotations for defining aspects, join points, and advice types (before, after, around).

3. Spring Data Access / Integration

- **JDBC with Spring**: Start with Spring's simplified JDBC template for interacting with databases.
- **Transaction Management**: Learn declarative transaction management with `@Transactional`.
- **Spring ORM (Object-Relational Mapping)**: Understand Spring's integration with JPA and Hibernate.
- **Introduction to Spring Data JPA**: Learn how Spring Data JPA abstracts JPA-based repositories for simpler CRUD operations.

4. Spring MVC (Model-View-Controller)

- **Basics of Spring MVC**: Understand the MVC architecture and components like controllers, models, and views.
- **Spring MVC Annotations**: Learn the key annotations (`@Controller`, `@RequestMapping`, etc.) to handle web requests.
- **Form Handling and Validation**: Explore form handling, binding, and validation using annotations like `@Valid` and `@ModelAttribute`.

- **RESTful Services with Spring MVC**: Learn to build REST APIs using `@RestController` and JSON/XML responses.

5. Spring Boot

- **Spring Boot Basics**: Learn what Spring Boot is, how it simplifies configuration, and the use of the Spring Initializr.
- **Auto-Configuration and Spring Boot Annotations**: Dive into `@SpringBootApplication` and auto-configuration concepts.
- **Embedded Servers**: Understand embedded servers (Tomcat, Jetty) and how to run standalone Spring Boot applications.
- **Spring Boot Starter Dependencies**: Familiarize yourself with pre-configured dependencies.
- **Spring Boot Actuator**: Learn how to monitor and manage your application.
- **Spring Boot Profiles**: Configure application environments using profiles (dev, test, prod).

6. Spring Data

- **Spring Data JPA Advanced**: Learn advanced features, including custom queries, pagination, and sorting.
- **NoSQL Databases**: Explore integrations with NoSQL databases like MongoDB, Redis, and Cassandra.

7. Spring Security

- **Authentication and Authorization**: Understand basic security concepts and how to implement login, roles, and permissions.
- **Spring Security Configurations**: Learn how to configure HTTP security, CORS, CSRF, and custom login forms.
- **OAuth2 and JWT**: Dive into OAuth2 for securing REST APIs, as well as JSON Web Tokens (JWT) for stateless authentication.

8. Spring REST and HATEOAS

- **Building RESTful APIs with Spring Boot**: Learn best practices for building REST APIs with Spring Boot.
- **Spring HATEOAS**: Introduce hypermedia-driven APIs, guiding clients with links to related resources.

9. Spring Testing

- **Unit Testing with Spring**: Test Spring components using JUnit and Mockito.
- **Integration Testing**: Write tests for Spring applications, focusing on loading the context and testing configurations.
- **MockMVC**: Test Spring MVC applications without starting the server.

10. Spring Cloud and Microservices

- **Spring Cloud Basics**: Understand microservices architecture and why Spring Cloud is useful.
- **Spring Cloud Config**: Learn centralized configuration management for distributed applications.
- **Service Discovery with Eureka**: Enable service discovery using Spring Cloud Netflix's Eureka.
- **Circuit Breakers and Resilience**: Implement fault tolerance with Hystrix or Resilience4j.
- **API Gateway with Spring Cloud Gateway**: Manage requests using an API Gateway.
- **Spring Cloud Sleuth and Zipkin**: Implement distributed tracing for microservices.

11. Spring WebFlux (Reactive Programming)

- **Introduction to Reactive Programming**: Understand the basics of reactive programming and why it's useful.
- **Spring WebFlux**: Learn to build reactive, non-blocking applications with WebFlux.
- **Reactive Repositories**: Explore reactive support in Spring Data for MongoDB, Redis, and more.

12. Spring Batch

- **Batch Processing Concepts**: Understand batch jobs, steps, and job repository concepts.
- **Spring Batch Basics**: Learn how to create, configure, and execute batch jobs.
- **Chunk-Oriented Processing**: Implement chunk-oriented batch jobs and explore retry, restart, and skip functionality.

13. Advanced Topics and Best Practices

- **Caching with Spring**: Use caching abstractions for performance optimization.
- **Caching in Hibernate**: Integrate Hibernate caching techniques.
- **Custom Annotations and Stereotypes**: Create custom annotations to standardize configurations.

- **Design Patterns with Spring**: Study design patterns commonly used in Spring (e.g., Singleton, Factory, Proxy).
- **Performance Optimization and Profiling**: Use Actuator, Metrics, and third-party tools to monitor and optimize performance.