Introduction to Spring

1. What is Spring Framework?

- **Spring** is a comprehensive **Java framework** for building robust, enterprise-grade applications.
- It simplifies **Java development** by offering a variety of tools for **dependency management**, **aspect-oriented programming**, **web development**, and more.

2. Why Spring?

- Other frameworks like **Struts** (web) and **Hibernate** (ORM) address specific concerns.
- **Spring** provides a **complete solution** for all application needs, including web, database, security, and transaction management.

3. Core Features of Spring:

- **POJO-Based Development**: Build applications using **Plain Old Java Objects** (POJOs), without relying heavily on frameworks.
- **Dependency Injection (DI)**: Handles object creation and dependency management, promoting **loose coupling**.
- **Aspect-Oriented Programming (AOP)**: Manages cross-cutting concerns (e.g., logging, security) separately from business logic.

4. Main Spring Modules:

- **Spring Core**: Provides the fundamental building blocks, including DI and BeanFactory.
- **Spring AOP**: Enables modularization of concerns like logging or transaction management.
- **Spring ORM**: Integration with Hibernate, JPA, and other ORM tools.
- **Spring MVC**: Facilitates web application development using the **Model-View-Controller** pattern.

- **Spring Boot**: A Spring extension for creating **standalone applications** with embedded servers (Tomcat, Jetty).

5. Why Spring?

- **Lightweight** and modular.
- Promotes **clean, maintainable code**.
- Supports a wide range of technologies (Web, ORM, Security).Spring has become a standard for Java development due to its flexibility and ease of integration with other frameworks and libraries.