# What is Spring?

1. Spring: A powerful Java framework used for building web applications. It provides comprehensive infrastructure support for developing robust Java applications easily and rapidly. Example: Spring's IoC container manages application components.

# What is Spring Boot?

2. Spring Boot: A project within the Spring ecosystem that simplifies the bootstrapping and development of new Spring applications. It provides default configurations to build Spring-powered applications quickly. Example: A Spring Boot application can be run with minimal setup using `@SpringBootApplication`.

# What is the relation between Spring platform and Spring Boot?

3. Relation between Spring platform and Spring Boot: Spring Boot is part of the Spring ecosystem and is built on top of the Spring framework, providing conventions that lead to less boilerplate configuration.

# What is the relation between Spring platform and Spring framework?

4. Relation between Spring platform and Spring framework: The Spring platform is an umbrella term that encompasses the Spring framework along with other projects like Spring Boot, Spring Data, etc. The Spring framework is the foundation providing core functionalities like Dependency Injection and AOP.

# What is Dependency Injection and how is it done in the Spring platform/framework?

5. Dependency Injection: It's a design pattern where a class receives its dependencies from external sources rather than creating them itself. In Spring, this is done through annotations like `@Autowired` or XML-based configurations.

# What is Inversion of Control (IoC) and how is it related to Spring?

6. Inversion of Control (IoC): A principle where the control of objects is transferred from the program to a container or framework. In Spring, IoC is implemented using the Spring IoC container which manages the lifecycle and configuration of application objects.

# Task 4:

A screenshot of a computer

Description automatically generated

A diagram of a service layer

Description automatically generated

- User Interface: This is the front-end application that users interact with. It can be built using frameworks like React or Angular.

- API Layer: This acts as an interface between the UI and the service layer, typically built with Spring Boot.

- Service Layer: Contains business logic, and is where the application's functions are implemented, often using the Spring Framework.

- Data Access Layer: Responsible for data persistence and retrieval, usually implemented with Spring Data.

- Database: The underlying database where all data is stored, such as MySQL or PostgreSQL.