1. What is Spring?

Ans. Spring is a powerful Java framework primarily used for building enterprise-level applications. It provides comprehensive infrastructure support for developing Java applications.

2. What is Spring Boot?

Ans. Spring Boot is an extension of the Spring framework that simplifies the process of building and deploying stand-alone, production-grade Spring-based applications. It offers auto-configuration and convention-over-configuration features to reduce boilerplate code.

3. What is the relation between Spring platform and Spring Boot?

Ans. Spring Boot is built on top of the Spring platform. It leverages many features of the Spring framework while providing additional tools and capabilities to streamline the development process.

4. What is the relation between Spring platform and Spring framework?

The Spring platform encompasses the entire Spring ecosystem, which includes the Spring framework along with other projects like Spring Boot, Spring Data, Spring Security, etc. The Spring framework, on the other hand, is the core component of the Spring platform, providing essential features like dependency injection and aspect-oriented programming.

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

Dependency Injection is a design pattern used to manage dependencies between objects. In Spring, dependencies are injected into a class rather than the class itself creating or looking up dependencies. This is typically done through constructor injection, setter injection, or field injection using annotations like @Autowired.

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

Inversion of Control (IoC) is a principle where the control of object creation and lifecycle is shifted from the application code to a container or framework. Spring implements IoC through its container, which manages the lifecycle of Java objects and their dependencies. This allows developers to focus on business logic rather than managing object creation and wiring dependencies.