**What is Spring?**

Spring is an open-source framework for building enterprise-level Java applications. It provides comprehensive infrastructure support and a vast array of services, including dependency injection, aspect-oriented programming, transaction management, and more, to simplify the development of Java applications.

**What is Spring Boot?**

Spring Boot is an extension of the Spring framework that aims to simplify the process of building production-ready applications. It provides conventions and pre-configured settings to reduce the setup and configuration overhead typically associated with Spring-based applications. Spring Boot includes embedded servers, auto-configuration, and other features to facilitate rapid application development.

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

Spring Boot is built on top of the Spring framework. While Spring provides a comprehensive set of features and tools for building enterprise Java applications, Spring Boot further simplifies the development process by providing conventions, defaults, and auto-configuration capabilities. Spring Boot is essentially a tool within the broader ecosystem of the Spring platform.

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

The Spring platform encompasses the entire ecosystem of Spring projects, including the core Spring framework along with various extensions and libraries such as Spring Boot, Spring Data, Spring Security, etc. The Spring framework is at the core of the platform, providing fundamental features like dependency injection, aspect-oriented programming, and more.

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

Dependency Injection (DI) is a design pattern in which objects are provided with their dependencies rather than creating them internally. In the Spring framework, DI is achieved through inversion of control (IoC) containers, where objects (beans) define their dependencies through configuration (XML, annotations, or Java-based configuration). The Spring container then injects these dependencies at runtime, allowing for loose coupling and easier unit testing.

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

Inversion of Control (IoC) is a principle in software engineering where the control flow of a program is inverted compared to traditional procedural programming. In IoC, the flow of control is shifted from the application code to a framework or container, which manages the lifecycle of application objects. Spring implements IoC through its container, which manages the creation, configuration, and assembly of application objects (beans). This allows for better modularization, easier testing, and improved maintainability of applications.