1. What is Spring?
   * Spring is a powerful, open-source framework for building Java applications, particularly those aimed at enterprise-level solutions. Developed to simplify Java development, Spring provides a comprehensive ecosystem for creating robust, scalable applications.
2. What is Spring Boot?
   * Spring Boot is a framework built on top of Spring that simplifies Java application development by minimizing configuration. It enables developers to create standalone applications with embedded servers (like Tomcat or Jetty), so there’s no need to deploy to an external server. Spring Boot uses convention over configuration, providing sensible defaults and auto-configuration, which means it automatically sets up the application based on dependencies in the project.
3. What is the relation between Spring platform and Spring Boot?
   * Spring Boot is part of the Spring platform. While Spring platform provides a comprehensive set of libraries and tools for building Java applications with extensive features like DI, transaction management and AOP, Spring Boot focuses on simplifying the development process by handling configuration automatically.
4. What is the relation between Spring platform and Spring framework?

- The Spring Platform is a comprehensive ecosystem that includes a variety of projects and tools for enterprise Java development, while the Spring Framework is the foundational component of this ecosystem.

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

**-Dependency Injection** is a design pattern in which an object receives its dependencies (objects it relies on) from an external source rather than creating them itself. This approach improves modularity, testability, and flexibility, as dependencies can be easily replaced or modified.

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

- Inversion of Control (IoC) is a principle in software design where the control over the creation and management of objects is transferred from the application code to a framework or container. This approach allows the framework to manage dependencies, enhance modularity and flexibility in application development.

In the Spring Framework, IoC is implemented through the IoC Container, which is responsible for creating, configuring, and managing the lifecycle of objects (known as beans) in the application. The container injects dependencies where needed, based on configurations or annotations, relieving developers from manual object management and configuration.