Task 3.

1. What is Spring?

   Ans: Spring is an open-source framework for Java applications, offering features like dependency injection, aspect-oriented programming, and database integration. It simplifies application development, promotes best practices, and is widely used in enterprise and web development. Spring Boot, a part of Spring, streamlines project setup, while Spring Cloud aids in building cloud-native applications.

2. What is Spring Boot?

   Ans: Spring Boot is a Java framework designed for fast and simple application development. It offers automated setup, built-in web servers, ready-made templates, and production-ready capabilities, streamlining the creation of robust applications with minimal manual configuration.

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

   Ans: Spring Boot, part of the Spring ecosystem, simplifies Spring application development by offering conventions and pre-configured features. "Spring Platform" can refer to the broader Spring ecosystem or the core Spring Framework. Spring Boot speeds up and streamlines Spring development.

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

   Ans: The Spring Framework serves as the heart of the Spring ecosystem, offering crucial utilities for Java applications. The term "Spring Platform" is not official, but it can be informally used to describe either the entire Spring ecosystem or just the Spring Framework itself, depending on the context.

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

   Dependency Injection (DI) in Spring is a design pattern that promotes loose coupling in software by injecting dependencies into classes from external sources, rather than having classes create their own dependencies. In Spring:

   - Annotate classes as Spring components.
   - Use @Autowired to declare dependencies.
   - Configure dependencies in XML, Java, or annotations.
   - Spring automatically manages and injects dependencies based on configuration.
   - Choose a constructor, field, or setter injection based on your needs.
   - Component scanning simplifies the registration of components.

   DI enhances code flexibility, maintainability, and testability by allowing components to focus on their core functionality while managing their dependencies externally.

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

Ans: Inversion of Control (IoC) is a design principle where control over how components are created and managed is shifted to an external framework or container. In the case of Spring, it acts as an IoC container, managing the lifecycle of components and their dependencies. This promotes loose coupling, modularity, and easier testing in software development.