

## **Lab 3-4, Sreytouch Lang (Jessica), StudentID:614145**

### **1. What is Spring?**

- ⇒ Spring is an open-source framework for building Java-based enterprise applications. It provides a comprehensive infrastructure for developing robust and maintainable software, especially for Java-based web and enterprise applications.

### **2. What is Spring Boot?**

- ⇒ Spring Boot is used for building various types of applications, including web applications, RESTful APIs, microservices, batch-processing applications, and more. It has gained popularity due to its ease of use, rapid development capabilities, and strong integration with the larger Spring ecosystem.

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

- ⇒ Spring Boot is a specialized project within the Spring platform that simplifies and streamlines the development of Spring-based applications. It relies on the core capabilities of the Spring Framework and integrates with other Spring projects to provide a cohesive and opinionated approach to application development, particularly for stand-alone and microservices-based applications.

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

- ⇒ The Spring Framework is the foundation upon which the entire Spring platform is built. It introduced many fundamental concepts and patterns to Java application development, including the concept of IoC/DI, which promotes loose coupling and testability.
- ⇒ Other projects within the Spring platform, such as Spring Boot, Spring Data, and Spring Security, are built on top of the Spring Framework. They leverage its core capabilities to provide higher-level functionality for specific use cases.
- ⇒ Spring Boot, for example, simplifies application setup and configuration by building on Spring Framework's principles and adding conventions and auto-configuration features.
- ⇒ Spring Data uses the Spring Framework's data access features to provide a unified and simplified approach to working with various data sources.
- ⇒ Spring Security builds on the Spring Framework's AOP capabilities to provide comprehensive security features.

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

- ⇒ Dependency Injection (DI) is a design pattern in which the dependencies of a class are injected from the outside rather than being created within the class itself. In Spring, DI is used to manage and resolve object dependencies. Spring provides various ways to achieve DI, with the most common approach being through constructor injection or setter injection.

⇒ Example (Constructor Injection in Spring):

```
```java

@Configuration
public class AppConfig {

    @Bean
    public UserService userService() {
        return new UserServiceImpl(); // Creating a UserService bean
    }

    @Bean
    public OrderService orderService() {
        return new OrderServiceImpl(userService());
    }
}
```
```

In this example, we define two beans (userService and orderService) in the Spring configuration. The orderService bean's constructor is injected with the userService bean, demonstrating Dependency Injection.

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

⇒ Inversion of Control (IoC) is a fundamental design principle that shifts control over object creation and management from application code to a framework or container. The Spring framework is closely related to IoC and provides tools and mechanisms for implementing it, particularly through its IoC container and Dependency Injection capabilities.