Day-26 Interview Questions

1. **What is the Spring framework, and why is it used in Java applications?**

The Spring framework is a comprehensive and lightweight framework that simplifies the development of Java applications. It provides various features, such as dependency injection, aspect-oriented programming, and transaction management, to improve application development and maintenance.

2. **Explain the core features of the Spring framework.**

The core features of the Spring framework include:

- Inversion of Control (IoC) container
- Dependency Injection (DI)
- Aspect-Oriented Programming (AOP)
- Data access and integration
- Transaction management
- Spring MVC for web applications
- Spring Security for security management

3. **What is the difference between Spring Boot and the traditional Spring framework?**

Spring Boot is a project within the Spring ecosystem that simplifies the setup and development of Spring applications. It provides auto-configuration, embedded web servers, and a streamlined approach to building Spring applications. In contrast, the traditional Spring framework requires more configuration and setup, making Spring Boot a more convenient choice for many developers.

4. **What is Dependency Injection, and why is it important in the Spring framework?**

Dependency Injection (DI) is a design pattern used in the Spring framework to inject dependencies (collaborating objects) into a class instead of the class creating or managing them. DI promotes loose coupling, making code more maintainable and testable.

5. Explain constructor injection and provide an example.

Constructor injection involves providing dependencies through a class constructor. For example:

```
public class MyClass {
    private Dependency dependency;

    public MyClass(Dependency dependency) {
        this.dependency = dependency;
    }
}
```

6. What does Inversion of Control (IoC) mean in the context of the Spring framework?

In IoC, the control of creating and managing objects (beans) is inverted from the application code to the framework's IoC container. This allows the framework to manage object lifecycles, relationships, and configurations.

7. How does the ApplicationContext differ from the BeanFactory in Spring's IoC container?

The ApplicationContext is a more feature-rich and advanced IoC container compared to the BeanFactory. It includes all the functionality of the BeanFactory and provides additional features like message sources, application event publishing, and aspect support.

8. How do you configure beans in XML and Java-based configurations in a Spring application?

In XML configuration, you define beans in an XML file using <bean> elements. In Java-based configuration, you use Java classes annotated with @Configuration and define beans using @Bean annotations.

9. What is the main role of the Spring IoC container?

The Spring IoC container is responsible for managing the lifecycle of Spring beans, creating and initializing them, and handling their dependencies. It follows the Inversion of Control principle.