Day 27-Interview Questions

1.Explain the bean lifecycle in Spring.

The Spring bean lifecycle involves the following phases:

- Instantiation: The container creates an instance of the bean.
- Population of Properties: Dependencies and properties are injected.
- Initialization Callbacks: @PostConstruct annotated methods or afterPropertiesSet method (if implementing InitializingBean) are invoked.
- Bean Usability: The bean is now available for use.
- Destruction Callbacks: @PreDestroy annotated methods or destroy method (if implementing DisposableBean) are invoked during container shutdown.

2.How can you specify custom initialization and destruction methods for a bean?

Custom initialization and destruction methods can be defined using @PostConstruct and @PreDestroy annotations, or by implementing the InitializingBean and DisposableBean interfaces.

3.Explain the difference between Singleton and Prototype scopes in Spring.

Singleton Scope: Only one instance of the bean is created per Spring container.
Prototype Scope: A new instance is created every time the bean is requested.

4.What is the default scope for a Spring bean?

The default scope for a Spring bean is Singleton.

5.How can you perform constructor injection using autowiring in Spring?

Constructor injection using autowiring can be achieved by placing the @Autowired annotation on the constructor.

```
@Autowired
public MyClass(MyDependency dependency) {
   this.dependency = dependency;
}
```

6.What does the @Autowired annotation do in Spring?

The @Autowired annotation is used for automatic dependency injection. It can be applied to fields, constructors, or methods, allowing Spring to automatically provide the necessary dependencies at runtime.