

1. What is Spring Framework?

Interview answer:

“Spring is a lightweight framework that provides comprehensive infrastructure support for Java applications, mainly focusing on dependency injection and aspect-oriented programming.”

2. What problem does Spring solve?

Interview answer:

“Spring reduces tight coupling, simplifies configuration, manages object lifecycle, and makes applications easier to test and maintain.”

3. What is IoC (Inversion of Control)?

Interview answer:

“In IoC, the control of object creation and dependency management is given to the Spring container instead of the programmer.”

4. What is Dependency Injection?

Interview answer:

“Dependency Injection is a design pattern where required objects are provided to a class from outside rather than the class creating them itself.”

5. Types of Dependency Injection?

Interview answer:

“Constructor injection, setter injection, and field injection. Constructor injection is preferred.”

6. Why constructor injection is preferred?

Interview answer:

“It ensures mandatory dependencies, supports immutability, and makes the class easier to test.”

7. What is Spring Container?

Interview answer:

“Spring container is responsible for creating, configuring, and managing Spring beans. Examples are ApplicationContext and BeanFactory.”

8. Difference between BeanFactory and ApplicationContext?

Interview answer:

“ApplicationContext is a superset of BeanFactory and provides features like event handling, internationalization, and eager bean initialization.”

9. What is a Spring Bean?

Interview answer:

“A Spring bean is an object that is created, configured, and managed by the Spring container.”

10. How do you define a Spring Bean?

Interview answer:

“Using XML configuration, Java-based configuration with @Bean, or annotations like @Component.”

11. What are common Spring Bean scopes?

Interview answer:

“Singleton, Prototype, Request, Session, and Application.”

12. Default scope of Spring Bean?

Interview answer:

“The default scope is Singleton.”

13. What is Singleton scope in Spring?

Interview answer:

“In Singleton scope, only one instance of the bean is created per Spring container.”

14. Difference between Spring Singleton and Java Singleton?

Interview answer:

“Spring Singleton is per container, while Java Singleton is per JVM.”

15. What is Bean lifecycle in Spring?

Interview answer:

“Bean lifecycle includes instantiation, dependency injection, initialization, and destruction, all managed by the Spring container.”

16. How do you customize bean lifecycle?

Interview answer:

“Using init-method, destroy-method, @PostConstruct, @PreDestroy, or implementing lifecycle interfaces.”

17. What is @Component?

Interview answer:

“@Component marks a class as a Spring-managed bean and enables component scanning.”

18. Difference between @Component, @Service, @Repository, @Controller?

Interview answer:

“They are specialization of @Component. @Repository provides exception translation, while others improve code readability and intent.”

19. What is component scanning?

Interview answer:

“Component scanning allows Spring to automatically detect and register beans using classpath scanning.”

20. What is @Autowired?

Interview answer:

“@Autowired is used to inject dependencies automatically by type.”

21. What if multiple beans of same type exist?

Interview answer:

“We use @Qualifier or @Primary to resolve ambiguity.”

22. What is @Qualifier?

Interview answer:

“@Qualifier is used to specify which bean should be injected when multiple beans of the same type exist.”

23. What is @Primary?

Interview answer:

“@Primary marks a bean as the default choice when multiple beans of the same type are available.”

24. Difference between @Autowired and @Inject?

Interview answer:

“@Autowired is Spring-specific, while @Inject is from Java specification. Functionally, they are similar.”

25. What is Spring Expression Language (SpEL)?

Interview answer:

“SpEL is used to query and manipulate object graphs at runtime in Spring configuration.”

26. What is @Value annotation?

Interview answer:

“@Value is used to inject values from property files, environment variables, or expressions.”

27. What is Java-based configuration?

Interview answer:

“It uses @Configuration and @Bean annotations to configure Spring beans using Java code instead of XML.”

28. What is @Configuration?

Interview answer:

“@Configuration indicates that a class contains bean definitions and is processed by the Spring container.”

29. How does Spring manage loose coupling?

Interview answer:

“By using dependency injection and interfaces, Spring ensures components are loosely coupled.”

30. Real-time use of Spring Core in projects?

Interview answer:

“In real projects, Spring Core is used to manage object creation, dependency injection, configuration, and lifecycle management across the application.”