Descriptive Exam 09

1. What is spring/spring framework?

Answer: Spring is one of the most widely used Java EE framework. Spring framework core concepts are "Dependency Injection" and "Aspect Oriented Programming".

Spring framework can be used in normal java applications also to achieve loose coupling between different components by implementing dependency injection and we can perform cross cutting tasks such as logging and authentication using spring support for aspect oriented programming.

I like spring because it provides a lot of features and different modules for specific tasks such as Spring MVC and Spring JDBC. Since it's an open source framework with a lot of online resources and active community members, working with Spring framework is easy and fun at same time.

2. What is IoC container / The Spring IoC container

Answer: It is the heart of the Spring Framework. The IoC container receives metadata from either an **XML file, Java annotations, or Java code**. The container gets its instructions on what objects to instantiate, configure, and assemble from simple Plain Old Java Objects (POJO) by reading the configuration metadata provided. These created objects through this process called **Spring Beans**.

The IoC container is responsible to instantiate, configure and assemble the objects. The IoC container gets informations from the XML file and works accordingly. The main tasks performed by IoC container are:

What is the role of IOC container in spring?

IOC container is responsible to:

- create the instance
- configure the instance, and
- assemble the dependencies

- to instantiate the application class
- to configure the object
- to assemble the dependencies between the objects

What are the types of IOC container in spring?

There are two types of IOC containers in spring framework.

1. BeanFactory
2. ApplicationContext

3. What is spring IoC?

Answer: Inversion of control is a process in which an object defines its dependencies without creating them. Ths object delegates the job of constructing such dependencies to an IoC container.

Inversion of control (IoC) is the principle where the control flow of a program is inverted: instead the programmer controls the flow of a program, the external sources (framework, services, other components) take control of it. As the name implies Inversion of control means now we have inverted the control of creating the object from our own using new operator to container or framework. Now it's the responsibility of container to create object as required.

The IoC container is responsible to instantiate, configure and assemble the objects. The IoC container gets informations from the XML file and works accordingly. The main tasks performed by IoC container are:

- to instantiate the application class
- to configure the object
- to assemble the dependencies between the objects

4. What is dependency injection injection in spring?

Answer: Dependency injection is a fundamental aspect of the spring framework, through which the spring container injects objects into other objects or dependencies. This allows for loose coupling of components and moves the responsibilty of managing components onto the container.

Dependency Injection in Spring can be done through constructors, setters or fields.

What is Dependency Injection?

Dependency injection is a pattern through which to implement IoC, where the control being inverted is the setting of object's dependencies.

The act of connecting objects with other objects, or "injecting" objects into other objects, is done by an assembler rather than by the objects themselves.

Dependency Injection in Spring

To set the *item* attribute in the example above, we can use metadata. Then, the container will read this metadata and use it to assemble beans at runtime.

5).Two ways to perform Dependency Injection in Spring framework

Spring framework provides two ways to inject dependency

- By Constructor
- By Setter method

5. What are the type of IoC?

Answer: There are two types of IoC

=>Dependency Injection

=>Dependency Lookup

6) What is AOP?

AOP is an acronym for Aspect Oriented Programming. It is a methodology that divides the program logic into pieces or parts or concerns.

It increases the modularity and the key unit is Aspect.

7) What are the advantages of spring AOP?

AOP enables you to dynamically add or remove concern before or after the business logic. It is **pluggable** and **easy to maintain**.

AOP breaks the program logic into distinct parts (called concerns). It is used to increase modularity by **cross-cutting concerns**.

A **cross-cutting concern** is a concern that can affect the whole application and should be centralized in one location in code as possible, such as transaction management, authentication, logging, security etc.

It provides the pluggable way to dynamically add the additional concern before, after or around the actual logic

8. What are the main features/concepts/terminologies of AOP?

What are the AOP terminology?

AOP terminologies or concepts are as follows:

- JoinPoint
- Advice
- Pointcut
- Aspect
- Introduction
- Target Object
- Interceptor
- AOP Proxy
- Weaving

9. What is AOP advice? What are the types of advice? How does it works?

Advice represents an action taken by an aspect at a particular join point. There are different types of advices:

- **Before Advice**: it executes before a join point.
- **After Returning Advice**: it executes after a joint point completes normally.
- **After Throwing Advice**: it executes if method exits by throwing an exception.
- **After (finally) Advice**: it executes after a join point regardless of join point exit whether normally or exceptional return.
- **Around Advice**: It executes before and after a join point.

What are the types of advice in AOP?

There are 5 types of advices in spring AOP.

1. Before Advice
2. After Advice
3. After Returning Advice
4. Throws Advice
5. Around Advice

9) What is JoinPoint?

JoinPoint is any point in your program such as field access, method execution, exception handling etc.

10. What is Spring means

Spring is a *lightweight* framework. It can be thought of as a *framework of frameworks* because it provides support to various frameworks such as Struts, Hibernate, Tapestry, EJB, JSF etc. The framework, in broader sense, can be defined as a structure where we find solution of the various technical problems.

The Spring framework comprises several modules such as IOC, AOP, DAO, Context, ORM, WEB, MVC etc.

It is a lightweight, loosely coupled and integrated framework for developing enterprise applications in java.

11) What are the advantages of spring framework?

1. Predefined Templates
2. Loose Coupling
3. Easy to test
4. Lightweight
5. Fast Development
6. Powerful Abstraction
7. Declarative support

12) What are the modules of spring framework?

1. Test
2. Spring Core Container
3. AOP, Aspects and Instrumentation

4. Data Access/Integration
5. Web

13) What is IOC and DI?

IOC (Inversion of Control) and DI (Dependency Injection) is a design pattern to provide loose coupling. It removes the dependency from the program.

14) What is Spring MVC?

A Spring MVC is a Java Framework which is used to develop dynamic web applications. It implements all the basic features of a core spring framework like Inversion of Control and Dependency Injection. It follows the Model-View-Controller design pattern.

- **Model** - A model contains the data of the application. Data can be a single object or a collection of objects.
- **Controller** - A controller contains the business logic of an application. Here, the @Controller annotation is used to mark the class as the controller.
- **View** - A view represents the provided information in a particular format. So, we can create a view page by using view technologies like JSP+JSTL, Apache Velocity, Thymeleaf, and FreeMarker.

15) What is the front controller of Spring MVC?

The front controller is a **DispatcherServlet** class present in **org.springframework.web.servlet** package. It dispatches the request to the appropriate controller and manages the flow of the application. It is required to specify the **DispatcherServlet** class in the web.xml file.

16) What are the advantages of Spring MVC Framework?

The following are the advantages of Spring MVC Framework : -

- **Separate roles** - The Spring MVC separates the application into three interconnected layers where each layer has its role.
- **Light-weight** - It uses light-weight servlet container to develop and deploy your application.

- **Powerful Configuration** - It provides a robust configuration for both framework and application classes that includes easy referencing across contexts, such as from web controllers to business objects and validators.
- **Rapid development** - The Spring MVC facilitates fast and parallel development.
- **Reusable business code** - Instead of creating new objects, it allows us to use the existing business objects.
- **Flexible Mapping** - It provides the specific annotations that easily redirect the page.

34) What does @Controller annotation?

The **@Controller** annotation marks the class as controller class. It is applied on the class.

@Controller

The @Controller annotation is used to indicate the class is a Spring controller. This annotation can be used to identify controllers for Spring MVC or Spring WebFlux.

35) What does @RequestMapping annotation?

The **@RequestMapping** annotation maps the request with the method. It is applied on the method.

36)Spring Dao Supports

The Data Access Object (DAO) support in Spring is aimed at making it easy to work with data access technologies like JDBC, Hibernate, JPA or JDO in a consistent way. This allows one to switch between the aforementioned persistence technologies fairly easily and it also allows one to code without worrying about catching exceptions that are specific to each technology.

- **What is Spring DAO?**

Spring DAO support is provided to work with data access technologies like JDBC, Hibernate in a consistent and easy way. For example we have `JdbcDaoSupport`, `HibernateDaoSupport`, `JdoDaoSupport` and `JpaDaoSupport` for respective technologies.

Spring DAO also provides consistency in exception hierarchy and we don’t need to catch specific exceptions.

37)What is a dao spring?

It's a design pattern in which a data access object (**DAO**) is an object that provides an abstract interface to some type of database or other persistence mechanisms. ... **Spring** data access framework is provided to integrate with different persistence frameworks like JDBC, Hibernate, JPA, iBatis etc.

What are design patterns used in Spring Framework?

- **Singleton Pattern:** Singleton-scoped beans
- **Factory Pattern:** Bean Factory classes
- **Prototype Pattern:** Prototype-scoped beans
- **Adapter Pattern:** Spring Web and Spring MVC
- **Proxy Pattern:** Spring Aspect Oriented Programming support
- **Template Method Pattern:** *JdbcTemplate*, *HibernateTemplate*, etc.
- **Front Controller:** Spring MVC *DispatcherServlet*
- **Data Access Object:** Spring DAO support
- **Model View Controller:** Spring MVC

• What are some of the important Spring annotations you have used?

Some of the Spring annotations that I have used in my project are:

- **@Controller** – for controller classes in Spring MVC project.
- **@RequestMapping** – for configuring URI mapping in controller handler methods. This is a very important annotation, so you should go through Spring MVC RequestMapping Annotation Examples
- **@ResponseBody** – for sending Object as response, usually for sending XML or JSON data as response.
- **@PathVariable** – for mapping dynamic values from the URI to handler method arguments.
- **@Autowired** – for autowiring dependencies in spring beans.
- **@Qualifier** – with @Autowired annotation to avoid confusion when multiple instances of bean type is present.
- **@Service** – for service classes.
- **@Scope** – for configuring scope of the spring bean.
- **@Configuration**, **@ComponentScan** and **@Bean** – for java based configurations.
- AspectJ annotations for configuring aspects and advices, **@Aspect**, **@Before**, **@After**, **@Around**, **@Pointcut** etc.

• What is Spring JdbcTemplate class and how to use it?

Spring Framework provides excellent integration with JDBC API and provides JdbcTemplate utility class that we can use to avoid bolier-plate code from our database operations logic such as Opening/Closing Connection, ResultSet, PreparedStatement etc.