Core Specific- Java Assignment 9

Q1. What is Spring Framework?

Ans:

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform.

A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

Q2. What are the features of Spring Framework?

Ans:

Few features of Spring are:

- Core technologies: dependency injection, events, resources, i18n, validation, data binding, type conversion, SpEL, AOP.
- Testing: mock objects, TestContext framework, Spring MVC Test, WebTestClient.
- Data Access: transactions, DAO support, JDBC, ORM, Marshalling XML.
- Spring MVC and Spring WebFlux web frameworks.
- Integration: remoting, JMS, JCA, JMX, email, tasks, scheduling, cache and observability.
- Languages: Kotlin, Groovy, dynamic languages.

Q3. What is a Spring configuration file?

Ans:

A Spring configuration file is an XML file that contains the classes information. It describes how those classes are configured as well as introduced to each other.

Core Specific- Java Assignment 9

Q4. What do you mean by IoC Container?

Ans:

Spring IoC (Inversion of Control) Container is the core of Spring Framework. It creates the objects, configures and assembles their dependencies, manages their entire life cycle. The Container uses Dependency Injection (DI) to manage the components that make up the application. It gets the information about the objects from a configuration file(XML) or Java Code or Java Annotations and Java POJO class. These objects are called Beans. Since the Controlling of Java objects and their lifecycle is not done by the developers, hence the name Inversion Of Control.
There are 2 types of IoC containers:

- BeanFactory
- ApplicationContext

That means if you want to use an IoC container in spring whether we need to use a BeanFactory or ApplicationContext. The BeanFactory is the most basic version of IoC containers, and the ApplicationContext extends the features of BeanFactory. The followings are some of the main features of Spring IoC,

- Creating Object for us,
- Managing our objects,
- Helping our application to be configurable,
- Managing dependencies

Q5. What do you understand by Dependency Injection?

Ans:

In object-oriented programming (OOP) software design, dependency injection is the process of supplying a resource that a given piece of code requires. The required resource, which is often a component of the application itself, is called a dependency.

Q6. Explain the difference between constructor and setter injection?

Ans:

Constructor-based DI fixes the order in which the dependencies need to be injected.

Setter-based DI helps us to inject the dependency only when it is required, as opposed to requiring it at construction time.

Spring code generation library doesn't support constructor injection so it will not be able to create proxy.

Core Specific- Java Assignment 9

Q7. What are Spring Beans?

Ans:

Spring Bean is nothing special, any object in the Spring framework that we initialize through Spring container is called Spring Bean. Any normal Java POJO class can be a Spring Bean if it's configured to be initialized via container by providing configuration metadata information.

Q8. What are the bean scopes available in Spring?

Ans:

There are five types of spring bean scopes:

1. **singleton** - only one instance of the spring bean will be created for the spring container. This is the default spring bean scope. While using this scope, make sure bean doesn't have shared instance variables otherwise it might lead to data inconsistency issues.
2. **prototype** – A new instance will be created every time the bean is requested from the spring container.
3. **request** – This is same as prototype scope , however it's meant to be used for web applications. A new instance of the bean will be created for each HTTP request.
4. **session** – A new bean will be created for each HTTP session by the container.
5. **global-session** – This is used to create global session beans for Portlet applications.

Q9. What is Autowiring and name the different modes of it?

Ans: Spring is an open-source application development framework of Java that allows you to create robust enterprise applications using Plain Old Java Objects (**POJO** in short). The Spring framework can inject dependencies automatically. The Spring container detects those dependencies specified in the configuration file and @ the relationship between the beans. This is referred to as autowiring in Spring. An autowired application requires fewer lines of code comparatively but at the same time, it provides very little flexibility to the programmer.

Spring supports the following autowiring modes:

no: It's the default autowiring mode. It means *no* autowiring.
byName: The *byName* mode injects the object dependency according to name of the bean. In such case, property name and bean name should be same. It internally calls setter method.
byType: The *byType* mode injects the object dependency according to type. So, it can have different property name and bean name. It internally calls setter method.
constructor: The *constructor* mode injects the dependency by calling the constructor of the class. It calls the constructor having large number of parameters.
autodetect: In this mode, Spring first tries to autowire by constructor. If this fails, it tries to autowire by using *byType* .

.

Q10. Explain Bean life cycle in Spring Bean Factory Container.

Ans:

Bean life cycle is managed by the spring container. When we run the program then, first of all, the spring container gets started. After that, the container creates the instance of a bean as per the request, and then dependencies are injected. And finally, the bean is destroyed when the spring container is closed.