**Assignment Questions 9**

💡 **Q1.What is Spring Framework?**

Spring Framework is a popular open-source Java framework that provides comprehensive support for building enterprise-grade applications. It offers features like dependency injection (IoC), aspect-oriented programming (AOP), data access, web development, security, and more. Spring promotes a modular and loosely coupled architecture, simplifying application development, improving testability, and facilitating integration with other technologies and frameworks.

💡 **Q2.What are the features of Spring Framework?**

Features of Spring Framework:

1. Dependency Injection (IoC): Manages object dependencies, enhancing flexibility and testability.

2. Aspect-Oriented Programming (AOP): Allows modularization of cross-cutting concerns.

3. Data Access: Offers support for various data access technologies like JDBC, JPA, and Hibernate.

4. MVC Framework: Provides a flexible Model-View-Controller architecture for web development.

5. Security: Facilitates securing applications through various mechanisms.

6. Transaction Management: Supports declarative and programmatic transaction management.

7. Integration: Enables integration with other technologies and frameworks.

💡 **Q3.What is a Spring configuration file?**

A Spring configuration file is an XML or Java-based file that defines the configuration of a Spring application. It contains bean definitions, dependency injection settings, aspect declarations, and other configurations required to wire components and manage the application's behavior.

💡 **Q4.What do you mean by IoC Container?**

IoC (Inversion of Control) Container is a core component of the Spring Framework. It manages the lifecycle of application objects, performs dependency injection, and provides the necessary components (beans) required by the application, effectively implementing the IoC design pattern.

💡 **Q5.What do you understand by Dependency Injection?**

Dependency Injection is a design pattern used in software development to achieve loose coupling between components. It allows the injection of required dependencies (objects or services) into a class rather than the class creating or managing those dependencies itself, promoting modularity and flexibility in the application.

💡 **Q6.Explain the difference between constructor and setter injection?**

Constructor Injection:

- Dependencies are provided through the class constructor.

- Immutable and mandatory dependencies are best suited for constructor injection.

- The class is initialized with all required dependencies during object creation.

- Safer and ensures that the object is in a valid state from the beginning.

Setter Injection:

- Dependencies are provided through setter methods.

- Mutable or optional dependencies are suitable for setter injection.

- Allows changing or reconfiguring dependencies after object creation.

- Offers flexibility but may leave the object in an incomplete state until all required dependencies are set.

💡 **Q7.What are Spring Beans?**

Spring Beans are the objects managed by the Spring IoC container. They are the fundamental building blocks of a Spring application, representing the various components and services that can be wired together, and their configurations are defined in the Spring configuration file.

💡 **Q8.What are the bean scopes available in Spring?**

In Spring, the following bean scopes are available:

1. Singleton: A single instance of the bean is created per Spring IoC container.

2. Prototype: A new instance of the bean is created every time it is requested.

3. Request: A new bean instance is created for each HTTP request.

4. Session: A new bean instance is created for each HTTP session.

5. Global Session: Similar to session scope but for global HTTP sessions (portlet-based applications).

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

Autowiring in Spring is a feature that allows automatic injection of dependencies into a bean without explicitly configuring them in the Spring configuration file. The different modes of autowiring are:

1. No: Default mode. No autowiring is applied, and dependencies are manually configured.

2. byName: Autowiring is done by matching bean names with property names.

3. byType: Autowiring is done by matching bean types with property types.

4. constructor: Autowiring is done by matching constructor arguments with bean types.

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

The Bean life cycle in Spring Bean Factory Container involves the following stages:

1. Instantiation: The container creates a new bean instance.

2. Population: Dependencies are injected into the bean.

3. Initialization: Bean initialization methods are called after dependency injection.

4. Use: The bean is available for use in the application.

5. Destruction: The container calls the bean's destruction methods when the application context is closed or the bean is no longer needed.