**Assignment Questions 9**

<aside> 💡 **Q1.What is Spring Framework?**

</aside>

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

</aside>

<aside> 💡 **Q3.What is a Spring configuration file?**

</aside>

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

</aside>

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

</aside>

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

</aside>

<aside> 💡 **Q7.What are Spring Beans?**

</aside>

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

</aside>

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

</aside>

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

</aside>

Q1. What is Spring Framework? A1. The Spring Framework is an open-source application framework for Java that provides comprehensive infrastructure support for developing Java applications. It offers features like dependency injection, aspect-oriented programming, and facilitates the development of modular and loosely coupled applications.

Q2. What are the features of Spring Framework? A2. The features of the Spring Framework include dependency injection (DI), aspect-oriented programming (AOP), robust transaction management, MVC web framework, integration with various technologies and frameworks, support for data access, and comprehensive testing capabilities.

Q3. What is a Spring configuration file? A3. A Spring configuration file is an XML or Java-based file that contains the configuration metadata for a Spring application. It defines beans, their dependencies, and other configuration settings required for the application to run.

Q4. What do you mean by IoC Container? A4. IoC (Inversion of Control) Container is a core component of the Spring Framework that manages the lifecycle of objects (beans) and their dependencies. It creates, configures, and injects dependencies into objects, allowing loose coupling and promoting modular and flexible application development.

Q5. What do you understand by Dependency Injection? A5. Dependency Injection (DI) is a design pattern implemented in the Spring Framework. It allows objects to be loosely coupled by injecting their dependencies instead of creating them internally. DI helps in achieving modularity, testability, and easier maintenance of code.

Q6. Explain the difference between constructor and setter injection? A6. Constructor injection involves passing dependencies through a class's constructor, while setter injection involves setting dependencies using setter methods. Constructor injection enforces the immutability of dependencies, while setter injection allows flexibility in changing dependencies.

Q7. What are Spring Beans? A7. Spring Beans are Java objects managed by the Spring IoC Container. They are the fundamental building blocks of a Spring application and are defined in the Spring configuration file. Beans are created, assembled, and managed by the container.

Q8. What are the bean scopes available in Spring? A8. The bean scopes available in Spring are singleton, prototype, request, session, and application. The singleton scope creates a single instance of a bean, while the prototype scope creates a new instance for each request. The request, session, and application scopes are specific to web applications.

Q9. What is Autowiring and name the different modes of it? A9. Autowiring is a feature in Spring that allows automatic injection of dependencies into beans without explicit configuration. The different modes of autowiring are "byName," "byType," "constructor," "autodetect," and "no" autowiring.

Q10. Explain Bean lifecycle in Spring Bean Factory Container. A10. The lifecycle of a bean in the Spring Bean Factory Container involves the following steps: bean instantiation, property population (dependency injection), initialization (calling init methods), and destruction (calling destroy methods). The container manages these steps, allowing customization and control over the bean's lifecycle.