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 : Features

* 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.

Q3.What is a Spring configuration file?

Ans : Spring configuration file is a file with . xml extension and the file contains information about the classes and interfaces and their dependencies. Using this file the spring container controls the life cycle of a spring bean and also Dependency Injection is achieved.

Q4.What do you mean by IoC Container?

Ans : IoC container is a framework for implementing automated dependency injection. It contains object creation for the longer ways to use and injects dependencies within the class.

Q5.What do you understand by Dependency Injection?

Ans :  Dependency Injection is a programming technique in which an object or function receives other objects or functions that it depends on. Dependency injection aims to separate the concerns of constructing objects and using them, leading to loosely coupled programs.

Q6.Explain the difference between constructor and setter injection?

Ans :

| Sr. No. | Key | Constructor based Injection | Setter based Injection |
| --- | --- | --- | --- |
| 1 | Circular | It doesn’t allow to create circular dependency | It doesn’t check the circular dependency |
| 2 | Ordering | 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. |
| 3 | MutilThread Environment | Combining with final fields, constructor injection gives extra safety in multithreaded environment | No extra benefit in setter injection |
| 4 | Spring Code generation Library | Spring code generation library doesn’t support constructor injection so it will not be able to create proxy. It will force you to use no-argument constructor. | Spring framework level code uses setter injection |
| 5 | Use Case | It should be used for mandatory dependencies | It should be used for optional dependencies. |

Q7.What are Spring Beans?

Ans : 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 : The latest version of the Spring framework defines 6 types of scopes:

* singleton.
* prototype.
* request.
* session.
* application.
* websocket.

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

Ans : Autowiring happens by placing an instance of one bean into the desired field in an instance of another bean. Both classes should be beans, i.e. they should be defined to live in the application context.

Autowiring is of 4 types

* 1) byName.
* 2) byType.
* 3) constructor.
* 4) autodetect.

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

Ans :

1. The Spring container instantiates the bean from the bean's definition in the XML file.
2. Spring populates all of the properties using the dependency injection, as specified in the bean definition.
3. The factory calls setBeanName() by passing the bean's ID if the bean implements the BeanNameAware interface.