1) What is Spring FrameWork??

Spring is a lightweight framework and it is a framework of frameworks because it provides support to various framework such as Structs, Hibernate , EJB and JSF etc.

It can be defined as a structure where we find solution of the various technical problems.

Spring framework comprises server module such as IOC, AOP, DAO, Context, ORM, Web, MVC etc.

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.

2) Explain Dependency Injection?

Dependency Injection (DI) is a design pattern used to implement IoC. It allows the creation of dependent objects outside of a class and provides those objects to a class through different ways. Using DI, we move the creation and binding of the dependent objects outside of the class that depends on them.

The Dependency Injection pattern involves 3 types of classes.

1. **Client Class:** The client class (dependent class) is a class which depends on the service class
2. **Service Class:** The service class (dependency) is a class that provides service to the client class.
3. **Injector Class:** The injector class injects the service class object into the client class.

**Types of Dependency Injection**

1. As you have seen above, the injector class injects the service (dependency) to the client (dependent). The injector class injects dependencies broadly in three ways: through a constructor, through a property, or through a method.
2. **Constructor Injection:** In the constructor injection, the injector supplies the service (dependency) through the client class constructor.
3. **Property Injection:** In the property injection (aka the Setter Injection), the injector supplies the dependency through a public property of the client class.
4. **Method Injection:** In this type of injection, the client class implements an interface which declares the method(s) to supply the dependency and the injector uses this interface to supply the dependency to the client class.

3) What is IOC Container?

IoC Container (a.k.a. DI Container) is a framework for implementing automatic dependency injection. It manages object creation and it's life-time, and also injects dependencies to the class.

The IoC container creates an object of the specified class and also injects all the dependency objects through a constructor, a property or a method at run time and disposes it at the appropriate time. This is done so that we don't have to create and manage objects manually.

All the containers must provide easy support for the following DI lifecycle.

* **Register:** The container must know which dependency to instantiate when it encounters a particular type. This process is called registration. Basically, it must include some way to register type-mapping.
* **Resolve:** When using the IoC container, we don't need to create objects manually. The container does it for us. This is called resolution. The container must include some methods to resolve the specified type; the container creates an object of the specified type, injects the required dependencies if any and returns the object.
* **Dispose:** The container must manage the lifetime of the dependent objects. Most IoC containers include different lifetimemanagers to manage an object's lifecycle and dispose it.

4) )What is BeanFactory?

The **BeanFactory**. The **BeanFactory** is the actual container which instantiates, configures, and manages a number of beans. These beans typically collaborate with one another, and thus have dependencies between themselves.

5)Explain Autowiring?

Autowiring feature of spring framework enables you to inject the object dependency implicitly. It internally uses setter or constructor injection.

Autowiring can't be used to inject primitive and string values. It works with reference only

Advantage of Autowiring :

* It requires the less code because we don't need to write the code to inject the dependency explicitly.

Disadvantage of Autowiring :

* No control of programmer.
* It can't be used for primitive and string values.

6) Different types of autowiring?

The XML-configuration-based autowiring functionality has five modes – no, byName, byType, constructor, and autodetect. The default mode is no.