**What is Spring framework?**

Spring is one of the most widely used open source Java EE framework. Spring provides Dependency Injection, IOC container, Aspect oriented programming, Spring MVC flow and several useful API for Java EE developer. Spring can be used to achieve loose coupling between different components by implementing dependency injection and we can perform cross cutting tasks such as logging and authentication using aspect oriented programming. Spring framework targets to make Java EE development easier to use and promote good programming practice by enabling a POJO- based programming model.

I like spring because it provides a lot features and different modules for specific task such as Spring MVC and Spring JDBC.

**What are the benefits of Spring Framework?**

Lightweight: It is lightweight when it comes to size and transparency. The basic version is around 2 MB.

Inversion od control (IOC): Loose coupling is achieved in Spring with the Inversion of Control Technique. The objects give their dependencies instead of creating or looking for dependent object. In other word, DI is one of the design pattern, which allows injecting dependency on object, instead of object resolving the dependency.

Aspect oriented (AOP): Separates application business logics from the system services.

Container: Spring contains and manages the life cycle and configuration of the application objects

MVC framework: Spring MVC framework can be used to create web applications as well as restful web services capable of returning XML as well as JSON response.

Also, it supports transaction management, JDBC operations, File uploading, Exception handing etc with very little configurations, either by using annotation or by spring bean configuration.

**What are the Spring framework modules?**

Spring Context – for dependency injection

Spring AOP- for aspect oriented programming

Spring DAO (data access object)- for database operations using DAO pattern.

Spring JDBC- for JDBC and Data Source support

Spring ORM- for ORM tools support

Spring Web Module- for creating web applications

Spring MVC – implementation for creating web applications, web services

**Explain the Core Container (Application Context) module?**

This is the basic Spring module, which provides the fundamental of the Spring framework, BeanFactory is the heart of any spring-based application. Spring framework was built on the top of this module, which makes the Spring container.

**What is Dependency Injection in Spring?**

DI is an aspect of Inversion of Control. The concept says that you do not create your objects but describe how they should be created. You don’t directly connect your component and services together, but describe which services are needed by which components in a configuration file. A container ( IOC container) is responsible for hooking it all up.

**BeanFactory- BeanFactory implementation example?**

A BeanFactory is an implementation of the factory pattern that applies Inversion of Control to separate the application’s configuration and dependencies from the actual application code.

The most commonly used Beanfactory implementation is the XmlBeanFactory class.

**XMLBeanFactory**

The most useful one is org.springframework.beans.factory.xml.XmlBeanFactory, which loads its beans based on the definitions contained in an XML file. This container reads the configuration metadata form an XML file and uses it to create a fully configured system or application.

**What are the different types of IoC (dependency injection)?**

Constructor based DI: It is accomplished when the container invokes a class constructor with a number of arguments, each representing a dependency on other class.

Setter- based DI: It is accomplished by the container calling setter methods on your beans after invoking a no-argument constructor or no- argument static factory method to instantiate your bean.

**Explain the AOP module?**

The AOP module is used for developing aspects for our Spring-enabled application. Much of the support has been provided by the AOP Alliance in order to ensure the interoperability between Spring and other AOP frameworks. This module also introduces metadata programming to Spring.

**Explain the JDBC abstraction and DAO module?**

With the JDBC abstraction and DAO module we can be sure that we keep up the database code clean and simple, and prevent problems that results from the failure to close the database resources. It provides a layer of meaningful exceptions on top of the error messages given by the database servers. It also makes use of Spring’s AOP module to provide transaction management services for objects in a Spring application.

Explain the object/relation mapping integration module?

Spring also support for using of an ORM tool over straight JDBC by providing ORM module. Spring provides support to tie into several popular ORM framework, including Hibernate, JDO. Spring’s transaction management support each of these ORM frameworks as well as JDBC.

Explain the Web Module?

The Spring web module is built on the application context module, providing a context that is appropriate for web-based applications. This module also contains support for several web-oriented tasks such as transparently handling multipart requests for the file uploads and programmatic binding of request parameters to your business objects. It also contains integration support with Jakarta Struts.

Explain the Spring MVC module:

MVC framework is provided by the Spring for building web applications. Spring can easily be integrated with other MVC frameworks, but Spring’s MVC framework is a better choice, since it uses IoC to provide for a clean separation of controller logic from business objects. With Spring MVC you can declaratively bind the request parameters to your business objects.

**Spring configuration file:**

Spring configuration file is an XML file. This file contains the classes information and describes how these classes are configured and introduced to each other.

**What is Spring IoC container?**

The Spring IoC responsible for creating the objects, managing them( with dependency injection (DI), wiring them together, configuring them, as also managing the complete lifecycle.

What are the benefits of IOC?

IOC or dependency injection minimizes the amount of code in an application. It makes easy to test application, since no singletons or JNDI lookup mechanism are required in unit test. Loose coupling is promoted with minimal effort and least intrusive mechanism.

What are the common implementations of the ApplicationContext?

The FileSystemXmlApplicationContext container loads the definitions of the beans from an XML file. The full path of the XML bean configuration file must be provided to the constructor.

The ClassPathXmlApplicationContext container also loads the definitions of the bean from an XML file. Here, you need to set CLASSPATH property because this container will look bean configuration XML file in CLASSPATH.

The WebXmlApplicationContext: container loads the XML file with defunitions of all beans from within a web application.

What is the difference between Bean Factory and ApplicationContext?

Both BeanFactory and ApplicationContext provides a way to get bean form Spring IOC container by getBean(“bean name”), but there is some difference in there working and features provided by them. One difference between bean factory and application context is that BeanFactory only instantiate bean when you call getBean() method while ApplicationContext instantiates SingleTon bean when the container is started, It doesn’t wait for getBean to be called.

What does a Spring application look like?

An interface that defines the functions.

The implementation that contains properties, its setter and getter methods, functions etc.

Spring AOP

The Spring configuration XML file

Client program that uses the function