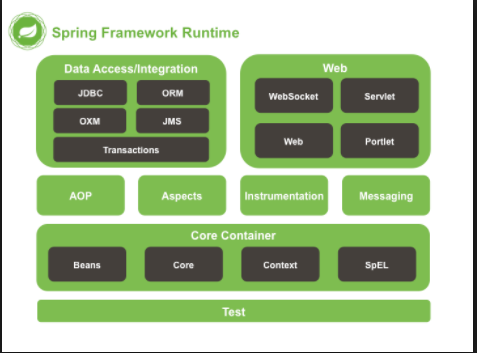
What is Spring Framework?

It is an application framework for Java Enterprise Application Development. It is light weight and loosely coupled. It has layered architecture which allows us to select components required. The most important feature is that it can be integrated with other frameworks like JSF, Struts, Hibernate easily.

Which is latest version of Spring framework?

Latest Version 5.0

Explain architecture of Spring framework?



There are Eight layers in Spring Framework

* Data Access/Integration

It consists of JDBC, ORM, OXM and JMS (For producing and consuming messages)

* Web

It consists of WebSocket ( Two way client-server communication), Servlet, Web (For web MVC application) and Portlet.

* Core Container

It consists of Core, Beans, Context and SpEL.

* + Core – Provides fundamental part of Spring Framework like DI and IOC.
  + Beans – Provides Bean factory. It is container for beans.
  + Context – it Provides Application Context. It is container for beans.
  + SpEL- Provides support for Expression Language.
* AOP

This layer is for cross-cutting concerns like logging, Transaction management and security.

* Aspects

Provides integration support for AspectJ which an important feature of Spring AOP.

* Instrumentation

This layer provides support to class instrumentation and classloader implementations

* Messaging

Provides support for STOMP (Streaming Text Orientated Messaging Protocol ).

What are advantage of Spring Framework?

* Layered architecture
* POJO based programming model which enables Continuous Integration.
* Open Source
* Light Weight

What are features of Spring Framework?

* Inversion Control - Loose coupling is achieved in Spring, with the Inversion of Control technique. The objects give their dependencies instead of creating or looking for dependent objects.
* AOP – Supports AOP.
* MVC Framework.
* Transaction management.
* Light Weight - Spring is lightweight when it comes to size and transparency. The basic version of spring framework is around 2MB.
* Exception Handling- Spring provides a convenient API to translate technology-specific exceptions (thrown by JDBC, Hibernate, or JDO) into consistent, unchecked exceptions.

What is Spring IOC Container?

It creates Objects, wires them, configures them and manages their life cycle. To manage these beans spring uses Dependency Injection. The container receives these instructions by reading configuration files (XML or Java Based Configuration Annotation).

What is Bean Factory?

It is basic Spring Container.

What is Application Context?

It is built upon Bean factory. It provides more feature on top of Bean Factory.

List out some implementations of Bean Factory and Application Context.

Bean Factory- XmlBeanFactory

Application Context – ClassPathApplicationContext, AnnotationConfigApplicationContext.

Differentiate between Bean Factory and Application Context

|  |  |
| --- | --- |
| **Bean Factory** | **Application Context** |
| Basic container | Advance container |
| Supports only two scopes singleton and prototype | Supports all five scopes |
| Uses lazy initialization | Uses eager initialization |
| Doesn’t supports annotation based dependency | Supports annotation based dependency |
| Doesn’t supports internationalization | Supports internationalization |

What is IOC?

The principle/Feature where instead of client doing activity of managing the object. Control is taken over by the container/Framework is called inversion of control. Dependency Injection is one flavor of IOC.

What is Dependency Injection?

Dependency Injection (DI) is a software design pattern that implements inversion of control for resolving dependencies. It is used to define Object dependencies between each other. This concept says that you do not create your objects but describe how they should be created.

What are different type of Dependency Injection?

There are three types of Dependency Injection

* Setter Injection
* Constructor Injection
* Interface Injection ( Avalon)

Note : Spring only supports setter and constructor based injection.

Differentiate between setter injection and constructor injection.

