1.What is mean by spring framework

------------------------------------

The Spring Framework (Spring) is an open-source application framework that provides infrastructure support for developing Java applications.

One of the most popular Java Enterprise Edition (Java EE) frameworks, Spring helps developers create high performing applications using plain old Java objects (POJOs).

2.what is mean by IOC and loosely coupled?

------------------------------------------

Spring IoC Container is the core of Spring Framework. It creates the objects, configures and assembles their dependencies, manages their entire life cycle.

The Container uses Dependency Injection(DI) to manage the components that make up the application.

loosely coupling

----------------

Loose coupling in Java means that the classes are independent of each other. The only knowledge one class has about the other class is what the other class has exposed through its interfaces in loose coupling

tight coupling

-----------------

When two classes are highly dependent on each other, it is called tight coupling. It occurs when a class takes too many responsibilities or where a change in one class

requires changes in the other class. In tight coupling, an object (parent object) creates another object (child object) for its usage.

3.Difference between Application Context and Bean Factory?

---------------------------------------------------------

The Benefactory is actual container which instantiates, configures and manages a number of beans these beans are typically collaborate with another

when we call the getBean("springbeanref").So when we call the getBean("springbeanref") at the time of spring bean life cycle starts.

-

The application context is the central interface within a spring application that is used for providing configuration information to the application

It implements Beanfactory interface

Its main function is to support the creation of big business application

WHEN CONTAINER STARTS.It won't wait until getBean("springbeanref") is called.

4.What is mean by Autowired? Explain Types?

---------------------------------------------

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.

Types

-----

no: It’s the default autowiring mode. It means no autowiring.

byName: The byName mode injects the object dependency according to name of the bean. In such a case, the property and bean name should be the same. It internally calls the setter method.

byType: The byType mode injects the object dependency according to type. So it can have a different property and bean name. It internally calls the setter method.

constructor: The constructor mode injects the dependency by calling the constructor of the class. It calls the constructor having a large number of parameters.

autodetect: In this mode, Spring first tries to autowire by the constructor. If this fails, it tries to autowire by using byType.

5.Explain Spring Bean life Cycle?

----------------------------------

Bean life cycle is managed by the spring container. When we run the program then, first of all, the spring container gets started. After that, the container creates the instance of a bean as per the request,

and then dependencies are injected. And finally, the bean is destroyed when the spring container is closed. Therefore, if we want to execute some code on the bean instantiation and just after closing the spring container,

then we can write that code inside the custom init() method and the destroy() method.

6.What is mean by Singleton how can we make singleton class?

--------------------------------------------------------------

Singleton class is a class that controls the object creation. It means the singleton class allows us to create a single object of the class, at a time. It is usually used to control access to resources, such as database connections or sockets.

It ensures that only one connection is made and a thread can access the connection at a time.

To create a singleton class, we must follow the steps, given below:

-------------------------------------------------------------------

.Ensure that only one instance of the class exists.

.Provide global access to that instance by:

.Declaring all constructors of the class to be private.

.Providing a static method that returns a reference to the instance. The lazy initialization concept is used to write the static methods.

.The instance is stored as a private static variable.

7.Types of Spring Bean Scopes?

--------------------------------

.Singleton: only one instance of the spring bean will be created for the spring container.this is the default spring bean scope.while using this scope,make sure bean doesn't have shared instance variables otherwise it might lead to data inconsistency issues

.Prototype: A new instance will be created every time the bean is requested from the spring container.

.Request: This is same as prototype scope however it's meant to be used for web application a new instance of the bean will be created each HTTP request.

.Session: A new bean will be created for each HTTP sessions by the container.

.Global-Session: This is used to create globalsession beans for portlet application.

8.Spring MVC Workflow?

-------------------------

1. Request

------------

The first step in the MVC flow is when a request is received by the Dispatcher Servlet.

2. Dispatcher Servlet

---------------------

Now, the Dispatcher Servlet will with the help of Handler Mapping understand the Controller class name associated with the received request. Once the Dispatcher Servlet knows which Controller will be able to handle the request, it will transfer the request to it.

3. Controller

--------------

The Controller will process the request based on appropriate methods and will return it to Model Data and View Name.

4. Model and View

--------------------

It will return the processed data to the Dispatcher Servlet.

5. View Resolver

------------------

Once Model and View receive the data, Dispatcher Servlet will transfer it to the View Resolver to get the actual view page.

6. View

------------

Finally, the Dispatcher Servlet will pass the Model object (results) to the view page. This is the final step of the flow where the results will be displayed.

9.Spring Annotations?

10.What is mean by @Qulifier?Example?

----------------------------------------

The @Qualifier annotation is used to resolve the autowiring conflict, when there are multiple beans of same type.

The @Qualifier annotation can be used on any class annotated with @Component or on method annotated with @Bean. This annotation can also be applied on constructor arguments or method parameters.

11.Spring XML based Configuration?EX:database,View Resolver

--------------------------------------------------------------

12.@Post vs @PUT

------------------

POST means "create new" as in "Here is the input for creating a user

PUT means "insert, replace if already exists"

13.HTTP StatusCodes ex:400,503--------

200 OK

201 Created

202 Accepted

204 No Content

400 Bad Request

401 Unauthorized

403 Forbidden

404 Not Found

500 Internal Server Error

502 Bad Gateway

503 Service Unavailable

14.What are the features of Spring Framework?

-----------------------------------------------

.Lightweight

.Inversion of Control (IoC)

.Aspect-oriented Programming (AOP)

.JDBC exception handling

.Transaction Management

.Dependency Injection

15.What do you mean by IoC (Inversion of Control) Container?

------------------------------------------------------------------

Spring IoC Container is the core of Spring Framework. It creates the objects, configures and assembles their dependencies, manages their entire life cycle. The Container uses Dependency Injection(DI) to manage the components that make up the application. These objects are called Beans.

16.What do you understand by Dependency Injection?

------------------------------------------------------

Dependency Injection is a fundamental aspect of the Spring framework, through which the Spring container “injects” objects into other objects or “dependencies”. Simply put, this allows for loose coupling of components and moves the responsibility of managing components onto the container

17.Explain the difference between constructor and setter injection?

------------------------------------------------------------------------

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.

18.What are Spring Beans?

----------------------------

Spring beans are the objects which are created and managed completely by spring container. These beans are the heart of the application. Beans can be defined in spring either by using XML configuration or by using Annotation.

19.How is the configuration meta data provided to the spring container?

------------------------------------------------------------------------

.XML based configuration file.

.Annotation-based configuration.

.Java-based configuration.

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

----------------------------------------------------------------

Bean life cycle is managed by the spring container. When we run the program then, first of all, the spring container gets started. After that, the container creates the instance of a bean as per the request, and then dependencies are injected.

And finally, the bean is destroyed when the spring container is closed.

24.What are the different components of a Spring application?

25.How is the dispatcher servlet instantiated

-----------------------------------------------

The dispatcher servlet is instantiated by means of servlet containers such as Tomcat. The Dispatcher Servlet should be defined in web.xml The DispatcherServlet is instantiated by Servlet containers like Tomcat

Ex: <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

26. What is the Spring MVC framework?

--------------------------------------------------------

Spring MVC is a Java framework that is used to develop web applications. It is built on a Model-View-Controller (MVC) pattern and possesses all the basic features of a spring framework, such as Dependency Injection, Inversion of Control. The architectural design of a Spring MVC can be used to develop flexible web applications.

It basically separates the different aspects of the application, like input logic, UI logic, and business logic.

27.What is the importance of the web.xml in Spring MVC

-------------------------------------------------------

In Spring MVC, web.xml used to be the place, where you needed to declare and configure Dispatcher Servlet, which is a Front Controller, receiving all the requests and dispatching to all the other components such as Controllers. Fortunately, Spring offers a convenient, XML-free way of declaring Dispatcher Servlet.

28. What is the importance of session scope?

----------------------------------------------------------------

Session scopes are used to create bean instances for HTTP sessions. This would mean that a single bean can be used for serving multiple HTTP requests. The scope of the bean can be defined by means of using scope attribute or using @Scope or @SessionScope annotations

29.What is the importance of @Required annotation

------------------------------------------------------------------------

The @Required annotation applies to bean property setter methods and it indicates that the affected bean property must be populated in XML configuration file at configuration time. Otherwise, the container throws a BeanInitializationException exception

https://www.interviewbit.com/spring-interview-questions/