ASSIGNMENT

1.what is IOC,Dependency Injection?

A IOC: IOC stands for Inversion Of Control .It is also known as Dependency Injection(DI).The spring container uses DI to manage the components that build up an application and these objects are called SpringBeans. Spring implements DI by either an XML configuration file or annotations.

Dependency Injection:- It is a method through which we can achieve inversion of control. It is a way to inject the dependency of a framework component by the following ways of spring are:

1. Setter/Getter injection
2. Constructor injection

Consider a class Employee

```
Class Employee{
   Private int id;
   Private String name;
   Private Address address;
    Employee(){
        Id=10;
        Name=”name”;
         Address=new Address();
   }
}
```

And consider class Address

```
Class Address{
  Private String street;
  Private String city;
Address(){
   Street=”test”;
    City=”test1”;
  }
}
```

- In the above code the address class values will be set only when the Employee class is instantiated, which is dependency of Address class on Employee class.
- And spring sloves this problem using Dependency injection concept by providing two ways to inject this dependency.
    1. Setter injection
    2. Constructor injection

Setter injection:- Setter method in Employee class which takes a reference of Address class.

```
  Public void setAddress(Address addr){
        This.address=addr;
    }
```

Constructor injection:-Constructor in Employee class which accepts Address.

```
   Employee(Address addr){
        This.address=addr;
      }
```

- In this way the address class values can be set independently using either setter/constructor injection
- The main goal of the Inversion of control and Dependency Injection is to remove dependencies of an application. This makes the system more decoupled and maintainable.

2.What is @Component,@Service,@Respository,@Transaction ?

A @Component: It is a generic stereotype for any Spring managed component.

- The use of @Component across the application to mark the beans as Spring’s managed components.
- Spring will only pick up and register beans with @Components, and doesn’t look for @Service and @Repository in general.

- They are registered in ApplicationContext because they are annotated with @Component.
- @Service and @Respository are special cases of @Component.They are technically same but we use them for the different purposes.

@Service: It annotates classes at the service layer,there isn't any other special use for this annotation.

@Respository: It annotates classes at the persistence layer, which will act as a database respository**.**

- The Spring provides PersistanceExceptionTranslationPostProcessor , which are required to add in our application context (already included if we're using Spring Boot):

  <bean class="org.spring.framework.dao.annotation.
  PersistanceExceptionTranslationPostProcessor"/>

- This bean post processor adds an advisor to any bean that's annotated with @Repository.

@Transaction: This annotation is metadata that specifies that an interface,class or method must have transactional semantics.

3.Differences between @Component and @bean ?

A

@Component	**@beans**
It is implicit mapping.	It is Explicit mapping.
The specialization for @Component are @Controller,@Repository and @Service.	It has no specialization.
@Configuration is not needed to use.	@Configuration is needed to use.
Preferable for component are scanning and auto-wiring.	For 3rd party libraries, we don't have source code . There @Beans is not possible.
It is Class level Annotation.	It is Method level Annotation.

It is not declared the decouples from defination	It is declared the decouples from defination
@Component auto detects and configures the beans using classpath scanning.	@bean explicity declares a single bean, rather than letting Spring do it automatically.

4.What is Spring Boot and What is @SpringBootApplication ?

A Spring Boot:-

- It hrlps to create a standalone application with less configuration.
- The Spring Boot is built on top of the conventional spring framework,widely used fto develop REST API's
- It provides embedded servers such as Tomcat and Jetty's.
- It makes it easy to quickly bootstrap and start developing a spring-based application.

@SpringBootApplication:-

- This annotation is a combination of following three Spring annotations and provides the functionality of all three with just one line of code.

@SpringBootApplication=@Configuration+@ComponentScan+ @EnableAutoConfiguration

5.How many Types of Autowire and What is default ?

A Autowire:

- Autowiring features of spring framework enables to inject the object dependency implicity. It internally uses setter or constructor injection.
- AutoWiring can't be used to inject primitive and string values.It works with reference only.
- There are many autowiring modes are:
  1. No
  2. byName
  3. byType
  4. constructor
  5. autodetect

6.What is qualifier used for ?

A @Qualifier:-

- The @Qualifier annotation is used to resolve the autowiring conflict, when there are multiple beans of same type.
- It can be used on any class annotated with @Component or an methods annotated with @Bean.
- This annotation can also be applied on constructor arguments or method parameters.

7.What is Scope for bean and What is default Scope ?

A Scope of Bean:

- Bean scope in Spring framework or Spring MVC is a scope for a bean managed by Spring IOC container.
- In spring managed environment bean(java classes) are created and wired by the Spring framework. Spring allows to define how those beans will be created.
- The scope Attribute is used to set the scope of the bean.The default scope of the bean is Singleton that is a unique bean instance will be returned for all subsequent getBean() calls and bean references.
- prototype creates a new bean instance each time when it is request**.**

Default Scope:

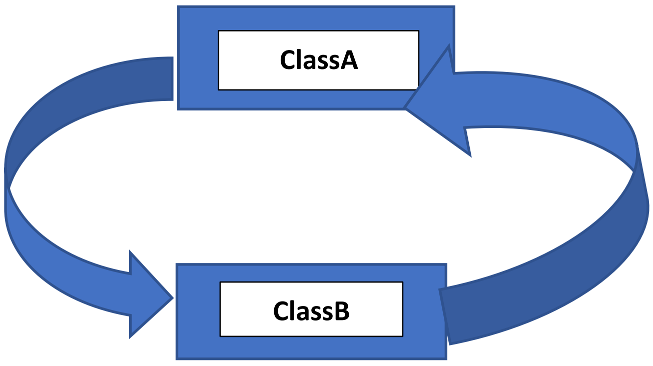
- The default scope of the bean is Singleton that is a unique bean instance will be returned for all subsequent getBean() calls and bean references.
- This scope impiles that Spring container will create an only shared instance of the class designated by this bean, so each time the bean is required the same object will be injected.

8.What is cyclic dependency issue in spring how to avoid it ?

A Circular dependencies is also known as Cyclic dependencies ours when two or more modules reference each other or Circular dependency in spring happens when two or more beans require instance of each other through Constructor dependency injections.

For Example:

- There is a **Class A** that requires an instance of **Class B** through constructor injection and Class B requires an instance of Class A through constructor injection.
- In that sort of configuration where beans for both classes need to be injected into each other, Spring container can't decide which bean should be created first.
- The Spring IOC container will detect this circular reference at runtime while trying to inject dependencies and throw a **BeanCurrentlyInCreationException.**



This could be a **direct reference**(A ->B ->A)

And the **indirect reference** (A ->B ->C->A**)**

9.What is the difference between ApplicationContext and BeanFactory and how many types of application context ?

A

ApplicationContext	**BeanFactory**
ApplicationContext Container is advanced than BeanFactory Container	The Bean Factory is a basic Container
ApplicationContext supports the features of AutoScanning	BeanFactory doesn't supports the features of AutoScanning
ApplicationContext Container creates objects of Singleton bean at the time of loading only It means there is early loading	Bean Factory will not create a bean object upto the request time It means Bean Factory Containerloads beans laziely

ApplicationContext Container supports all the beans scope	Bean Factory Container supports only two scopes are: • Singleton • prototype
ApplicationContext is the central interface within a Spring application that is used for providing Configuration information to the application	Bean Factory is the root interface for accessing the Spring Container

- It provides basic features in addition to enterprise specific functionalites which are:
  - Publishing events to registered listereners by resolving property files.
  - Methods for accessing application components.
  - Supports Internationalization.
  - Loading File resources in a generic fashion.
- It is because of these additional features,developers prefer to use ApplicationContext over BeanFactory.
- There are different types of Application containers provided by Spring for different requiements are
  1. AnnotationConfigApplicationContextContainer
  2. AnnotationConfigWebApplicationContext
  3. XmlWebApplicationContext

10. How to write constructor injection in spring?

A Constructor injection:-

- The Constructoe based Depency injection is a type of Spring Dependency injection Where object's constructor is used to inject dependencies.
- This type of injection is safer as the objects won't get created if the dependencies aren't available or dependencies cannot be resolved.
- The other type of dependency injections are Setter injection and Getter injection.
- Constructor in Employee class which accepts Address.

```
Employee(Address addr){
        This.address=addr;
    }
```

- In this way the address class values can be set independently using either setter/constructor injection