**Spring Application : IOC**

Java Configuration :

Spring says : Need not to create objects by yourself,

springContainer ( Object of AnnotationConfigApplicationContext /ApplicationContext) will create object for you, How?

Spring Container need a configuration file

**There are two ways of creating the configuration file :**

**1. Java Configuration File**

Configuration file is a Java Class which is annotated with an Annotation @Configuration

Inside that class we create public method to return the object

All methods will be annotated with @Bean inside configuration class, will be called by Spring Container and those objects will be recognised by the name specified with name attibute of @Bean Annotation like this @Bean(name="object Name")

How we can use the objects created by Spring Container

Launch SpringContainer by instantiating AnnonationConfigApplicationContext(ConfigurationFileName)

call getBean("ObjectName") method of AnnotationConfigApplicationContext

**Scenario :** If we are the producer of components as well and we have so many, lets sat 50 classes, do we need to create 50 @Bean annotated methods to return object of all 50 classes ?

**Solution :** Make use of @ComponentScan(basepackages={"package1","package2",...."packagen"}) in Configuration file

and make use of @Component for presentation Layer classes, @Service for Service layer classes and @Repository for Persistence Layer classes. Need not to create @Bean annotated method

**2. XML Configuration File**

Create the xml file with bean schema and add <bean id="" class=""> tag for all the classes whoes life cycle you want to be maintained by Spring Container.

**Scenario :** If we are the producer of components as well and we have so many, lets sat 50 classes, do we need to create 50 @Bean annotated methods to return object of all 50 classes ?

**Solution :** Make use of context schema along with bean schema in xml configuration and add two tags

<context:annotation-config>

<context:component-scan>

and make use of @Component for presentation Layer classes, @Service for Service layer classes and @Repository for Persistence Layer classes. Need not to create @Bean annotated method