Spring is a open source light weighted layered architecture framework.

Spring core

IOC : Inversion of control : it is a programming pattern or design pattern. In place of creating object explicitly allow to create and maintain by container. If container will create it will maintain property.

Rather than push the object (Emloyee emp = new Employee()) pull the objet from a container whenever required and use it and leave it.

In MVC project web container will create the object of servlet and jsp but not model layer.

EJB . EJB container will create the object of those classes if class is type of ejb.

Spring framework providing the features to create the object of normal class ie POJO (Plain Old Java Object). The class not to extends or implements any pre-defined class except Object. with Spring container are part of jar files.

IOC is a concept

DI : Dependency Injection : DI is a implementation of IOC.

Spring framework support two type of DI

Constructor based DI

Setter based DI

Using XML

Using Annotation

Generally Spring container do the di for Primitive property. If class contains complex property like address then we have to ref using xml file with explicitly DI for complex object.

Autowired : autowired is use to enable the DI for the complex object implicitly or it is replacement of ref in property tag or constructor args.

byType : in xml file you have to provide the DI for only one object ie only one bean tag.

byName : if more than one bean tag for address available then we have to use byName.

In byName id name in xml file and reference name in pojo class must be same.

@component : This annotation we to use on pojo or java bean class.

@autowired annotation we have to use on complex property.

By default these annotation are not enable so we have to enable using <component-scan tag in xml file.

These annotation we can enable using xml file or using java configuration class with @configuration annotation.