

Spring Core

#2

Prerequisites of Spring:

- Core Java
- JDBC
- Servlet & JSP
- Important web and database related terms

Html, CSS, DBMS(SQL, MongoDB, Oracle, etc.)

What is Spring?

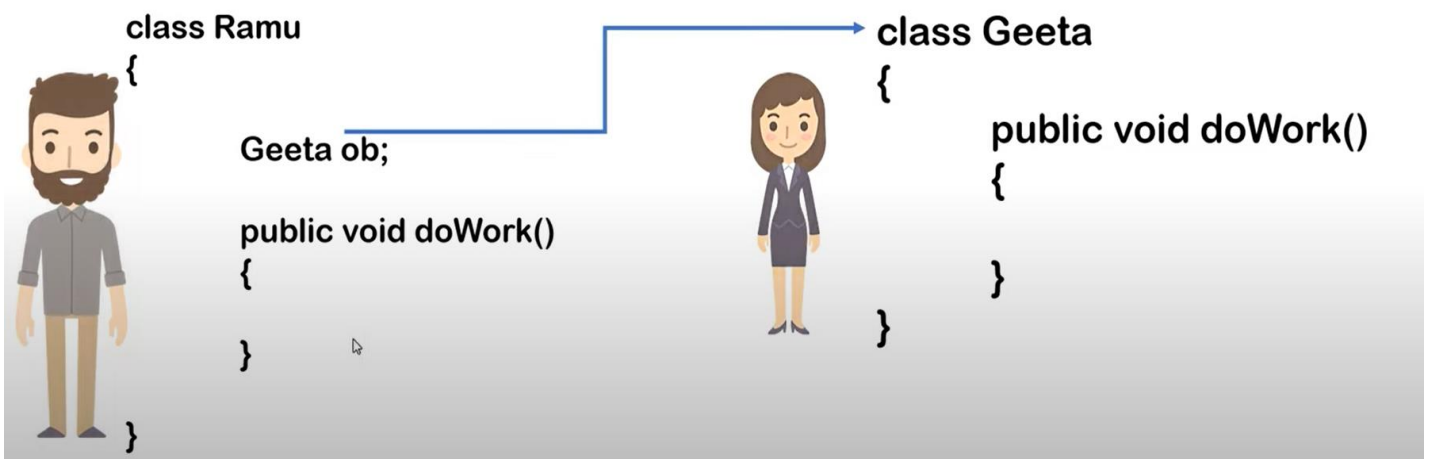
- Spring is a Dependency Injection framework to make java Application loosely Coupled.

Loosely Coupled: Work should be easy, we should be able to change the code easily, to achieve loosely coupled we remove the tightly coupled.

- Spring Framework makes the easy development of Java EE Application.
- It was Developed by **Rod Johnson** in 2003

Dependency Injection:

It is design pattern



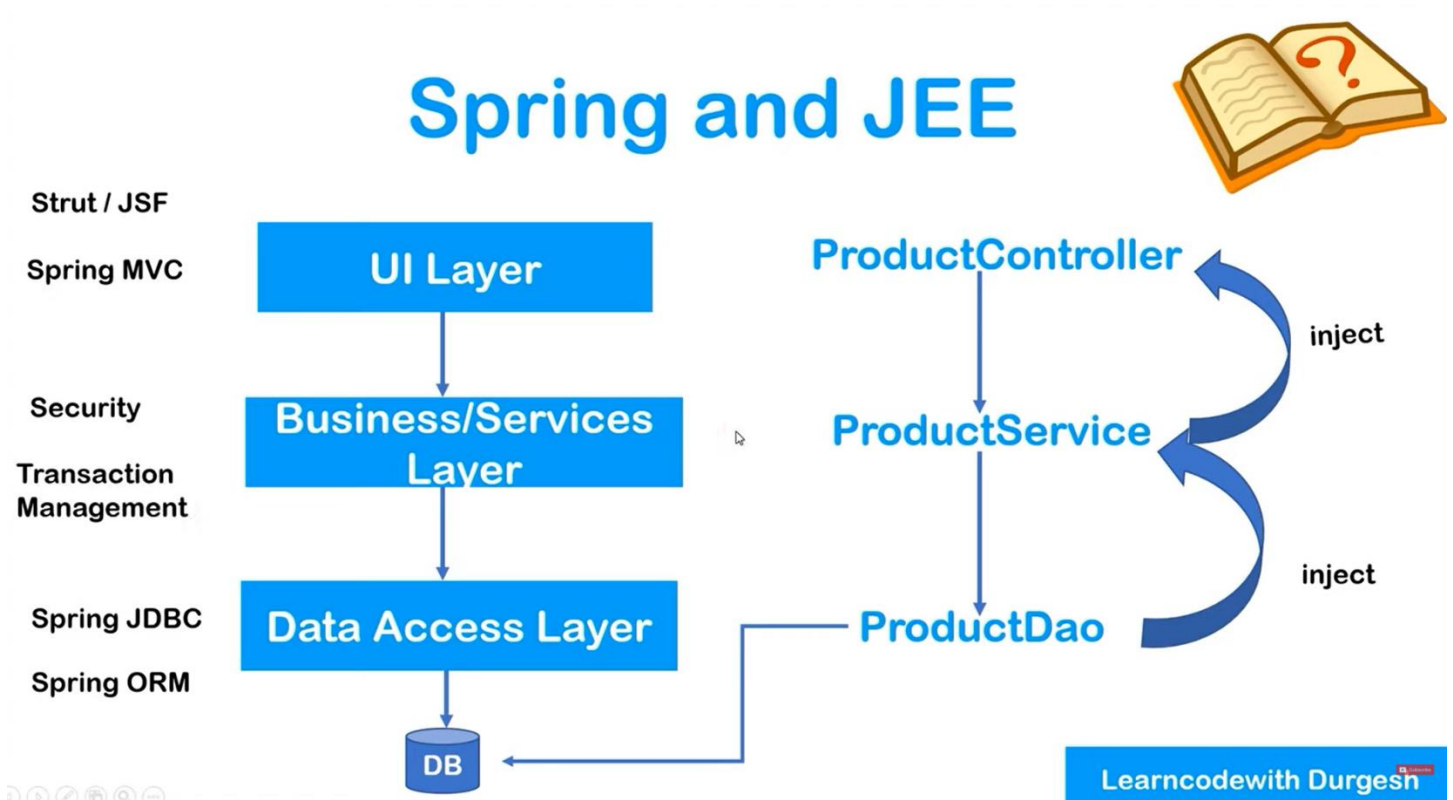
Here Ramu is totally depends on Geeta and Geeta is also a class and she is helping Ramu to the work it is called Dependency.

Dependency mean's one class is depend on second class to perform the work then we have to provide the Geeta's object in Ramu's class

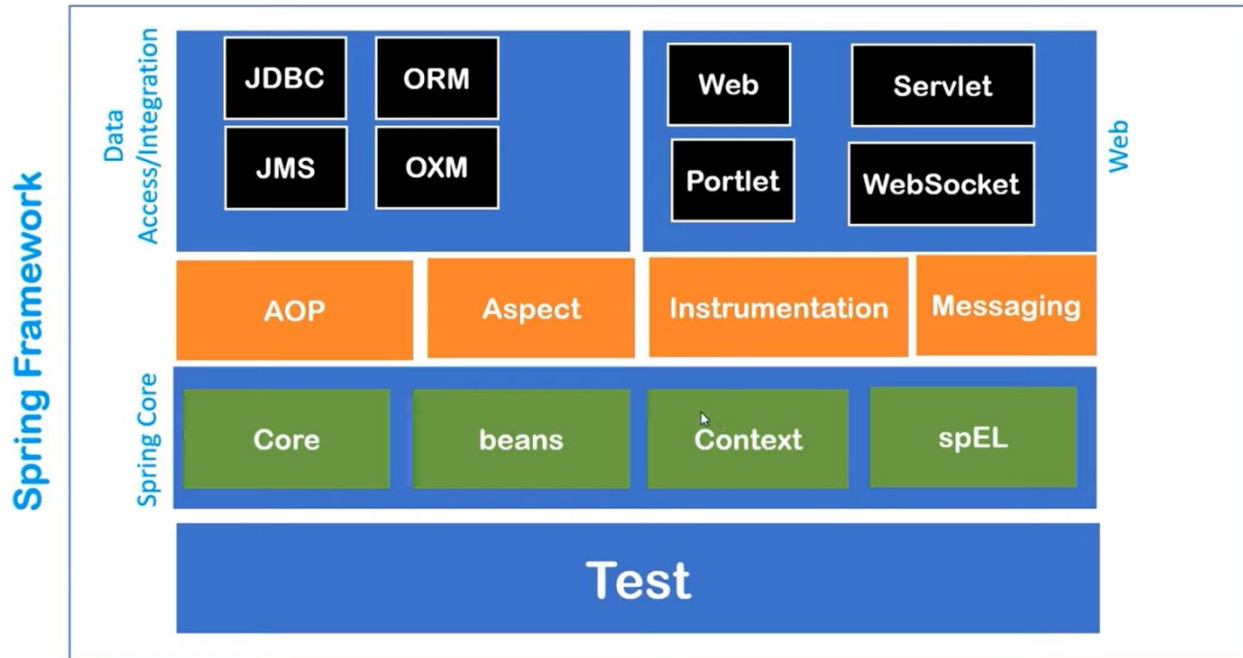
Note

- If we create the object of Geeta in Ramu's class with the New keyword then it will be the highly/tightly coupled. if we perform again that then we will have to create and change the code by own and compile
- Now we will use spring to do the same work because spring has capability to create the object of second class and inject in first class if Ramu is depending on Geeta then spring can create the Geeta's object and inject in Ramu's class.

This Technique is called an inversion of Control



Spring Modules



IoC Container

1. [IoC Container](#)
2. [Using BeanFactory](#)
3. [Using ApplicationContext](#)

The IoC container is responsible to instantiate, configure and assemble the objects. The IoC container gets information's from the XML file and works accordingly. The main tasks performed by IoC container are:

- to instantiate the application class
- to configure the object
- to assemble the dependencies between the objects

Basically, in simple word IOC Container's work is:

1. Create the object
2. Holed them in a memory
3. Inject them in another object as required.

There are two types of IoC containers. They are:

1. **BeanFactory**
2. **ApplicationContext**

Difference between BeanFactory and the ApplicationContext:

The org.springframework.beans.factory.**BeanFactory** and the org.springframework.context.**ApplicationContext** interfaces acts as the IoC container. The ApplicationContext interface is built on top of the BeanFactory interface. It adds some extra functionality than BeanFactory such as simple integration with Spring's AOP, message resource handling (for I18N), event propagation, application layer specific context (e.g. WebApplicationContext) for web application. So it is better to use ApplicationContext than BeanFactory.

Using ApplicationContext

The ClassPathXmlApplicationContext class is the implementation class of ApplicationContext interface. We need to instantiate the ClassPathXmlApplicationContext class to use the ApplicationContext as given below:

```
ApplicationContext context =  
    new ClassPathXmlApplicationContext("applicationContext.xml");
```

The constructor of ClassPathXmlApplicationContext class receives string, so we can pass the name of the xml file to create the instance of ApplicationContext.