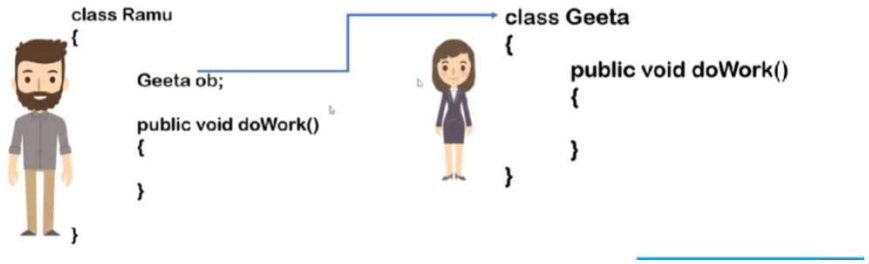
- Spring :1) Spring it is lightweight dependency injection framework which is used to make loosely coupled java applications.
  2)Loosely coupled : means we can easily make changes if required.
  3)It provide IOC(inversion of control) used for dependency injection.
  4)It make easy development of java EE applications.
  5)it is developed by Rod Johnson 2003.
  6)it support or allow integration of various frameworks such as Struct,Hiberanate,EJB,JSP
  7)it support various modules such as IOC,ORM,WEB MVC and AOP.

- Dependancy Injection : -

1) It is design pattern used to develop java applications.
2) makes the code loosely coupled so easy to maintain.
3) makes the code easy to test.

[as we have multiple clasess every time we need to create ob using new it help to avoid this]



As in order to excute method ramu class is totally depend on Geet[it tightly coupled].and if we create object with new keyword then in that scanrio every time we need to modify the code.so now will take help of spring because it can directly inject object of geeta into ramu.[Spring create object of that class by own and inject to ramu called dependency injection]

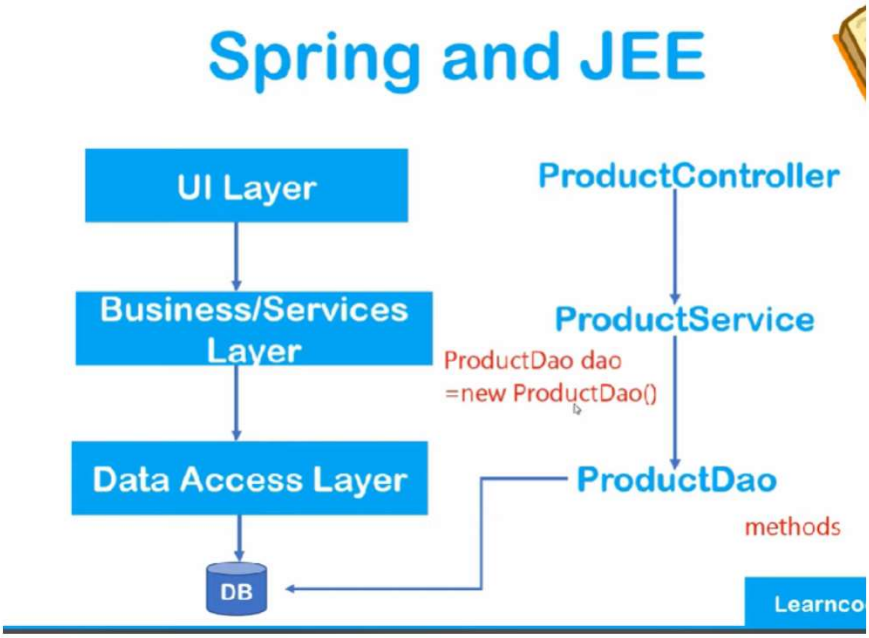
This above process called as Inversion of control as we are giving control to the spring and spring dynamically creating objects and injecting it.[**object creation control will be given to spring.]**

- **Layers** : in any java application

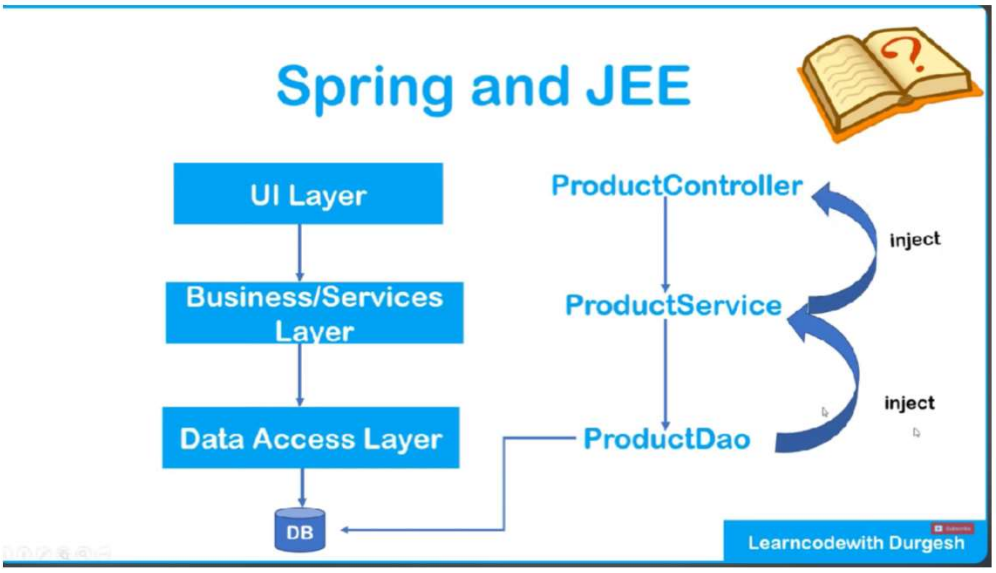
1)UI Layer : it has class productcontroller who hadle the request.

2)Buisness Layer : here Buisness logic will be written.

3)Data Access Layer : ProductDaoclass to acess data.



Now lets say we have to acess methods of productdao in ProductService class so will create object in that class with new keyword now it is tightly coupled .so suppose we made any change in ProductDao class so we need to recompile application.Now will take help of spring.



During Development of application at various layers we can use various modules

1)Spring JDBC: To connect with database [data acess layer]

2)Security ,Transaction Management : -Buisness layer.

3)Spring MVC ,Struct/JSF : UI Layer

❖ Advantages of Spring :

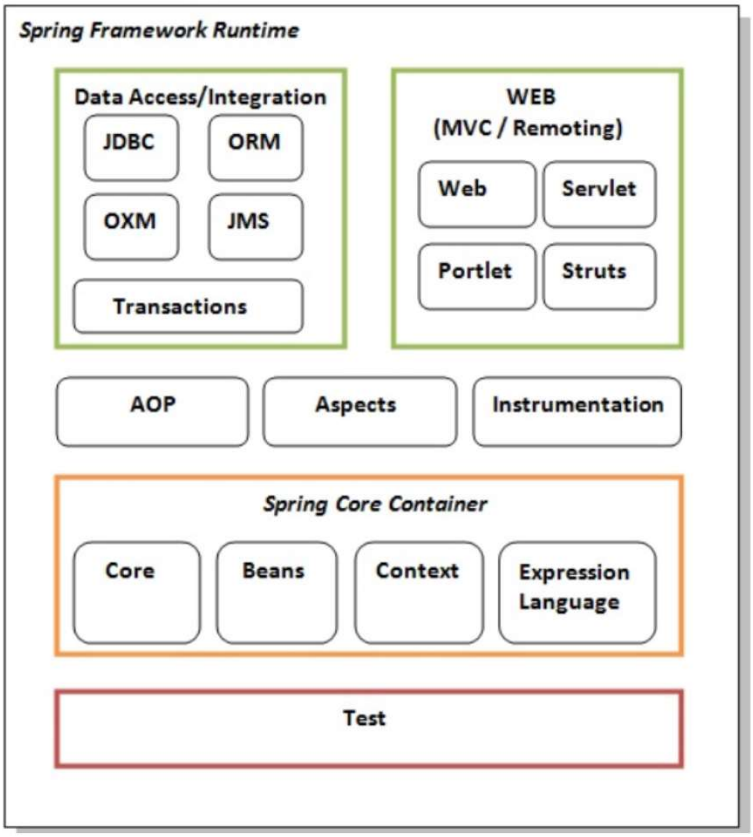
1)Fast Development

2)Lightweight

3)lossely coupled

4)easy to test.

❖ Spring Modules :



- **Core Modules** : core ,bean,context,spring Expression Language [include basic concepts] :-
  1)Core +Bean : - fundamentals of framework.provides Dependancy injection and Inversion of control.

  2)Context : inherits features from Bean and adds things like resource loading ,event propogation.supports EJB, JMS, Basic Remoting.

  3)SEL :-manipulate object value at runtime.[add remove] object graph. It is an extension to the EL defined in JSP. It provides support to setting and getting

property values, method invocation, accessing collections and indexers, named variables, logical and arithmetic operators, retrieval of objects by name etc

- **AOP** (Aspect Oriented Programming] + Aspects +instrumentation : it allow to define method interceptors,point cuts .overall it help to decouple code so that we can maintain functionality separately.
  2)The aspects module provides support to integration with AspectJ.
  3)The instrumentation module provides support to class instrumentation and classloader implementations.

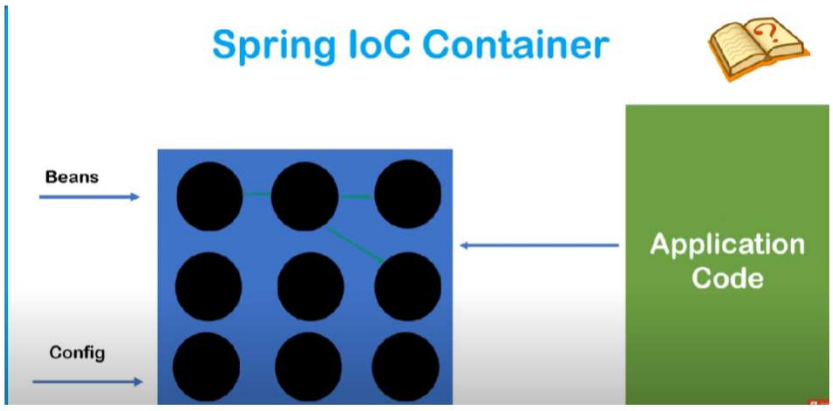
- **Data Acess** : This group comprises of JDBC, ORM, OXM, JMS and Transaction modules. These modules basically provide support to interact with the database.

- **WEB** : This group comprises of Web, Web-Servlet, Web-Struts and Web-Portlet. These modules provide support to create web application.

❖ **Steps to create Spring :**

- create the class
- create the xml file to provide the values
- create the test class
- Load the spring jar files
- Run the test class

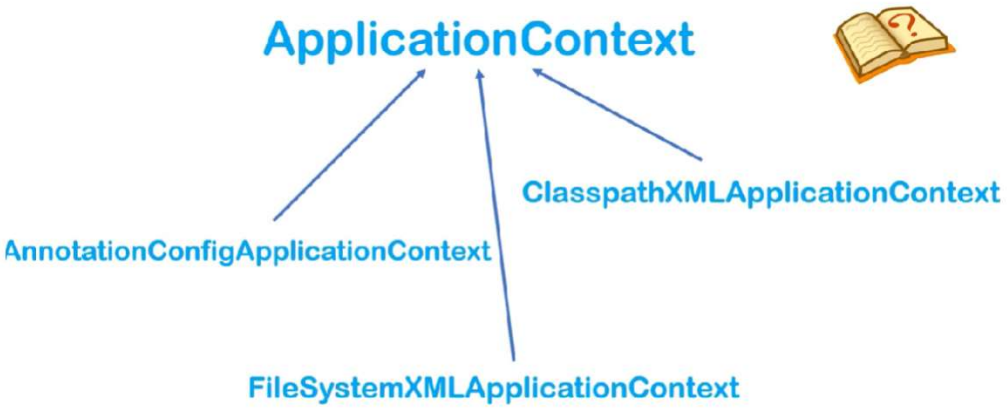
❖ **Spring IOC Container**: it is predefined program or component comes with spring Responsible of object creation ,hold object into memory and also it can inject one object into anouther object.[means maintain lifecycle of object]->just we need to provide 2 imp things beansor pojo [classes and its dependency info] and configuration information.



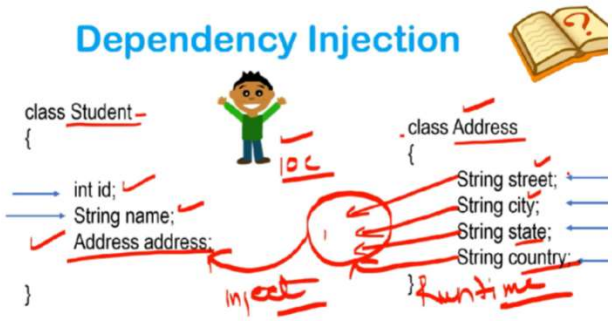
- The **IoC container** : - is responsible to instantiate, configure and assemble the objects. The IoC container gets informations from the XML file and works accordingly. The main tasks performed by IoC container are:

  1)to instantiate the application class
  2)to configure the object
  3)to assemble the dependencies between the objects

  Two types :
  1)Bean factory :
  2)Application Context : extends bean factory has all properties of it along with that it provide some additional properties like 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.



❖ **Ways of Injectecting Dependencies** :

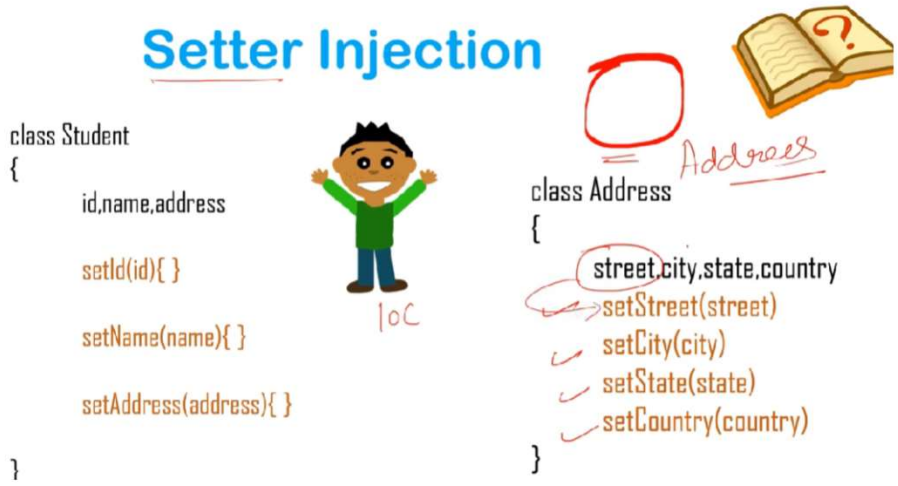


Ioc container will create object of address automatically inject values for attributes at runtime also it will inject values for attribute of student and provide ready object to us.

- **Dependancy Injection can be done by IOC contatiner in Two Ways :**

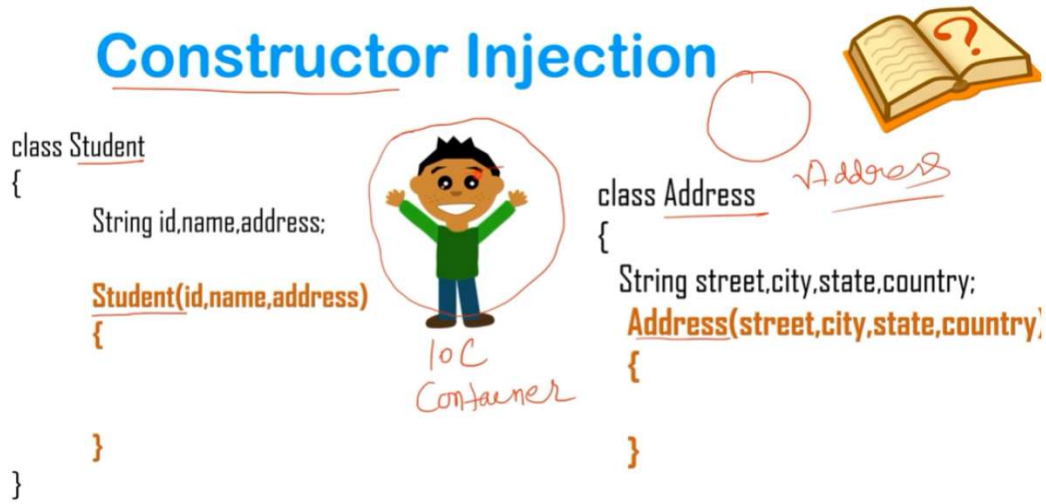
1)Setter Injection or Property Injection :

- Used to set values .
- When creating object of address IOC container will automatically call Setter methods. [same wil happen for Student object]



2)Constructer Injection :

object will take values from configutration file.



[bean is just object or class defined in configuration file.name of beans its count &dependency will be provided from our side to IOC Container.]

<bean>(no of classes)

<bean tage> ->class 1

<bean tag > class 2

<bean>

- **Types of data Supported by IOC :**

1)Primitive data types

2)collection :list ,map ,set.

3)Reference type :user defined.