What is Spring :

* Spring is dependency injection framework to make java application loosely coupled.
* Spring framework makes the easy development of JavaEE application
* Was developed by Rod Johnson in 2003

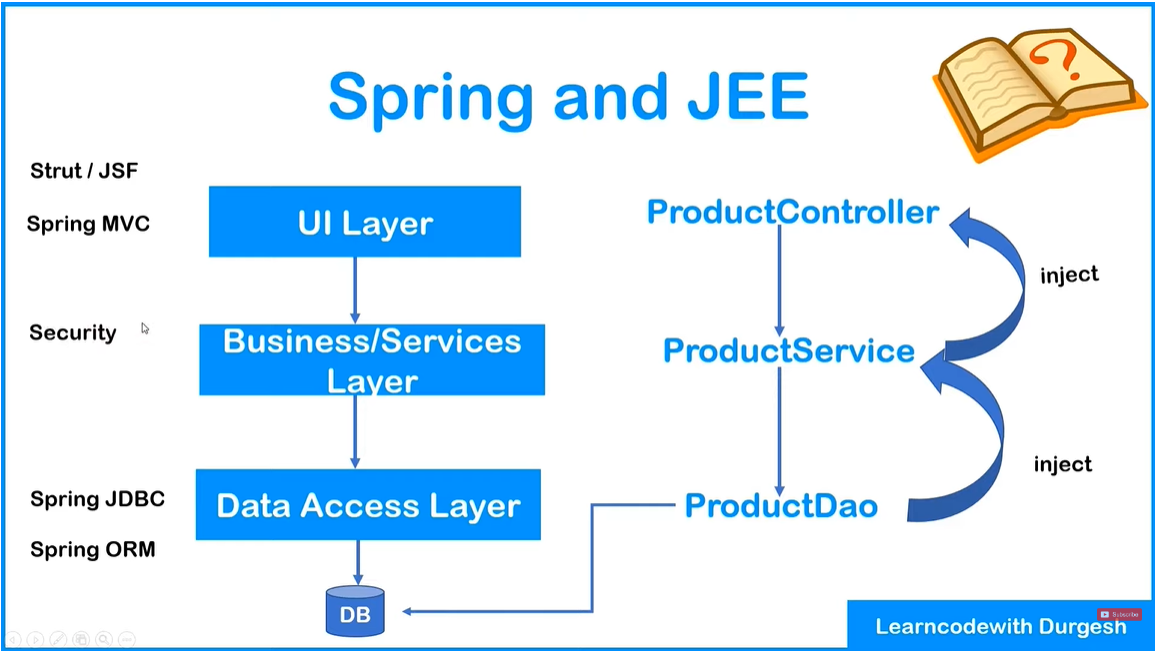
Dependency Injection

* It’s design pattern
* **A** object is depend on **B** object to completes its task, at that time injecting B object into A is known as dependency injection

IOC (Inversion of Control)

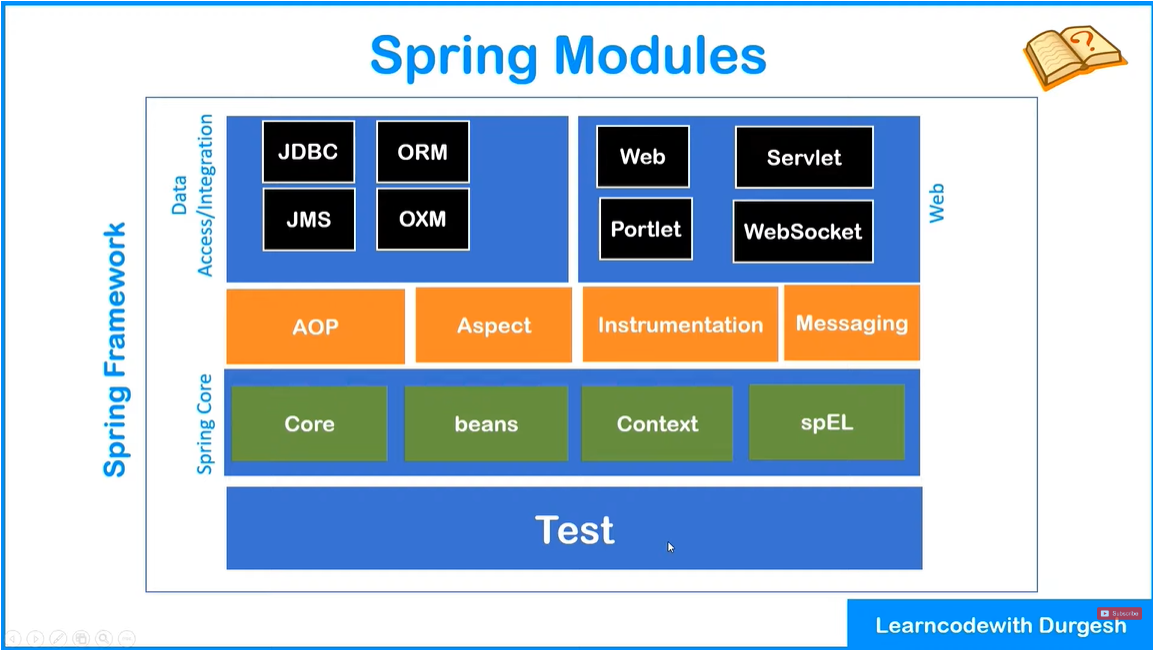
* The whole process of dependency injection is known as IOC
* Rather that creating object by our self we are giving control to spring to create object for us

Spring and JEE



**Transaction Management**

**Spring Modules**

****

**Spring IOC Container**

**The complete life cycle of objects from creation to destruction is maintained by IOC Container**

**It needs two thing Beans & Config.**

**Task perform by IOC container:**

1. **Create Object**
2. **Hold them in memory**
3. **Inject them in another object as required**

**Application Context**

* **It is interface which represent IOC container (It has all properties of bean factory)**
  + **ClassPathXmlApplicationContext *applicationContext.xml Configuration***
  + **AnnotationConfigApplicationContext @annotation**
  + **FileSysteXMLApplicationContext**

**Dependency Injection**

* **Setter Injection**
* **Constructor Injection**

**Configuration File**

* **Declare beans and dependencies**

**Data types (dependencies)**

1. Primitive data type
   1. Byte, short, int, char, float, double, long, boolean.
2. Collection Type
   1. List, Set, Map & Properties
3. Reference type
   1. Class object

Life Cycle Methods of Beans

* Spring provides two important methods to every bean (we can change the name of these methods but signature must be same)

1. Public void init() -Initialization code, Loading config, Connecting db, WebServices etc.
2. Public void destroy() -Clean/close up code
3. Flow - properties will set –> init method call -> injection ops -> destroy method call

* Ways to call implement method
  1. XML Configuration
     + ‘method-init = “method\_name”’
     + ‘mthod-ddestroy = “method\_name”
  2. Interface implementation
     + Implement these interfaces to bean - InitializatingBean & DisposableBean
  3. Using Annotation (in/above java 9 we have to add dependencies )
     + @PostConstruct
     + @PreDestroy
     + Note- to enable all annotation (<context:annotation-config />)

Autowiring in Spring

* Feature of spring framework in which spring container inject dependencies automatically
* It can only be used for object/reference (not primitive and string)

1. XML
   1. Autowiring Modes
      1. No
      2. byName - reference variable and bean name in xml file must be same
         1. eg., <bean name=”address”> then private Address address
      3. byType - will check type, but ran into ambiguity when more than one bean present
      4. Constructor -internally uses byType
         1. To use it byName – below @Autowired @Qualifier(“name\_of\_bean”)
2. Annotation
   1. @Autowired

Spring Standalone Collection

* Usefull when you have to use same values of collection in multiple places in configuration (.xml file)
* <util:list list-class=”java.uitl.LinkedList” id=”linkedList> <value></value> </util:list>