Spring - Dependency Injection (DI)

DI is a framework which provides loose coupling in code. Here loose coupling mean **no hard coding of object.** Instead of hard coding we will be injecting these object from some configurations (xml based, java based, annotation based).

DI is nothing but a design patterns which give you the light weight loosely couple enterprise application in result.

What is an injection? -- Inserting of an object/ value into a class whenever it is required by the help of third party configurations.

Note: For in-detail classes/ discussions you can feel free to contact me.

There are two way of achieving Dependency Injection:

- By Constructor
- By Setter method

Constructor Injection: Injecting any object/ value through constructor of a class. For ex: I have a Test class with parameterized constructor in it and I am initializing parameter of that constructor from external source i.e. XML file. See how

```
public class Test {
private int id;
public Employee(int id) {
this.id = id;
}
```

XML configuration file: //here now 100 will be assigned to id of Test class

```
<br/><bean id="e" class="com.Test "></br/><br/><constructor-arg value="100" type="int"></constructor-arg></bean>
```

```
Setter method Injection: Injecting any object/ value through setter method of a
class. For ex: I have a Test class with setter method in it and I am initializing
parameter of that setter method from external source i.e. XML file. See how
public class Test {
private int id;
public int getId() {
  return id;
}
public void setId(int id) {
  this.id = id;
}
}
XML configuration file: //here now 100 will be assigned to id of Test class
<bean id="obj" class="com.Test">
cproperty name="id">
<value>100</value>
</property>
</bean>
```

Please add on comments and like this lesson, if it gave you some revision or refreshment on Spring - Dependency Injection (DI) concepts.