**Important note: Before going further you are requested to go through theory of Spring Dependency Injection mentioned in JavaLive Notes on Spring Core thoroughly so that you can understand the concept very well. (\*\*This activity i.e. going through theory part of respective subject is applicable for each of our project.)**

**What this project does??**

This project explains DI in spring framework of follwoing types

i) DI by using constructor used in case of mandatory situation.

ii) DI by using setter method used in case of non mandatory situation.

**Note that this is xml based spring project having configuration file viz. ' spring-config.xml' in resources folder.**

**Steps to create project:-**

1. Create quick start maven project. Refer file 'CreateQuickStartMavenProjectInEclipse.docx' in project 'SpringCoreDIWithAutowiring' for more details.

2. Add require dependencies for spring as shown in pom.xml file.

3. Create required packages and add the files.

**Functioning of the project**

As mentioned earlier, this project deals with explaining dependency injection types (i.e. by using setter method and by using constructor) in spring. In this project please carefully go through packages com.javaLive.beans and com.javaLive.main.

We have three classes viz. Student, RoomConstructor and RoomSetter in com.javaLive.beans package. There is has-a relationship between Student and RoomSetter(as name suggest we have injected Student object by using setter method) and Student and RoomConstructor(as name suggest we have injected Student object by using constructor).

com.javaLive.main.MainClass.java contains code which get and displays beans 'roomConstructor' and ' roomSetter' from application context which ultimately displays respective Student details

To understand the relation and functioning carefully go through spring configuration file viz 'spring-config.xml' in resources folder. We have passed student object via <constructor-arg> tags ref attribute in case of constructor based DI and using <property> tags ref attribute in case of setter based DI.