**Important note: Before going further you are requested to go through theory of Spring Dependency Injection with look up method 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 how DI carried out using Look Up method in spring.

Need of Lookup Method:

**We require lookup method functionality when we refer prototype bean from singleton bean.**

**By using lookup method we get new instance of prototype bean that is referred by given singleton bean each time when we refer the given singleton bean.**

**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 this project 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**

This project provides comparative analysis of behavior of spring container with and without 'look up method'. Package com.javaLive.beansWithMethodInjection contains class 'SingletonBean2.java' with partial lookup method code i.e. abstract method 'getPrototypeBean() ' (Rest of the code in included in spring-config.xml file under definition of bean class 'SingletonBean2' with <lookup-method> tag.) Thus when we call singleton bean with id 'singletonBean2' in spring-config file, each time fresh instance of bean 'prototypeBean2' will get created.

Package com.javaLive.beansWithoutMethodInjection contains normal code without any look up method. Which behaves normally and both beans ' singletonBean1' and ' prototypeBean1' created only once at the time of loading spring application context.