**Important note: Before going further you are requested to go through theory of spring bean lifecycle methods 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 methods with @PostConstruct and @PreDestroy annotations plays role in spring bean life cycle.

**Note that this is alternate(and in some extent more advance as uses annotations) way to custom init and custom destroy methods which we have seen in project 'SpringCoreBeanLifeCycleCustomInitAndDestroyMethods'.**

**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 'SprinDIWithAutowiring' 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 is mainly divided in three parts:

1. com.javaLive.beans package contains a file Book.java which declared as bean in spring-config.xml file. **This class defines methods with @PostConstruct and @PreDestroy annotations. These methods are main part of this project.**

2. spring-config.xml file in resources folder contains bean definition for above class.

**\*\*Note during definition of this bean we haven't mentioned any other attribute as in case of spring-config file of project 'SpringCoreBeanLifeCycleCustomInitAndDestroyMethods' as we have used annotations for the respective methods in Book.java class.**

3. com.javaLive.main package contains MainClass.java class which get the bean from spring application context and displays its details. When we run this file, we will come to know in which sequence methods with @PostConstruct and @PreDestroy methods of this bean file get called.