**Exercise 6: Configuring Beans with Annotations**

**CODE:**

**applicationContext.xml file: -**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd">  
  
 <context:component-scan base-package="org.Library"/>  
  
</beans>

**BookService Class: -**

package org.Library.service;  
import org.Library.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BookService {  
 private BookRepository bookRepository;  
  
 @Autowired  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
   
 public void addBook(String title){  
 System.*out*.println("Saving book...");  
 bookRepository.saveBook(title);  
 }  
}

**BookRepository Class: -**

package org.Library.repository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public class BookRepository {  
 public void saveBook(String title){  
 System.*out*.println("Saved Book: "+title);  
 }  
}

**LoggingAspect Class: -**

package org.Library.Aspect;  
import org.aspectj.lang.JoinPoint;  
import org.aspectj.lang.annotation.After;  
import org.aspectj.lang.annotation.Aspect;  
import org.aspectj.lang.annotation.Before;  
import org.springframework.context.annotation.EnableAspectJAutoProxy;  
import org.springframework.stereotype.Component;  
  
@Component  
@Aspect  
@EnableAspectJAutoProxy  
public class LoggingAspect {  
 @Before("execution(\* org.Library.service.\*.\*(..))")  
 public void logBeforeMethod(JoinPoint joinPoint){  
 String methodName=joinPoint.getSignature().getName();  
 System.*out*.println("[BEFORE] Method "+methodName+" is about to execute.");  
 }  
 @After("execution(\* org.Library.service.\*.\*(..))")  
 public void logAfterMethod(JoinPoint joinPoint){  
 String methodName=joinPoint.getSignature().getName();  
 System.*out*.println("[AFTER] Method "+methodName+" has completed execution.");  
 }  
}

**LibraryManagementApplication Class: -**

package org.Library;  
import org.Library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class LibraryManagementApplication {  
 public static void main(String[] args) {  
 ApplicationContext context= new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService=(BookService) context.getBean("bookService");  
 bookService.addBook("All about Spring Core");  
 }  
}

**OUTPUT:**

