**Exercise 8: Implementing Basic AOP with Spring**

**Scenario:**

The library management application requires basic AOP functionality to separate cross-cutting concerns like logging and transaction management.

**BookRepository.java**

package com.library;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

    public void save(String title) {

        System.out.println("SAVING BOOK: " + title.toUpperCase());

    }

    public String find(String title) {

        System.out.println("LOOKING UP BOOK: " + title.toUpperCase());

        return title;

    }

}

**BookService.java**

package com.library;

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("ADDING BOOK: " + title.toUpperCase());

        bookRepository.save(title);

    }

    public String findBook(String title) {

        System.out.println("FINDING BOOK: " + title.toUpperCase());

        return bookRepository.find(title);

    }

}

**LibraryManagementApplication.java**

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

    public static void main(String[] args) {

        System.out.println("STARTING LIBRARY MANAGEMENT APPLICATION");

        ApplicationContext context =

                new ClassPathXmlApplicationContext("applicationContext.xml");

        BookService bookService = context.getBean(BookService.class);

        bookService.addBook("Pro Spring AOP");

        String foundBook = bookService.findBook("Clean Code");

        System.out.println("BOOK FOUND: " + foundBook.toUpperCase());

        System.out.println("APPLICATION COMPLETED SUCCESSFULLY");

    }

}

**applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

       xmlns:context="http://www.springframework.org/schema/context"

       xmlns:aop="http://www.springframework.org/schema/aop"

       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

       xsi:schemaLocation="

         http://www.springframework.org/schema/beans

         http://www.springframework.org/schema/beans/spring-beans.xsd

         http://www.springframework.org/schema/context

         http://www.springframework.org/schema/context/spring-context.xsd

         http://www.springframework.org/schema/aop

         http://www.springframework.org/schema/aop/spring-aop.xsd">

    <!-- Enable component scanning -->

    <context:component-scan base-package="com.library" />

    <!-- Enable Spring AOP auto-proxy -->

    <aop:aspectj-autoproxy />

</beans>

**pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

         xsi:schemaLocation="

           http://maven.apache.org/POM/4.0.0

           http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.ex8</groupId>

  <artifactId>ex8</artifactId>

  <version>1.0-SNAPSHOT</version>

  <properties>

    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <maven.compiler.source>1.8</maven.compiler.source>

    <maven.compiler.target>1.8</maven.compiler.target>

  </properties>

  <dependencies>

    <!-- Spring Context (for IoC container) -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-context</artifactId>

      <version>5.3.30</version>

    </dependency>

    <!-- Spring AOP (for aspect-oriented programming) -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-aop</artifactId>

      <version>5.3.30</version>

    </dependency>

    <!-- AspectJ Weaver (required for Spring AOP) -->

    <dependency>

      <groupId>org.aspectj</groupId>

      <artifactId>aspectjweaver</artifactId>

      <version>1.9.21</version>

    </dependency>

    <!-- JUnit for testing -->

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.13.2</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <!-- Maven Compiler Plugin -->

      <plugin>

        <groupId>org.apache.maven.plugins</groupId>

        <artifactId>maven-compiler-plugin</artifactId>

        <version>3.10.1</version>

        <configuration>

          <source>1.8</source>

          <target>1.8</target>

        </configuration>

      </plugin>

    </plugins>

  </build>

</project>

**Output:**

