**Exercise 2: Implementing Dependency Injection**

**Scenario:**

In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.

**Spring application is using XML-based configuration**

**App.java :-**

package org.example1;  
  
import org.example1.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class App {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService service = context.getBean("bookService", BookService.class);  
 service.processBook();  
 }  
}

BookService.java :-

package org.example1.service;  
  
import org.example1.repository.BookRepository;  
import org.example1.repository.NotificationService;  
  
public class BookService {  
 private BookRepository bookRepository; // via constructor  
 private NotificationService notificationService; // via setter  
  
 // Constructor Injection  
 public BookService(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 // Setter Injection  
 public void setNotificationService(NotificationService notificationService) {  
 this.notificationService = notificationService;  
 }  
  
 public void processBook() {  
 System.*out*.println("Book Info: " + bookRepository.getBookInfo());  
 if (notificationService != null) {  
 notificationService.sendNotification();  
 }  
 }  
}

**BookRepository.java :-**

package org.example1.repository;  
  
public class BookRepository {  
 public String getBookInfo() {  
 return "Spring Boot in Practice";  
 }  
}

**application.xml :-**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 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">  
  
 <!-- Bean Definitions -->  
 <bean id="bookRepository" class="org.example1.repository.BookRepository" />  
  
 <bean id="notificationService" class="org.example1.repository.NotificationService" />  
  
 <bean id="bookService" class="org.example1.service.BookService">  
 <constructor-arg ref="bookRepository"/>  
 <property name="notificationService" ref="notificationService"/>  
 </bean>  
  
</beans>

**Pom.xml :-**

<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>org.example1</groupId>  
 <artifactId>LibraryManagement</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <name>LibraryManagement</name>  
 <url>http://maven.apache.org</url>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>3.8.1</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>6.0.12</version>  
 </dependency>  
 </dependencies>  
</project>

**Output :-**

