**Spring Core and Maven**

**Exercise 2: Implementing Dependency Injection**

**To manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI**

* **src/main/resources/applicationContext.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

https://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- Define BookRepository bean -->

<bean id="bookRepository" class="com.library.repository.BookRepository" />

<!-- Define BookService bean and inject BookRepository -->

<bean id="bookService" class="com.library.service.BookService">

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

* **src/main/java/com/library/service/BookService.java**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter method for Dependency Injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository; }

public void listBooks() {

bookRepository.displayBooks();

}

}

* **src/main/java/com/library/LibraryApp.java**

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryApp {

public static void main(String[] args) {

// Load application context from XML

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

// Retrieve BookService bean

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

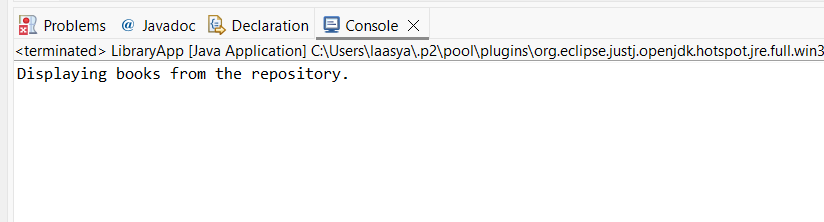
// Call method to list books

bookService.listBooks();

}

}

**OUTPUT:**

****