Before starting this task, I completed the basic Spring project setup in Task 1, which included the following:

~ Created a Maven project named LibraryManagement.

~Added Spring dependencies (spring-context and commons-logging) in pom.xml.

~ Created Java packages and classes:

~ com.library.repository.BookRepository with a save() method.

~ com.library.service.BookService with an addBook() method.

~ Created applicationContext.xml to define basic Spring beans.

~ Created a App.java class to load Spring context and call service methods.

Step 1: Modify applicationContext.xml

Location: src/main/resources/applicationContext.xml

Make sure it looks like this:

<?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">

<!-- Define BookRepository Bean -->

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

<!-- Define BookService Bean with DI -->

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

<!-- Inject BookRepository into BookService -->

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

</bean>

</beans>

**.** This configures Spring Dependency Injection via setter injection.

Step 2: Make Sure BookService Has a Setter

📁 File: BookService.java

Package: com.library.service

Your class should have this setter:

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

. If it already exists (which it should from Exercise 1), you're good to go.

Step 3: Test the Configuration

📁 File: App.java

Package: com.library

Use this code to test DI:

package com.library;

import com.library.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");

// Get the BookService bean, which should have BookRepository injected

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

// Call method to verify DI

bookService.addBook();

}

}

Step 4: Run the App

Right-click App.java

Choose Run 'App.main()'

