**EXERCISE 2: IMPLEMENTING DEPENDENCY INJECTION**

**Step 1: Modify the XML Configuration**

<?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 and wire BookRepository -->

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

<!-- Inject BookRepository bean into BookService using setter injection -->

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

</bean>

</beans>

**Step 2: Update the BookService Class**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter for BookRepository

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void performSomeService() {

System.out.println("Performing service in BookService");

bookRepository.displayRepositoryInfo();

}

}

**Step 3: Test the Configuration**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApplication {

public static void main(String[] args) {

// Load the Spring context

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

// Retrieve the BookService bean

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

// Use the BookService to perform some action

bookService.performSomeService();

}

}