**Exercise 2: Implementing Dependency Injection**

*BookService.java:*

package com.example.library.service;

import com.example.library.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    // Setter method for DI

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void addBook(String title) {

        System.out.println("Adding book: " + title);

        bookRepository.saveBook(title);

    }

}

*BookRepository.java:*

package com.example.library.repository;

public class BookRepository {

    public void saveBook(String title) {

        System.out.println("Book \"" + title + "\" saved to the repository.");

    }

}

*LibraryApp.java:*

package com.example.library;

import com.example.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryApp {

    public static void main(String[] args) {

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

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

        bookService.addBook("1984 by George Orwell");

    }

}

*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

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

    <!-- Repository Bean -->

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

    <!-- Service Bean, with Dependency Injection via setter -->

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

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

    </bean>

</beans>

Output:

