**Spring Core and Maven**

Exercise 1: Configuring a Basic Spring Application

Scenario:

Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

Steps:

1. Set Up a Spring Project:
   * Create a Maven project named LibraryManagement.
   * Add Spring Core dependencies in the pom.xml file.
2. Configure the Application Context:
   * Create an XML configuration file named applicationContext.xml in the src/main/resources directory.
   * Define beans for BookService and BookRepository in the XML file.
3. Define Service and Repository Classes:
   * Create a package com.library.service and add a class BookService.
   * Create a package com.library.repository and add a class BookRepository.
4. Run the Application:
   * Create a main class to load the Spring context and test the configuration.

**CODE:**

**BOOKService.java:**

package com.library.service;  
import com.library.repository.BookRepository;  
public class BookService {  
 private BookRepository bookRepository;  
 public BookService(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void displayBooks() {  
 System.*out*.println("[BookService] Displaying book list...");  
 bookRepository.fetchBooks();  
 }  
 }

**BookRepository.java:**

package com.library.repository;  
public class BookRepository {  
 public void fetchBooks() {  
 System.*out*.println("[BookRepository] 1. The Alchemist by Paulo Coelho");  
 System.*out*.println("[BookRepository] 2. Clean Code by Robert C. Martin");  
 }  
 }

**MainApp.java:**

package com.library;  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
 public class MainApp {  
 public static void main(String[] args) {  
 System.*out*.println("[MainApp] Library Management App started...");  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService = (BookService) context.getBean("bookService");  
 bookService.displayBooks();  
 System.*out*.println("[MainApp] App finished successfully.");  
 }  
 }

**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>com.library</groupId>  
 <artifactId>LibraryManagement</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <dependencies>  
 <!-- Spring Core for ApplicationContext and beans -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <!-- Compiler plugin to use Java 8+ -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.8.1</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**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.library.repository.BookRepository"/>  
 <!-- Service Bean with constructor injection -->  
 <bean id="bookService" class="com.library.service.BookService">  
 <constructor-arg ref="bookRepository"/>  
 </bean>  
</beans>

**OUTPUT:**

