**Exercise 1: Configuring a Basic Spring Application**

**Solution:**

**Step 1:** Create a New Maven Project

**Step 2:** Add Spring Core Dependency in pom.xml

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.32</version>

</dependency>

</dependencies>

**Step 3:** Create the Required Packages

**Step 4:** Create Java Classes

BookRepository.java (Inside com.library.repository)

package com.library.repository;

public class BookRepository {

public void displayRepo() {

System.*out*.println("BookRepository method called");

}

}

BookService.java (Inside com.library.service)

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter for Dependency Injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void displayService() {

System.*out*.println("BookService method called");

bookRepository.displayRepo();

}

}

**Step 5:** Create applicationContext.xml

<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">

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

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

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

</bean>

</beans>

**Step 6:** Create the Main Class

LibraryManagementApplication.java (Inside com.library)

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

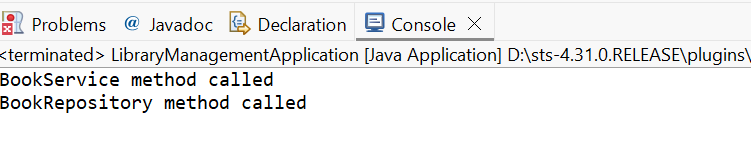
service.displayService();

}

}

**Step 7:** Run and Test the Application

**Output:**



**Exercise 2: Implementing Dependency Injection**

**Solution:**

**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">

<!-- Bean definition for BookRepository -->

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

<!-- Bean definition for BookService with setter injection -->

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

<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 method required for Dependency Injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void displayService() {

System.*out*.println("BookService method called");

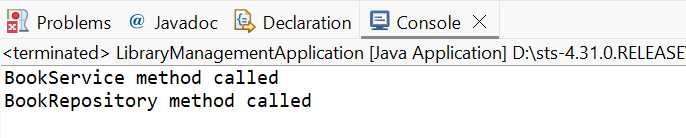
bookRepository.displayRepo(); // Call repository method

}

}

**Step 3:** Run the LibraryManagementApplication.java

**Output:**

****

**Exercise 4: Creating and Configuring a Maven Project**

**Solution:**

**Step 1:** Create a New Maven Project

**Step 2:** Add Spring Dependencies in pom.xml

<dependencies>

<!-- Spring Core (Context) -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.32</version>

</dependency>

<!-- Spring AOP -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.32</version>

</dependency>

<!-- Spring Web MVC (for future web app support) -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.32</version>

</dependency>

</dependencies>

**Step 3:** Configure Maven Compiler Plugin

Inside the same pom.xml, below the <dependencies> section, add the <build> section:

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.10.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

**Step 4:** Update the Project

**Step 5:** Project is now configured