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

class BookRepository {

public void saveBook(String bookName) {

System.out.println("BookRepository: Saving book - " + bookName);

}

}

class BookService {

public void addBook(String bookName) {

System.out.println("BookService: Adding book - " + bookName);

BookRepository bookRepository = new BookRepository();

bookRepository.saveBook(bookName);

}

}

public class Exercise1\_LibraryManagementApplication {

public static void main(String[] args) {

BookService service = new BookService();

service.addBook("Java Programming");

}

}

A white background with red text

AI-generated content may be incorrect.

**Exercise 2: Implementing Dependency Injection**

class BookRepository {

public void saveBook(String bookName) {

System.out.println("BookRepository: Saving book - " + bookName);

}

}

class BookService {

private BookRepository bookRepository;

// Setter for Dependency Injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("BookService: Adding book - " + bookName);

bookRepository.saveBook(bookName);

}

}

public class Exercise2\_LibraryManagementApplication {

public static void main(String[] args) {

// Simulating Spring Dependency Injection

BookRepository repository = new BookRepository();

BookService service = new BookService();

service.setBookRepository(repository);

service.addBook("Design Patterns");

}

}



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

<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</version>

<dependencies>

<!-- Spring Core Context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.33</version>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Java Compiler Plugin -->

<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>

</project>

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

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

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

</bean>

</beans>

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("BookRepository: Saving book - " + bookName);

}

}

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("BookService: Adding book - " + bookName);

bookRepository.saveBook(bookName);

}

}

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.addBook("Spring Framework Mastery");

}

}

A close up of words

AI-generated content may be incorrect.