**Module 5 - Spring Core and Maven**

**Exeercise-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: o Create a Maven project named LibraryManagement. o Add Spring Core dependencies in the pom.xml file.**

**2. Configure the Application Context: o Create an XML configuration file named applicationContext.xml in the src/main/resources directory. o Define beans for BookService and BookRepository in the XML file.**

**3. Define Service and Repository Classes: o Create a package com.library.service and add a class BookService. o Create a package com.library.repository and add a class BookRepository.**

**4. Run the Application: o Create a main class to load the Spring context and test the configuration.**

**Step 1: Set Up a Spring Project**

**Create Maven Project named LibraryManagement.**

**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 -->**

**<dependency>**

**<groupId>org.springframework</groupId>**

**<artifactId>spring-context</artifactId>**

**<version>5.3.36</version>**

**</dependency>**

**</dependencies>**

**</project>**

**Step 2: Configure the Application Context**

**File: src/main/resources/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">**

**<!-- Define BookRepository bean -->**

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

**<!-- Define BookService bean -->**

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

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

**</bean>**

**</beans>**

**Step 3: Define Service and Repository Classes**

**File: com/library/repository/BookRepository.java**

**package com.library.repository;**

**public class BookRepository {**

**public void saveBook() {**

**System.out.println("Book saved to database.");**

**}**

**}**

**File: com/library/service/BookService.java**

**java**

**CopyEdit**

**package com.library.service;**

**import com.library.repository.BookRepository;**

**public class BookService {**

**private BookRepository bookRepository;**

**// Setter Injection**

**public void setBookRepository(BookRepository bookRepository) {**

**this.bookRepository = bookRepository;**

**}**

**public void addBook() {**

**System.out.println("Adding book...");**

**bookRepository.saveBook();**

**}**

**}**

**Step 4: Run the Application**

**File: LibraryManagementApplication.java**

**import com.library.service.BookService;**

**import org.springframework.context.ApplicationContext;**

**import org.springframework.context.support.ClassPathXmlApplicationContext;**

**public class LibraryManagementApplication {**

**public static void main(String[] args) {**

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

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

**bookService.addBook();**

**}**

**}**

**Exercise 2**

**Implementing Dependency Injection Scenario: In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.**

**Steps: 1. Modify the XML Configuration: o Update applicationContext.xml to wire BookRepository into BookService.**

**2. Update the BookService Class: Ensure that BookService class has a setter method for BookRepository.**

**3. Test the Configuration: Run the LibraryManagementApplication main class to verify the dependency injection.**

**Step 1: Modify XML Configuration**

**Done in applicationContext.xml:**

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

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

**</bean>**

**Step 2: Update BookService Class**

**public void setBookRepository(BookRepository bookRepository) {**

**this.bookRepository = bookRepository;**

**}**

**Step 3: Test the Configuration**

**Run LibraryManagementApplication.java  
Expected Output:**

**Adding book...**

**Book saved to database.**

**Exercise 4**

**Creating and Configuring a Maven Project Scenario: You need to set up a new Maven project for the library management application and add Spring dependencies.**

**Steps: 1. Create a New Maven Project: o Create a new Maven project named LibraryManagement.**

**2. Add Spring Dependencies in pom.xml: o Include dependencies for Spring Context, Spring AOP, and Spring WebMVC.**

**3. Configure Maven Plugins: o Configure the Maven Compiler Plugin for Java version 1.8 in the pom.xml file.**

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

* **Project Name: LibraryManagement**
* **Project Structure:**

**LibraryManagement/**

**├── pom.xml**

**├── src/**

**│ ├── main/**

**│ │ ├── java/**

**│ │ └── resources/**

**│ └── test/**

**│ └── java/**

**Step 2: Add Spring Dependencies in 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>**

**<properties>**

**<java.version>1.8</java.version>**

**</properties>**

**<dependencies>**

**<!-- Spring Context -->**

**<dependency>**

**<groupId>org.springframework</groupId>**

**<artifactId>spring-context</artifactId>**

**<version>5.3.36</version>**

**</dependency>**

**<!-- Spring AOP -->**

**<dependency>**

**<groupId>org.springframework</groupId>**

**<artifactId>spring-aop</artifactId>**

**<version>5.3.36</version>**

**</dependency>**

**<!-- Spring Web MVC -->**

**<dependency>**

**<groupId>org.springframework</groupId>**

**<artifactId>spring-webmvc</artifactId>**

**<version>5.3.36</version>**

**</dependency>**

**</dependencies>**

**<!-- Maven Compiler Plugin -->**

**<build>**

**<plugins>**

**<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>**