# **WEEK 3 Mandatory HandsOn**

# **Spring core and Maven Exercises - 1,2,3**

**Questions:**

**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:

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.

**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:

o Ensure that BookService class has a setter method for BookRepository.

3. Test the Configuration:

o Run the LibraryManagementApplication main class to verify the dependency injection.

**Exercise 3: Implementing Logging with Spring AOP**

Scenario:

The library management application requires logging capabilities to track method execution times.

Steps:

1. Add Spring AOP Dependency:

o Update pom.xml to include Spring AOP dependency.

2. Create an Aspect for Logging:

o Create a package com.library.aspect and add a class LoggingAspect with a method to log execution times.

3. Enable AspectJ Support:

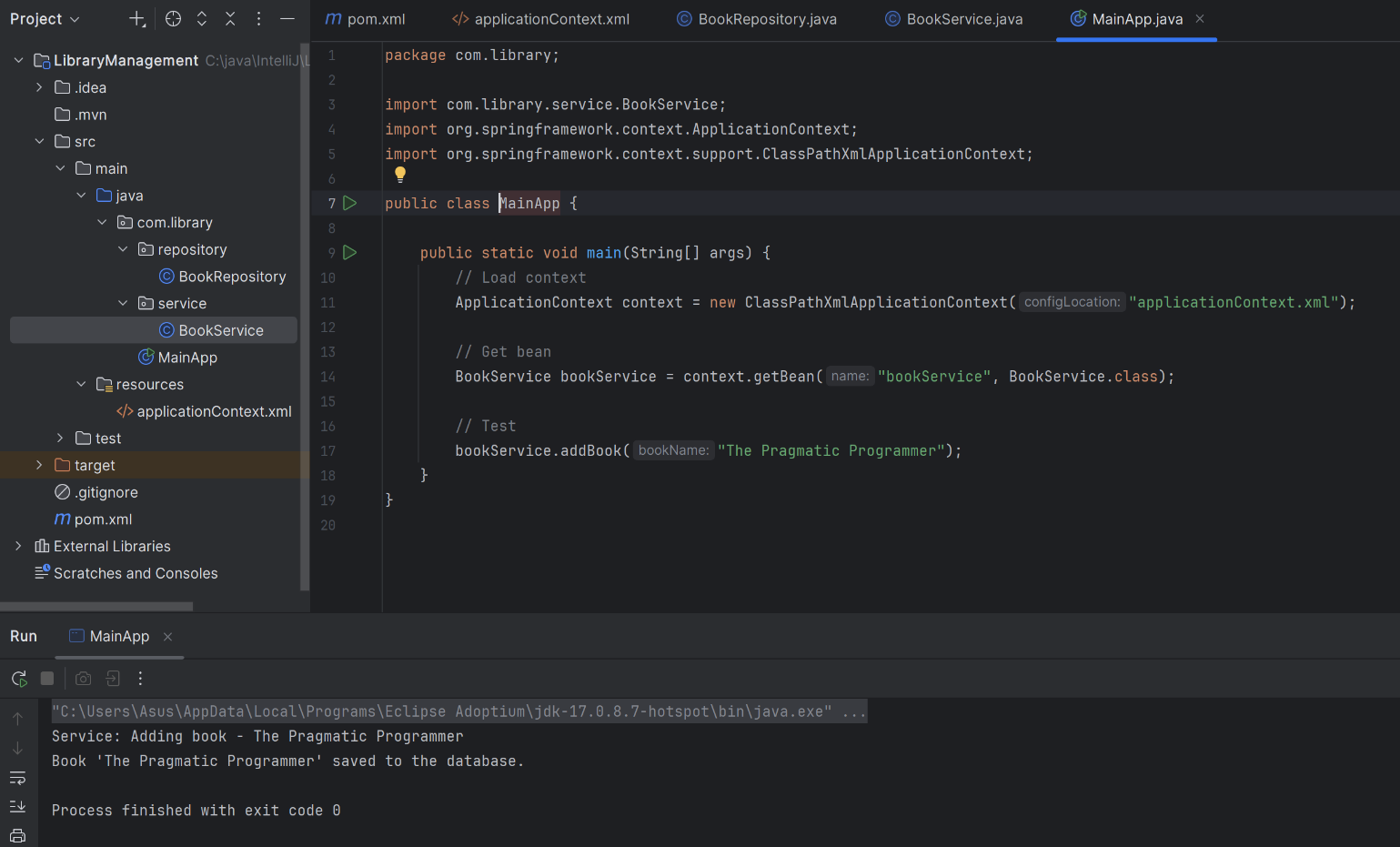
o Update applicationContext.xml to enable AspectJ support and register the aspect.

4. Test the Aspect:

o Run the LibraryManagementApplication main class and observe the console for log messages indicating method execution times**.**

**Solution**

**CODE: Inside the github repository.  
Output:**

****