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

BookRepository.java

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook() {  
 System.*out*.println("Book saved to the repository.");  
 }  
}

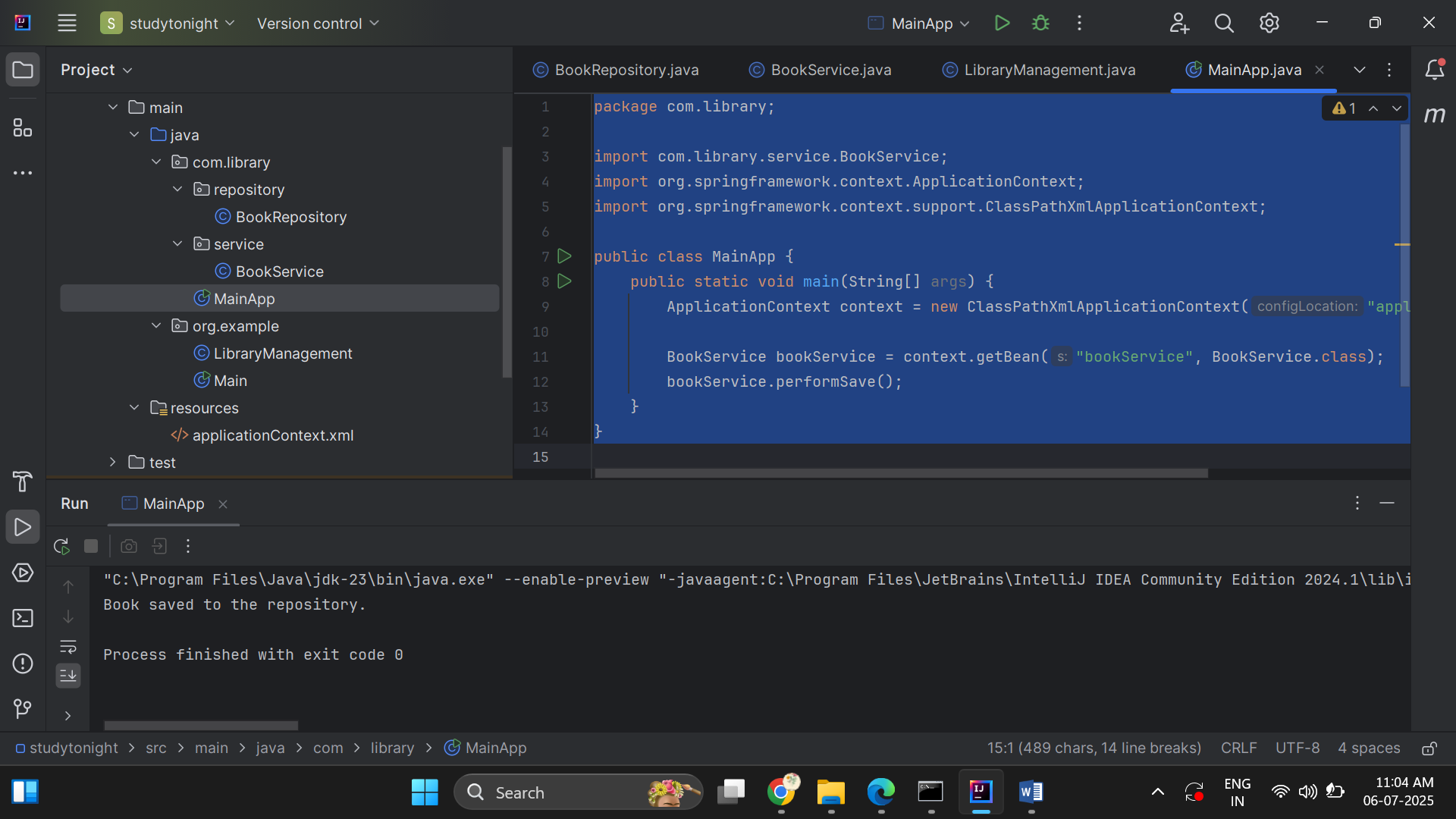
BookService.java

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 performSave() {  
 bookRepository.saveBook();  
 }  
}

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) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = context.getBean("bookService", BookService.class);  
 bookService.performSave();  
 }  
}

output:



**Exercise 2: Implementing Dependency Injection**

Repository.java

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook() {  
 System.*out*.println("Book saved to the repository.");  
 }  
 public void addBook(){  
 System.*out*.println("A book has been added to the library repository!!");  
 }  
}

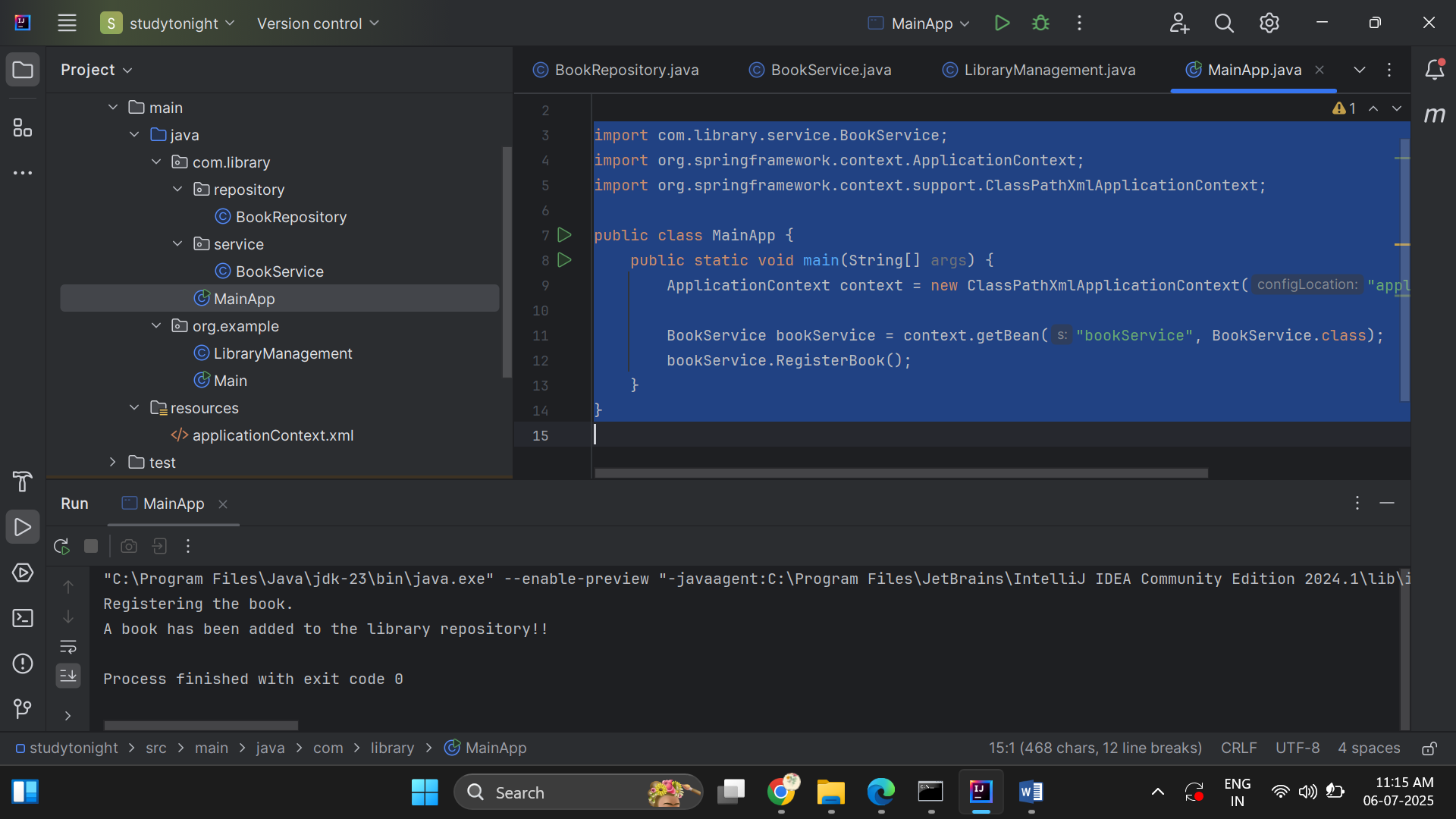
bookService.java(update)

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 RegisterBook(){  
 System.*out*.println("Registering the book.");  
 bookRepository.addBook();  
  
 }  
  
  
}

MainApp.java

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) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = context.getBean("bookService", BookService.class);  
 bookService.RegisterBook();  
 }  
}

output:



**Exercise 5: Configuring the Spring IoC Container**

It is responsible for:

* Creating objects (beans)
* Injecting dependencies
* Managing the complete life cycle of beans

**Repository.java**

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook() {  
 System.*out*.println("Book saved to the repository.");  
 }  
 public void addBook(){  
 System.*out*.println("A book has been added to the library repository!!");  
 }  
 public void store() {  
 System.*out*.println("Book stored in the BookRepository 📚 ");  
 }  
}

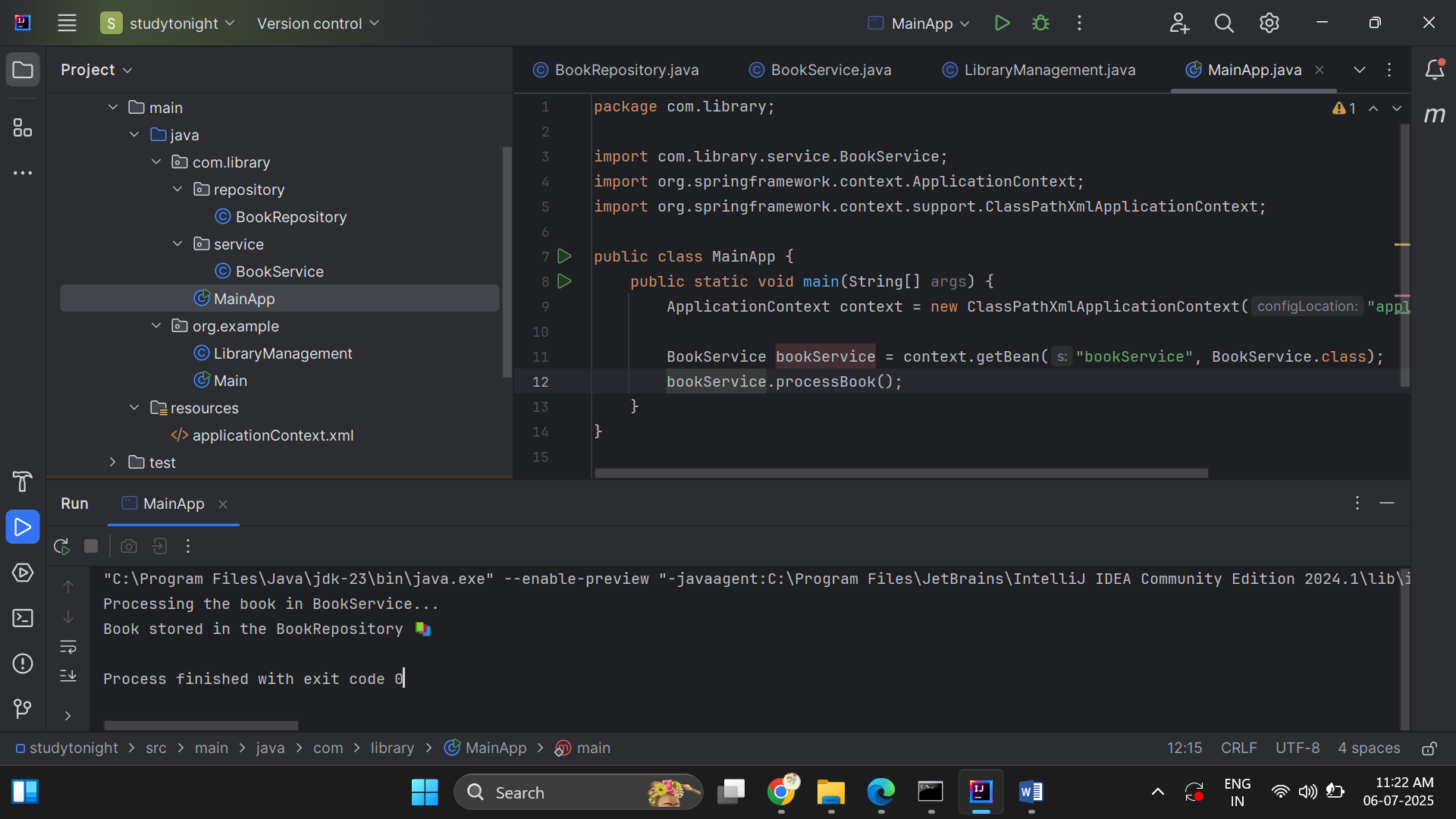
**BookService.java**

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 RegisterBook(){  
 System.*out*.println("Registering the book.");  
 bookRepository.addBook();  
  
 }  
 public void processBook() {  
 System.*out*.println("Processing the book in BookService...");  
 bookRepository.store();  
 }  
  
  
}

**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) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = context.getBean("bookService", BookService.class);  
 bookService.processBook();  
 }  
}

Output:



**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-SNAPSHOT</version>  
  
 <!-- Step 3: Maven Compiler Plugin -->  
 <properties>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 </properties>  
  
 <dependencies>  
 <!--Spring Context (Core + Beans + Context) -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
  
 <!--Spring AOP -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
  
 <!--Spring Web MVC -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.30</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>

**Exercise 7: Implementing Constructor and Setter Injection**

Logger.java

package com.library.util;  
  
public class Logger {  
 public void Logger(String message){  
 System.*out*.println("Message :" +message);  
 }  
}

BookRepository.java

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook() {  
 System.*out*.println("Book saved to the repository.");  
 }  
 public void addBook(){  
 System.*out*.println("A book has been added to the library repository!!");  
 }  
 public void store() {  
 System.*out*.println("Book stored in the BookRepository 📚 ");  
 }  
}

BookService.java

package com.library.service;  
  
import com.library.repository.BookRepository;  
import com.library.util.Logger;  
  
public class BookService {  
  
 private BookRepository bookRepository;  
 private Logger logger;  
  
 // Constructor injection  
 public BookService(Logger logger) {  
 this.logger = logger;  
 }  
  
 // Setter injection  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void processBook() {  
 logger.Logger("Processing book...");  
 bookRepository.saveBook();  
 }  
}

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">  
  
 <!-- Logger Bean -->  
 <bean id="logger" class="com.library.util.Logger" />  
  
 <!-- BookRepository Bean -->  
 <bean id="bookRepository" class="com.library.repository.BookRepository" />  
  
 <!-- BookService Bean with constructor and setter injection -->  
 <bean id="bookService" class="com.library.service.BookService">  
 <!-- Constructor injection for logger -->  
 <constructor-arg ref="logger" />  
 <!-- Setter injection for bookRepository -->  
 <property name="bookRepository" ref="bookRepository" />  
 </bean>  
  
</beans>

Output:  
