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

**Scenario:**

The library management application requires both constructor and setter injection for better control over bean initialization.

**App.java :-**

package org.example1;  
  
import org.example1.model.Book;  
import org.example1.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class App {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = context.getBean("bookService", BookService.class);  
 bookService.addBook(new Book("Clean Code"));  
 }  
}

**Book.java :-**

package org.example1.model;  
  
public class Book {  
 private String title;  
  
 public Book(String title) {  
 this.title = title;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
}

**BookRepository.java :-**

package org.example1.repository;  
  
import org.example1.model.Book;  
  
public class BookRepository {  
 public void save(Book book) {  
 System.*out*.println("Repository: Saved book - " + book.getTitle());  
 }  
}

**BookService.java :-**

package org.example1.service;  
  
import org.example1.model.Book;  
import org.example1.repository.BookRepository;  
  
public class BookService {  
 private BookRepository bookRepository;  
 private String libraryName;  
  
 // Constructor injection  
 public BookService(String libraryName) {  
 this.libraryName = libraryName;  
 }  
  
 // Setter injection  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void addBook(Book book) {  
 System.*out*.println("[" + libraryName + "] Adding Book: " + book.getTitle());  
 bookRepository.save(book);  
 }  
}

**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  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <!-- BookRepository Bean -->  
 <bean id="bookRepository" class="org.example1.repository.BookRepository"/>  
  
 <!-- BookService Bean -->  
 <bean id="bookService" class="org.example1.service.BookService">  
 <!-- Constructor Injection -->  
 <constructor-arg value="Central Library"/>  
 <!-- Setter Injection -->  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
  
</beans>

**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>org.example1</groupId>  
 <artifactId>LibraryManagement</artifactId>  
 <version>1.0</version>  
 <packaging>war</packaging>  
  
 <properties>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <finalName>LibraryManagement</finalName>  
 <plugins>  
 <!-- Compiler Plugin -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.11.0</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Output :-**

