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

**Scenario:**

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

**BookRepository.java**

package com.library;

public class BookRepository {

    public void save(String title) {

        System.out.println("SAVING BOOK: " + title.toUpperCase());

    }

    public String find(String title) {

        System.out.println("LOOKING UP BOOK: " + title.toUpperCase());

        return title;

    }

}

**BookService.java**

package com.library;

public class BookService {

    private String serviceName;

    private BookRepository bookRepository;

    // Constructor for constructor injection

    public BookService(String serviceName) {

        this.serviceName = serviceName;

        System.out.println("CONSTRUCTOR INJECTION: Service Name is " + this.serviceName.toUpperCase());

    }

    // Setter for setter injection

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

        System.out.println("SETTER INJECTION: BookRepository injected");

    }

    public void addBook(String title) {

        System.out.println("ADDING BOOK: " + title.toUpperCase());

        bookRepository.save(title);

    }

    public String findBook(String title) {

        System.out.println("FINDING BOOK: " + title.toUpperCase());

        return bookRepository.find(title);

    }

}

**LibraryManagementApplication.java**

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

    public static void main(String[] args) {

        System.out.println("STARTING LIBRARY MANAGEMENT APPLICATION");

        // Load Spring context from XML

        ApplicationContext context =

                new ClassPathXmlApplicationContext("applicationContext.xml");

        // Get BookService bean

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

        // Test methods

        bookService.addBook("Spring Framework Essentials");

        String foundBook = bookService.findBook("Java Concurrency in Practice");

        System.out.println("BOOK FOUND: " + foundBook.toUpperCase());

        System.out.println("APPLICATION COMPLETED SUCCESSFULLY");

    }

}

**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.BookRepository" />

    <!-- Define BookService bean with constructor and setter injection -->

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

        <!-- Constructor Injection -->

        <constructor-arg value="Library Service Bean" />

        <!-- Setter Injection -->

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

    </bean>

</beans>

**pom.xml**

<?xml version="1.0" encoding="UTF-8"?>

<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.ex6</groupId>

  <artifactId>ex6</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>ex6</name>

  <url>http://www.example.com</url>

  <properties>

    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <maven.compiler.source>1.8</maven.compiler.source>

    <maven.compiler.target>1.8</maven.compiler.target>

  </properties>

  <dependencies>

    <!-- Spring Context (IoC container, @Service, @Repository, etc.) -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-context</artifactId>

      <version>5.3.30</version>

    </dependency>

    <!-- Spring Beans (needed for core container features) -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-beans</artifactId>

      <version>5.3.30</version>

    </dependency>

    <!-- JUnit for testing -->

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.13.2</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <!-- Maven 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>

**Output:**

