**WEEK 3 SOLUTONS**

**Spring Core and Maven:**

**Exercise 7: Configuring the Spring IoC Container**

**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 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.5.3</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
 <groupId>com.exercise</groupId>  
 <artifactId>one</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>LibraryManagement</name>  
 <description>Library Management Project</description>  
 <properties>  
 <java.version>24</java.version>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.34</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>6.2.7</version>  
 </dependency>  
 <!-- Spring AOP -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 </dependency>  
  
 <!-- AspectJ Weaver -->  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>6.2.6</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.14.0</version>  
 <configuration>  
 <source>8</source>  
 <target>8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**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"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop.xsd">  
  
 <!-- Enable AspectJ auto proxying -->  
 <aop:aspectj-autoproxy />  
 <!-- Register Aspect -->  
 <bean id="bookRepository" class="com.exercise.one.repository.BookRepository"/>  
  
 <bean id="bookService" class="com.exercise.one.service.BookService">  
 <!-- Constructor Injection -->  
 <constructor-arg ref="bookRepository"/>  
  
 <!-- Setter Injection -->  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>

</beans>

**BookRepository.java:**

package com.exercise.one.repository;  
  
public class BookRepository {  
 public void saveBook(String s){  
 System.*out*.println(s+" ,Book Saved");  
 }  
}

**BookService.java:**

package com.exercise.one.service;  
  
import com.exercise.one.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@Service  
public class BookService {  
 private BookRepository bookRepository;  
 public List<String> books = new ArrayList<>();  
 public BookService(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 System.*out*.println("Constructor injection done");  
 }  
 public void setBookRepository(BookRepository bookRepository){  
 this.bookRepository=bookRepository;  
 System.*out*.println("Setter injection done");  
 }  
 public void displayAllBooks(){  
 System.*out*.println(books);  
 }  
 public void addBook(String s){  
 books.add(s);  
 System.*out*.println("Book Added to library.");  
 bookRepository.saveBook(s);  
 }  
}

**Librarian.java:**

package com.exercise.one;  
import com.exercise.one.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class Librarian {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService bookService = context.getBean(BookService.class);  
 bookService.addBook("The Alchemist");  
 }

}

**LoggingAspect.java**

package com.exercise.one.aspect;  
  
import org.aspectj.lang.ProceedingJoinPoint;  
import org.aspectj.lang.annotation.Around;  
import org.aspectj.lang.annotation.Aspect;  
  
@Aspect  
public class LoggingAspect {  
  
 @Around("execution(\* com.exercise.one.service.\*.\*(..))")  
 public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {  
 long start = System.*currentTimeMillis*();  
  
 Object result = joinPoint.proceed(); // Run the method  
  
 long end = System.*currentTimeMillis*();  
 System.*out*.println("Executed method: " + joinPoint.getSignature().getName() +  
 " in " + (end - start) + " ms");  
  
 return result;  
 }  
}

Output:

