

EXERCISE -1

1. Set Up a Spring Project (Maven)

Project Name: LibraryManagement

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>com.library</groupId>
```

```
<artifactId>LibraryManagement</artifactId>
```

```
<version>1.0-SNAPSHOT</version>
```

```
<dependencies>
```

```
<!-- Spring Core -->
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-context</artifactId>
```

```
<version>5.3.29</version>
```

```
</dependency>
```

```
</dependencies>
```

```
</project>
```

2. Create XML Configuration File

```
<?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">
```

```
<bean id="bookRepository" class="com.library.repository.BookRepository" />

<bean id="bookService" class="com.library.service.BookService">
  <property name="bookRepository" ref="bookRepository"/>
</bean>

</beans>
```

BookRepository Class

```
package com.library.repository;

public class BookRepository {
    public String getBookTitle() {
        return "Effective Java";
    }
}
```

BookService Class

```
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void printBookTitle() {
        System.out.println("Book Title: " + bookRepository.getBookTitle());
    }
}
```

Main Class to Run the App

```
package com.library;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.library.service.BookService;

public class MainApp {
    public static void main(String[] args) {
```

```

        ApplicationContext context = new
ClassPathXmlApplicationContext("applicationContext.xml");

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

```

Output

Book Title: Effective Java

EXERCISE -2

BookService.java

```
package com.library.service;
```

```
import com.library.repository.BookRepository;
```

```

public class BookService {
    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void printBookTitle() {
        System.out.println("Book Title: " + bookRepository.getBookTitle());
    }
}

```

MainApp.java

```
package com.library;
```

```

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.library.service.BookService;

```

```

public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context = new
ClassPathXmlApplicationContext("applicationContext.xml");

        BookService bookService = (BookService) context.getBean("bookService");
        bookService.printBookTitle(); // Should output: Book Title: Effective Java
    }
}

```

EXERCISE -3

Package: com.library.aspect

File: LoggingAspect.java

```
package com.library.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.library.service.*(..))")
    public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {
        long start = System.currentTimeMillis();

        Object result = joinPoint.proceed();

        long timeTaken = System.currentTimeMillis() - start;
        System.out.println("Execution of " + joinPoint.getSignature() + " took " + timeTaken + "
ms");

        return result;
    }
}
```

MainApp.java

```
package com.library;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.library.service.BookService;

public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context = new
        ClassPathXmlApplicationContext("applicationContext.xml");

        BookService bookService = (BookService) context.getBean("bookService");
        bookService.printBookTitle();
    }
}
```

Output

Book Title: Effective Java

Execution of void com.library.service.BookService.printBookTitle() took 2 ms

EXERCISE -4

New Maven Project

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>com.library</groupId>
  <artifactId>LibraryManagement</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <!-- Spring Context -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.3.29</version>
    </dependency>

    <!-- Spring AOP -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-aop</artifactId>
      <version>5.3.29</version>
    </dependency>

    <!-- Spring WebMVC -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.3.29</version>
    </dependency>

    <!-- AspectJ Weaver (needed for AOP) -->
    <dependency>
      <groupId>org.aspectj</groupId>
      <artifactId>aspectjweaver</artifactId>
      <version>1.9.21</version>
    </dependency>
  </dependencies>

  <!-- 3. Configure Maven Compiler Plugin -->
  <build>
    <plugins>
```

```

<!-- Compiler Plugin to use Java 1.8 -->
<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>

```

EXERCISE-5

Update the BookService Class

```

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void displayBookTitle() {
        String title = bookRepository.getBookTitle();
        System.out.println("Book Title: " + title);
    }
}

```

Main

```

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");

    // Retrieve and use the BookService bean
    BookService bookService = (BookService) context.getBean("bookService");
    bookService.displayBookTitle();
}
}

```

OUTPUT

Book Title: Effective Java

EXERCISE -6

Annotate Classes

```

package com.library.service;

import com.library.repository.BookRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class BookService {

    private BookRepository bookRepository;

    @Autowired
    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void displayBookTitle() {
        System.out.println("Book Title: " + bookRepository.getBookTitle());
    }
}

```

```

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository
public class BookRepository {

    public String getBookTitle() {
        return "Clean Code";
    }
}

```

```
}  
}
```

```
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.class);  
        bookService.displayBookTitle();  
    }  
}
```

Output

Book Title: Clean Code

EXERCISE -7

BookService Class

```
package com.library.service;  
  
import com.library.repository.BookRepository;  
  
public class BookService {  
    private BookRepository bookRepository;  
    private String message;  
  
    public BookService(BookRepository bookRepository) {  
        this.bookRepository = bookRepository;  
    }  
  
    public void setMessage(String message) {  
        this.message = message;  
    }  
}
```



```

        public void displayBookDetails() {
            System.out.println("Book Title: " + bookRepository.getBookTitle());
            System.out.println("Message: " + message);
        }
    }
}

```

```

package com.library.repository;

```

```

public class BookRepository {
    public String getBookTitle() {
        return "The Pragmatic Programmer";
    }
}

```

MAINAPP

```

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

```

OUTPUT

Book Title: The Pragmatic Programmer

Message: Injected via setter

EXERCISE -8

Package: com.library.aspect

File: LoggingAspect.java

```

package com.library.aspect;

```

```

import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.After;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;

```

```

@Aspect
public class LoggingAspect {

    @Before("execution(* com.library.service.*(..))")
    public void beforeAdvice(JoinPoint joinPoint) {
        System.out.println("Before Method: " + joinPoint.getSignature().getName());
    }

    @After("execution(* com.library.service.*(..))")
    public void afterAdvice(JoinPoint joinPoint) {
        System.out.println("After Method: " + joinPoint.getSignature().getName());
    }
}

```

BookService

```

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;
    private String message;

    public BookService(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public void displayBookDetails() {
        System.out.println("Book Title: " + bookRepository.getBookTitle());
        System.out.println("Message: " + message);
    }
}

```

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

```

Output

```

Before Method: displayBookDetails
Book Title: The Pragmatic Programmer
Message: Hello from Spring AOP
After Method: displayBookDetails

```

EXERCISE -7

```

com.library.entity
Book.java

```

```

package com.library.entity;

```

```

import jakarta.persistence.*;

```

```

@Entity

```

```

public class Book {

```

```

    @Id

```

```

    @GeneratedValue(strategy = GenerationType.IDENTITY)

```

```

    private Long id;

```

```

    private String title;

```

```

    private String author;

```

```

    private String isbn;

```

```

    public Book() {}

```

```

    public Book(String title, String author, String isbn) {

```

```

        this.title = title;

```

```

        this.author = author;

```

```

        this.isbn = isbn;

```

```

    }

```

```

    public Long getId() { return id; }

```

```

    public void setId(Long id) { this.id = id; }

```

```

    public String getTitle() { return title; }

```

```

    public void setTitle(String title) { this.title = title; }

```

```

    public String getAuthor() { return author; }

```

```
    public void setAuthor(String author) { this.author = author; }

    public String getIsbn() { return isbn; }
    public void setIsbn(String isbn) { this.isbn = isbn; }
}
```

LibraryManagementApplication.java

```
package com.library;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class LibraryManagementApplication {
    public static void main(String[] args) {
        SpringApplication.run(LibraryManagementApplication.class, args);
    }
}
```

POST http://localhost:8080/api/books
Content-Type: application/json

```
{
  "title": "Clean Code",
  "author": "Robert C. Martin",
  "isbn": "9780132350884"
}
```