**Exercise 1: Basic Spring Application**

**BookService.java**

package com.library.service;

public class BookService {

public void manageBooks() {

System.out.println("Managing books...");

}

}

**BookRepository.java**

package com.library.repository;

public class BookRepository {

public void saveBook() {

System.out.println("Saving book...");

}

}

**LibraryManagementApplication.java**

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

bookService.manageBooks();

}

}

**Exercise 2: Dependency Injection**

**BookService.java** (with Dependency Injection)

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 manageBooks() {

bookRepository.saveBook();

System.out.println("Managing books...");

}

}

**Exercise 3: Logging with Spring AOP**

**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.BookService.manageBooks(..))")

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long start = System.currentTimeMillis();

Object proceed = joinPoint.proceed();

long executionTime = System.currentTimeMillis() - start;

System.out.println(joinPoint.getSignature() + " executed in " + executionTime + "ms");

return proceed;

}

}

**Exercise 4: ApplicationContext and BeanFactory**

**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 manageBooks() {

bookRepository.saveBook();

System.out.println("Managing books...");

}

}

**BookRepository.java**

package com.library.repository;

public class BookRepository {

public void saveBook() {

System.out.println("Saving book...");

}

}

**LibraryManagementApplication.java** (using BeanFactory)

package com.library;

import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.core.io.ClassPathResource;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

BeanFactory factory = new XmlBeanFactory(new ClassPathResource("applicationContext.xml"));

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

bookService.manageBooks();

}

}

**Exercise 5: Spring IoC Container**

**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 manageBooks() {

bookRepository.saveBook();

System.out.println("Managing books...");

}

}

**BookRepository.java**

package com.library.repository;

public class BookRepository {

public void saveBook() {

System.out.println("Saving book...");

}

}

**LibraryManagementApplication.java**

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

bookService.manageBooks();

}

}

**Exercise 6: Beans with Annotations**

**BookService.java** (with Annotations)

package com.library.service;

import org.springframework.stereotype.Service;

@Service

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void manageBooks() {

bookRepository.saveBook();

System.out.println("Managing books...");

}

}

**BookRepository.java** (with Annotations)

java

Copy code

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

public void saveBook() {

System.out.println("Saving book...");

}

}

**LibraryManagementApplication.java**

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

bookService.manageBooks();

}

}

**Exercise 7: Constructor and Setter Injection**

**BookService.java** (with Constructor Injection)

java

Copy code

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public BookService(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void manageBooks() {

bookRepository.saveBook();

System.out.println("Managing books...");

}

}

**LibraryManagementApplication.java**

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

bookService.manageBooks();

}

}