**Exercise 1: Configuring a Basic Spring Application:**

**BookRepository.java:**

public class BookRepository {

public void save() {

System.out.println("Book saved to repository.");

}

}

**BookService.java:**

public class BookService {

public void addBook() {

System.out.println("BookService: Adding a book...");

}

}

**LibraryManagementApplication.java:**

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

service.addBook();

}

}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Exercise 2: Implementing Dependency Injection:**

**BookRepository.java:**

public class BookRepository {

public void save() {

System.out.println("Book saved to repository.");

}

}

**BookService.java:**

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook() {

System.out.println("BookService: Adding a book...");

bookRepository.save();

}

}

**LibraryManagementApplication.java:**

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

service.addBook();

}

}

**Output:**

**A screen shot of a number

AI-generated content may be incorrect.**

**Exercise 3: Implementing Logging with Spring AOP:**

**BookRepository.java:**

public class BookRepository {

public void save() {

System.out.println("Book saved to repository.");

}

}

**BookService.java:**

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook() {

System.out.println("BookService: Adding a book...");

bookRepository.save();

}

}

**LoggingAspect.java:**

import org.aspectj.lang.ProceedingJoinPoint;

public class LoggingAspect {

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long start = System.currentTimeMillis();

Object result = joinPoint.proceed();

long end = System.currentTimeMillis();

System.out.println("Execution time: " + (end - start) + " ms");

return result;

}

}

**LibraryManagementApplication.java:**

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

service.addBook();

}

}

**Output:**

**A computer screen with white lines and numbers

AI-generated content may be incorrect.**

**Exercise 4: Creating and Configuring a Maven Project:**

**LibraryManagementApplication.java:**

public class LibraryManagementApplication {

public static void main(String[] args) {

System.out.println("Maven project setup completed successfully.");

}

}

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Exercise 5: Configuring the Spring IoC Container:**

**BookRepository.java:**

public class BookRepository {

public void save() {

System.out.println("Book saved to repository.");

}

}

**BookService.java:**

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook() {

System.out.println("BookService: Adding a book...");

bookRepository.save();

}

}

**LibraryManagementApplication.java:**

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

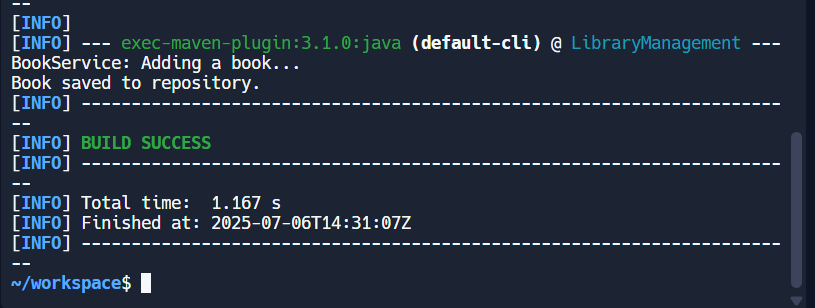
ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

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

service.addBook();

}

}**Output:**

****

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

**BookRepository.java:**

public class BookRepository {

public void save() {

System.out.println("Book saved to repository.");

}

}

**BookService.java:**

public class BookService {

private BookRepository bookRepository;

private String description;

public BookService(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void setDescription(String description) {

this.description = description;

}

public void addBook() {

System.out.println("BookService: " + description);

bookRepository.save();

}

}

**LibraryManagementApplication.java:**

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

service.addBook();

}

}

**Output:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Exercise 9: Creating a Spring Boot Application:**

**BookRepository.java:**

package com.example.library;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Long> {

}

**BookService.java:**

package com.example.library;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class BookService {

@Autowired

private BookRepository bookRepository;

public List<Book> getAllBooks() {

return bookRepository.findAll();

}

public Book addBook(String title, String author) {

return bookRepository.save(new Book(title, author));

}

}

**BookService.java:**

package com.example.library;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

@Entity

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String author;

public Book() {}

public Book(String title, String author) {

this.title = title;

this.author = author;

}

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;

}

}

**LibraryManagementApplication.java:**

package com.example.library;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication(

exclude = {

org.springframework.boot.autoconfigure.r2dbc.R2dbcAutoConfiguration.class

}

)

public class LibraryManagementApplication {

public static void main(String[] args) {

SpringApplication.run(LibraryManagementApplication.class, args);

}

}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**