Ex-1

**Configuring a Basic Spring Application**

**Bookrepo.java code-**

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Book '" + bookName + "' saved to the repository.");

}

}

Bookservice.java code-

package com.library.service;

public class BookService {

public void addBook(String bookName) {

System.out.println("Adding book: " + bookName);

}

}

LibraryManagementApplication.java code-

package com.library;

import com.library.service.BookService;

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

bookService.addBook("The Great Gatsby");

}

}

OUTPUT---

Adding book: The Great Gatsby

**Exercise 2: Implementing Dependency Injection**

NotificationService.java-- package com.library.service;

public class NotificationService {

public void sendConfirmation(String bookName) {

System.out.println("Notification: Book '" + bookName + "' has been added to the library.");

}

}

Bookservice.java code—

package com.library.service;

public class BookService {

private NotificationService notificationService;

// Setter injection

public void setNotificationService(NotificationService notificationService) {

this.notificationService = notificationService;

}

public void addBook(String bookName) {

System.out.println("Adding book: " + bookName);

notificationService.sendConfirmation(bookName);

}

}

Main class---

package com.library;

import com.library.service.BookService;

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

bookService.addBook("To Kill a Mockingbird");

}

}

OUTPUT

Adding book: To Kill a Mockingbird

Notification: Book 'To Kill a Mockingbird' has been added to the library.

EXERCISE -4

BOOKSERVICE.java

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter for DI

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("Adding book: " + bookName);

bookRepository.saveBook(bookName);

}

}

Bookrepo.java

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Book '" + bookName + "' saved to the repository.");

}

}

Librarymanagement.java

package com.library;

import com.library.service.BookService;

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

bookService.addBook("Pride and Prejudice");

}

}

Output-- Adding book: Pride and Prejudice

Book 'Pride and Prejudice' saved to the repository.