Exercise 2: Implementing Dependency Injection

# Scenario:

In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.

# 1. Modify the XML Configuration

Update applicationContext.xml to wire BookRepository into BookService:

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

# 2. Update the BookService Class

Ensure that BookService has a setter method for BookRepository:

public class BookService {  
 private BookRepository bookRepository;  
  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void addBook(String bookName) {  
 System.out.println("Adding book: " + bookName);  
 bookRepository.saveBook(bookName);  
 }  
}

# 3. Test the Configuration

Run the LibraryManagementApplication main class to verify the dependency injection:

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
BookService bookService = (BookService) context.getBean("bookService");  
bookService.addBook("1984");

# Expected Output:

Adding book: 1984  
Book saved: 1984