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

**Code:**

package com.library.repository;

public class BookRepository {

    public void saveBook(String title) {

        System.out.println("Book '" + title + "' saved to the database.");

    }

}

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    // Setter-based Dependency Injection

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void addBook(String title) {

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

        bookRepository.saveBook(title);

    }

}

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.addBook("The Alchemist");

    }

}

**Output:**

A screen shot of a computer

AI-generated content may be incorrect.

**Exercise 2: Implementing Dependency Injection**

**Code:**

package com.example.main;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.service.BookService;

public class LibraryManagementApplication {

    public static void main(String[] args) {

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

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

        bookService.displayBooks();

    }

}

package com.example.repository;

import java.util.Arrays;

import java.util.List;

public class BookRepository {

    public List<String> findAll() {

        return Arrays.asList("Book A", "Book B", "Book C");

    }

}

package com.example.service;

import com.example.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void displayBooks() {

        bookRepository.findAll().forEach(System.out::println);

    }

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

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

**Code:**

<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.example.library</groupId>

    <artifactId>LibraryManagement</artifactId>

    <version>1.0-SNAPSHOT</version>

    <properties>

        <java.version>1.8</java.version>

        <spring.version>5.3.30</spring.version> <!-- Or latest compatible -->

    </properties>

    <dependencies>

        <!-- Spring Context -->

        <dependency>

            <groupId>org.springframework</groupId>

            <artifactId>spring-context</artifactId>

            <version>${spring.version}</version>

        </dependency>

        <!-- Spring AOP -->

        <dependency>

            <groupId>org.springframework</groupId>

            <artifactId>spring-aop</artifactId>

            <version>${spring.version}</version>

        </dependency>

        <!-- Spring WebMVC -->

        <dependency>

            <groupId>org.springframework</groupId>

            <artifactId>spring-webmvc</artifactId>

            <version>${spring.version}</version>

        </dependency>

        <!-- Optional: Servlet API (for web apps) -->

        <dependency>

            <groupId>javax.servlet</groupId>

            <artifactId>javax.servlet-api</artifactId>

            <version>4.0.1</version>

            <scope>provided</scope>

        </dependency>

        <!-- JUnit 4 dependency -->

        <dependency>

            <groupId>junit</groupId>

            <artifactId>junit</artifactId>

            <version>4.13.2</version>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <!-- Compiler plugin -->

            <plugin>

                <groupId>org.apache.maven.plugins</groupId>

                <artifactId>maven-compiler-plugin</artifactId>

                <version>3.10.1</version>

                <configuration>

                    <source>1.8</source>

                    <target>1.8</target>

                </configuration>

            </plugin>

        </plugins>

    </build>

</project>

**Output:**

A screen shot of a computer

AI-generated content may be incorrect.