## Spring Core - Exercise 1: Basic Spring Configuration

pom.xml dependency:  
<dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
</dependency>  
  
Bean class - HelloService.java:  
public class HelloService {  
 public void sayHello() {  
 System.out.println("Hello from Spring!");  
 }  
}  
  
applicationContext.xml:  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="helloService" class="com.example.HelloService" />  
</beans>  
  
MainApp.java:  
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");  
 HelloService service = (HelloService) context.getBean("helloService");  
 service.sayHello();  
 }  
}

## Spring Core - Exercise 2: Dependency Injection

Service class - GreetingService.java:  
public class GreetingService {  
 public void greet(String name) {  
 System.out.println("Hello, " + name);  
 }  
}  
  
Dependent class - Client.java:  
public class Client {  
 private GreetingService greetingService;  
  
 public void setGreetingService(GreetingService greetingService) {  
 this.greetingService = greetingService;  
 }  
  
 public void sendGreeting(String name) {  
 greetingService.greet(name);  
 }  
}  
  
applicationContext.xml:  
<bean id="greetingService" class="com.example.GreetingService" />  
<bean id="client" class="com.example.Client">  
 <property name="greetingService" ref="greetingService" />  
</bean>  
  
MainApp.java:  
ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
Client client = (Client) context.getBean("client");  
client.sendGreeting("World");

## Spring Core - Exercise 4: Maven Project Structure

Typical Maven Directory Structure:  
my-spring-project/  
├── pom.xml  
└── src/  
 ├── main/  
 │ ├── java/  
 │ │ └── com/example/  
 │ │ ├── HelloService.java  
 │ │ ├── GreetingService.java  
 │ │ ├── Client.java  
 │ │ └── MainApp.java  
 │ └── resources/  
 │ └── applicationContext.xml  
 └── test/  
 └── java/

HelloService.java:  
package com.example;  
  
public class HelloService {  
 public void sayHello() {  
 System.out.println("Hello from Spring Core + Maven!");  
 }  
}

applicationContext.xml:  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="helloService" class="com.example.HelloService" />  
</beans>

MainApp.java:  
package com.example;  
  
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");  
 HelloService service = (HelloService) context.getBean("helloService");  
 service.sayHello();  
 }  
}

