Q1.

package com.spring.MavenSpring12;

public interface Sim {

void call();

void data();

}

**package** com.spring.MavenSpring12;

**public** **class** Airtel **implements** Sim {

@Override

**public** **void** call() {

System.***out***.println("Calling from Airtel");

}

@Override

**public** **void** data() {

System.***out***.println("Data from Airtel");

}

}

**package** com.spring.MavenSpring12;

**public** **class** Vodafone **implements** Sim {

@Override

**public** **void** call() {

System.***out***.println("Calling from Vodafone");

}

@Override

**public** **void** data() {

System.***out***.println("Data from Vodafone");

}

}

Main Class

**package** com.spring.MavenSpring12;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** Mobile {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ApplicationContext context = **new** ClassPathXmlApplicationContext("beans.xml");

Sim sim=(Sim) context.getBean("air");

Sim sim1=(Sim) context.getBean("voda");

sim.call();

sim.data();

sim1.call();

sim1.data();

}

}

Beans.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<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 definitions here -->

<bean id=*"air"* class=*"com.spring.MavenSpring12.Airtel"*></bean>

<bean id=*"voda"* class=*"com.spring.MavenSpring12.Vodafone"*></bean>

</beans>

Q2. package com.example.studentapp.model;

public class Student {

private String name;

private int rollno;

private String email;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getRollno() {

return rollno;

}

public void setRollno(int rollno) {

this.rollno = rollno;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public void display() {

System.out.println("Student Details:");

System.out.println("Name: " + name);

System.out.println("Roll No: " + rollno);

System.out.println("Email: " + email);

System.out.println("--------------------------");

}

}

<?xml version="1.0" encoding="UTF-8"?>

<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">

<!-- Student Bean 1 -->

<bean id="student1" class="com.example.studentapp.model.Student">

<property name="name" value="Alice Johnson"/>

<property name="rollno" value="101"/>

<property name="email" value="alice@example.com"/>

</bean>

<!-- Student Bean 2 -->

<bean id="student2" class="com.example.studentapp.model.Student">

<property name="name" value="Bob Smith"/>

<property name="rollno" value="102"/>

<property name="email" value="bob@example.com"/>

</bean>

</beans>

package com.example.studentapp;

import com.example.studentapp.model.Student;

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");

Student student1 = (Student) context.getBean("student1");

Student student2 = (Student) context.getBean("student2");

student1.display();

student2.display();

// Alternatively, accessing via getters

System.out.println("Student1 Name: " + student1.getName());

System.out.println("Student2 Email: " + student2.getEmail());

}

}

<dependencies>

<!-- Spring Core and Context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.32</version>

</dependency>

</dependencies>

Use **constructor injection** if you want to inject values via the constructor

<bean id="student1" class="com.example.studentapp.model.Student">

<constructor-arg value="Alice Johnson"/>

<constructor-arg value="101"/>

<constructor-arg value="alice@example.com"/>

</bean>

| **Style** | **Required Constructor** | **XML Tag Used** |
| --- | --- | --- |
| Setter Injection | No-arg constructor | <property> |
| Constructor Injection | Parameterized constructor | <constructor-arg> |
| Q3. |  |  |

package com.example.vehicleapp.model;

public interface Vehicle {

void start();

void stop();

}

package com.example.vehicleapp.model;

public class Car implements Vehicle {

private String name;

private int id;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

@Override

public void start() {

System.out.println("Car Started: Name = " + name + ", ID = " + id);

}

@Override

public void stop() {

System.out.println("Car Stopped: Name = " + name + ", ID = " + id);

}

}

package com.example.vehicleapp.model;

public class Bike implements Vehicle {

private String name;

private int id;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

@Override

public void start() {

System.out.println("Bike Started: Name = " + name + ", ID = " + id);

}

@Override

public void stop() {

System.out.println("Bike Stopped: Name = " + name + ", ID = " + id);

}

}

<?xml version="1.0" encoding="UTF-8"?>

<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">

<!-- Car Bean -->

<bean id="car" class="com.example.vehicleapp.model.Car">

<property name="name" value="Tesla Model S"/>

<property name="id" value="101"/>

</bean>

<!-- Bike Bean -->

<bean id="bike" class="com.example.vehicleapp.model.Bike">

<property name="name" value="Yamaha R15"/>

<property name="id" value="202"/>

</bean>

</beans>

package com.example.vehicleapp;

import com.example.vehicleapp.model.Vehicle;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Transport {

public static void main(String[] args) {

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

Vehicle car = (Vehicle) context.getBean("car");

Vehicle bike = (Vehicle) context.getBean("bike");

car.start();

car.stop();

bike.start();

bike.stop();

}

}

Q4.For Practice

package p1;

public interface Branch {

void Branchname();

}

package p1;

public class CSE implements Branch{

@Override

public void Branchname() {

System.out.println("I am from CSE");

}

}

package p1;

public class CSIT implements Branch{

@Override

public void Branchname() {

System.out.println("I am from CSIT");

}

}

package p1;

public class Student {

int RegdNo;

String name;

Branch B;

Student(int RegdNo,String name, Branch B)

{

this.RegdNo=RegdNo;

this.name=name;

this.B=B;

}

void Studentdetails()

{

System.out.println(RegdNo+" " + name+" "+ B);

}

}

package p1;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

class ApplicationMain {

public static void main(String[] args) {

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

Student s1=(Student) context.getBean("s1");

s1.Studentdetails();

}

}

Config.xml

<?xml version="1.0" encoding="UTF-8"?>

<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="stud" class="p1.Student">

<constructor-arg value="101" type="int"></constructor-arg>

<constructor-arg value="Akash" type="String"></constructor-arg>

</bean>

</beans>