1) Implement different types of dependency injections by using bean configuration.

\* Create Car Class and their properties with the help of the Setter dependency injection technique

Class

//1) Implement different types of dependency injections by using bean configuration.

//\* Create Car Class and their properties with the help of the Setter dependency injection technique

//\* Bus and their properties with the help of the Constructor dependency injection technique.

**package** SpringAssignment;

**public** **class** Car {

**private** String Car\_Name;

**private** String colour;

**private** String model;

//Generating Setters and Getters

**public** String getCar\_Name() {

**return** Car\_Name;

}

**public** **void** setCar\_Name(String car\_Name) {

Car\_Name = car\_Name;

}

**public** String getColour() {

**return** colour;

}

**public** **void** setColour(String colour) {

**this**.colour = colour;

}

**public** String getModel() {

**return** model;

}

**public** **void** setModel(String model) {

**this**.model = model;

}

@Override

**public** String toString() {

**return** "Car [Car\_Name=" + Car\_Name + ", colour=" + colour + ", model=" + model + "]";

}

}

Configuration File

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean name="car1" class="SpringAssignment.Car">

<property name="Car\_Name" value="Lamborghini"/>

<property name="colour" value="Red"/>

<property name="model" value="2023 Lamborghini Urus"/>

</bean>

</beans>

Main Class

**package** SpringAssignment;

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

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

// Main1 file for running the file

**public** **class** Main1

{

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

{

System.***out***.println( "The execution of project has started" );

ApplicationContext context = **new** ClassPathXmlApplicationContext("con1.xml");

Car car = (Car)context.getBean("car1"); //typecasting

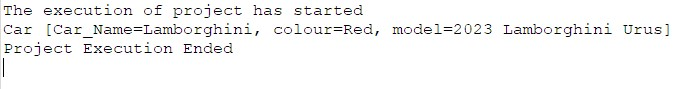
System.***out***.println(car); //printing the values

System.***out***.println("Project Execution Ended");

}

}

Output: -



\* Bus Class and their properties with the help of the Constructor dependency injection technique.

Class

**package** SpringAssignment;

**public** **class** Bus {

**private** String busName;

**private** String busColour;

**private** **int** busNum;

**public** Bus(String busName, String busColour, **int** busNum) {

**super**();

**this**.busName = busName;

**this**.busColour = busColour;

**this**.busNum = busNum;

}

@Override

**public** String toString() {

**return** busName+" " + busColour+" "+busNum;

}

}

Configuration File

<?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">

<!--Bus bus = new Bus();-->

<bean class="SpringAssignment.Bus" id = "bus1">

<constructor-arg value= "School Bus"></constructor-arg>

<constructor-arg value= "Yellow"></constructor-arg>

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

</bean>

</beans>

Main Class

package SpringAssignment;

import org.springframework.context.\*;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main2

{

public static void main(String[] args)

{

System.out.println( "The execution of project has started" );

ApplicationContext context = new

ClassPathXmlApplicationContext("SpringAssignment/con2.xml");

Bus bus = context.getBean("bus1", Bus.class);

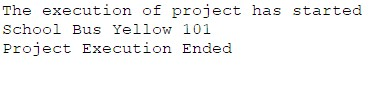
System.out.println(bus);

System.out.println("Project Execution Ended");

}

}

Output: -



2 ) Create Employee and Address class. Inject Address object into Employee by the help of @autowired annotation.

Employee Class

**package** SpringAssignment;

**import** org.springframework.beans.factory.annotation.Autowired;

**public** **class** Emp {

@Autowired

**private** Address address;

**public** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

System.***out***.print("setting value");

**this**.address = address;

}

@Override

**public** String toString() {

**return** "Emp [address=" + address + "]";

}

**public** Emp() {

**super**();

}

}

Address Class

**package** SpringAssignment;

**public** **class** Address {

**private** String area;

**private** String city;

**private** String State;

//getters and setters

**public** String getArea() {

**return** area;

}

**public** **void** setArea(String area) {

**this**.area = area;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getState() {

**return** State;

}

**public** **void** setState(String state) {

State = state;

}

@Override

**public** String toString() {

**return** "Address [area=" + area + ", city=" + city + ", State=" + State

+ "]";

}

}

Configuration File

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:p="http://www.springframework.org/schema/p"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:annotation-config />

<bean class="SpringAssignment.Address" name="address">

<property name="area" value="Aya Nagar"/>

<property name="city" value="New Delhi"/>

<property name="State" value="Delhi"/>

</bean>

<bean class="SpringAssignment.Emp" name="emp" />

</beans>

Main Class

package SpringAssignment;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test

{

public static void main( String[] args )

{

ApplicationContext context = new ClassPathXmlApplicationContext

("SpringAssignment/AutoConfi.xml");

Emp emp = context.getBean("emp", Emp.class); //object

System.out.println(emp); //printing the values

}

Output: -

