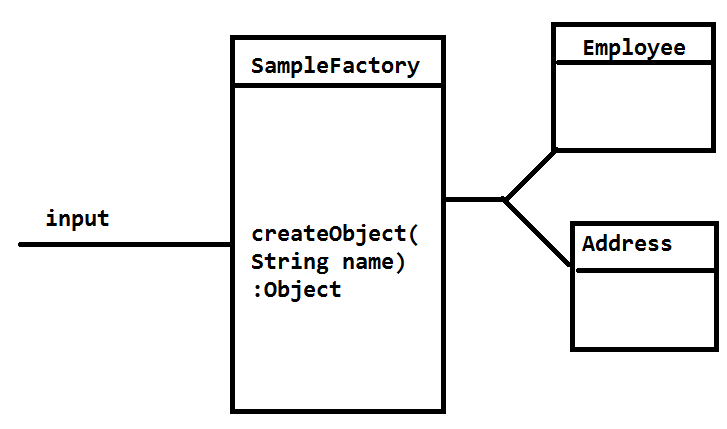
**Spring Instance Factory and Static Factory Design pattern:-**

**Spring supports in-built factory design pattern in 2 ways instance factory and static factory.**

**Here factory method returns object type. It will take dynamic input based on that, it will create the object. In this case we need to configure only Factory class in spring configuration file.**

**Ex:**

****

**Code:**

**package** com.app.module;

**public** **class** Employee {

**private** **int** empId;

**private** String empName;

**public** Employee() {

**super**();

}

**public** Employee(**int** empId, String empName) {

**super**();

**this**.empId = empId;

**this**.empName = empName;

}

**public** **int** getEmpId() {

**return** empId;

}

**public** **void** setEmpId(**int** empId) {

**this**.empId = empId;

}

**public** String getEmpName() {

**return** empName;

}

**public** **void** setEmpName(String empName) {

**this**.empName = empName;

}

@Override

**public** String toString() {

**return** "Employee [empId=" + empId + ", empName=" + empName + "]";

}

}

**package** com.app.module;

**import** java.util.List;

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

**private** **int** addrId;

**private** String loc;

**public** Address() {

**super**();

}

**public** Address(**int** addrId, String loc) {

**super**();

**this**.addrId = addrId;

**this**.loc = loc;

}

**public** **int** getAddrId() {

**return** addrId;

}

**public** **void** setAddrId(**int** addrId) {

**this**.addrId = addrId;

}

**public** String getLoc() {

**return** loc;

}

**public** **void** setLoc(String loc) {

**this**.loc = loc;

}

@Override

**public** String toString() {

**return** "Address [addrId=" + addrId + ", loc=" + loc + "]";

}

}

**package** com.app.module;

**public** **class** SimpleFactory {

**public** **static** Object createObject(String name){

**if**("emp".equalsIgnoreCase(name)){

**return** **new** Employee(10,"ABCD");

}**else** **if**("addr".equalsIgnoreCase(name)){

**return** **new** Address(200, "HYD");

}

**return** **null**;

}

}

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

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

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

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"*>

<!--

Java code :

Object sf=SampleFactory.createObject("emp");

-->

<bean class=*"com.app.module.SimpleFactory"* name=*"sf"* factory-method=*"createObject"*>

<constructor-arg>

<value>addr</value>

</constructor-arg>

</bean>

</beans>

Test.java

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

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

Object ob=context.getBean("sf");

**if**(ob **instanceof** Employee){

Employee emp=(Employee)ob;

System.*out*.println(emp);

}**else** **if**(ob **instanceof** Address){

Address add=(Address)ob;

System.*out*.println(add);

}

System.*out*.println("done");

}

Output:

Address [addrId=200, loc=HYD]

Done

For Instance factory changed cod is:

Factory class:

**package** com.app.module;

**public** **class** SimpleFactory {

**public** Object createObject(String name){

**if**("emp".equalsIgnoreCase(name)){

**return** **new** Employee(10,"ABCD");

}**else** **if**("addr".equalsIgnoreCase(name)){

**return** **new** Address(200, "HYD");

}

**return** **null**;

}

}

**XML Code:**

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

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

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

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"*>

<!--

Java code :

SimpleFactory sf=new SimpleFactory();

-->

<bean class=*"com.app.module.SimpleFactory"* name=*"sf1"* >

</bean>

<!-- Object sf=sf1.createObject("emp"); -->

<bean name=*"sf"* factory-bean=*"sf1"* factory-method=*"createObject"*>

<constructor-arg>

<value>emp</value>

</constructor-arg>

</bean>

</beans>

Output:

Employee [empId=10, empName=ABCD]

done