**STEPS**

* **create the class**
* **create the xml file to provide the values**
* **create the test class**
* **Load the spring jar files**
* **Run the test class**

Steps to create spring application

Let's see the 5 steps to create the first spring application.

1) Create Java class

This is the simple java bean class containing the name property only.

1. **package** com.javatpoint;
3. **public** **class** Student {
4. **private** String name;
6. **public** String getName() {
7. **return** name;
8. }
10. **public** **void** setName(String name) {
11. **this**.name = name;
12. }
14. **public** **void** displayInfo(){
15. System.out.println("Hello: "+name);
16. }
17. }

This is simple bean class, containing only one property name with its getters and setters method. This class contains one extra method named displayInfo() that prints the student name by the hello message.

2) Create the xml file

In case of myeclipse IDE, you don't need to create the xml file as myeclipse does this for yourselves. Open the applicationContext.xml file, and write the following code:

1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans
3. xmlns=*"http://www.springframework.org/schema/beans"*
4. xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*
5. xmlns:p=*"http://www.springframework.org/schema/p"*
6. xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd"*>
8. <bean id="studentbean" **class**="com.javatpoint.Student">
9. <property name="name" value="Vimal Jaiswal"></property>
10. </bean>
12. </beans>

The **bean** element is used to define the bean for the given class. The **property** subelement of bean specifies the property of the Student class named name. The value specified in the property element will be set in the Student class object by the IOC container.

3) Create the test class

Create the java class e.g. Test. Here we are getting the object of Student class from the IOC container using the getBean() method of BeanFactory. Let's see the code of test class.

2. **public** **class** Test {
3. **public** **static** **void** main(String[] args) {
4. ApplicationContext context =
5. **new** ClassPathXmlApplicationContext("applicationContext.xml");
6. );
8. Student student=(Student)context.getBean("studentbean");
9. student.displayInfo();
10. }
11. }

**getBean()**, which returns the object of the associated class.