Dependany Injection:

In our java class datamembers are nothing but the dependancies and assigning values to them is Dependancy Injection.

Types of DI:

1)Primitive Type

2)secondary type

3)Collection Type

This Dependancy Injection can be achieved in 2 ways –

1. Setter Injection
2. Constructor Injection

In Spring these dependancies will be injected by Spring IOC container to our Spring class (bean).

Spring container will come to know about injection type from spring configuration file.

Setter Injection :

In Setter Injection, container uses setter methods in POJO class to inject dependancies.

In this, our class must contain setter methods otherwise we cannot inject dependancies.

Constructor Injection :

In Constructor Injection, container uses constructor in class to inject dependancies.

Note:

If both Constructor & Setter Injections are applied on same property then Constructor Injection will be overridden by Setter Injection. Since, Constructor Injection takes place when object is created.

Setter Injection

Constructor Injection

Partial injection of dependancies is possible.

Partial injection of dependancies is not possible.

Setter injection will override constructor injected values.

Constructor injection will not override Setter injected values.

Setter injection makes bean class object as mutable (changeble).

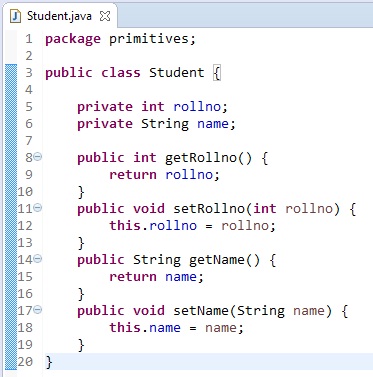
Constructor injection makes bean class object as immutable (unchangeble).

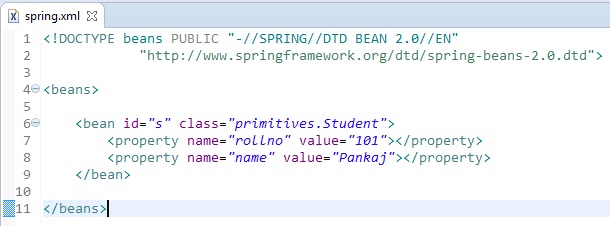
Setter Injection–primitives:

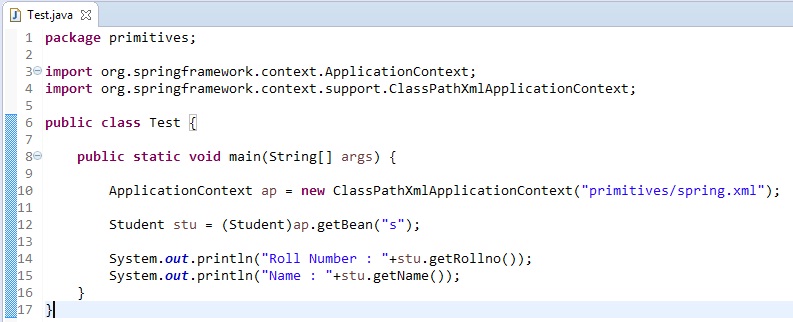
Files required : 1)Student.java

2) Test.java

3)spring.xml +SpringJars







Output :



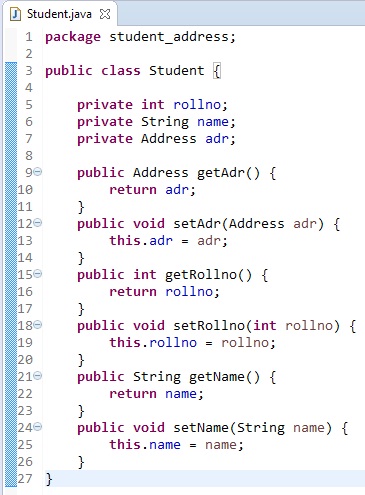
Setter Injection –secondary:

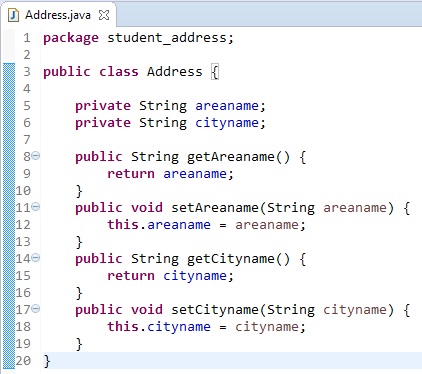
Files required : 1) Student.java

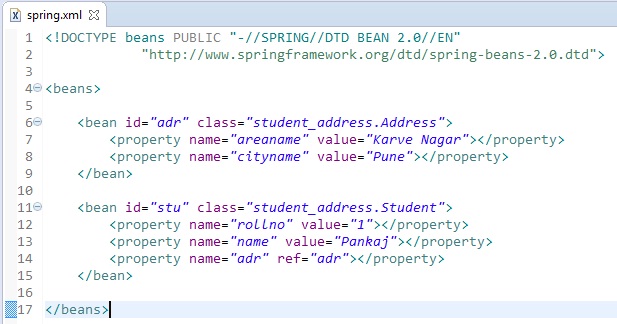
2) Address.java

3) Test.java

4) spring.xml +Spring Jars

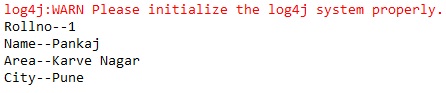








Output :

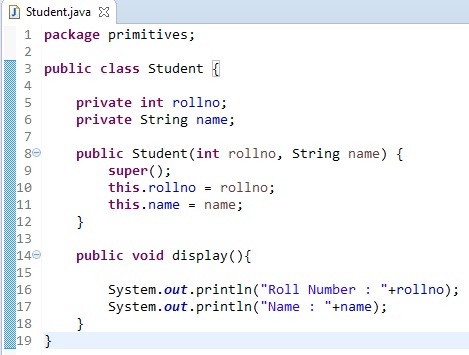


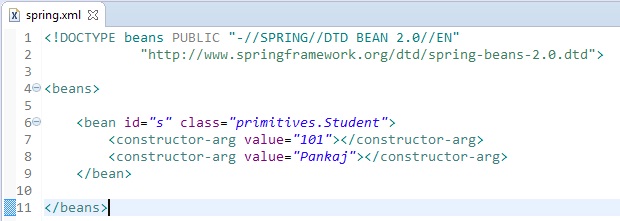
Constructor Injection –primitives:

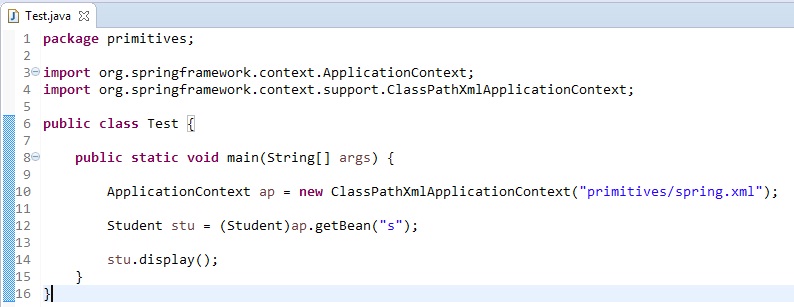
Files required : 1)Student.java

2) Test.java

3)spring.xml +SpringJars







Output :



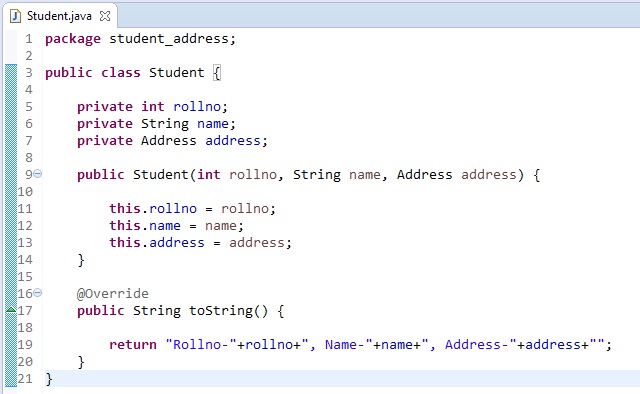
Constructor Injection –secondary:

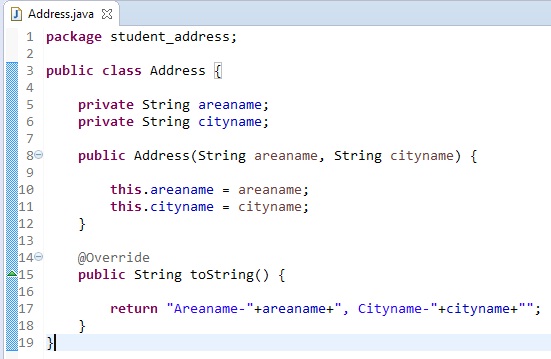
Files required : 1)Student.java

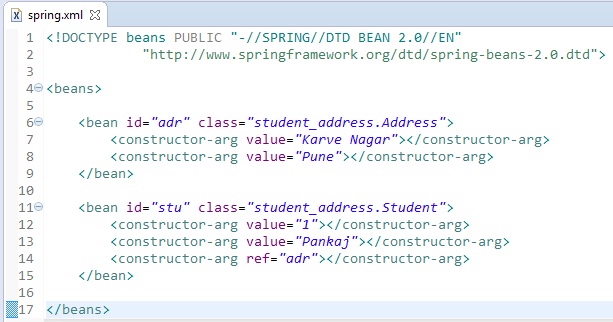
2) Address.java

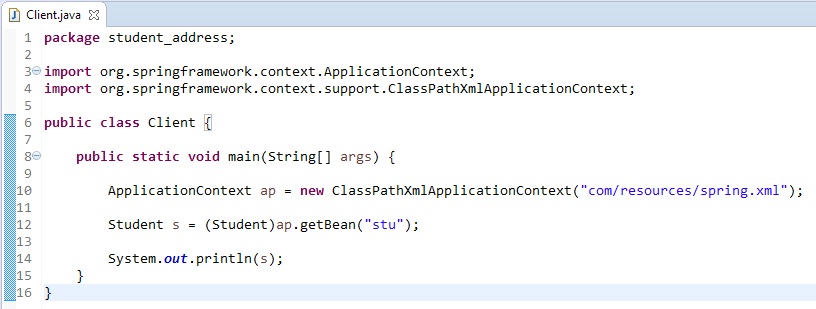
3) Client.java

4)spring.xml +SpringJars









`

Output :

