

# PRACTICAL 7(C) – SPRING DEPENDENCY INJECTION USING CONSTRUCTOR

## AIM

To demonstrate Dependency Injection in Spring Framework using Constructor Injection.

## THEORY

Constructor Injection is a type of Dependency Injection where the Spring container injects dependencies through the constructor of the class. It ensures that required dependencies are provided at the time of object creation.

## SOURCE CODE – Address.java

```
package springdemo;

public class Address {
    private String city;

    public Address(String city) {
        this.city = city;
    }

    public String getCity() {
        return city;
    }
}
```

## SOURCE CODE – Student.java

```
package springdemo;

public class Student {
    private int id;
    private Address address;

    public Student(int id, Address address) {
        this.id = id;
        this.address = address;
    }

    public void display() {
        System.out.println("Student ID: " + id);
        System.out.println("City: " + address.getCity());
    }
}
```

## SOURCE CODE – applicationContext.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="address" class="springdemo.Address">
        <constructor-arg value="Pune"/>
    </bean>

    <bean id="student" class="springdemo.Student">
        <constructor-arg value="202"/>
        <constructor-arg ref="address"/>
    </bean>
</beans>

```

## SOURCE CODE – MainApp.java

```

package springdemo;

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 s = (Student) context.getBean("student");
        s.display();
    }
}

```

## NETBEANS 7 – EXECUTION STEPS

1. Use the same Spring project and libraries from Practical 7(A)
2. Replace or add Address.java and Student.java with constructor-based versions
3. Update applicationContext.xml with constructor-arg tags
4. Run MainApp.java using Shift + F6

## OUTPUT

Student ID: 202 City: Pune

## RESULT

Thus, dependency injection using constructor was successfully implemented in Spring Framework.