

1.Basic Spring Application With Constructor Injection

MainDemo.java

```
package com.example;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class MainDemo {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        ApplicationContext context = new ClassPathXmlApplicationContext("config.xml");  
        student s = context.getBean("S",student.class);  
        System.out.println(s);  
        s.Read();  
    }  
}
```

Student.java

```
package com.example;

public class student {
    int Eno;
    String Ename;

    public student(int eno, String ename) {
        super();
        Eno = eno;
        Ename = ename;
    }

    public student(int eno) {
        super();
        Eno = eno;
    }

    @Override
    public String toString() {
        return "student [Eno=" + Eno + ", Ename=" + Ename + "]";
    }

    public void Read() {
        System.out.println("This is Employee class method");
    }
}
```

Config.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
https://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="S" class="com.example.student">
        <constructor-arg type="int" value="123"></constructor-arg>
        <constructor-arg type="String" value="Ramya"></constructor-arg>
    </bean>
</beans>
```

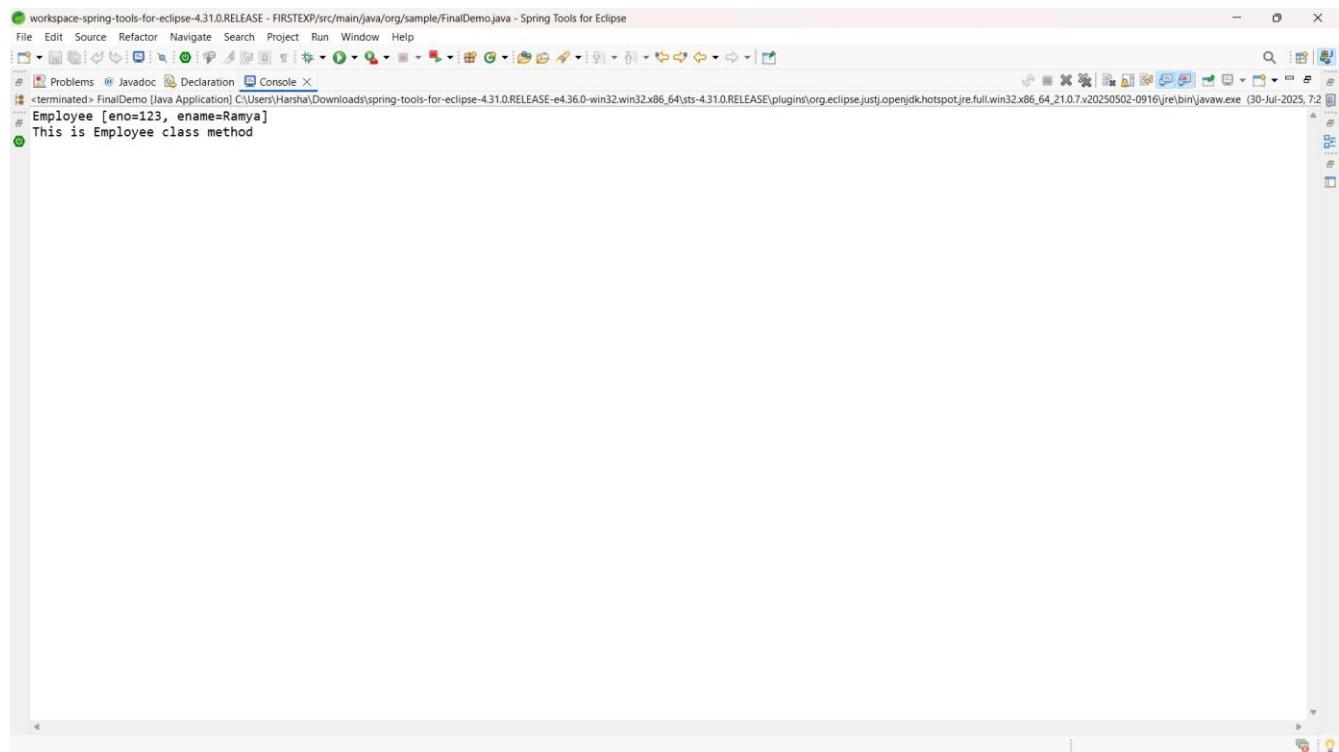
Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>nnn</groupId>
  <artifactId>SecondLab</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-core</artifactId>
      <version>6.2.8</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>6.2.8</version>
    </dependency>
  </dependencies>
</project>
```

Output :



The screenshot shows the Eclipse IDE interface with the title "workspace-spring-tools-for-eclipse-4.31.0.RELEASE - FIRSTEXP/src/main/java/org/sample/FinalDemo.java - Spring Tools for Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, Cut, Copy, Paste, Find, and Delete. The left sidebar has tabs for Problems, Javadoc, Declaration, and Console. The main workspace shows a single Java file "FinalDemo.java" with the following code:

```
Employee [eno=123, ename=Ramya]
This is Employee class method
```

The "Console" tab shows the output of the application, which includes the output of the constructor and the call to the class method.

2.Spring Application With Setter Injection

MainDemo.java

```
package com.programs;

import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.context.ApplicationContext;
public class FinalDemo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ApplicationContext context = new ClassPathXmlApplicationContext("App.xml");
        Employee e1= context.getBean("e",Employee.class);
        System.out.println(e1);
    }
}
```

Employee.java

```
package com.programs;

public class Employee {

    int Eno;
    String Ename;

    public int getEno() {
        return Eno;
    }

    public void setEno(int eno) {
        Eno = eno;
    }

    public String getEname() {
        return Ename;
    }

    public void setEname(String ename) {
        Ename = ename;
    }

    @Override

    public String toString() {
        return "Employee [Eno=" + Eno + ", Ename=" + Ename + "]";
    }
}
```

App.xml(config.xml 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
https://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="e" class="com.programs.Employee">
        <property name="eno" value="356"></property>
        <property name="ename" value="Ramya"></property>
    </bean>
</beans>
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>hii</groupId>
  <artifactId>FirstLab</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-core</artifactId>
      <version>6.2.9</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>6.2.9</version>
    </dependency>
  </dependencies>
</project>
```

OUTPUT :

