

Constructor Injection

➤ **Example:**

1. Create the Employee class.

```
12  public class Employee {
13
14      private String name;
15      private int age;
16      private double salary;
17
18      //injecting name,age and salary in the constructor
19      public Employee(String name, int age, double salary) {
20          this.name = name;
21          this.age = age;
22          this.salary = salary;
23      }
24
25      public String getName() {
26          return name;
27      }
28
29      public int getAge() {
30          return age;
31      }
32
33      public double getSalary() {
34          return salary;
35      }
36
37      @Override
38      public String toString() {
39          return "Employee [name=" + name + ", age=" + age + ", salary=" + salary
40              + " ]";
41      }
42  }
```

2. Create the Company class(POJO).

```

12 public class Company {
13
14     private String name;
15     private String compnayType;
16     private Employee employee;
17
18     //injecting name,compnayType and employee in the constructor
19     public Company(String name, String compnayType, Employee employee) {
20
21         this.name = name;
22         this.compnayType = compnayType;
23         this.employee = employee;
24     }
25
26     public Employee getEmployee() {
27         return employee;
28     }
29
30     public String getName() {
31         return name;
32     }
33
34     public String getCompnayType() {
35         return compnayType;
36     }
37
38     @Override
39     public String toString() {
40         return "Company [name=" + name + ", compnayType=" + compnayType
41             + ", employee=" + employee + "]";
42     }
43 }

```

3. Create the Configuration file and add Company and Employee POJO's to the configuration file.

```

10 <beans xmlns="http://www.springframework.org/schema/beans"
11     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
12     xsi:schemaLocation="http://www.springframework.org/schema/beans
13     http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
14
15     <bean id="employee" class="com.java2learn.core.Employee">
16         <constructor-arg name="name" value="Rock"></constructor-arg>
17         <constructor-arg name="age">
18             <value>52</value>
19         </constructor-arg>
20         <constructor-arg name="salary" value="521352.4"></constructor-arg>
21
22     </bean>
23

```

```

25     <bean id="company" class="com.java2learn.core.Company">
26         <constructor-arg name="name" value="GOOGLE"></constructor-arg>
27         <constructor-arg name="compnayType">
28             <value>PRODUCT</value>
29         </constructor-arg>
30         <constructor-arg ref="employee"></constructor-arg>
31     </bean>
32
33 </beans>

```

4. Create the client and run it.

```

12 import org.springframework.context.ApplicationContext;
13 import org.springframework.context.support.FileSystemXmlApplicationContext;
14 import com.java2learn.core.Company;
15 public class ApplicationContextDemo {
16
17     public static void main(String[] args) {
18
19         //ApplicationContext applicationContext=new
20         ClassPathXmlApplicationContext("Spring-Core.xml");
21         //(or)
22         ApplicationContext applicationContext=new
23         FileSystemXmlApplicationContext("classpath:Spring-Core.xml");
24         Company company=(Company)applicationContext.getBean("company");
25         System.out.println("Company :: "+company);
26         System.out.println("Employee :: "+company.getEmployee());
27         System.out.println("Employee
28         Salary :: "+company.getEmployee().getSalary());
29     }
30 }

```

➤ Output:

```

Company ::Company [name=GOOGLE, compnayType=PRODUCT, employee=Employee
[name=Rock, age=52, salary=521352.4]]
Employee::Employee [name=Rock, age=52, salary=521352.4]
Employee Salary::521352.4

```