**SPRING**

**Fundamental**

* C, C++ and Java are Programming language.
* JDBC, Servlet, JSP, EJP, JPA, java mail, JTA and etc. are java Technology.
* Struts, Spring and Hibernate and etc. are java Framework.

**Programming Language**

* It is directly installed software having features to develop Software Application.
* It defines syntax (Rules) and semantics (Structure) of programming.
* These are uses to create software technologies, Framework, Tools, DB Software, Operating System and etc.
* These are raw materials of Software programming.

E.g.: C, C++, C# and java

**SPRING CORE MODULE**

* Base module for other modules.
* Gives BeanFactory containers as IOC container.
* When used alone we can develop standalone Apps.
* When used along with other modules and technologies.
* We can develop module layer services classes having logic

By taking advantages of Dependency Injection.

* **Dependency Injection**
* If the underlying Server/Container/Framework, runtime environment is dynamically assigning dependent value to resource (classes/Objects) then it is called **Dependency Injection**.
* The Spring Container designed for supporting five modes

Of dependency Injection.

1. Setter Injection

2. Constructor Injection

3. Aware Injection/Interface Injection

4. Lookup method Injection

5. Method Injection/Method Replacer

* **Setter Injection**
* If spring container uses setXxx() method to assign dependent objects/value to main objects then it is called **Setter Injection**.

**Sample Code**

public class WishGenerator{

private String name;

private Date date;

//setter method for Setter Injection

public void setName(String name){

this.name=name;

}

public void setDate(Date date){

this.date =date;

}

//Business Method

public String wishGeneratorMsg(){

int hour=date.getHour();

if (hour<=12)

return "Good Morning" +name;

else if(hour<=16)

return "Good Afternoon" +name;

else if (hour<=20)

return "Good Evening" +name;

else

return "Good Night" +name

}//method

}//class

* **applicationContext.xml**

<beans>

-------

-------//DTD or XSD statements

<bean id="dt" class="java.util.Date"/>

<bean id=""wg" class="pkg.WishGenerator"/>

<property name="name" value="raja"/>

<property name="date" value="dt"/>

</beans>

* When the above code is given by Spring Container,

It creates the Bean class Object using 0-param constructor

and inject "raja" ,"Date object" values to name ,date properties

of WishGenerator by calling setName("raja") and setDate(dt) method.

* **Constructor Injection**
* If Spring Container uses parameterized constructor for bean instantiation and initialization (assigning dependent values) then it is called Constructor Injection.

**WishGenerator.java**

public class WishGenerator{

private String name;

private Date date;

//2-parameter constructor

public WishGenerator(String name,Date date){

this.name=name;

this.date=date;

}

//Business logic

public String generatorWishMsg(){

---------

---------

}

}

**applicationContext.xml(Spring bean cfg file)**

<beans>

----------//DTD or XSD statement

----------

<bean id="dt" class="java.util.Date"/>

<bean id="wg" class="pkg.WishGenerator"/>

<constructor-args value="raja"/>

<constructor-args value="dt"/>

</beans>

* Since Constructor-arg it two times hence it call 2-param constructor.

**Note**: When the above code given to spring container it create Date class object using 0-param constructor but create Wish Generator class object using 2-args constructor having “raja” ,“dt” as the argument values.

* In this this process Dependency Injection also take place.