**3🡪SpringBoot,mysql,JavaBasic**

**4🡪debugging+Project Work and git and GitHub+**

**5🡪Coding full day**

**6🡪coding +revision**

**Spring Framework**

**Its open source java framework**

**We can develop standalone and enterprise application**

**Adventages🡪**

1. its lightweight Application
2. Flexible configuration
3. Dependency injection
4. Simplified database Access
5. Testing easy
6. Spring provide security module
7. Integration capability

**---------------------------------------------------------------------**

**Spring Container->**

1. its like heart of Spring
2. Its core component of spring
3. Responsibility-
4. manage Bean Object(delete create object)
5. Manage bean lifecycle
6. Dependency Injection
7. AOP achieved
8. Integration

**-----------------------------------------------------------------------------------**

**Types of containers🡪**

* **Bean Factory(old)**
* **Application factory(new interface)**

**Spring Container working🡪**

**Java Bean class🡪**This is Special type of POJO

Its serializable (can be saved and restored) class, has a no-argument constructor, and follows certain naming rules for methods (like get/set). It's often used in frameworks because it can be easily integrated.

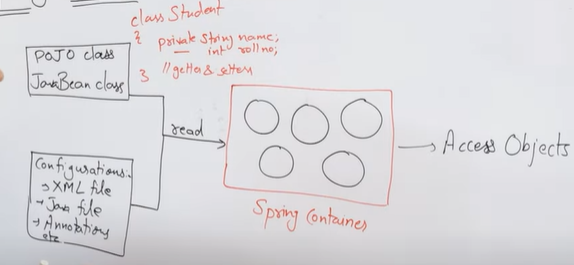
POJO class(Plain old java Object )🡪simple java class with filed and getter/setters used for data representation without framework dependencies

**Class Student{Private String name;**

**//getter and setter**

**}**

**Then configuration provide 🡪instruction provide to Spring Container like what will be id of object, object name**

**We can configuration through XML file, java File, Annotation**

**For Spring Container we need JAR files**

A close-up of a document

Description automatically generated

---------------------------------------------------------------------------------------

**Using XML file🡪**

Public class Student{

Private String name;

}//getter setter method Auto Generate

1. Then we should configure it ->
2. Create new file applicationcontext.xml //Spring config file
3. A screen shot of a computer

   Description automatically generatedAdd external schema then edit it

When we create this bean class means we instructed to Spring Container to do action

A white background with black text

Description automatically generatedA screenshot of a computer code

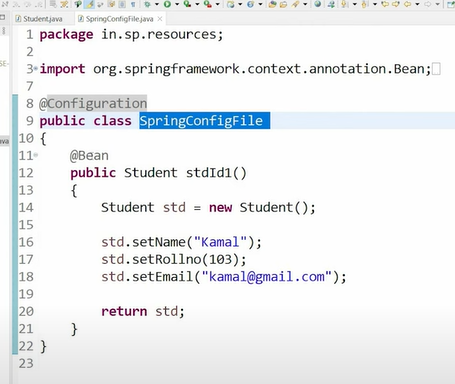
Description automatically generatedWe can add value using Property tag like name age and all

-------------------------------------------------

**We should need to jar files**

A screen shot of a computer code

Description automatically generated--------------------------------------------------------------

**A computer screen shot of a program

Description automatically generatedSetup Using Java File** 🡪

---------------------------------------------

**A screenshot of a computer code

Description automatically generatedAnnotation based Configuration🡪**

In annotation we should create on file eighter xml or java compulsory

I created XML file and necessary schema added

Now in last time we create object in bean but In this condtion we create object in class Student

Using @component Annotation

A computer screen shot of a program

Description automatically generatedA screenshot of a computer program

Description automatically generatedStep1🡨------- 🡪step2

A computer screen shot of a program

Description automatically generated

* **@Value**

@Value("${property.name}")

private String propertyName;

Value is used to injects the values

* **@Component**

public class MyService { }

Marks a class as a Spring bean(its manged by Spring IOC container)

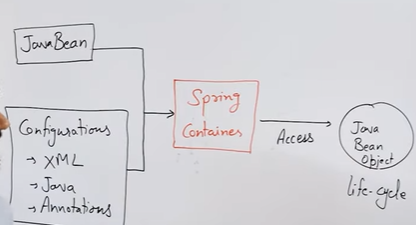
* **@ComponentScan**

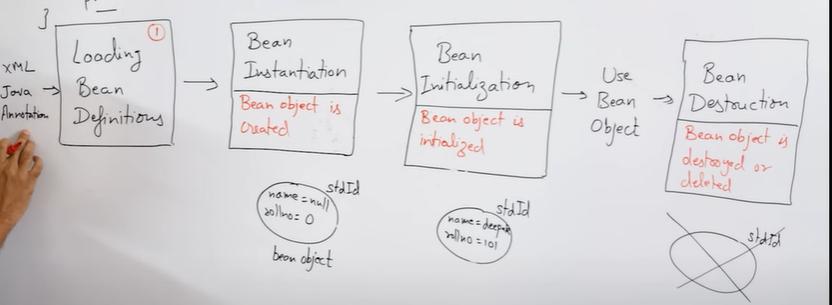
@ComponentScan("com.example")

Tells Spring to scan the specified package(s) for components (like @Component, @Service, @Repository).

**Java Bean 🡪**

its java class that follow following conventions

* 1.it should have Private Properties
* it should provide getter and setter method
* it should have public no-arg Constructor or default constructor
* When we create java bean object then its called java been object

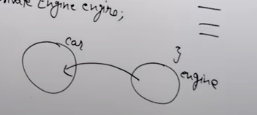
**Java Bean Object Life Cycle in 4 phase**

1. Loading Bean Definition->from xml ,java, annotation
2. Bean Instantiation🡪bean object is created

Bean Initialization🡪bean object initialize

1. Use bean Object🡪using Context.getBeans
2. Bean Destruction🡪distroy object destroy()

But we can again use using bean past processor

**Dependency Injection🡪**

* It is an design pattern
* Its main task is to inject dependency mean inject one object into another Object
* A text on a white background

  Description automatically generatedA close-up of a car

  Description automatically generatedEg.

**Why we need🡪**

* **If you not used dependency injection it will be hardcode**
* **Hardcoding means if we putting directly values in code rather than using Variables or external configuration**

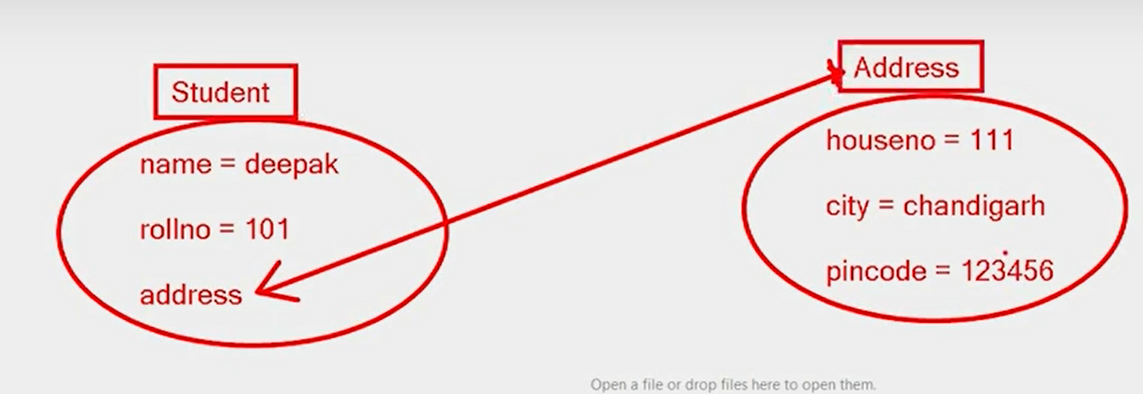
**A computer screen with white text and red letters

Description automatically generated**

* **It is used to achieve loose coupling in java**
* **If we create object of class engine using new keyword it will be one class depend on other so Inject using xml, java,**

**We can achieve dependency injection two ways->**

1. **Setter method DI**
2. **Constructor DI**

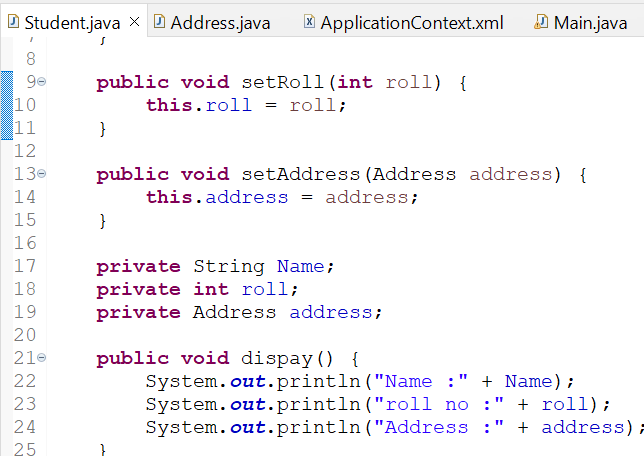
****

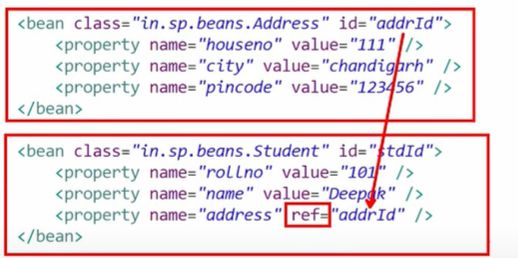
**EG. Program**

A computer code with text

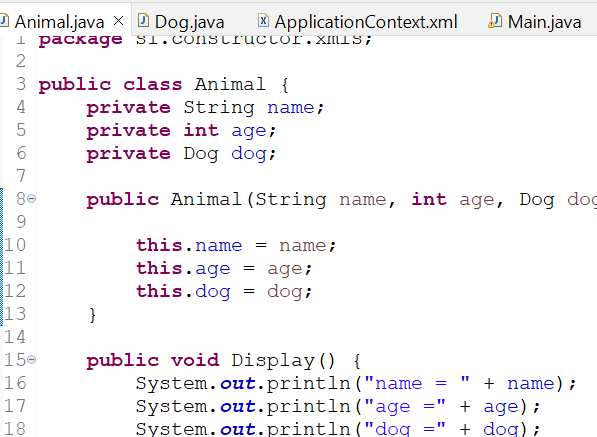
Description automatically generatedA computer code on a white background

Description automatically generatedA screenshot of a computer program

Description automatically generated

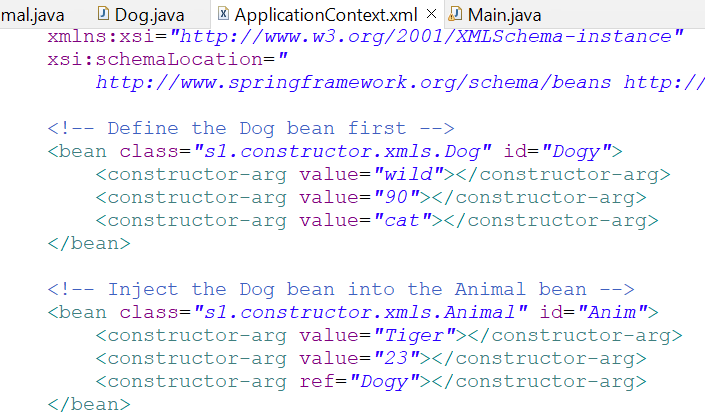


Using Constructor also we can acchive

**A screenshot of a computer code

Description automatically generatedConstructor dependencies injection🡪**

**A screen shot of a computer

Description automatically generated**

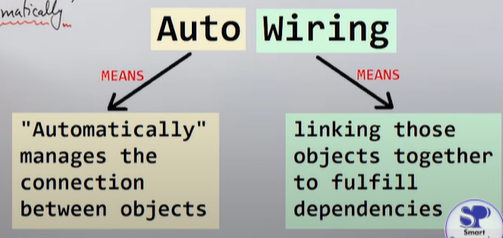
**Its very easy to do but difference is between constructor and setter is setter code is understandable and constructor code is not we only put value there but in setter we can put value also**

**A screenshot of a computer program

Description automatically generatedSpring Configuration using java🡪**

**A screenshot of a computer program

Description automatically generated**

**Auto wiring 🡪**

What is Auto wiring and what is advantages and disadvantages and how to achive?

🡪 its feature of Spring framework we can achieve DI automatically

🡪we can achieve by

1. Annotation based 🡪@Autowired
2. Xml based 🡪@Autowired Attribute

A screenshot of a computer

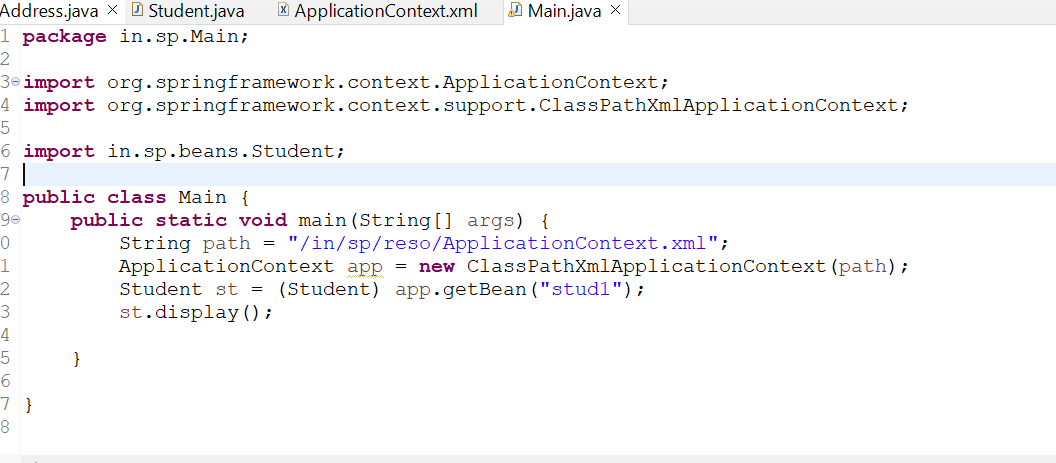
Description automatically generatedA screen shot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generatedXML Based Autowired🡪

**Maven 🡪**

It is a build tool which Automates everything related to building of java Projects (automates compile, deploy, clean,t ext so on…)

**Responsibility🡪**

1. Create Project Structure Automatic
2. Download required dependencies
3. Prepare documentation
4. Compiles the source code
5. Starts or stops server
6. Packaging project in jar,war,EAR file

**Working of Maven🡪**

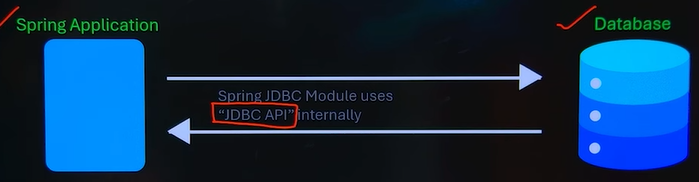
A diagram of a software development

Description automatically generated

1. create bean class
2. create configFile add xml schema
3. Pom.xml add Spring Context dependcy
4. Then Create Main and call

**What is Spring-jdbc Module || what classes provide by spring-jdbc module**

**🡪 we connect Spring Application with database using JDBC and for this Spring provides one module i.e. “Spring JDBC module”**

****

**Jdbc Module Provide 3 classes for database connection and operation**

1. **Driver manager data source**
2. **Jdbc templet**
3. **Named parameter JDBC template**

**----------------------------------------------------------**

* **Driver manager data source**

It is implemented class of “Data Source” interface (data source is present in “javax.sql” package )

**It is used for🡪**

Database configuration

Driver loading

Connection creation

---------------------------------------------------

* **“Jdbc Template” class**

1. It is central class in Spring-jdbc module
2. It is used for database operation(Crud Operation)

**Methods**

1. Update(insert,update,delete Sql quires)
2. Query(queryForList(),queryForMap(),ForObject(),RowSet)

* **queryForList() // if we retrieve data from database and result will be store in list**
* **queryForMap() // result will store map**
* **queryForObject()// result in bean object**
* **queryForRowSet() //result in set**

**Step 1->dependency pom.xml**

**dependency🡪SpringContext,Spring jdbc ,mysql connector**

**Step 2🡪configuration file with java**

* **A screenshot of a computer program

  Description automatically generated** You see 🡪drivermanagerdatasouce is main class and mydata() is method,it contains
* Object create
* Class load using setDriverclass name
* url set
* username
* password
* jdbcTemplate also main class contains different operations

A screenshot of a computer program

Description automatically generated

here we used jdbc template class because class contains lots of operations

**Named parameter JDBC template🡪**

**(?,?,?) in this positional paramemter we can add named paramerter for better understanding**

