**Spring AOP-**

What is Spring AOP?

AOP stands Aspect Oriented Programming.

Java is OOP. We code using classes and object, we can create object of one class in another class and use all functionalities.

Cross cutting concerns -> AOP.

Technical aspect 🡪

Traditional Approach

Authenticate

Public void Authenticate()

Class Shopping{

obj

Public void buyProduct(){

}

}

Using Spring AOP

Class Shopping{

(join point before method execution)

Public void buyProduct(){

}

(join point after method execution)

}

@Aspect

Authenticate

@After()

@Before(“”)

authenticate(){

}

successMessage(){

}

Imp Terminologies of Spring AOP 🡪

1. **Aspect**- It helps in implementing modularization of cross cutting concern that cuts across multiple classes. This Aspect is a nothing but a class only.
2. **Join Point**- It is only point in our program such as **method execution**, exception handling, field access etc.

**Spring supports method execution join points**

1. **Advice –** Advice represent an action taken by an Aspect at particular join point (set statements and in simple words we can say it’s just a method)

Advices are also of multiple types

1. After
2. Before
3. After returning
4. After throwing
5. **Pointcut –** It is just considered as expression to match with join points. It is expression pattern which use to check specific join point where we will have to implement advice.

Class Payment{

@Aspect

Class Login{

(advice)

Public void login(){

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

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

}

}

Join Point

Public void buyProduct(){

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

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

}

}

**Spring MVC 🡪**

* **Windows application**
* **Web application🡪** 
  + **Static web pages application (HTML)**
  + **Dynamic web application**

**Spring MVC is used to develop dynamic web applications. Spring MVC is derived from Spring therefore all core functionality of Spring core will be inherited so development will become very easy.**

**Web Application we require server.**

**What is server?**

**Request(AWS arc)**

Logic

Server

Client

**response**

Data

Jsp -Java server pages

Servelet – is also a java class.

Spring MVC internally going to implement jsp servlet approach.

**MVC - > design pattern**

Model -> POJO which is going to get data from database

View -> presentation layer

Controller -> servlet – java class

Jsp Servlet Approach

Controller 1

Spring MVC approach

Controller 2

(data and , view2 info)

View Resolver

FrontController

(Servlet)

View2

View1

View3

Controller 3

Controller 1