

## Lecture 12 : Observer Design Pattern

Page No.	
Date	

### # Introduction

- Let's take scenario ki jab bhi  $\rightarrow$  koi youtuber new video upload krta hai toh uske subscribers ko notification jaati hai. Toh humara account and youtuber ka account aapna mein kaise interact karte.

- Yeh ek specific type ki problem solve karta hai; and problem is - do object hai jismein se jab bhi ek object change honga toh dusre object janna chalte hai ki woh kb change hua aur kya value thi

### # Observer Design Pattern

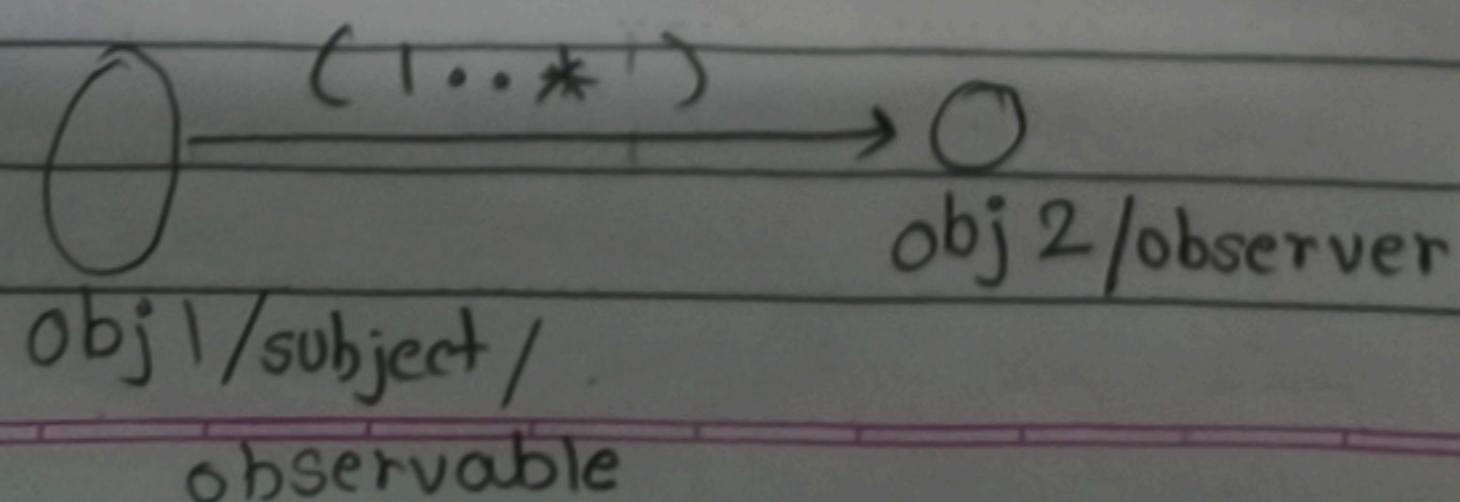
- Humare pass bahot saare objects hai jo janna chalte hai ki obj1 ki internal state kab change hoti hai.

- Real life ex: YouTube ; jb bhi hum kisi channel ko subscribe karte toh humme notification aata hai new video uploaded

Technique to implement -

#### 1] Polling (Big NO)

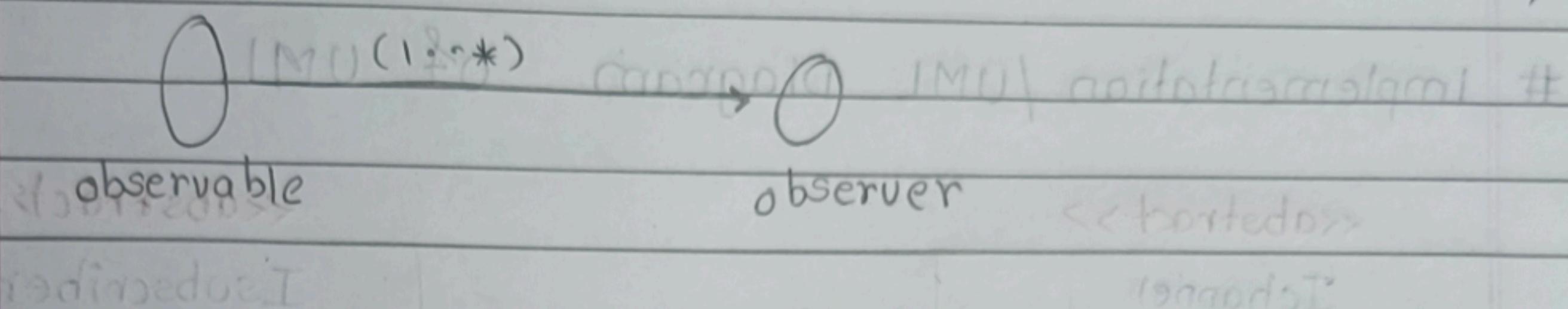
Yaha par observer job hai baar baar (some time which is fixed) observable se puchega has your value change and we need to determine frequency / time interval



+ observable &  
many observers

## 2] Pushing - We use this

Yaha par observable ka kaam hota hai ki observer ko batao jab jab <sup>uska</sup> mera state change ho raha hai

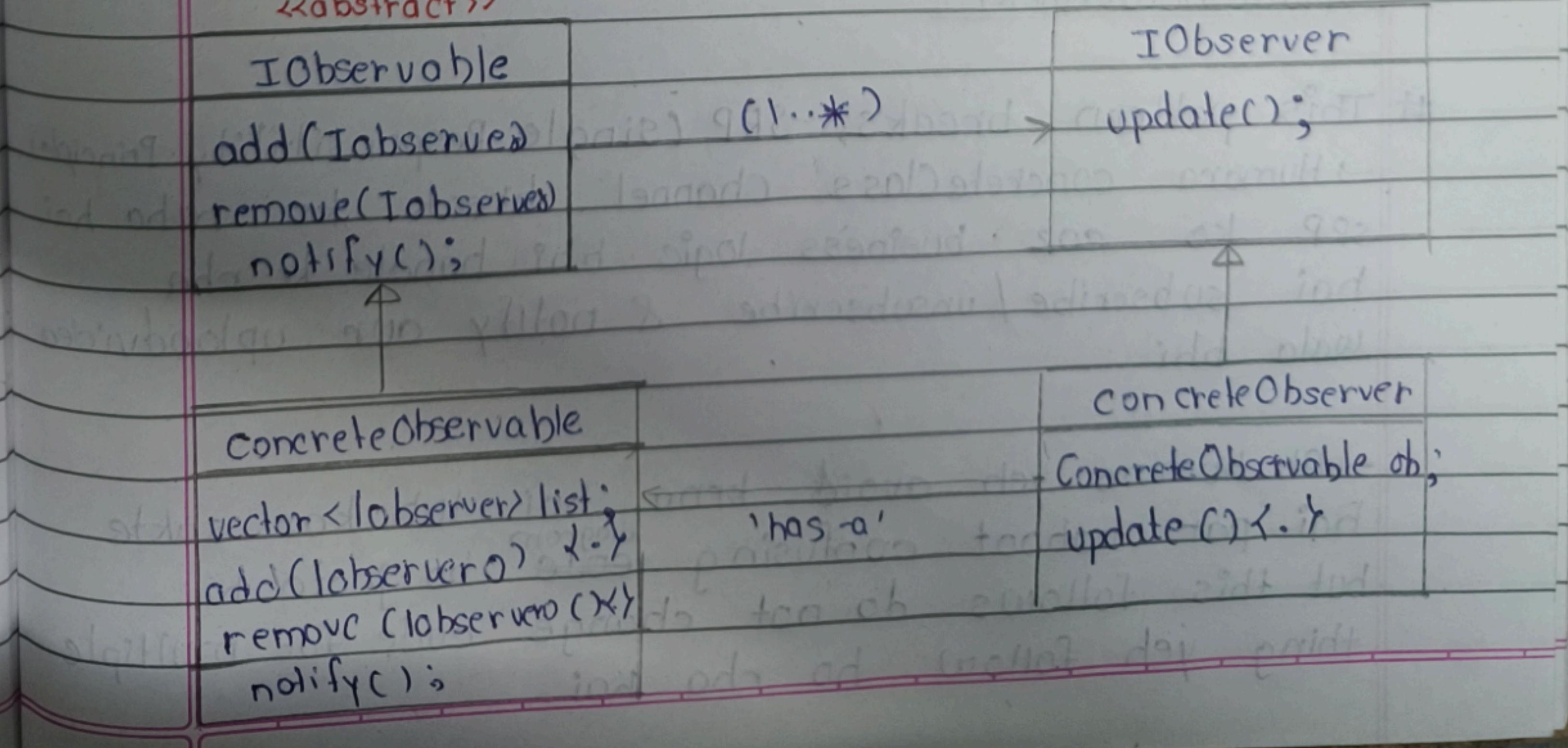


- Ab observable ek list maintain karega jo ha par saare observer store karega And jab bhi observable ki state change hoti hai toh woh list mein se sabko notify kar deta hai ki meri value change ho gayi hai

## Naming Convention

<<abstract>> aise upar likhne ki jagah we will write I + class name in UML. "I" denotes interface i.e. abstract class in CPP

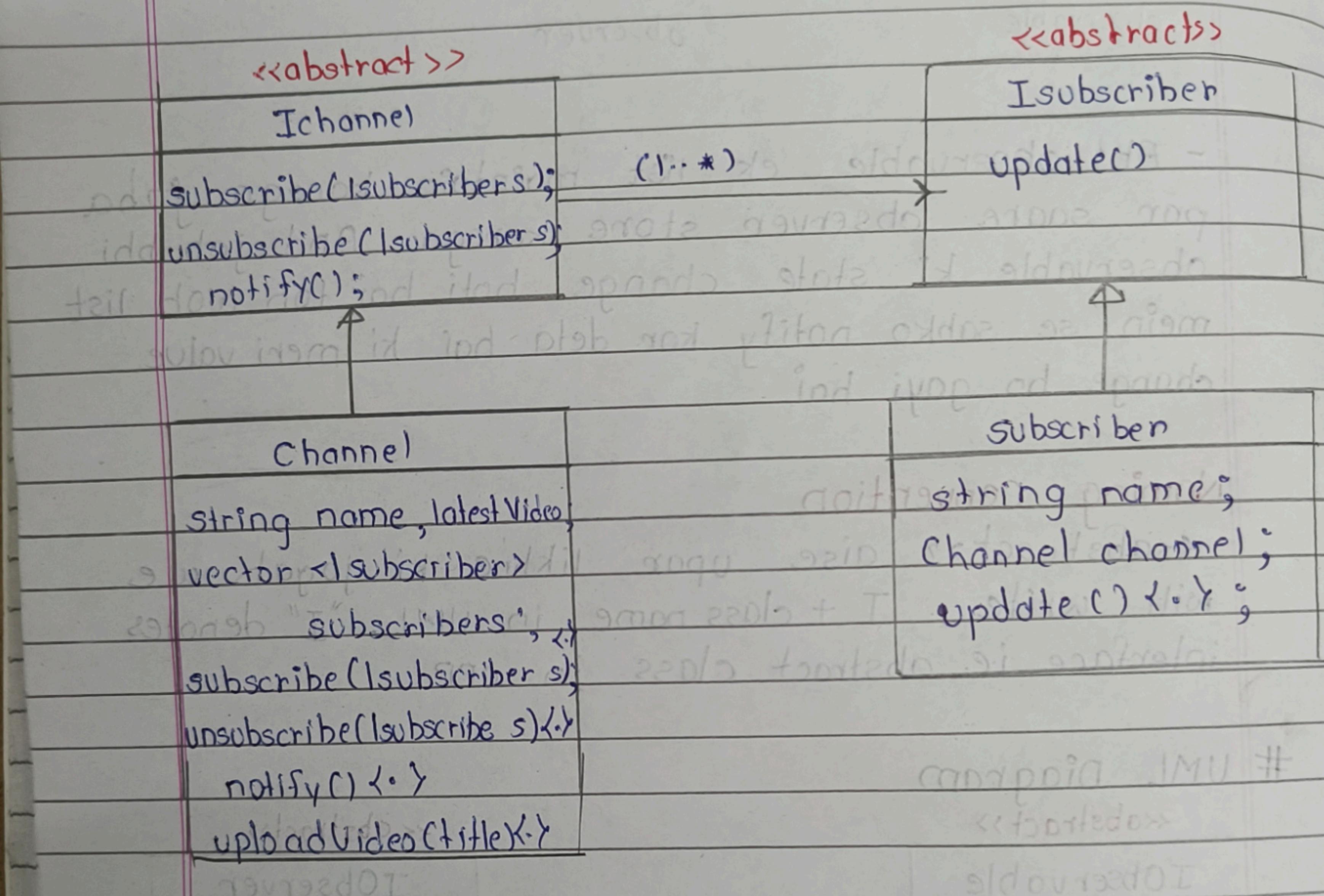
## # UML Diagram



## # Defination

↳ Defines a one-to-many relationship btw objects so that when one object changes state, all of its dependents are notified, and updated automatically.

## # Implementation / UML diagram of UML



# This pattern breaks SRP (Single Responsibility Principle)  
→ Humara concrete class Channel break kr raha hai  
SRP ko coz business logic bhi handle kr rha  
hai subscribe / unsubscribe & notify aur uploadvideo  
wala bhi

→ Ab humme yeh avoid krna hais toh kar sakte  
hai par bahot confusing code ho jaata hai  
but this follows do not change code for multiple  
thing yeh follow ho rha hai.

→ So for this pattern it is always a trade off whether to follow SRP or keep it less complicated.

## # Real-Life Example

- 1) Notification Service
- 2) Event Handling