OOPs Concept - Part II & Doubt Clearing Session

Course on OOPs Basics

Love Babbar • Lesson 2 • May 21, 2022

→ OOPS :-

↑

↳ Access modifiers

zoom session → OOPS

↳ class

↳ Obj

↳ Pillars

↳ Poly

↳ In

↳ type

↳ Anom

↓

diam

↓

4_5

→ ↳ pi

↳ Iqms

→ Access modifiers :-

- private
- default
- protected
- public

what - ?

- scope
  - field / method
  - class

→ private :- [only within class]

Yes or No

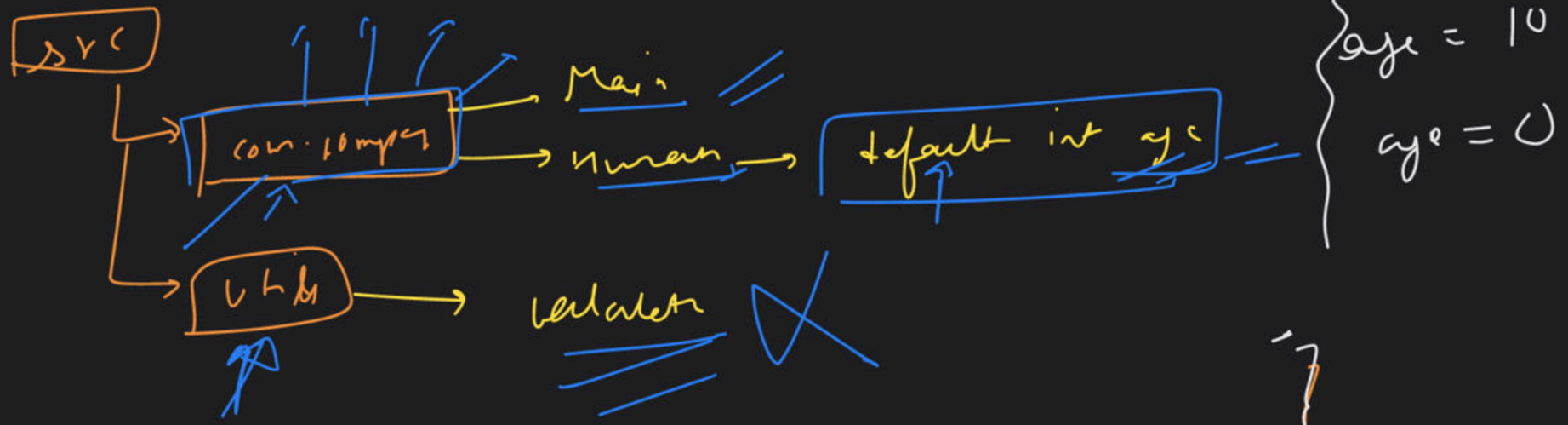
```
class
{
private int
age;

work()
{ love = this.age;
}
```

→ Default :- → access modifier

[access is possible only within the same package

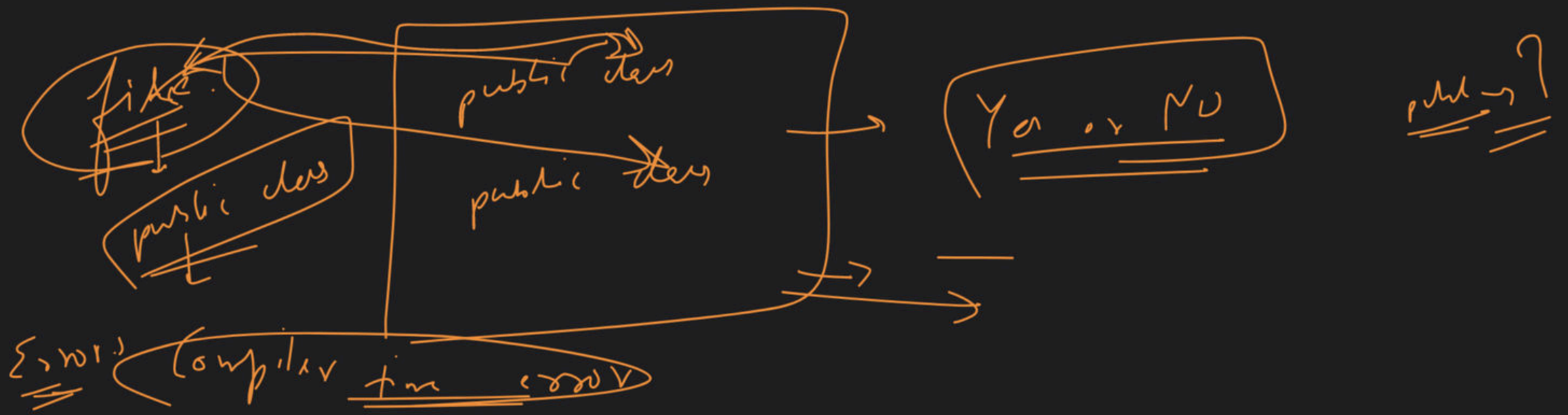
class A
{
private int age;
}

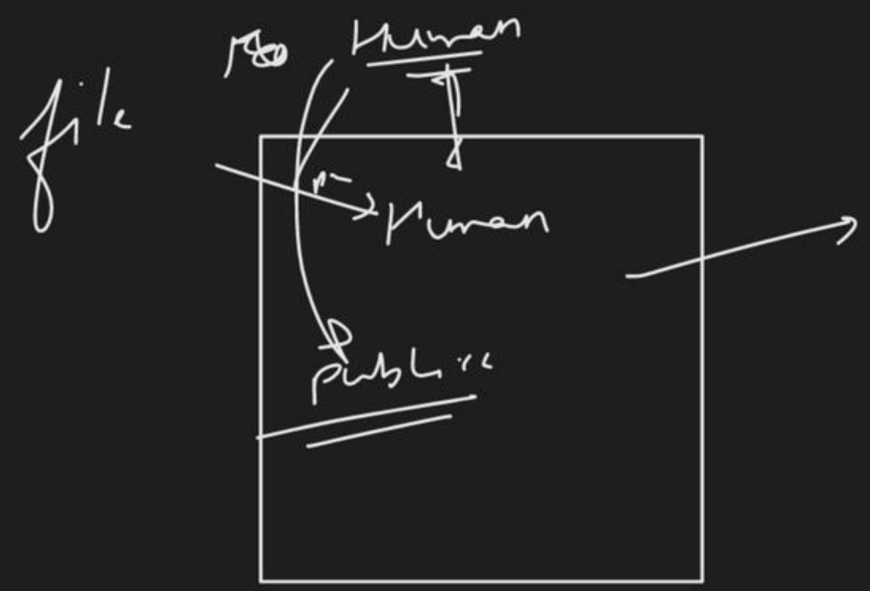


→ Public :- access level is within / Outside the package

→ Protected :- access level is within / Outside the package

cond → only through child class





file -> 1 Public class

protected

Human

female

Human ← Bivkoop

accept

female obj

obj.weight

parent

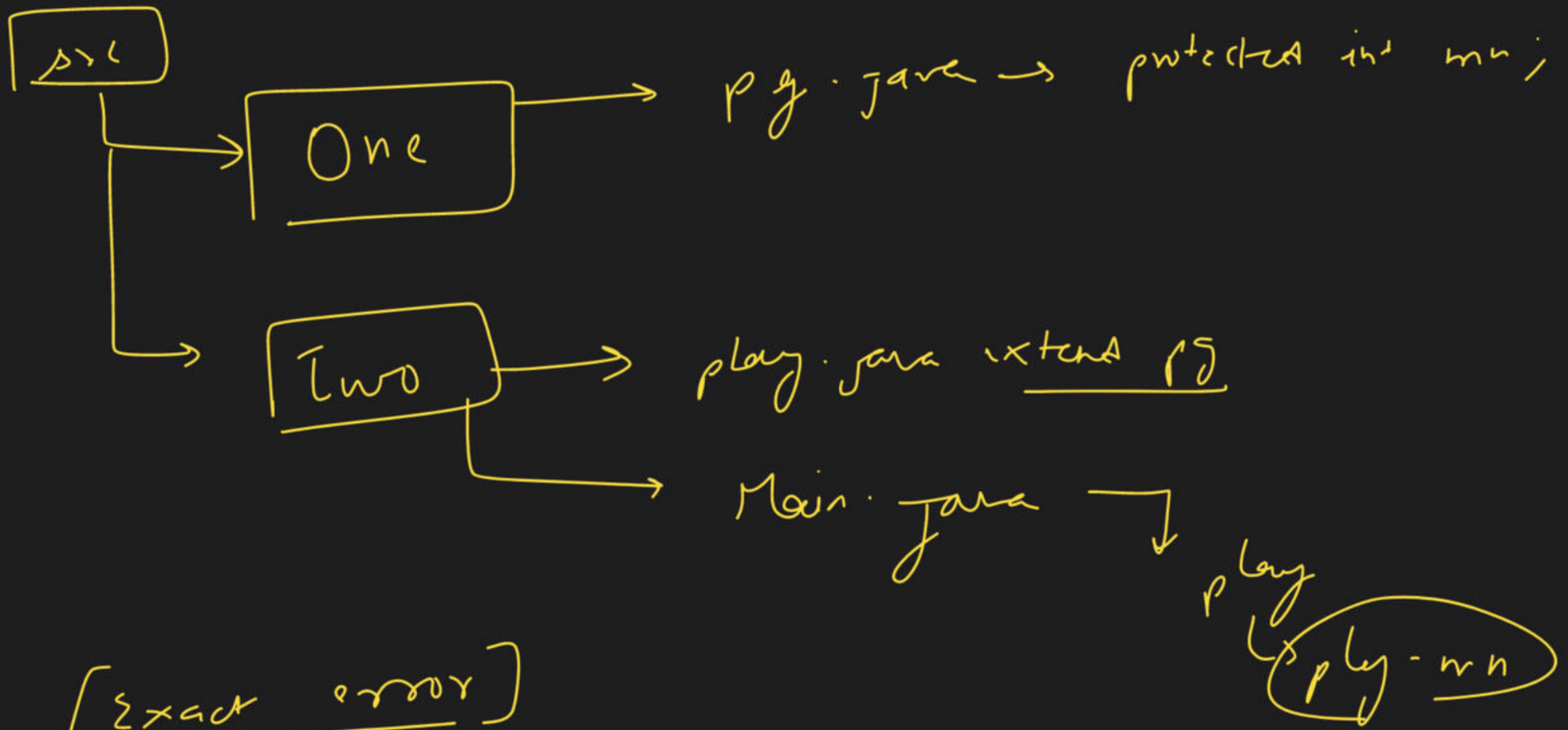
protected A;

Child obj = new Child();

obj = A;

Access modifier	within class	within package	outside package by child class only	outside package
Private	Yes	No	No	No
Default	Yes	Yes	No	No
Protected	Yes	Yes	Yes	No
Public	Yes	Yes	Yes	Yes

16 cases



[Exact error]

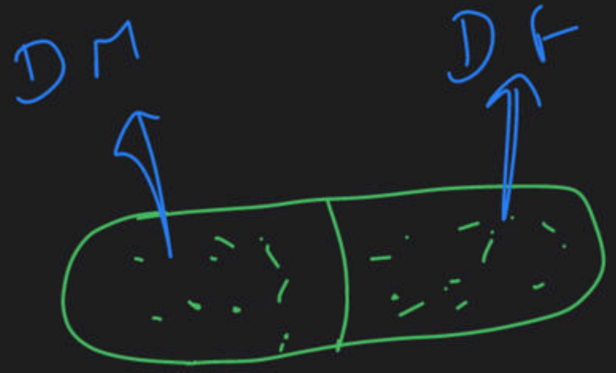
Piyush → Discord

→ AM → Access Modifiers

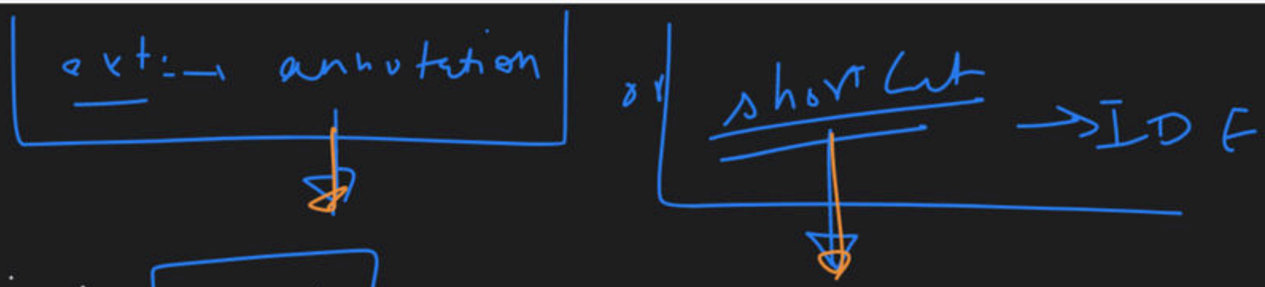
→ Encapsulation :-

?

[wrapping up of data member & function in a single unit]



→ fully encapsulated class → [all DM private]

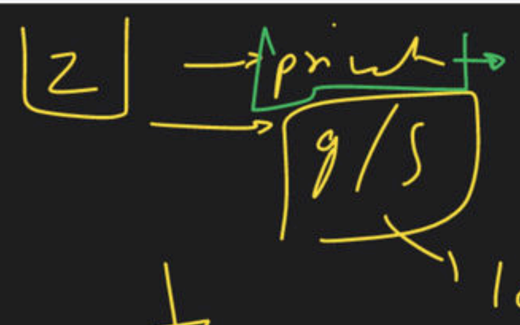



```
class Human
{
    private int age;
    private int wt;

    // getter & setter
    public int getAge()
    {
        return age;
    }
    public void setAge(int v)
    {
        this.age = v;
    }
}
```

Encapsulation

Adv:- 1.

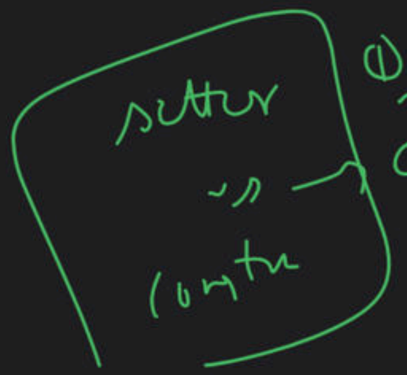


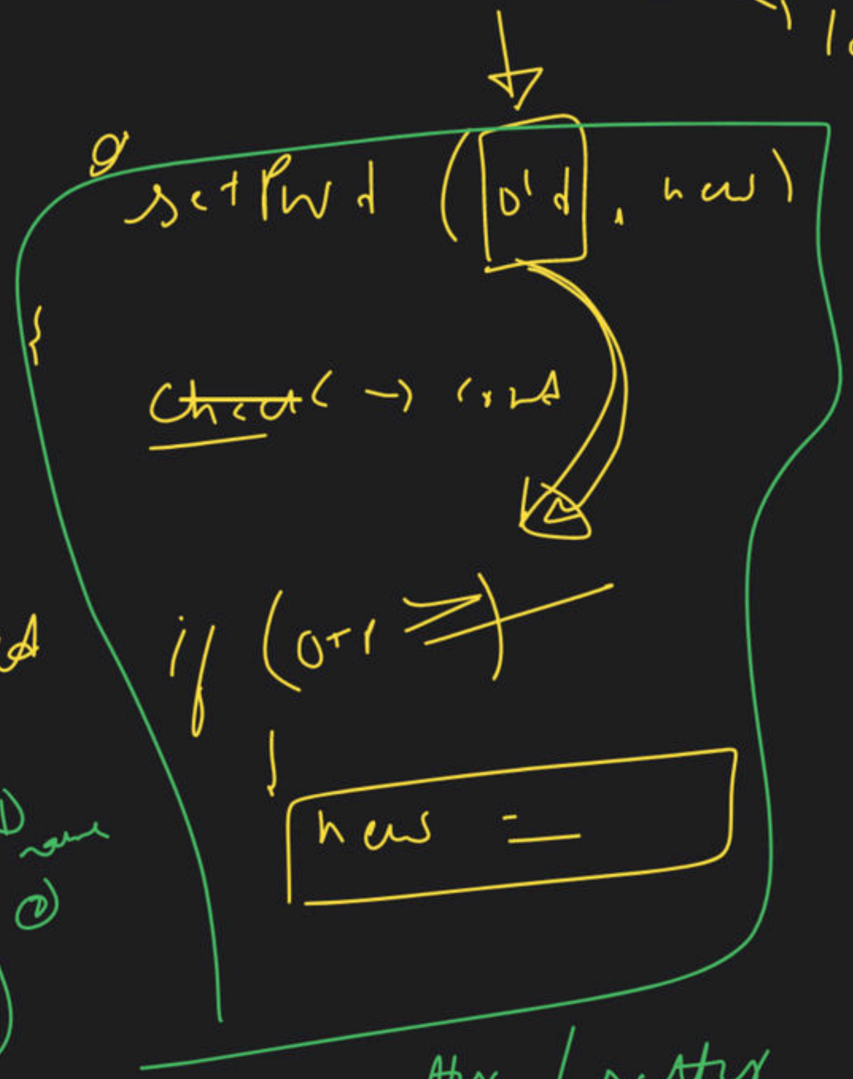
↳ used for data hiding / Security

↳ read-only class → Ex → Real world

↳ write-only class

↳ easy & fast





public int age = 0

this.age = 10

getter ✗

setter

→ chg → cond^n

if (____)

Rajiv . public verify & set

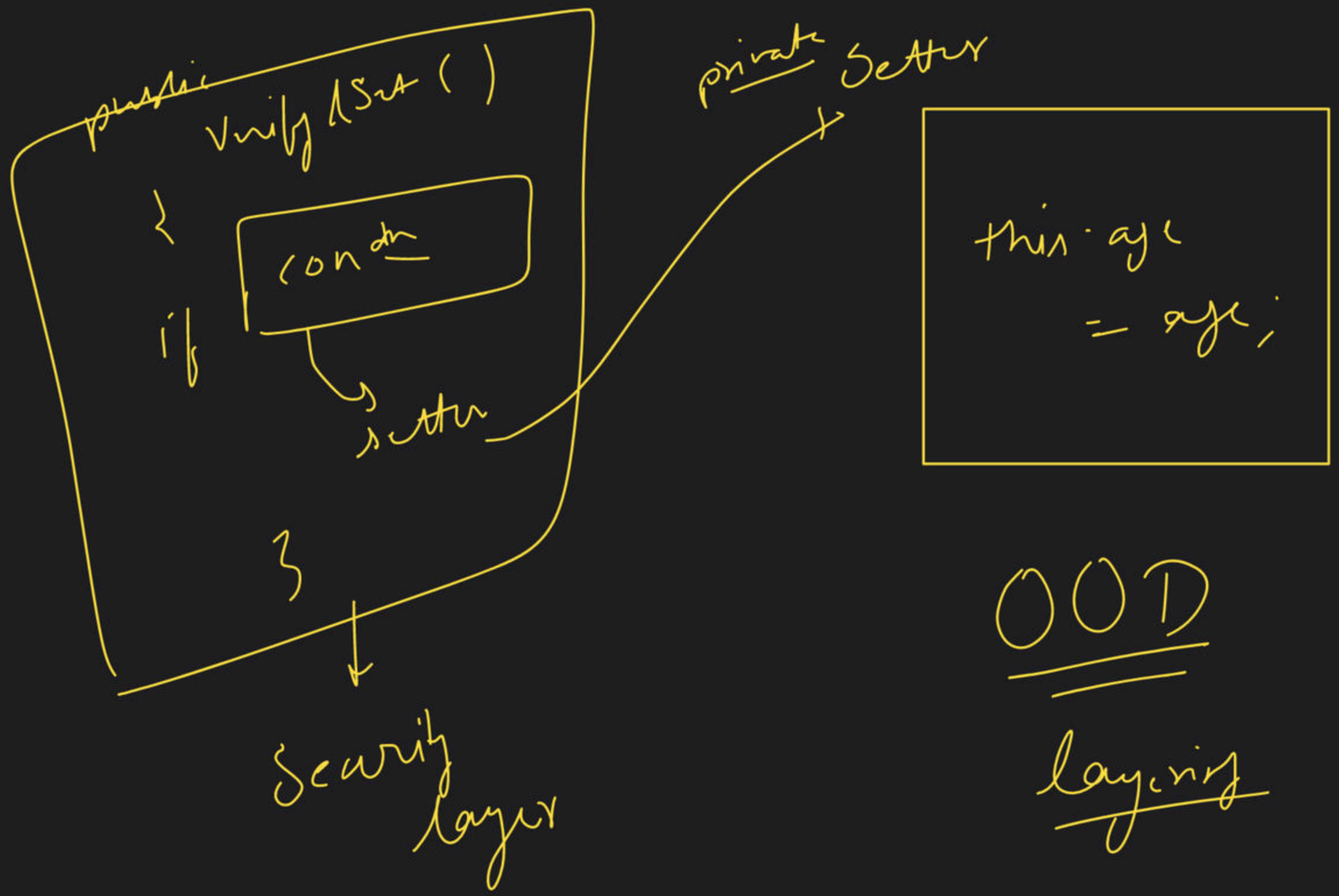
setM ( )

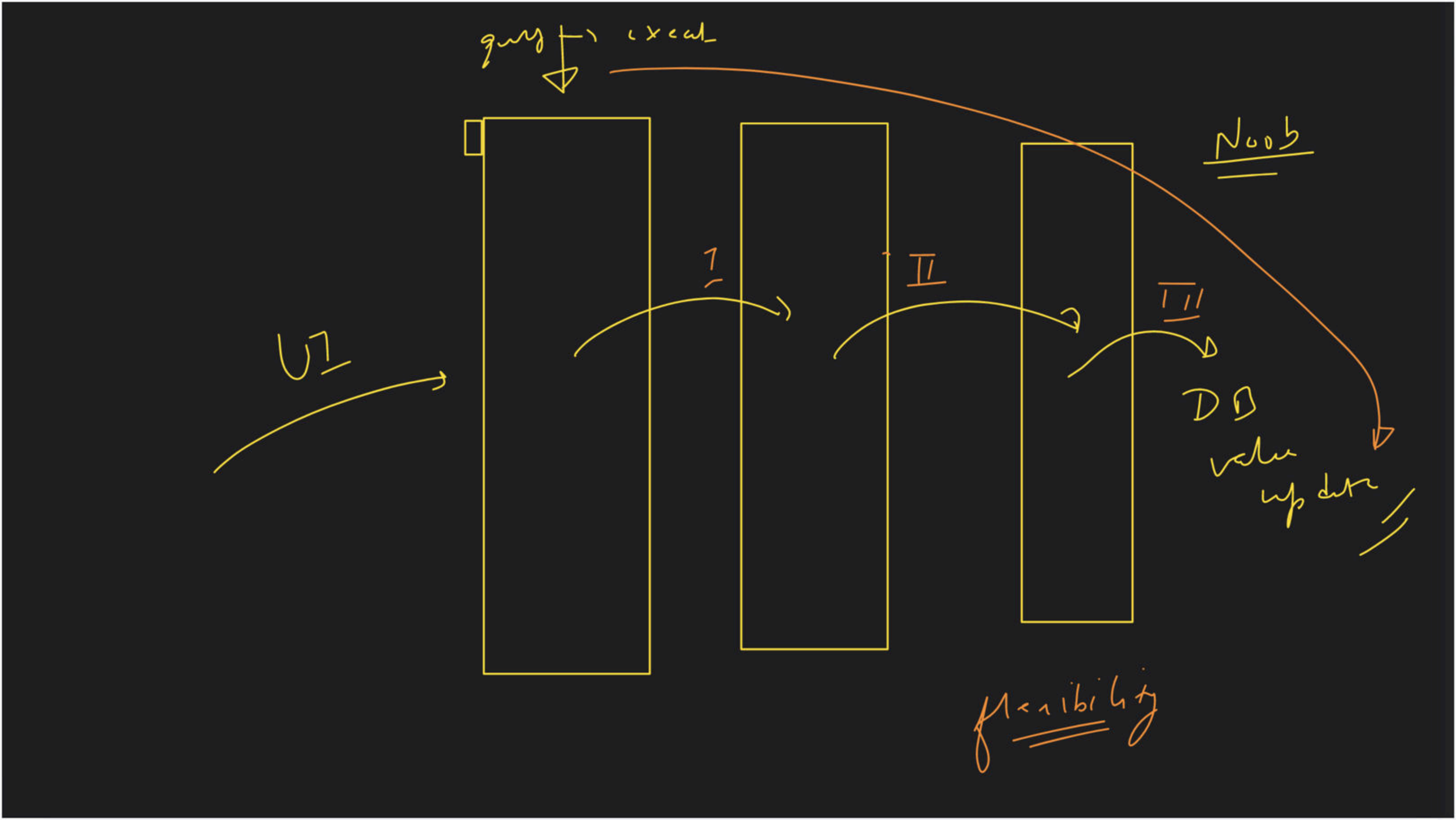
print setAge( )

{ this.age = age;
}

Java

2nd


public
verifyUser()
{
if
condn
setter
}
private
Setter
this.age
= age;
Security
layer
OOD
layering


query → excel
Noob
UI
I
II
III
DB
value
update
flexibility

→ what is ?

→ Adv :- ?

→ 2 Rules → Encapsulation

→ DM - private

→ getter / Setter

OOPs

- P →
- A →
- E →

→ Data Hiding :-

→ Abstraction: what -> ?

- essential details → hi show krenge
- non-essential hide

Ex :- Car

Abstraction can be achieved

→ abstract class

→ Interfaces

Imp points :-

① abstract class Shape { }

public abstract class Shape { }

② abstract method → declare w/o implementation

③ abstract → may or may not have all abstract method

some concrete method

→ declare & define

(i) no object of abstract class -> Why ? explain

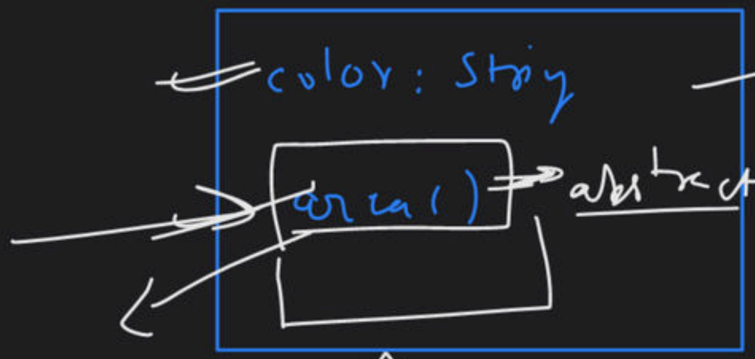
(j) abstract class -> q -> Y

(k) abstract class -> P

-> D

Area

Shape



Blue Print

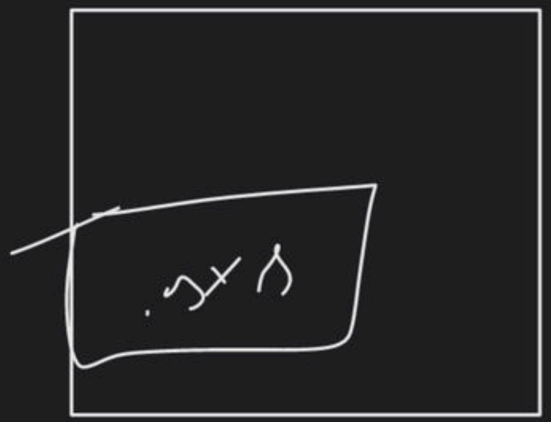
Square → S x S

Rect → l x b

l = b

Ao?

Square



~~Polymorphism~~

class Square extends Shape

{

@Override

area()

{ s x s;

}

}

@Override -> mandatory

NJ

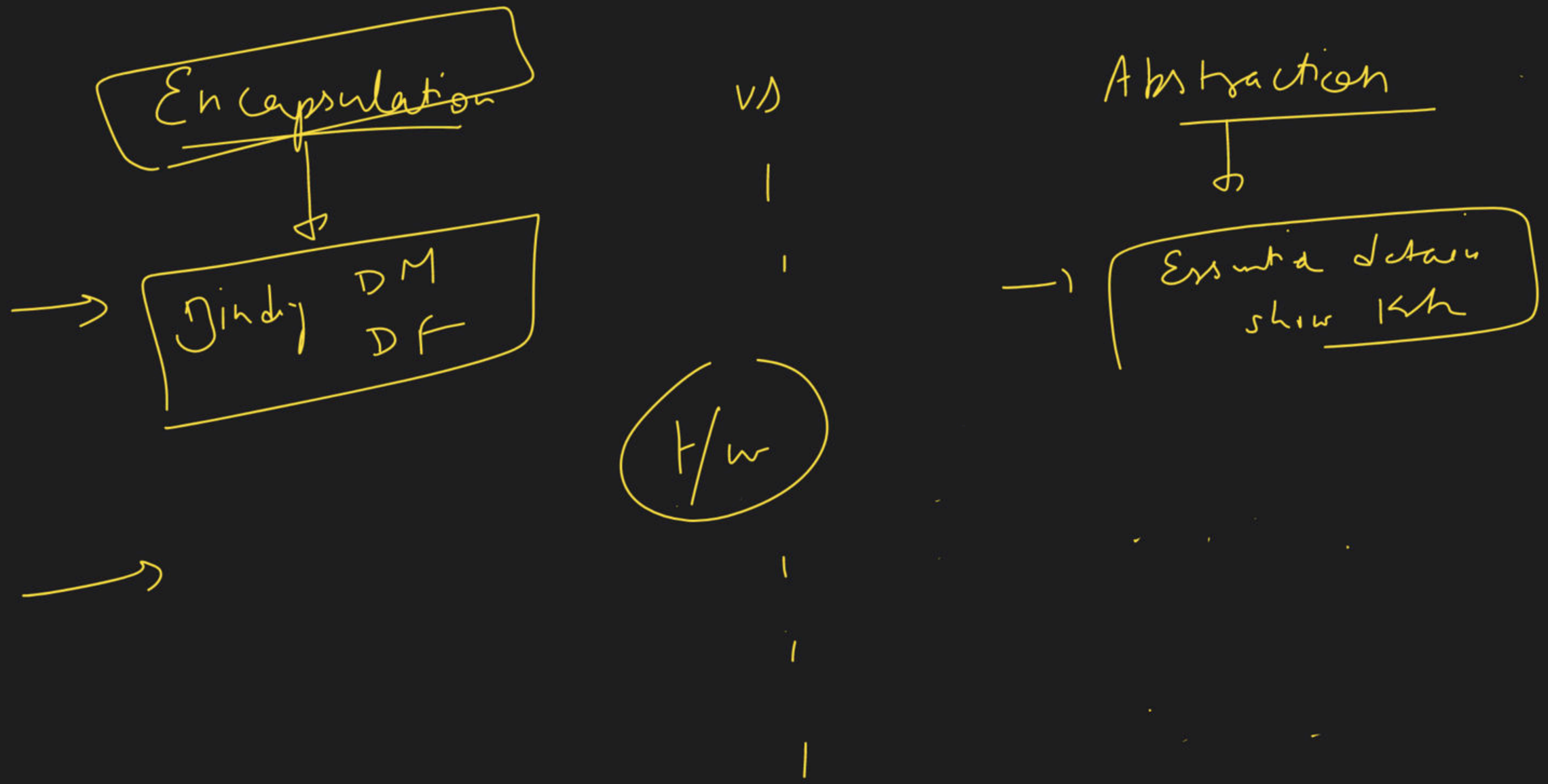
??:/

2 min Break:

Best Practice

↳ Readability

↳ force / signature → Error

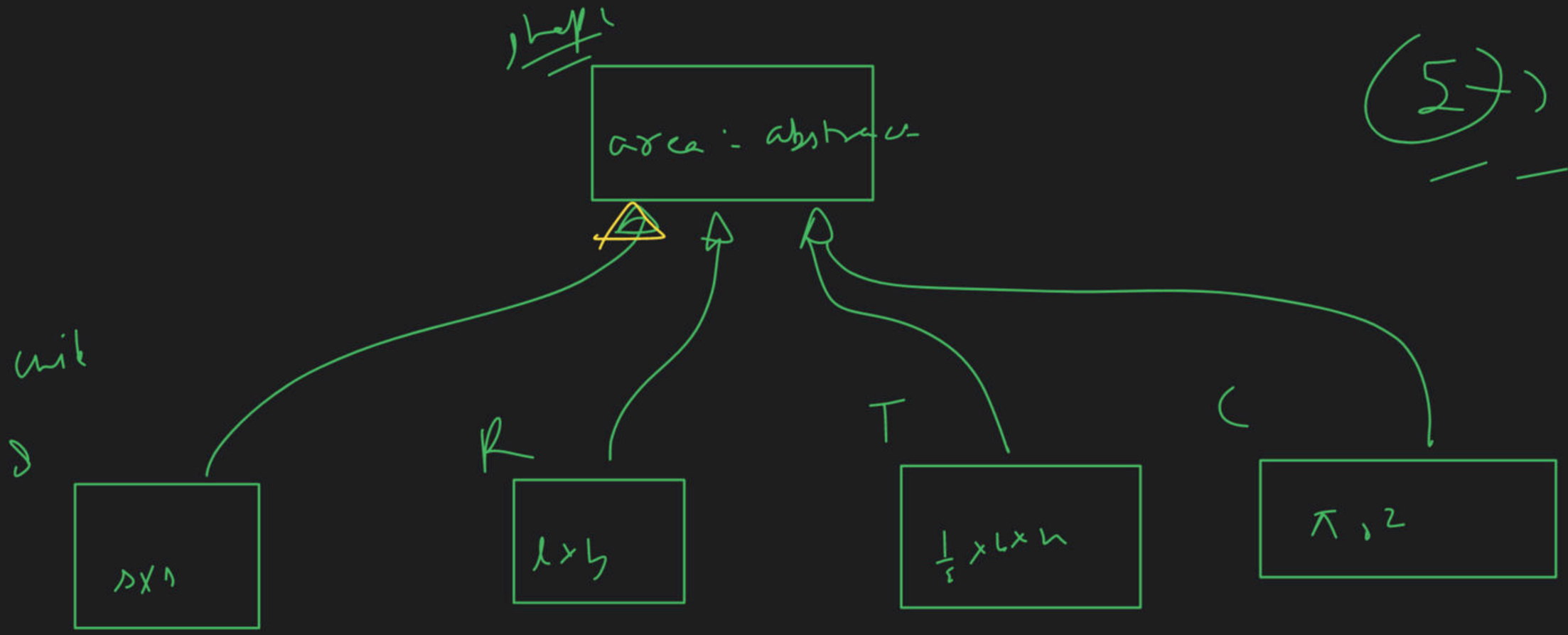

Encapsulation
VS
Abstraction
Jindy
DM
DF
H/w
Essential details
show Kro

Ans → adv:-

→ ① It reduces complexity of viewing things

Easy to Understand

Re-usability


area : abstract
unit
R
T
C
πr2

→ Interfaces :-

100% / Complete / Total / Pure Abstraction

all method

↓

w/o implementation

↓

Keyword → implements

(ch)

H/W

public Interface Area
{

public void Square();

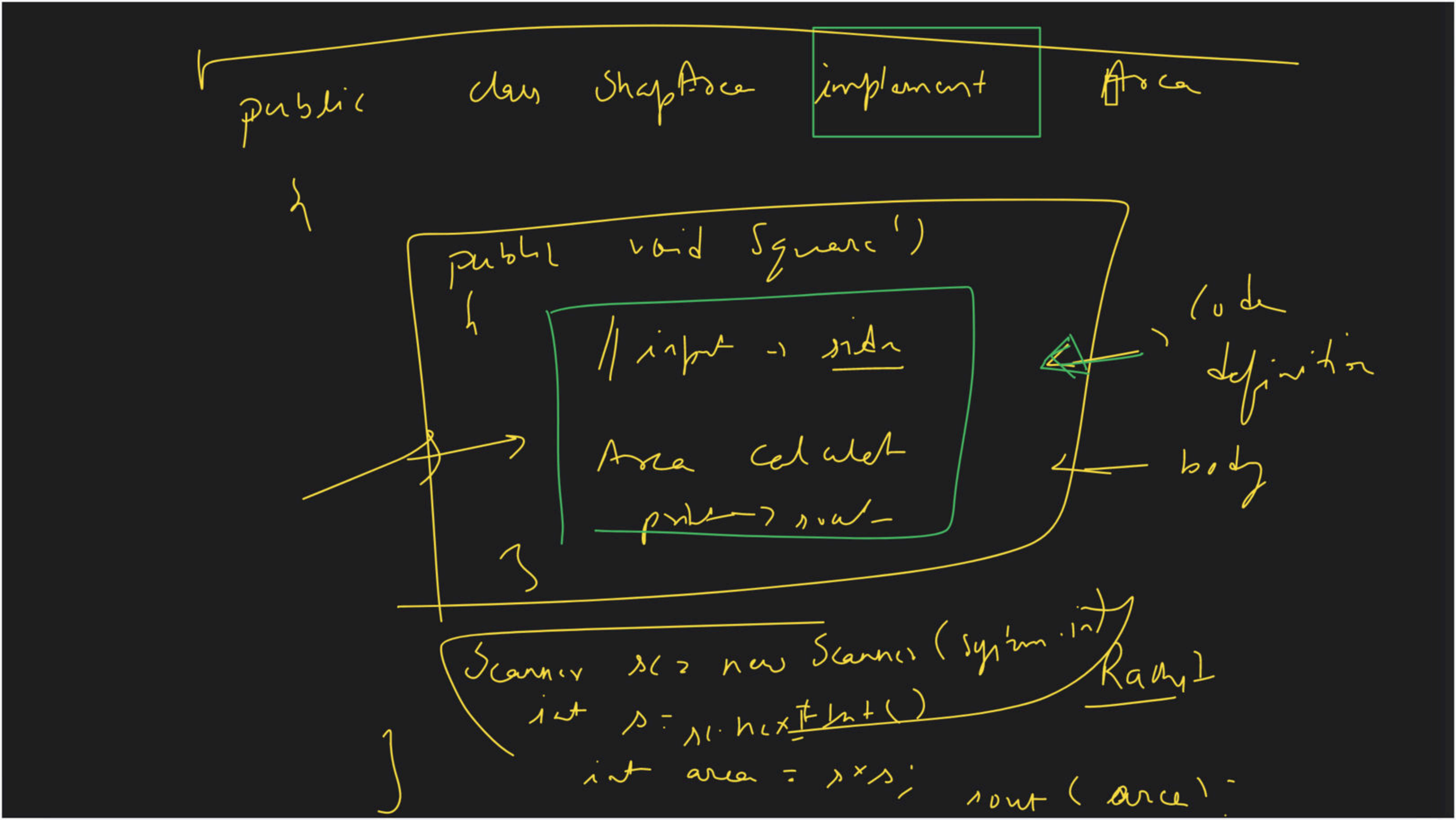
Circle();

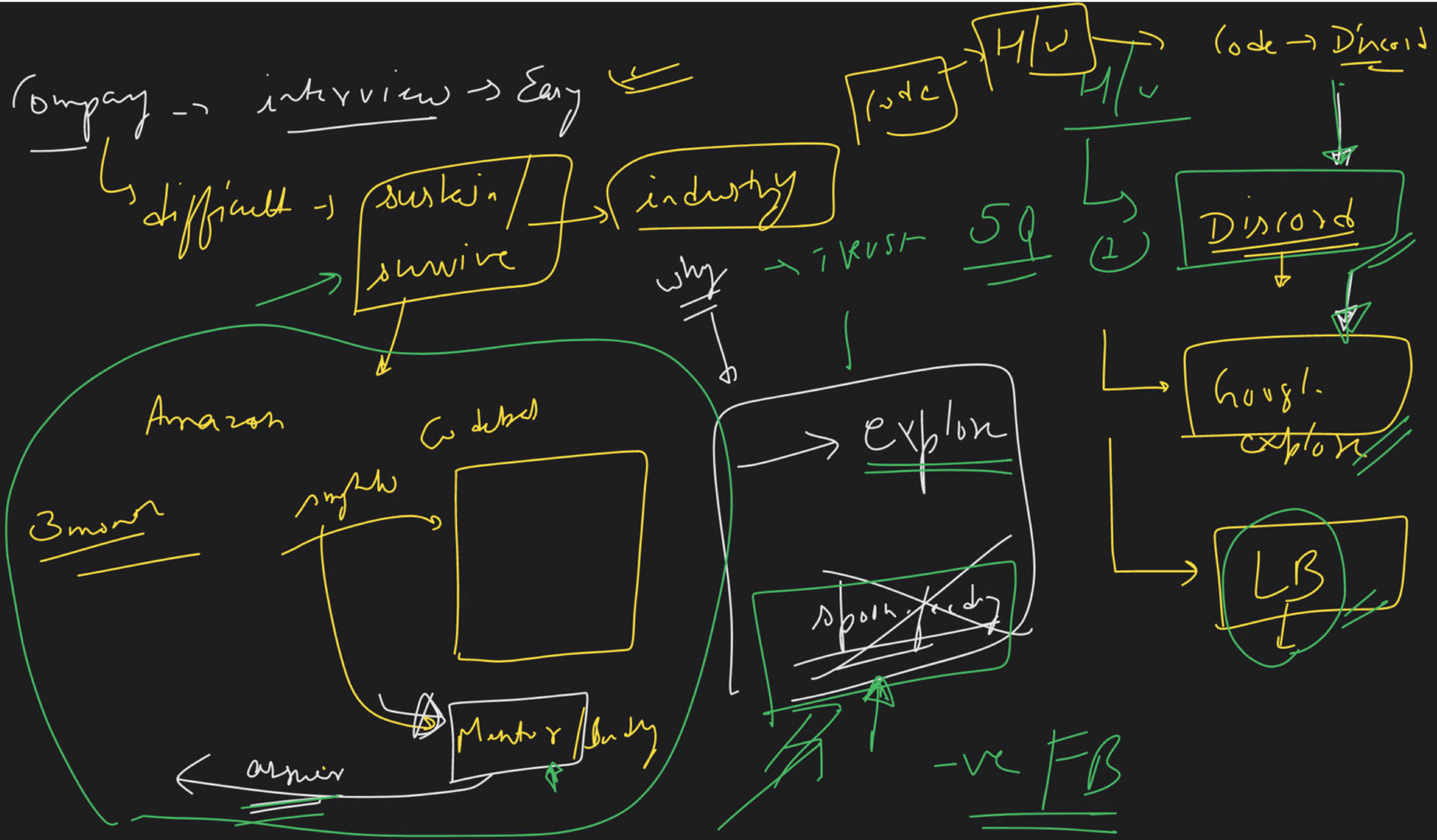
Tri();

Rec();

}

widely used in market


public class ShapArea implement Area
{
public void Square()
{
// input → side
Area calculate
problem sout
}
code definition
body
Scanner sc = new Scanner (System.in)
int s = sc.nextInt()
int area = s*s;
Ramya
sout (area);
}


interview → Easy
difficult → sustain / survive
industry
why
TRUST 50
Code → H/w
H/w
Code → Discord
Discord
explore
LB
Amazon
Mentor / Buddy
answer
explore
-ve FB

Interface

Keyword

(1) .Interface

(2) child class has to implement the Interface

(3) support multiple Inheritance

VS

Java

Sukanya

abstract class

Abstract

(2) child class ——— extend the abstract class

(3) ✗

class & Object → Structure

Inheritance → Types → Ambiguity —

Poly → types → Overloading —

→ Overriding —

Encapsulation → Adv → Rules → DH —

Abstract → Abstract —

→ Interface —

Human → easy → Blame

H/W

Example

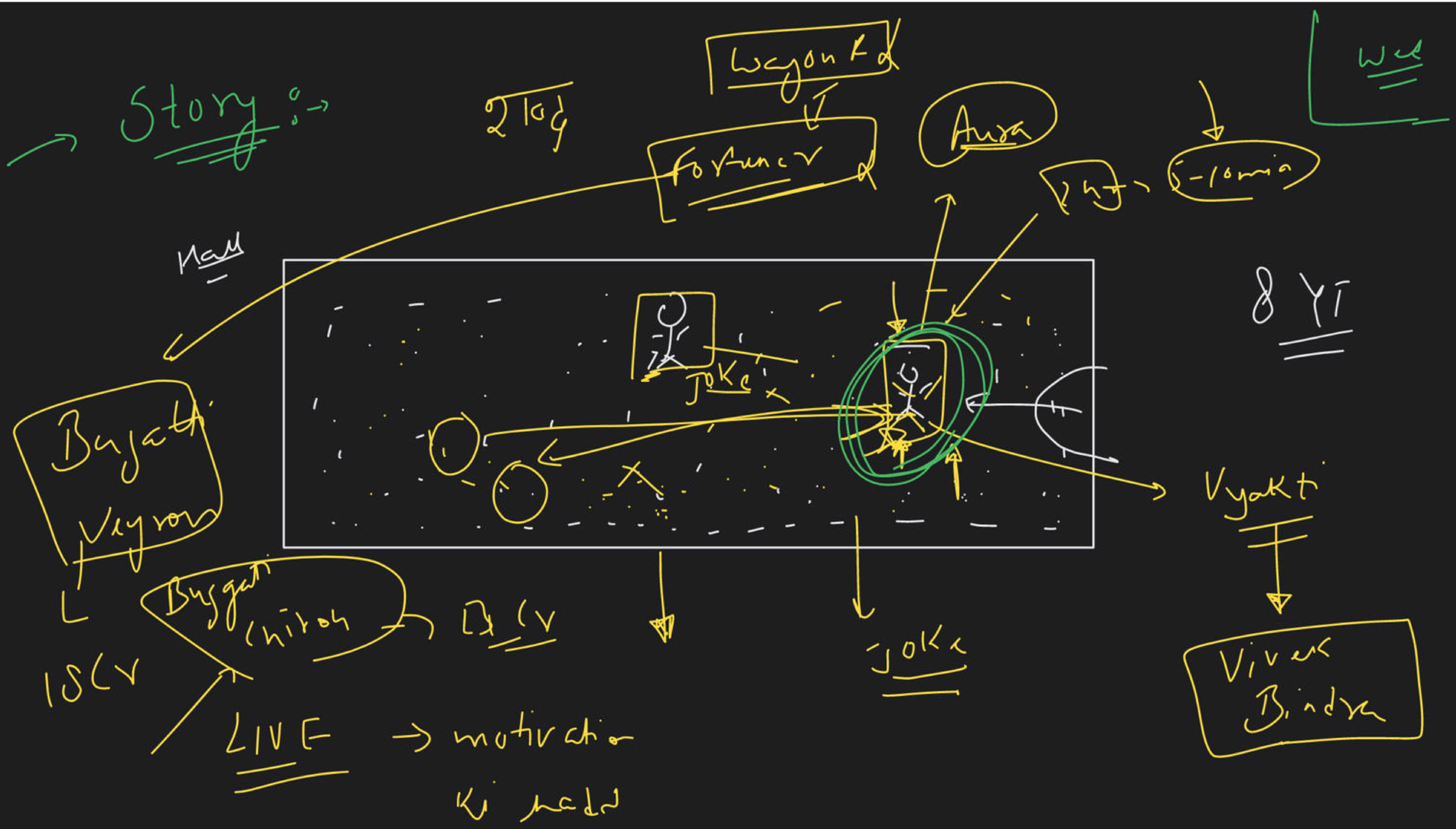
→ IDE

→ Code

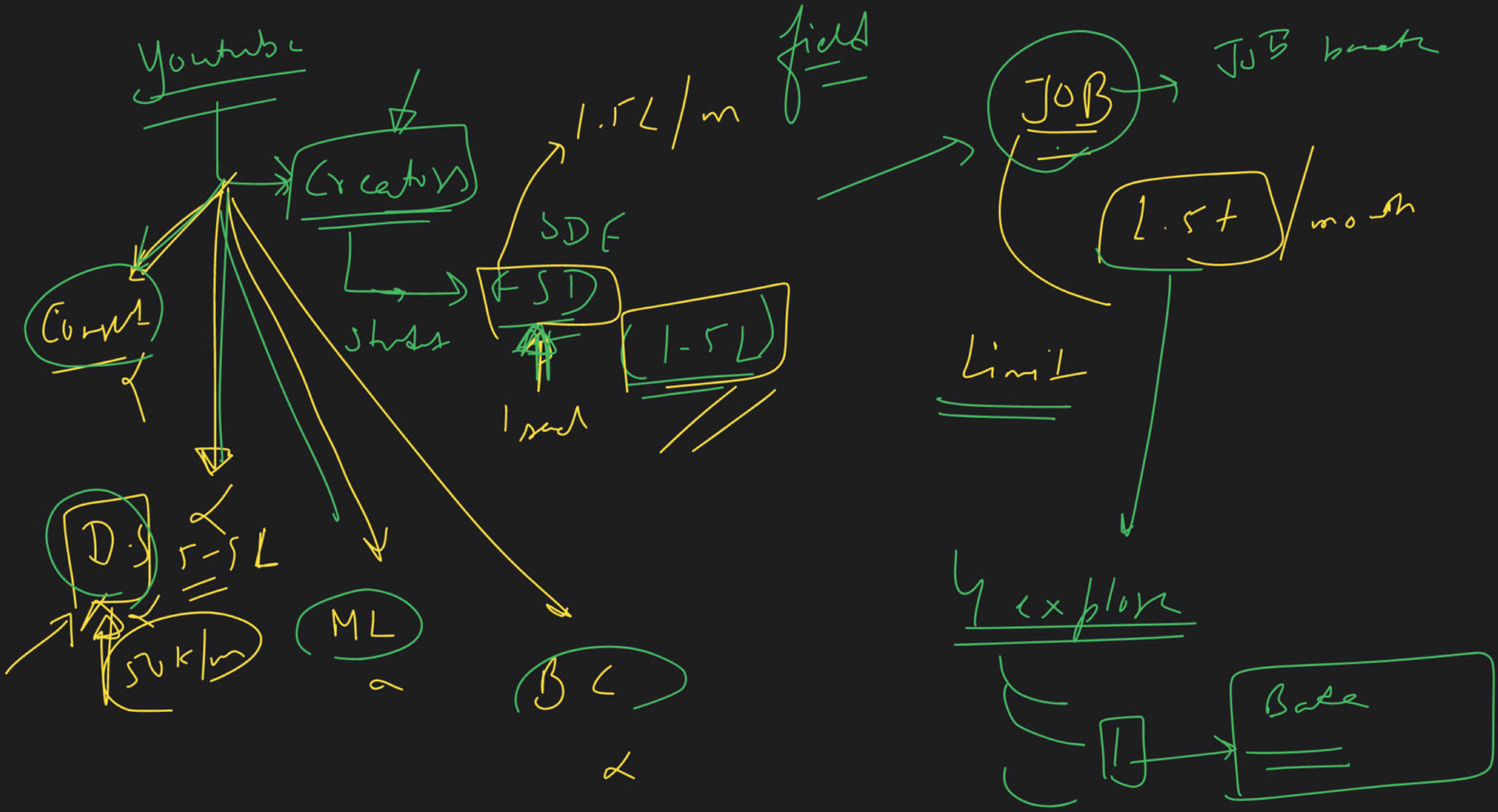
→ Run

→

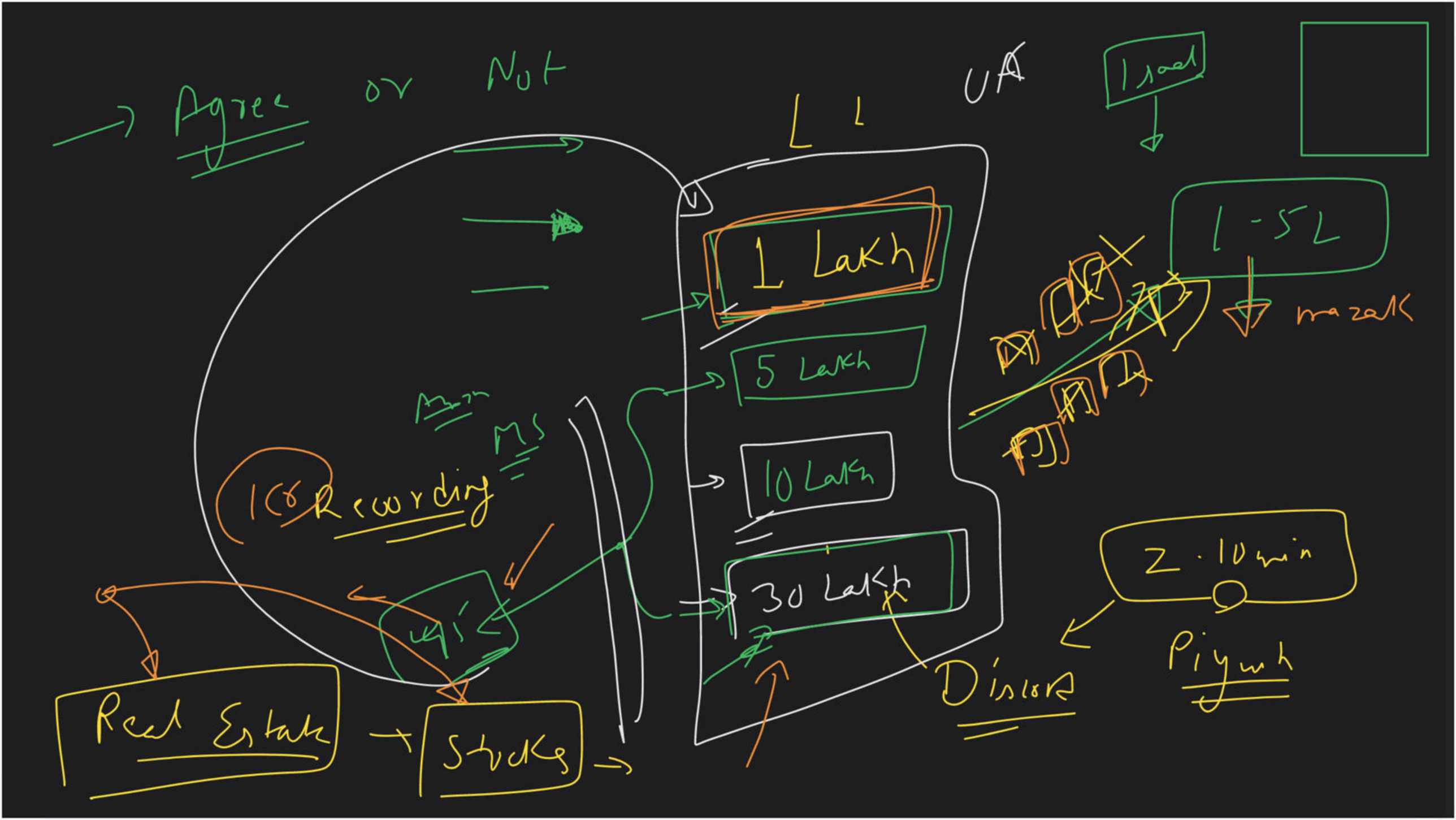
Confidence ↑

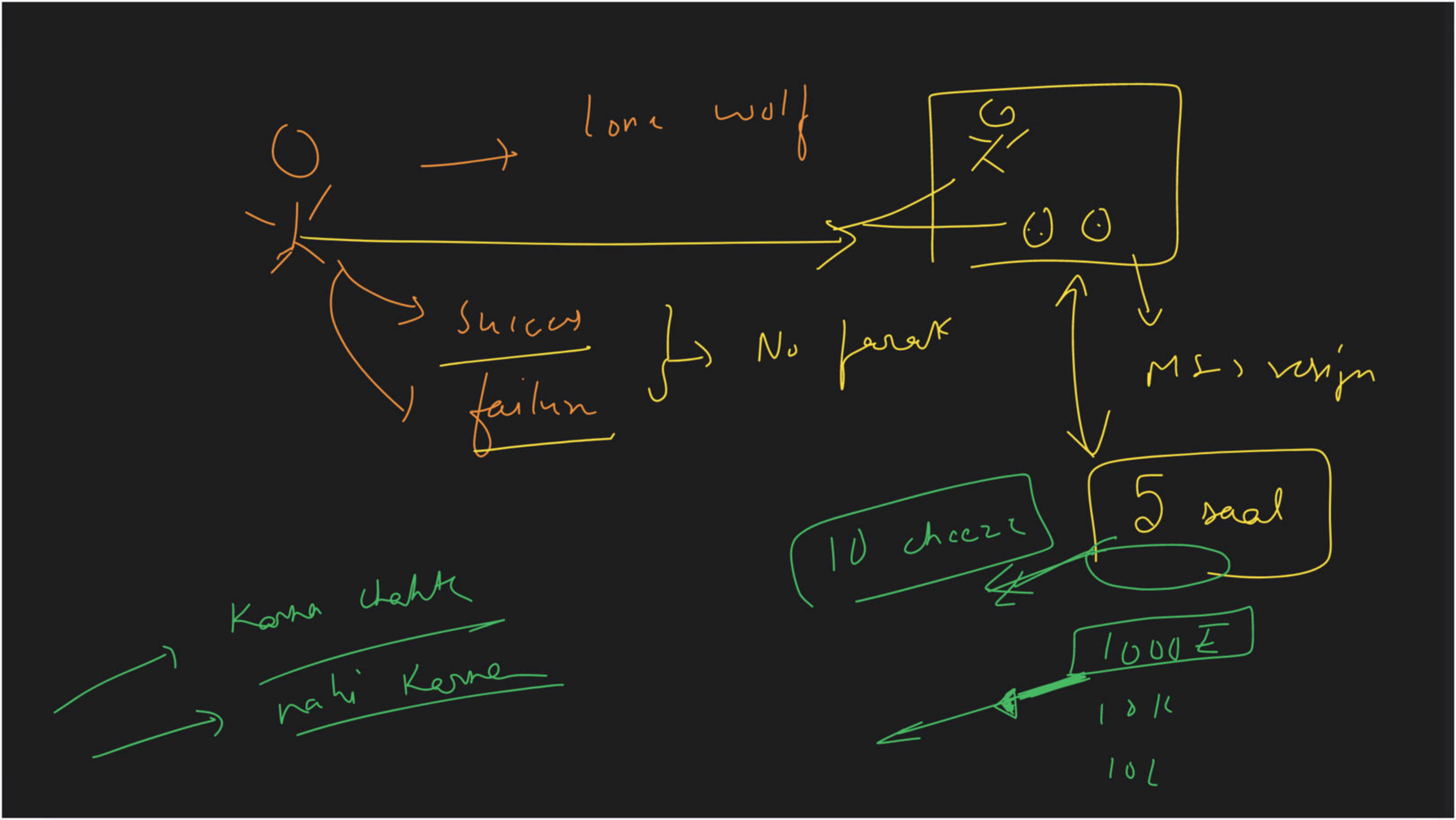


Story :
2 lac
Wagon R
Fortuner
Aura
Mall
8 Yr
Bugatti Veyron
Joke
Bugatti Chiron
15 Cr
Joke
Vyakti
Vivek Bindra
LIVE → motivation ki madad



Youtube
Creators
FSD
SDE
1.5L/m
field
1-5L
JOB
1.5+
month
Limit
explore
ML
BC
5-5L
50k/m


Agree or Not
L L
UA
Load
1 Lakh
5 Lakh
10 Lakh
30 Lakh
1 - 5L
mazak
Amazon
MS
ICICI Recording
Real Estate
Stocks
Divisor
2 . 10 min
Piyush


lone wolf
success
failure
No forest
MLS version
5 saal
10 chezz
Karna chahta
nahi Karna
1000 E
10 L