Recap

OOPS:-

BackTracking sheet

Mil gyi - ?

Pisliyd

OOPs - II

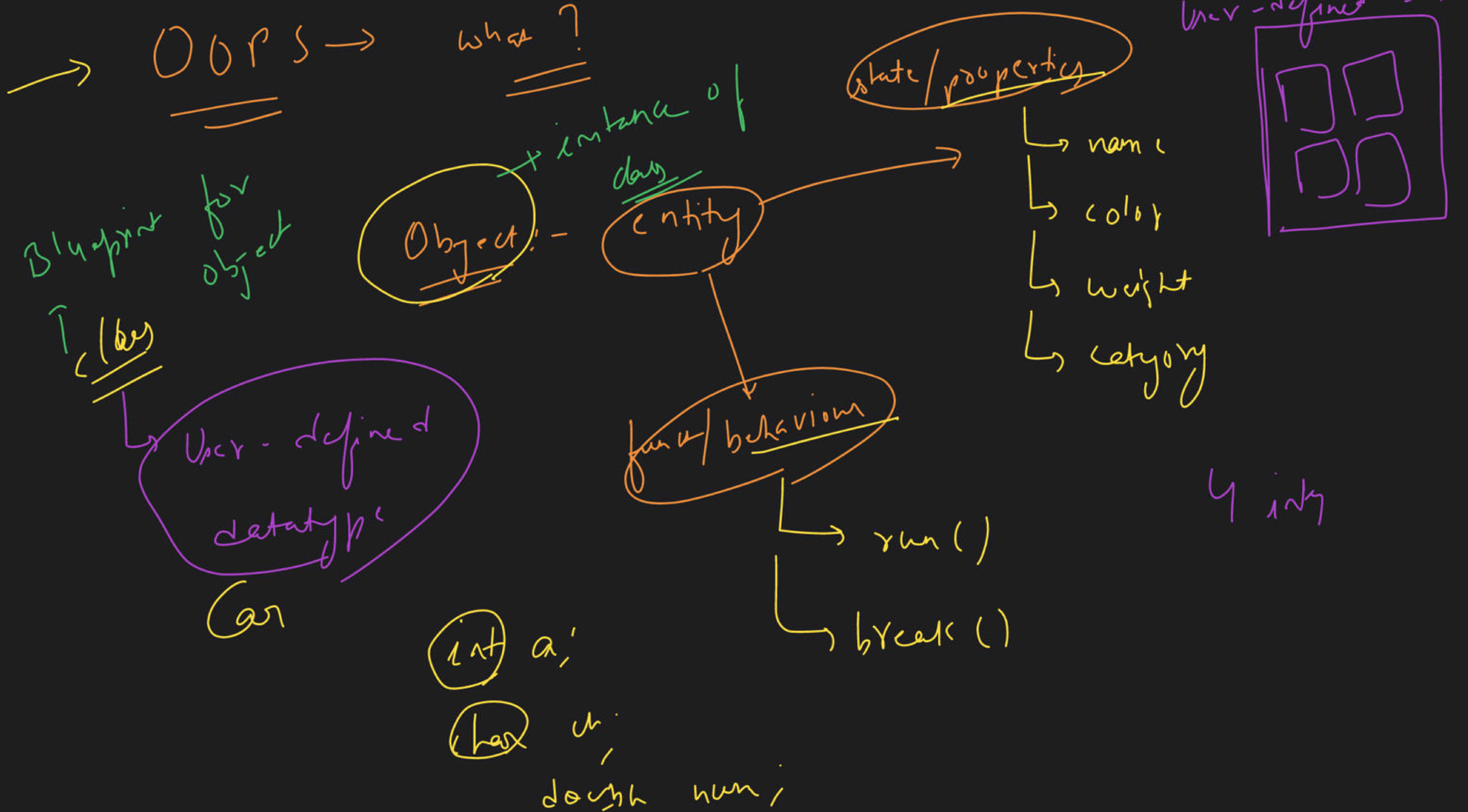
Foundation Course on Data Structures & Algorithm - Part I

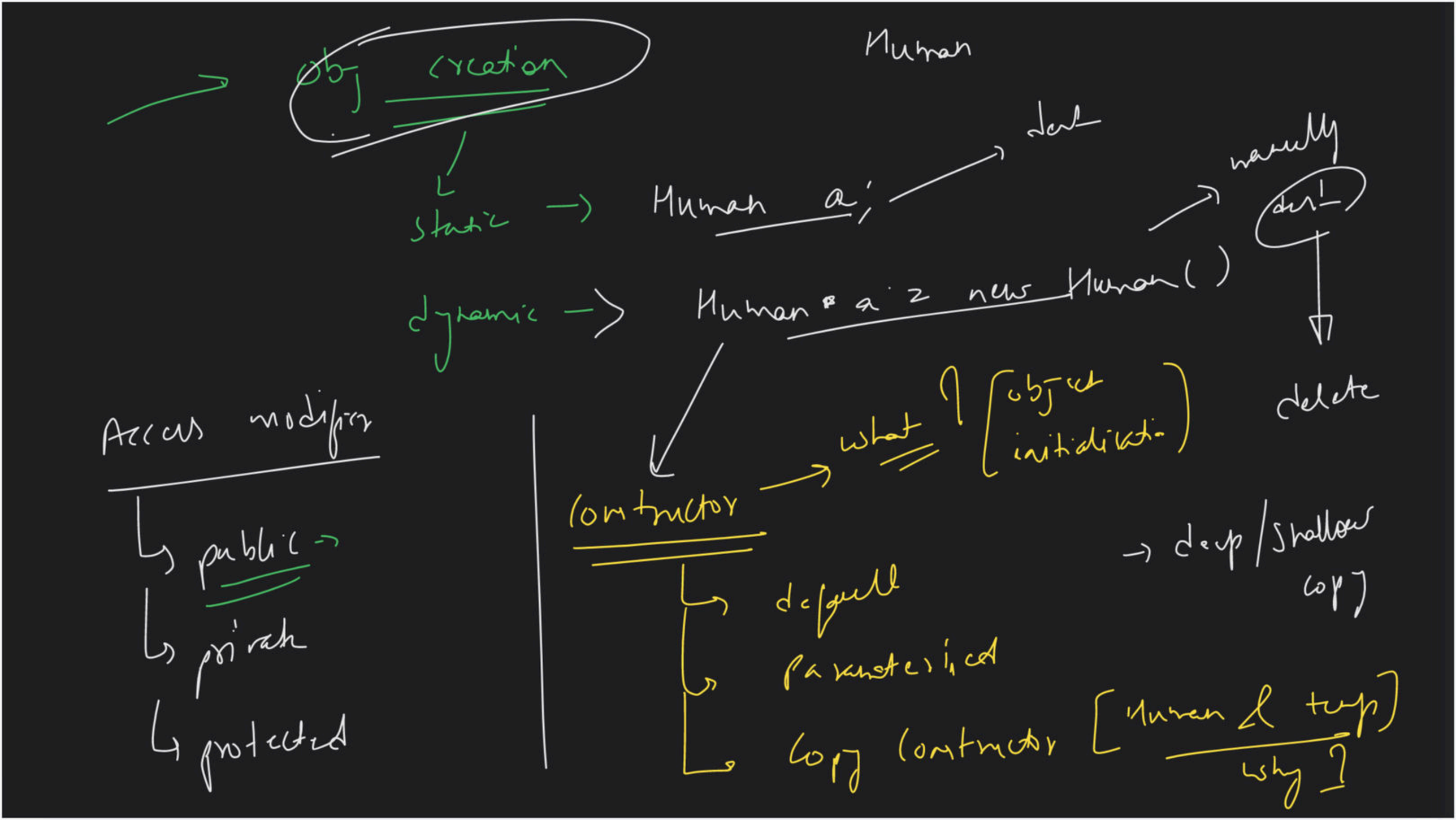
7:48 8:00 -> 8:59 -> 9:00

Priyal

1 - 2 min

Love Babbar • Lesson 27 • June 11, 2022


OOPS → what?
Blueprint for object
class
User-defined datatype
Car
Object :- entity
instance of
class
state/property
name
color
weight
category
func/behaviour
run()
break()
int a;
char ch;
double num;
User-defined DT
4 ints


Human
obj creation
static
Human a;
dot
manually
dot
dynamic
Human * a = new Human()
delete
Access modifiers
public
private
protected
constructor
what ? (object initialization)
default
parameterized
Copy Constructor [Human & temp]
why ?
deep / shallow copy

	public	private	protected
within class	✓	✓	✓
derived class	?	?	?
outside class	✓	X	X

Empty class

```
class Human
{

};
```

```
Human a;
cout << sizeof(a);
```

→ ans → 1

error or not ?

Babbar

Codehelp → YT →

→ OOPS → 4 pillar of OOPS

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

Encapsulation → Data Hiding / Information Hiding

↳ Binding data members / function together in a capsule

Why ?

single entity

↳ Security

↳ clean code

↳ Readability
↳ Reusability
↳ extensibility

Fully

full encapsulation

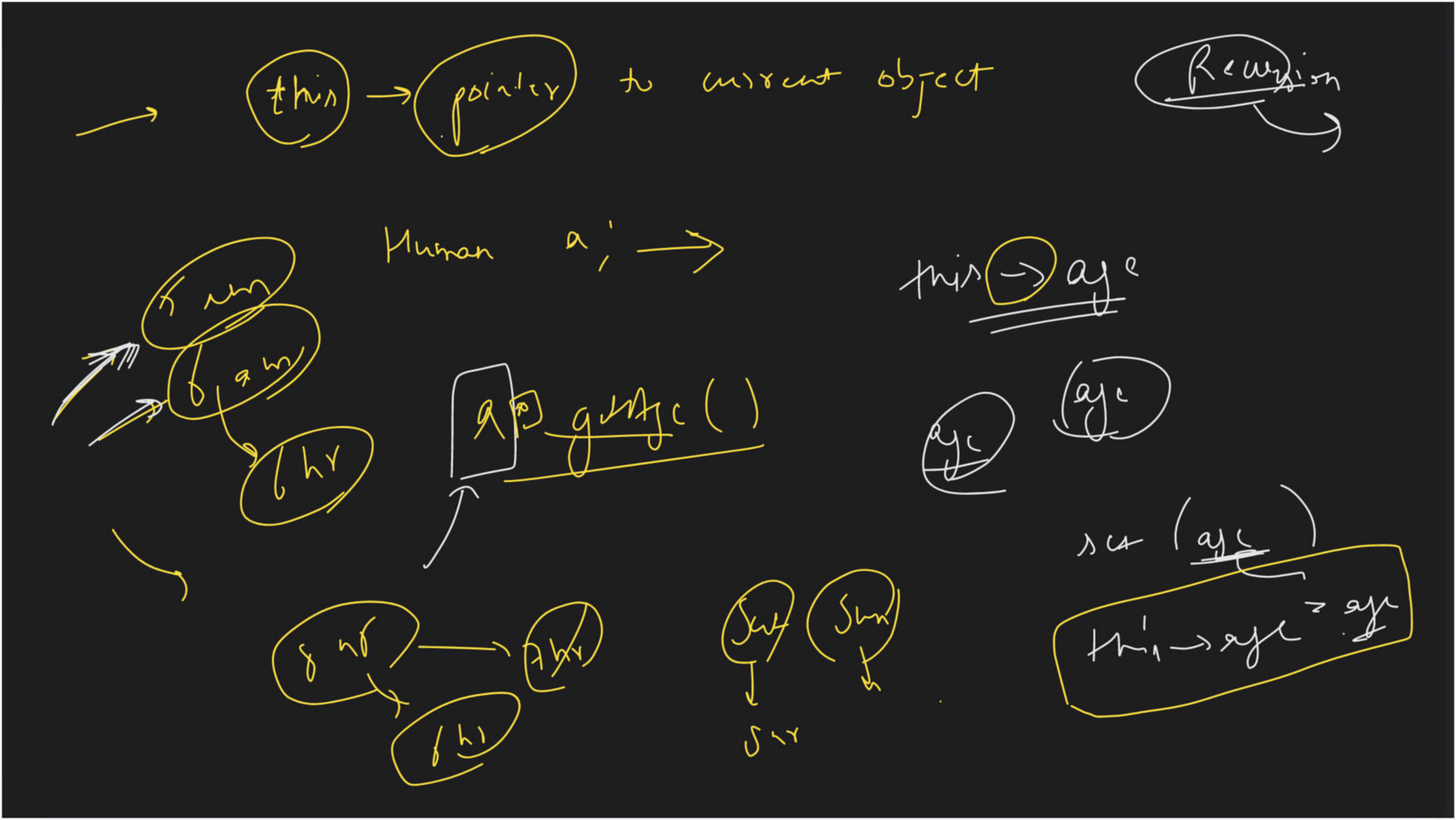
100%

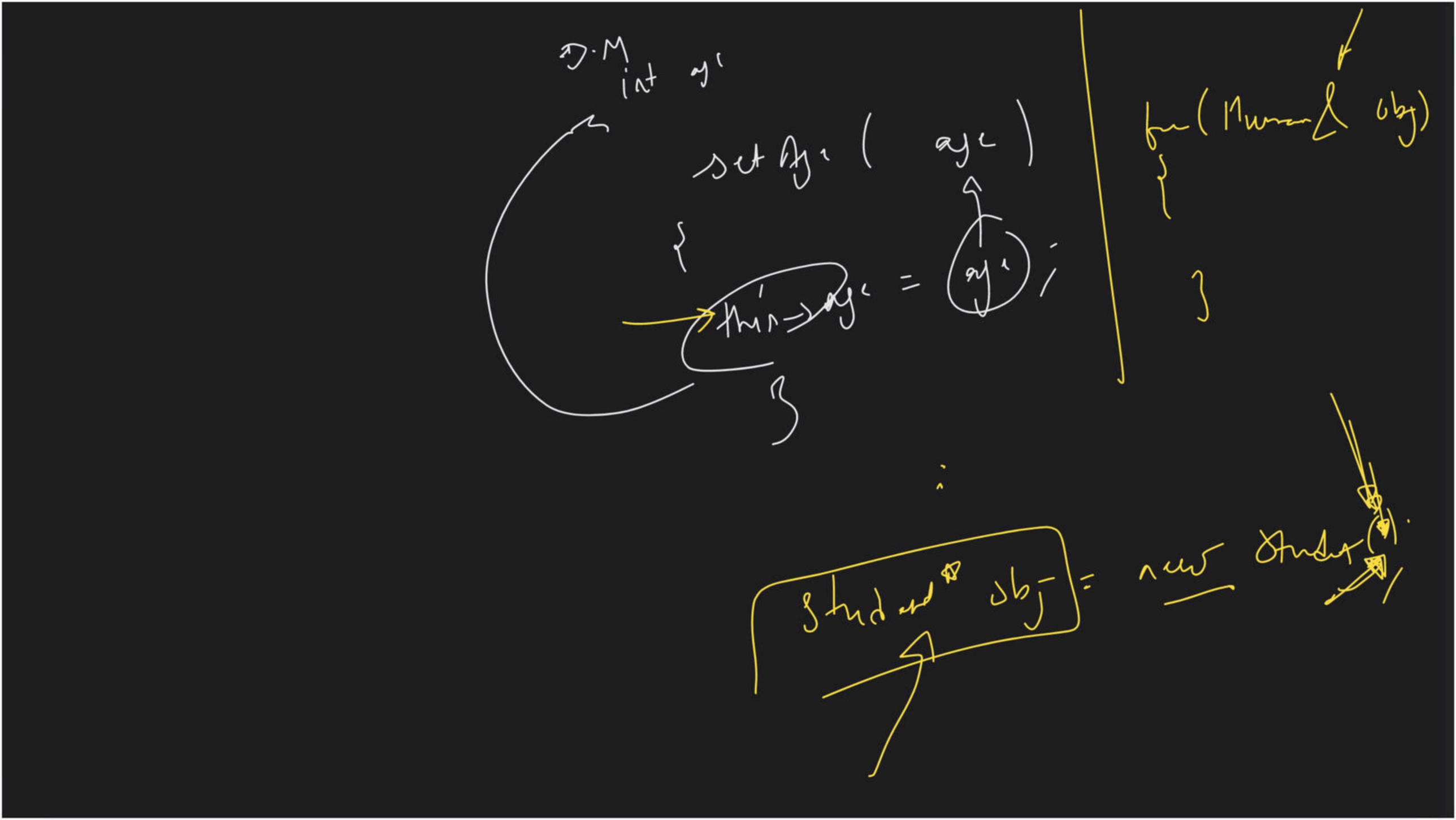
class Human

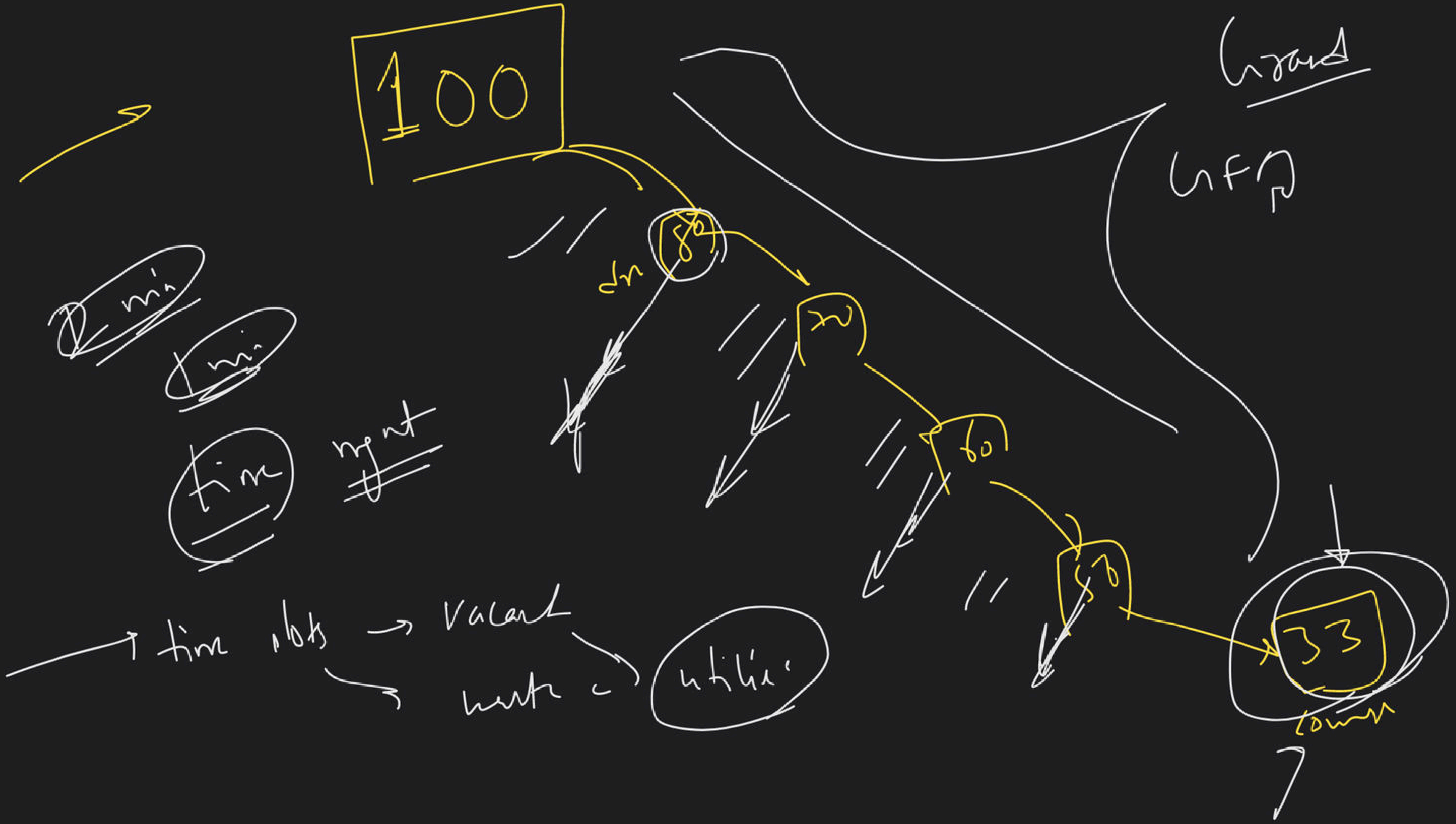
DM → all private

DF

3

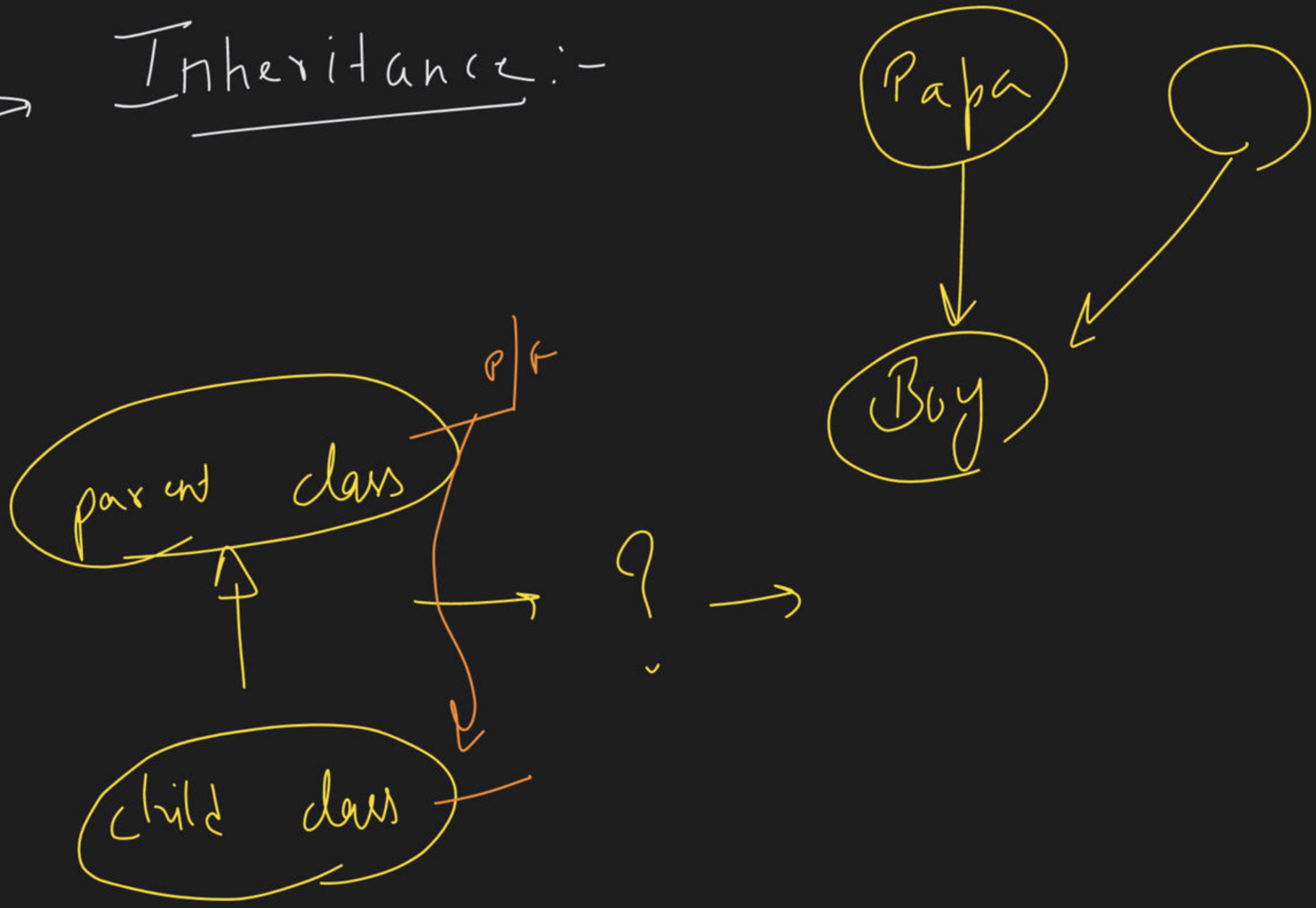

this
pointer
to current object
Human a;
this->age
sa(age)
this->age = age;

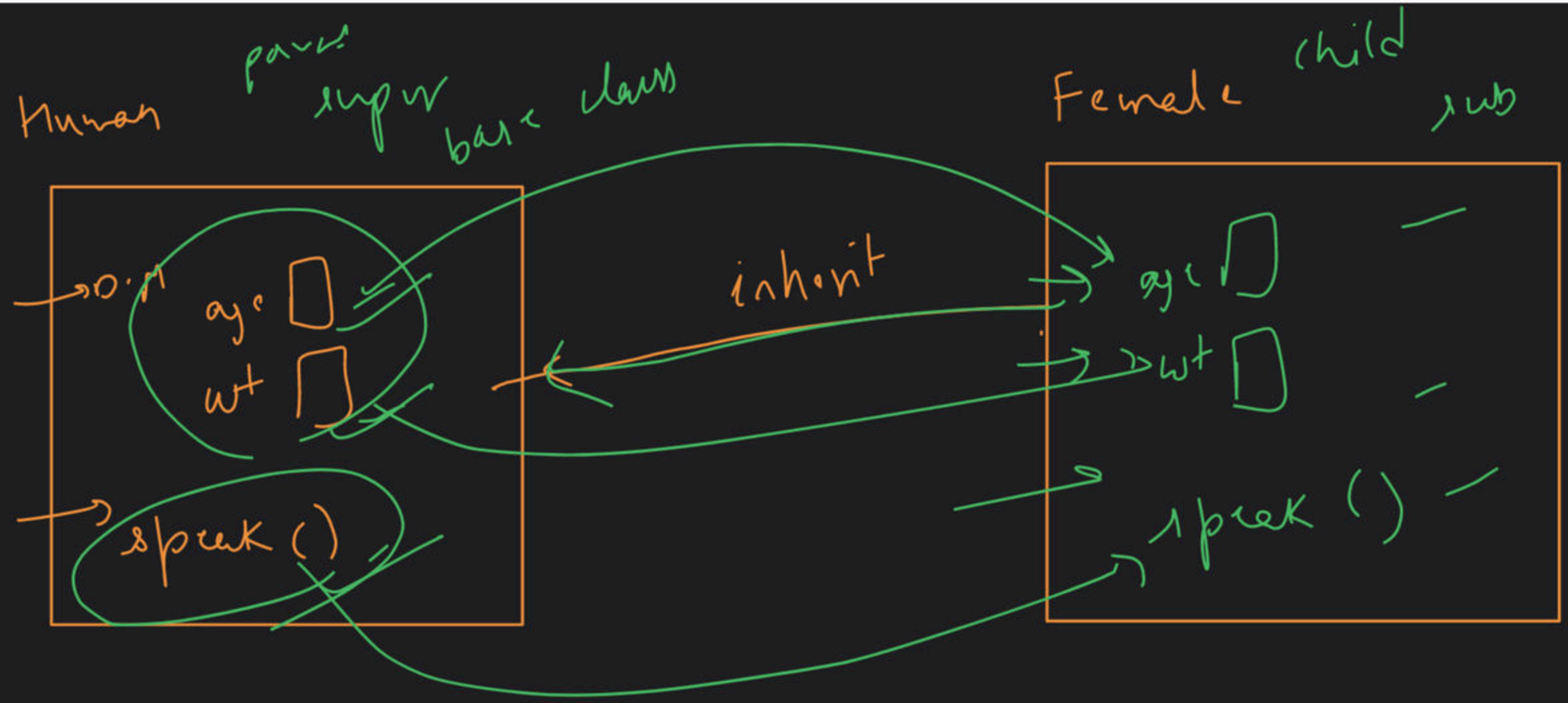

D.M
int ag;
setAge(age)
{
this->age = age;
}
fun(Human &obj)
{
}
Student obj = new Student();

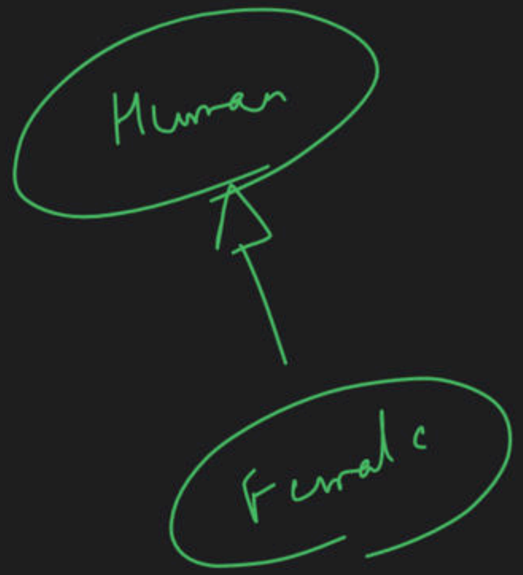

100
80
70
60
50
33
time
time slots
vacant

→ encapsulation → Imp → public → Adv:-

Inheritance :-

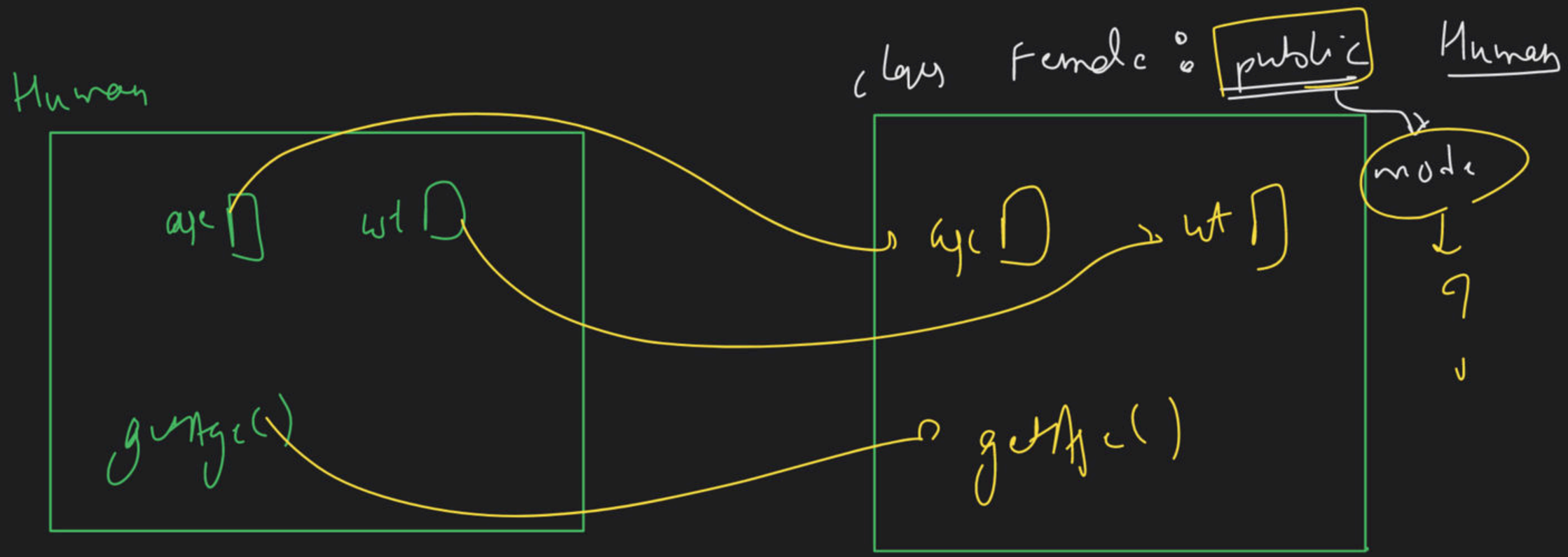


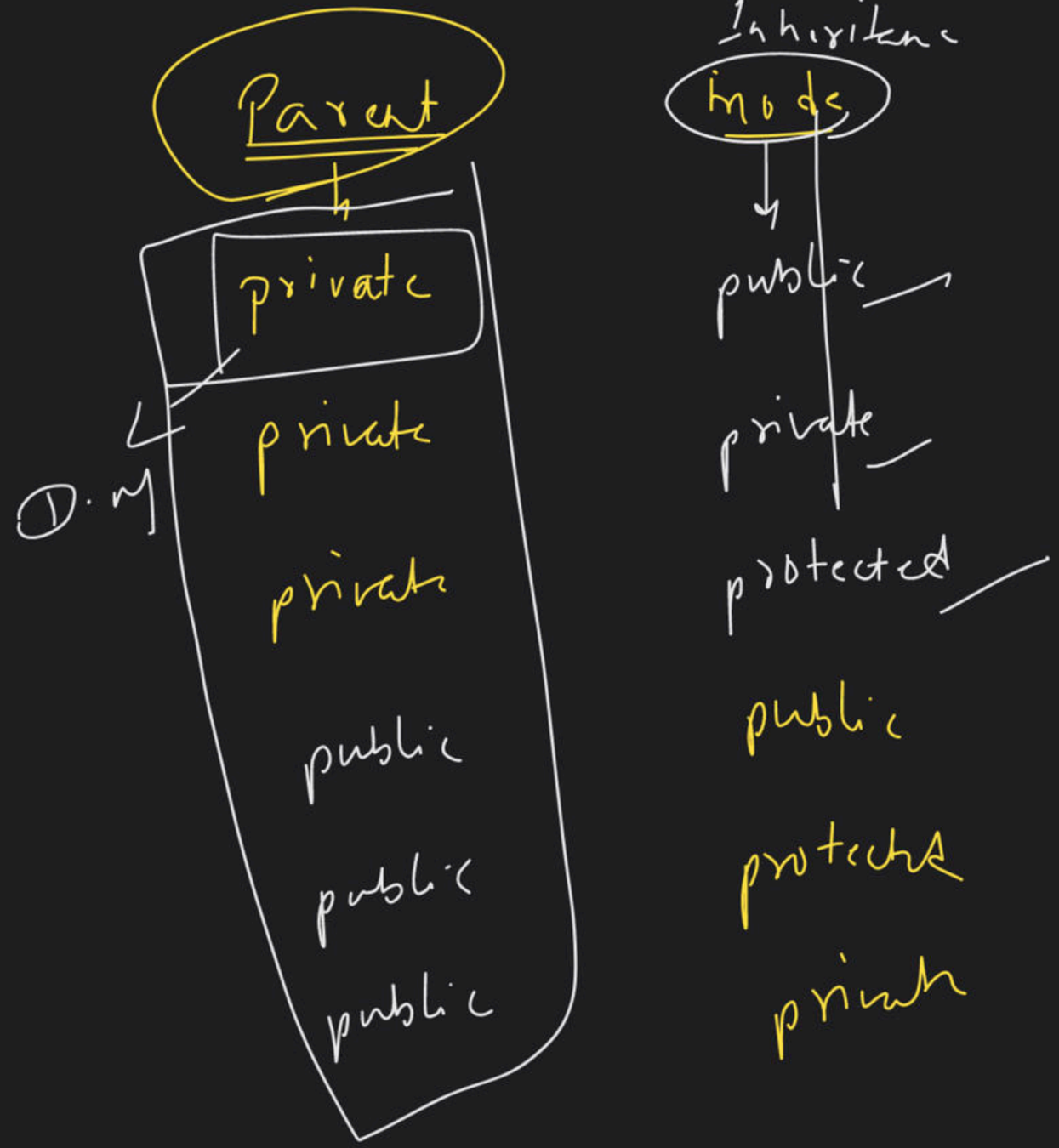




Adv:-

→ Reusable
Code


Human
class Female : public Human
mode
age
wt
getAge()

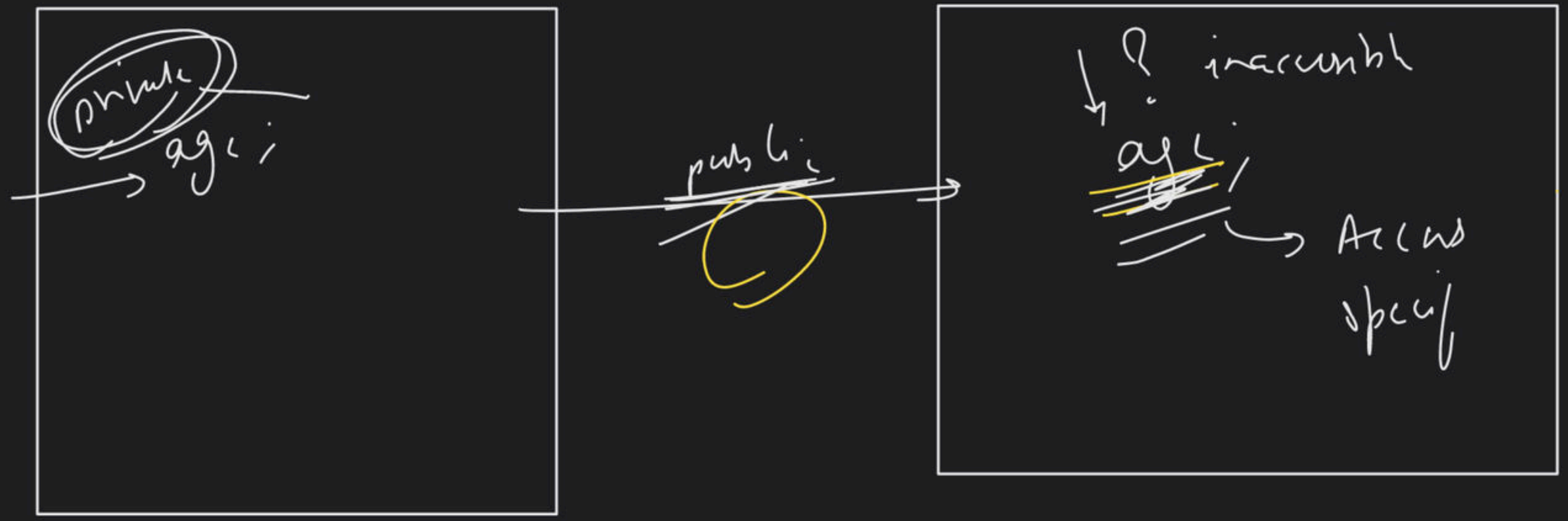


Parent	mode
protected	public
protected	private
protected	protected

class Female : [ ] Human

mode

Human



parent

private

private

private

mode

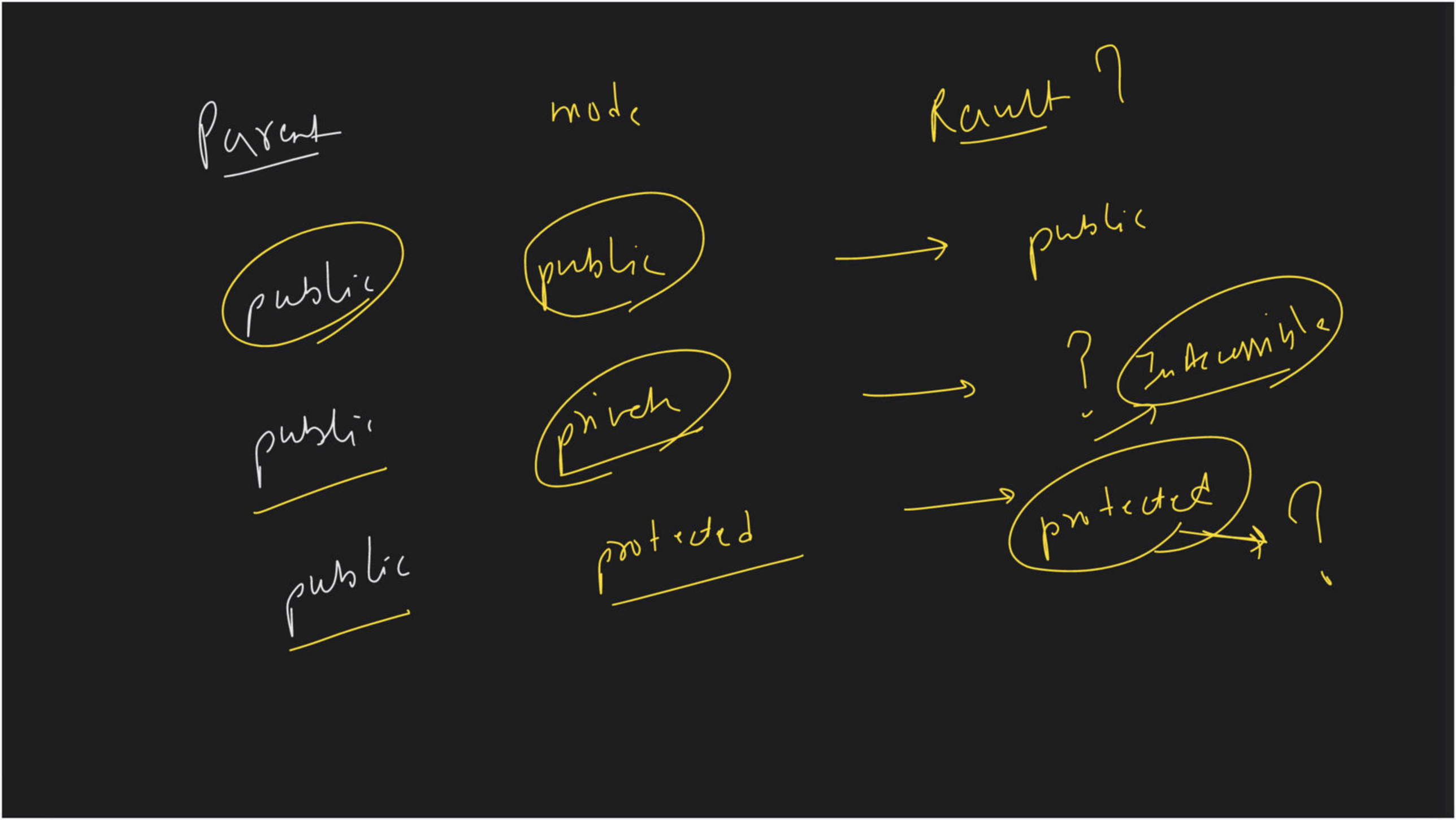
public

private

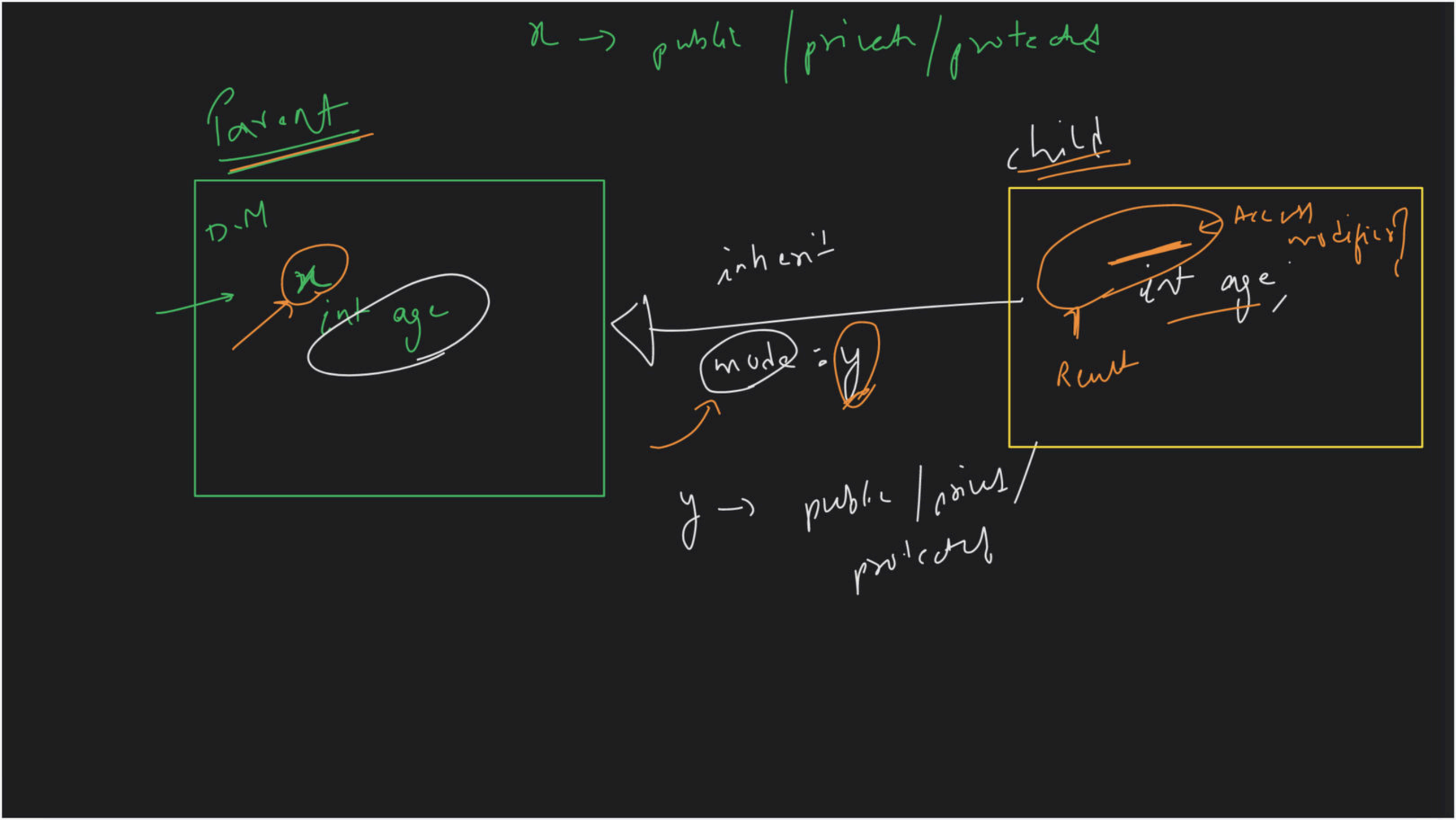
protected

Result ?

→ ~~private~~ inAccessible

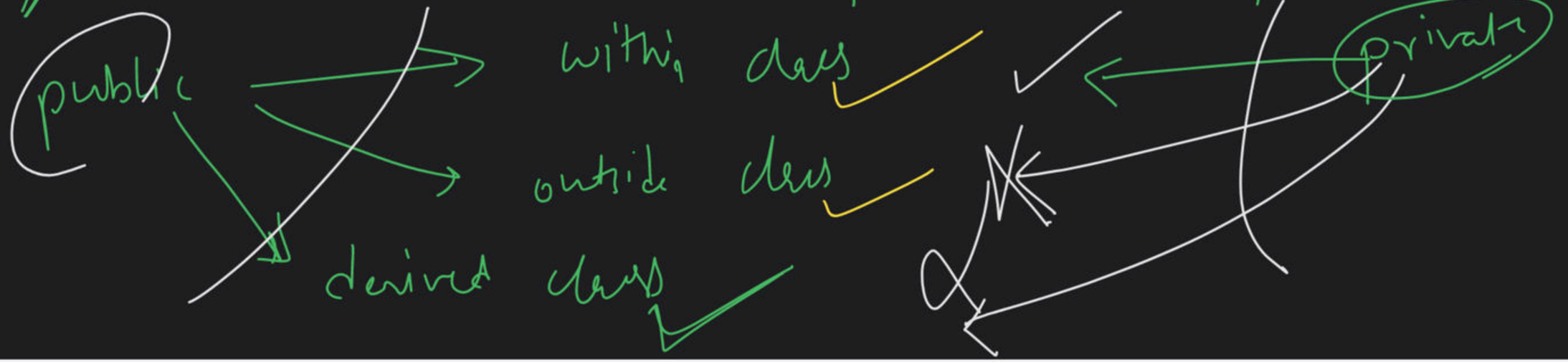

Parent
mode
Result ?
public
public
public
public
public
private
protected
?
Inaccessible
protected
?

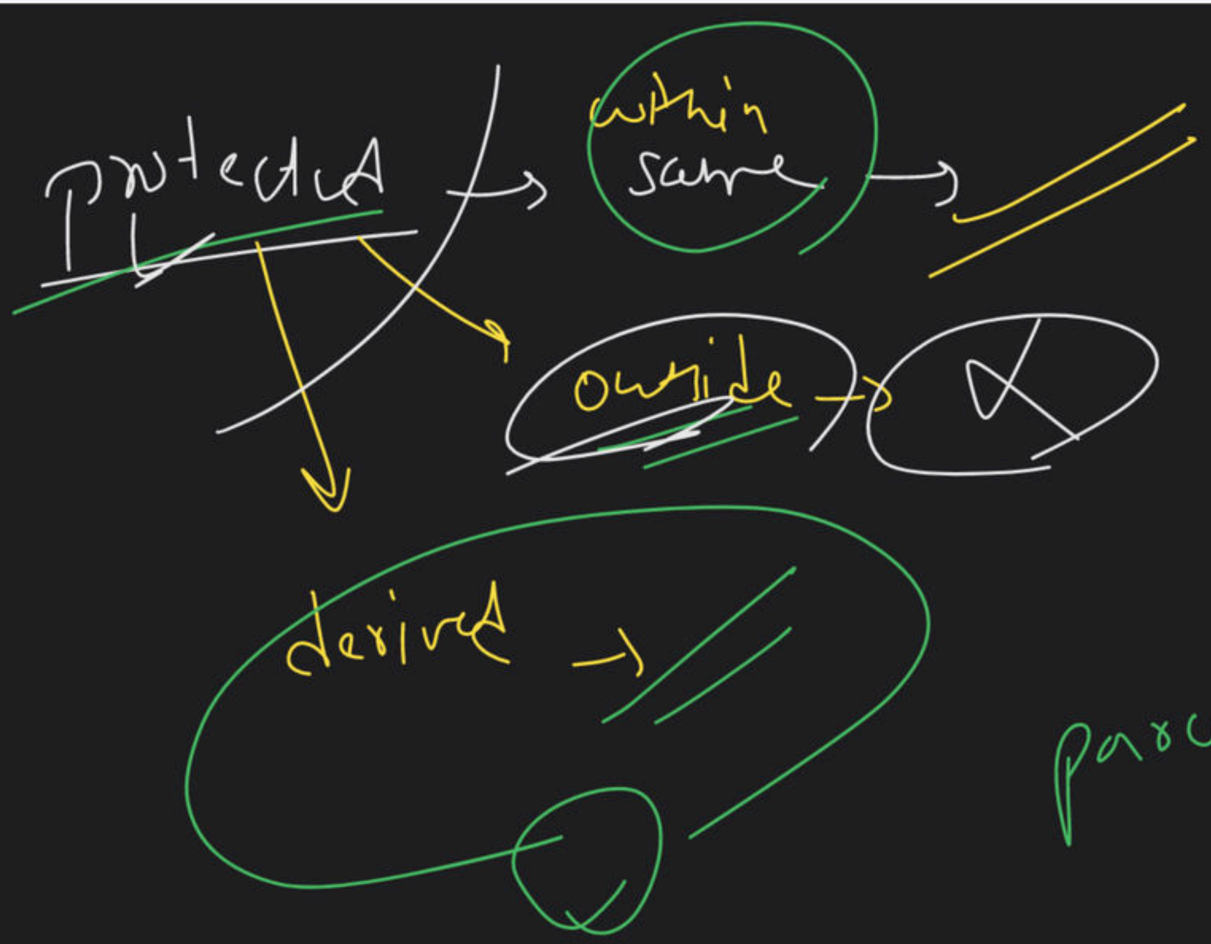
mode	Result
public	protected
private	private
protected	protected


x → public / private / protected
Parent
D.M
x
int age
inherit
mode = y
child
Access modifier?
int age
Result
y → public / private / protected

Grand child →

Parent →	public	private	protected
mode ↓ public	Public	private	protected
private	NA	NA	NA
protected	protected	private	protected





private

public

Base class

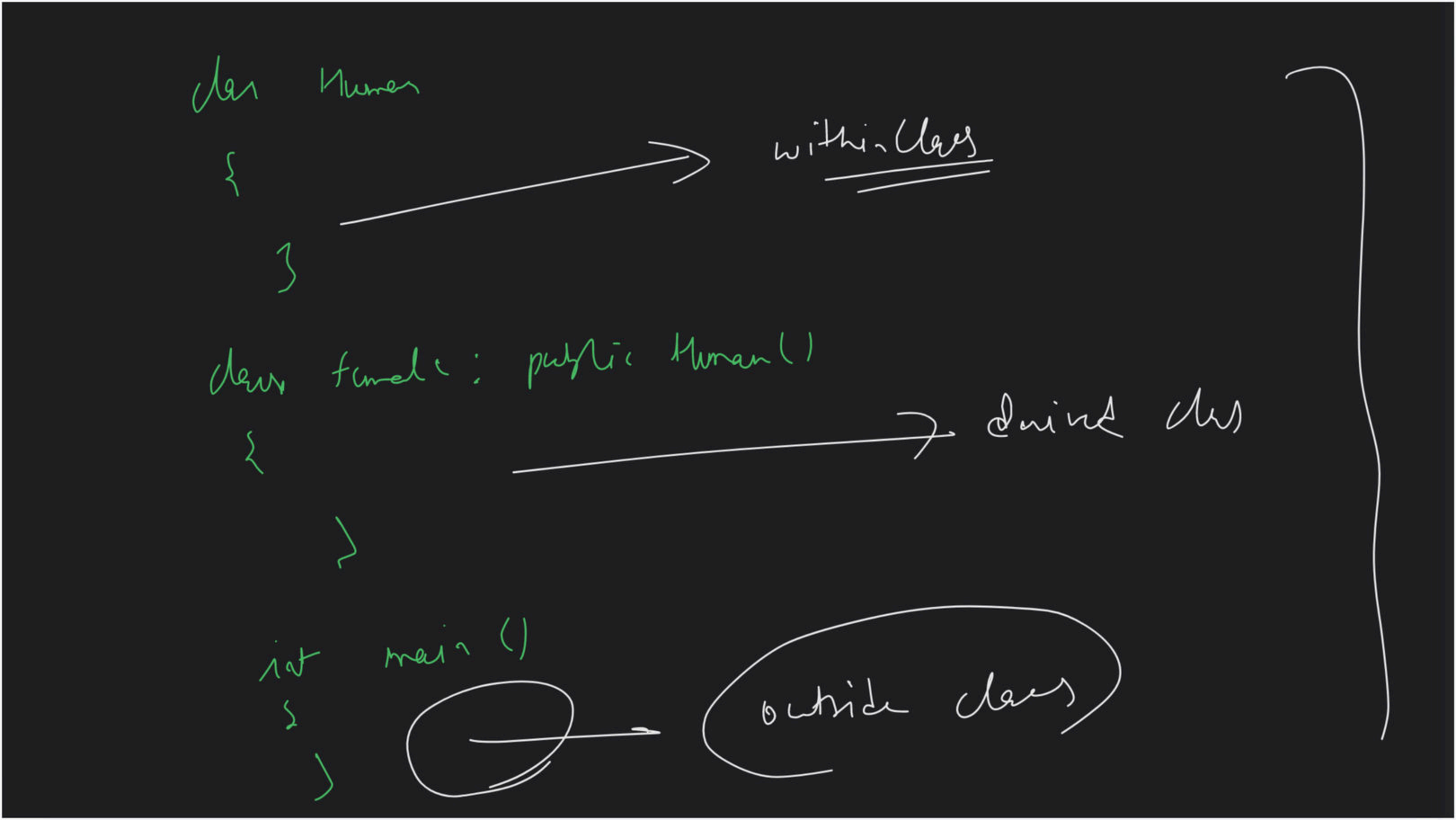
protected

parent

Human

Female

derived class


class Human
{
}
within class
class female : public Human()
{
}
derived class
int main()
{
}
outside class

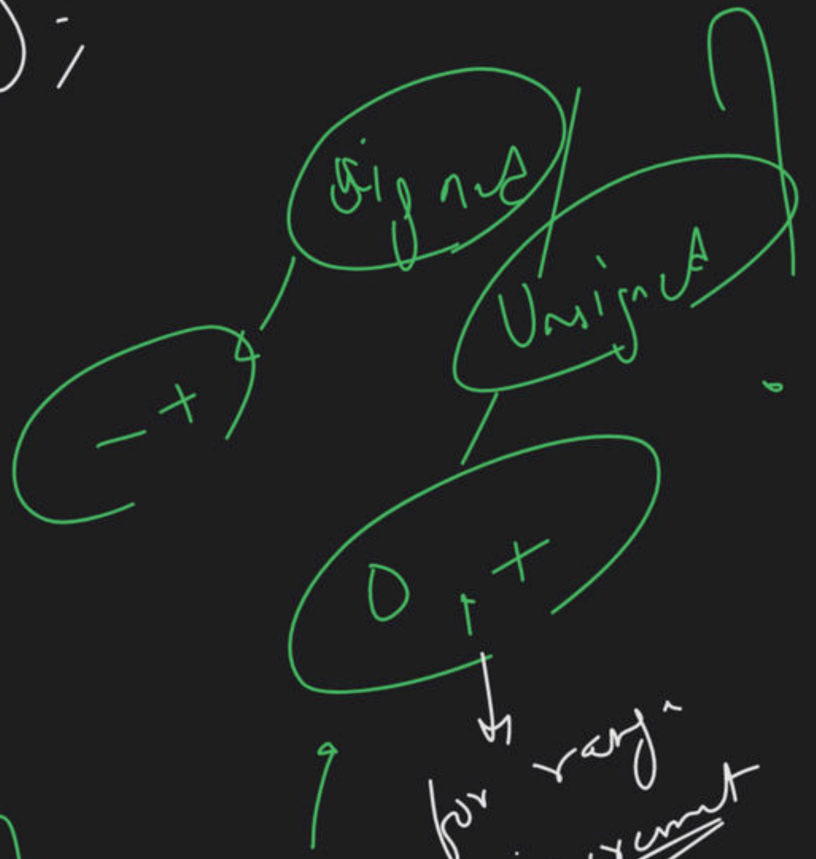
Ramnarayan

```
Human* a = new Human();
```

```
a -> age = 3;
a -> wt = 3;
a -> getAge();
```

int a = 3

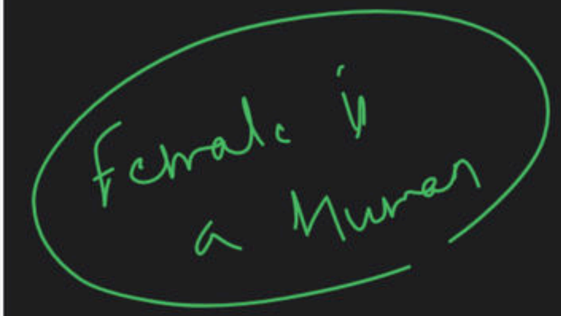
[— 9 + n]

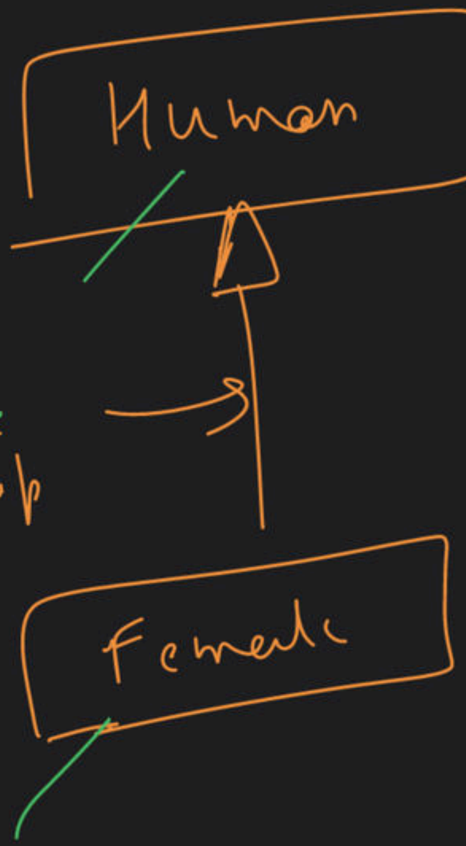


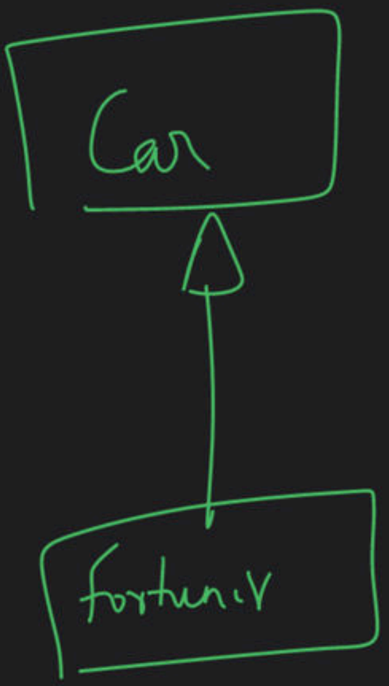
Inheritance ? → How

Adv:-

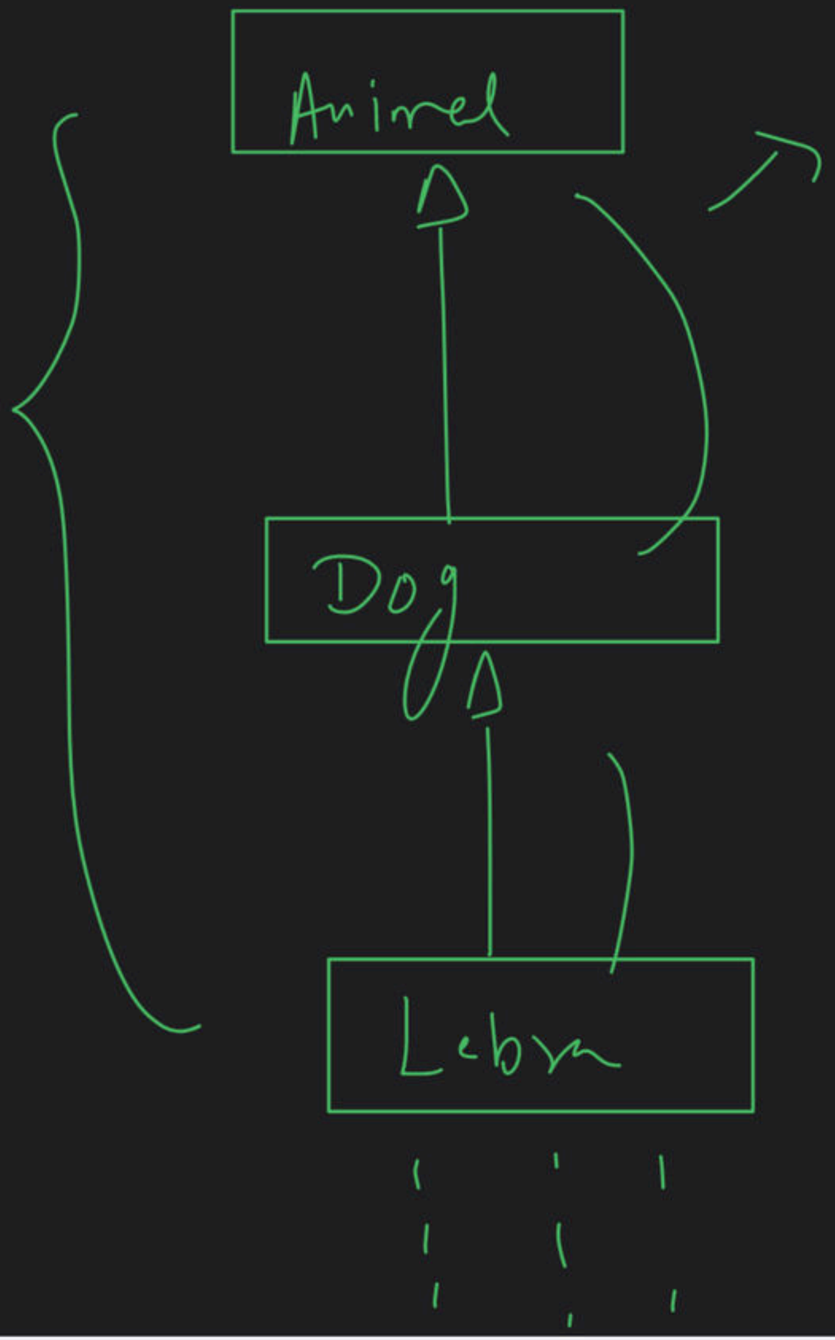
types → Single Inheritance







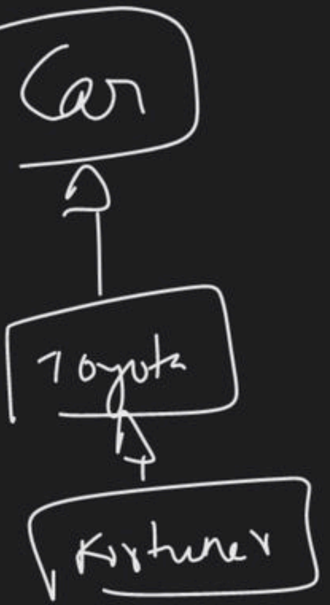
(2) Multilevel



Labra is a Dog

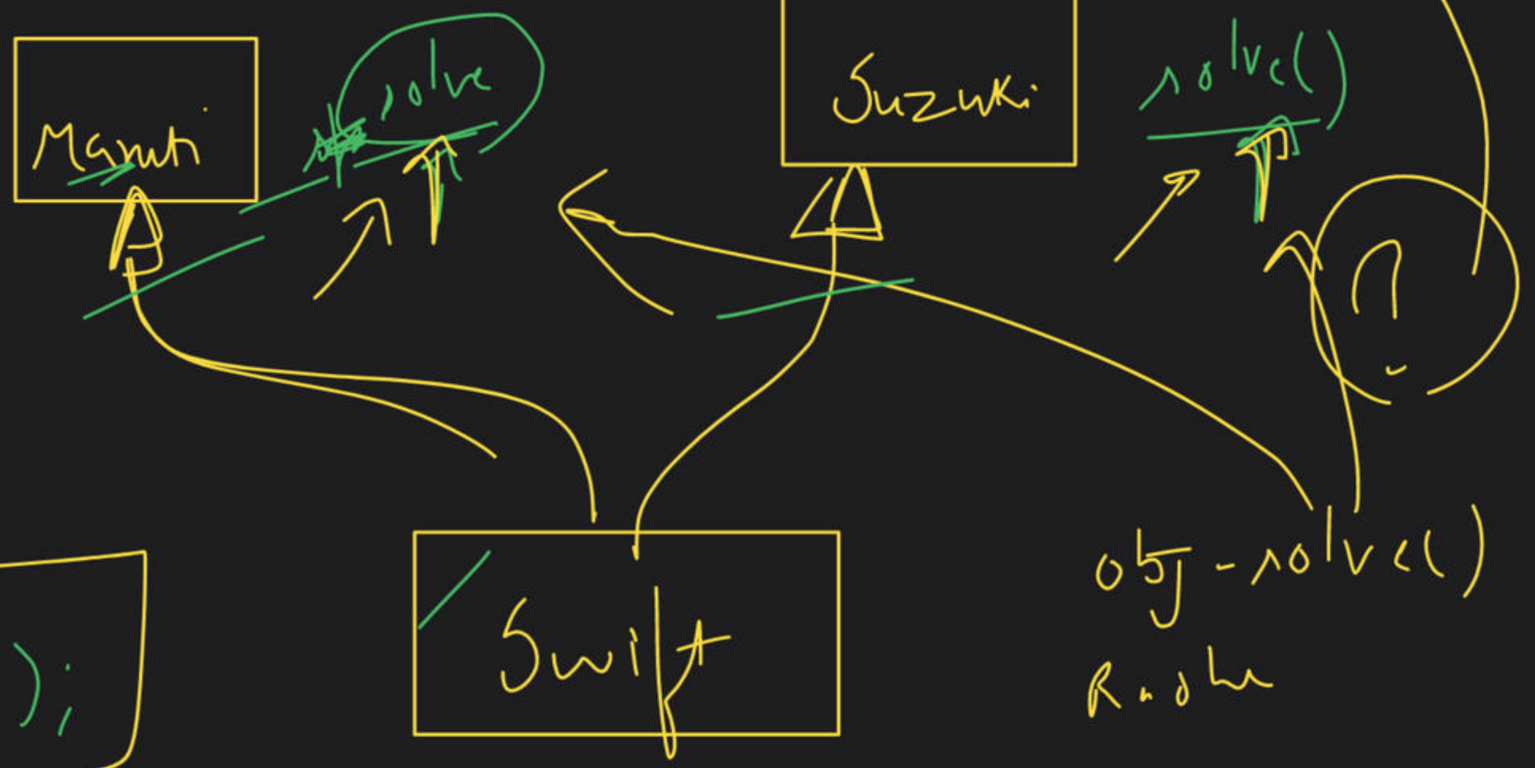
Dog is a Animal

(Ex)



③ Multiple:-

Imp → Inheritance Ambiguity



Swift obj;

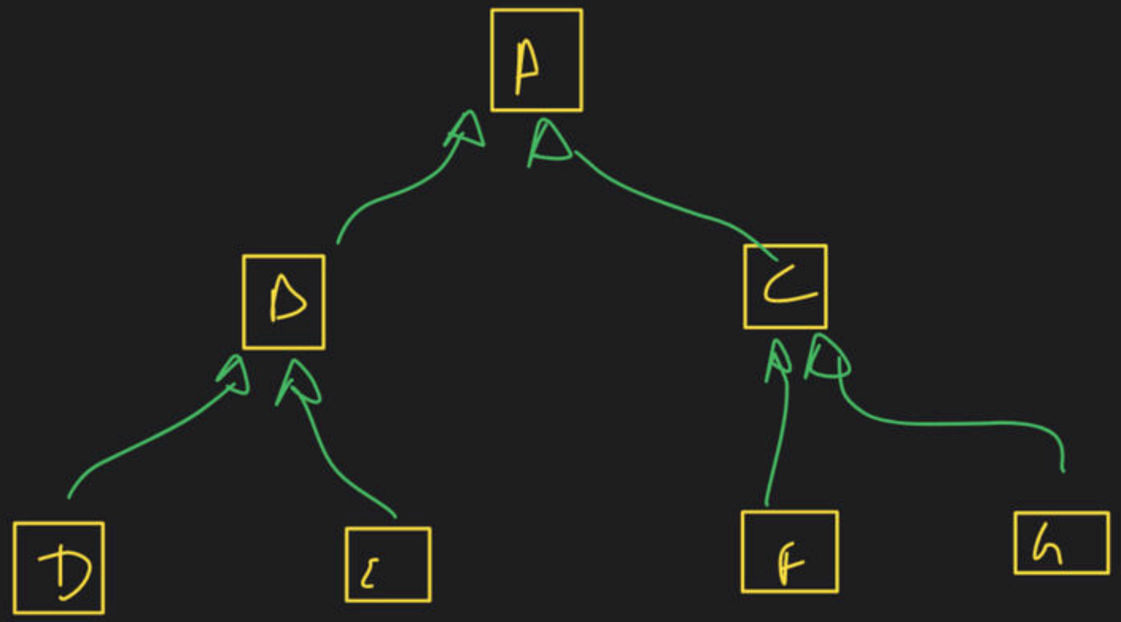
obj.Maruti::solve();

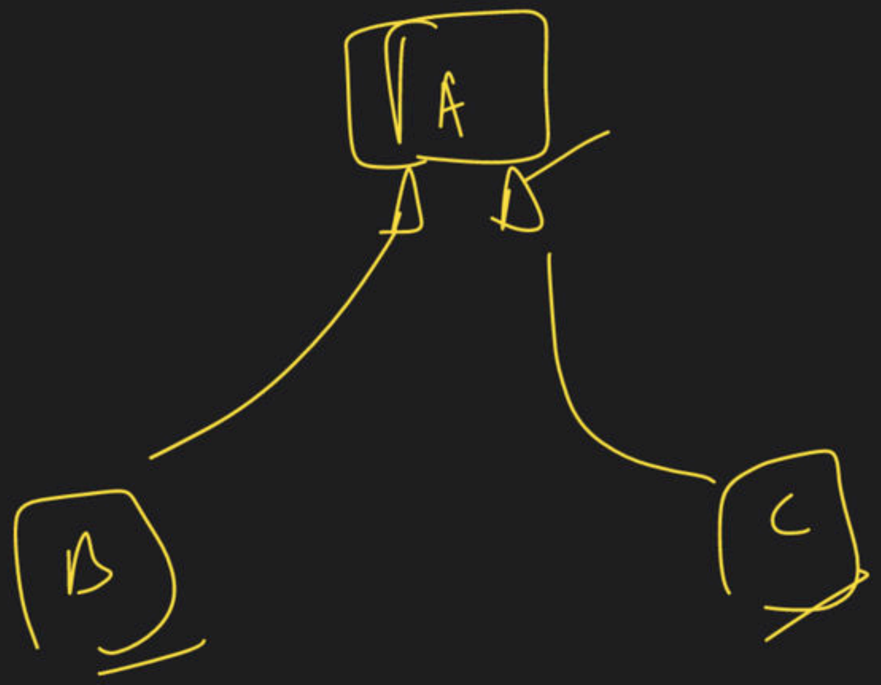
class Swift : public Maruti, Suzuki

scope resolution operator

→ Hierarchical :-

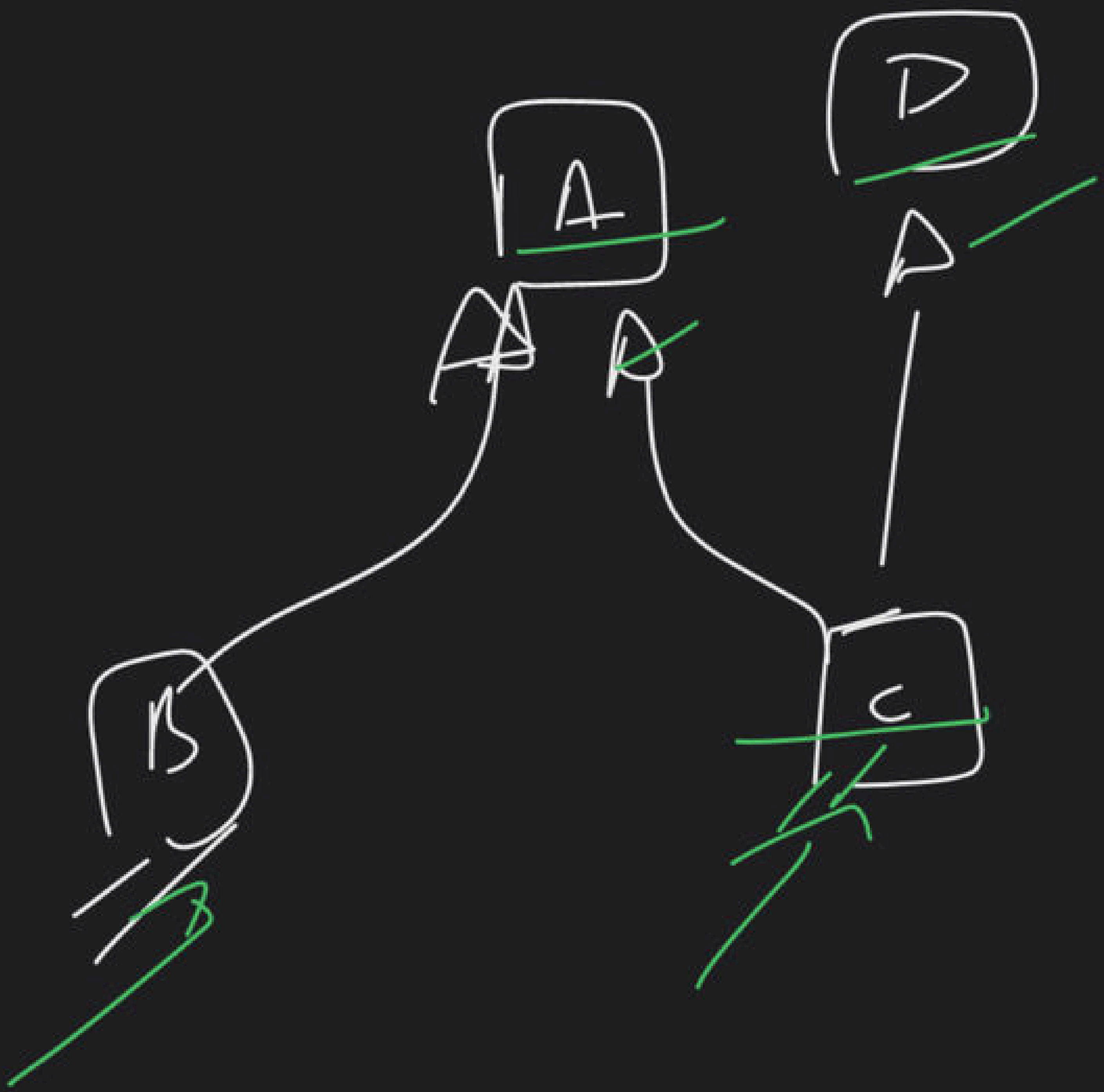
One class serves as Parent class for more than 1 class



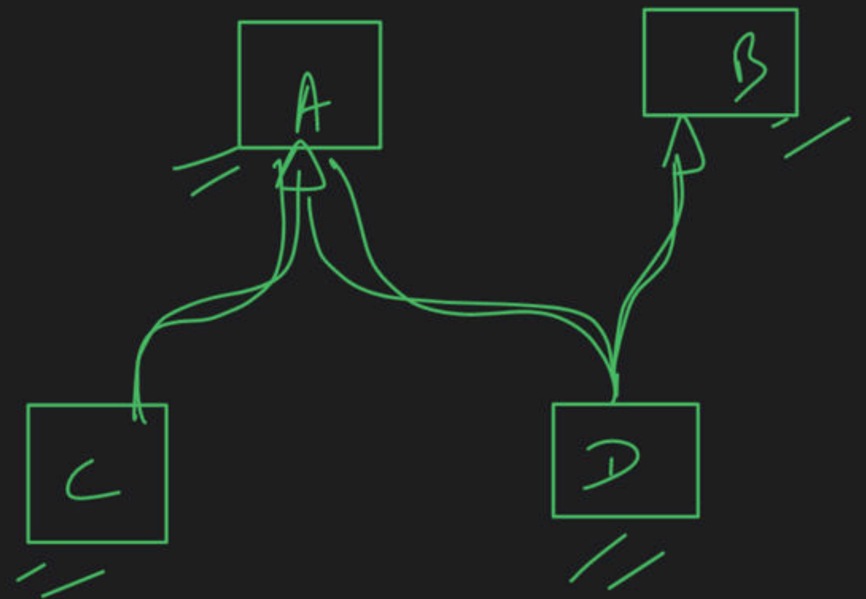



```
class A
{

};

class B : public A
{

};

class C : public A
{

};
```

→ Hybrid Inheritance → Combination of more than 1 type of inheritance



Swift ⊗ → new




```
class A
{
};
class B
{
};
```

```
class C : public A
{
};
```

```
class D : public A , public B
{
};
```

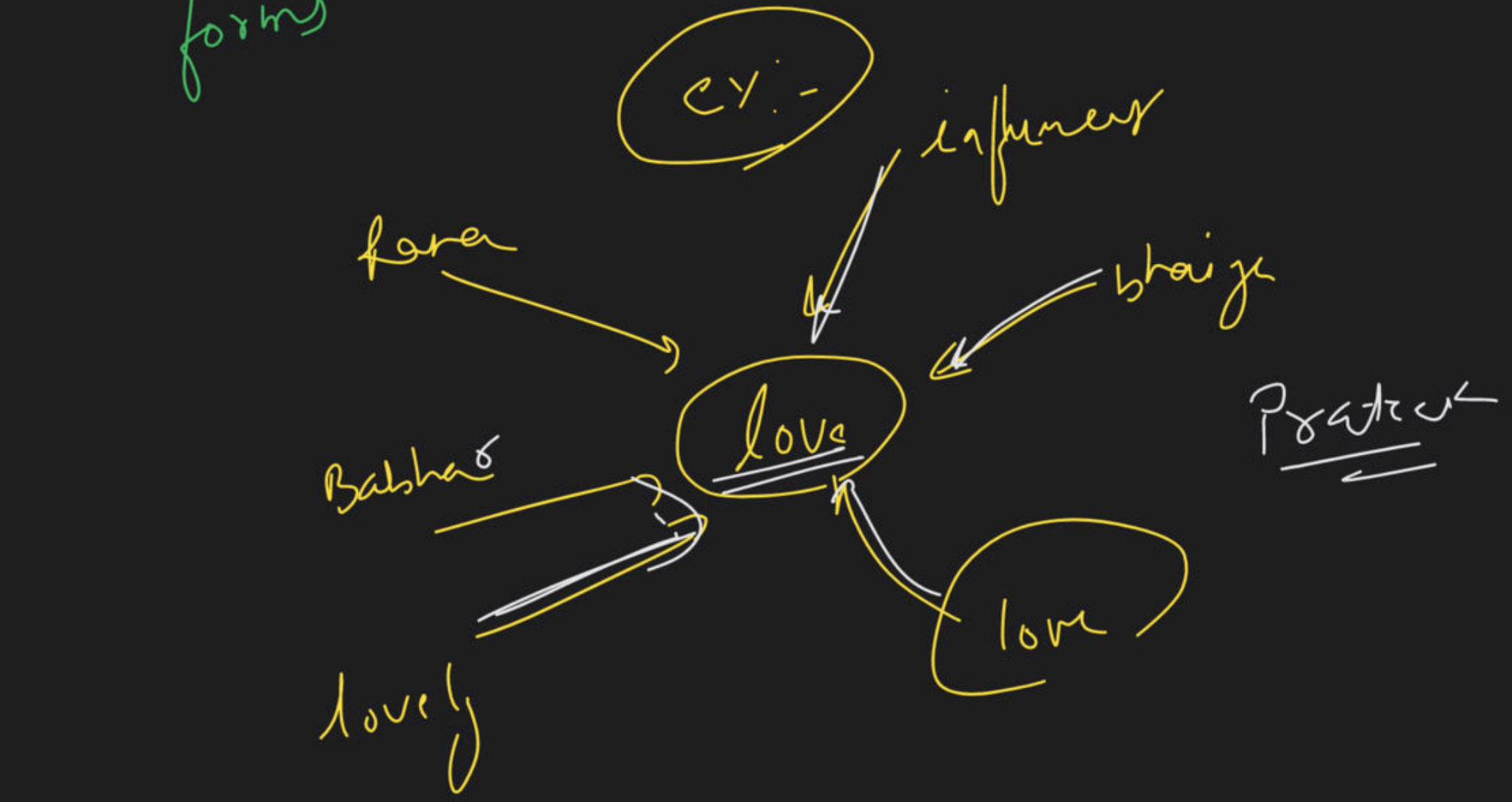
2 min Break!

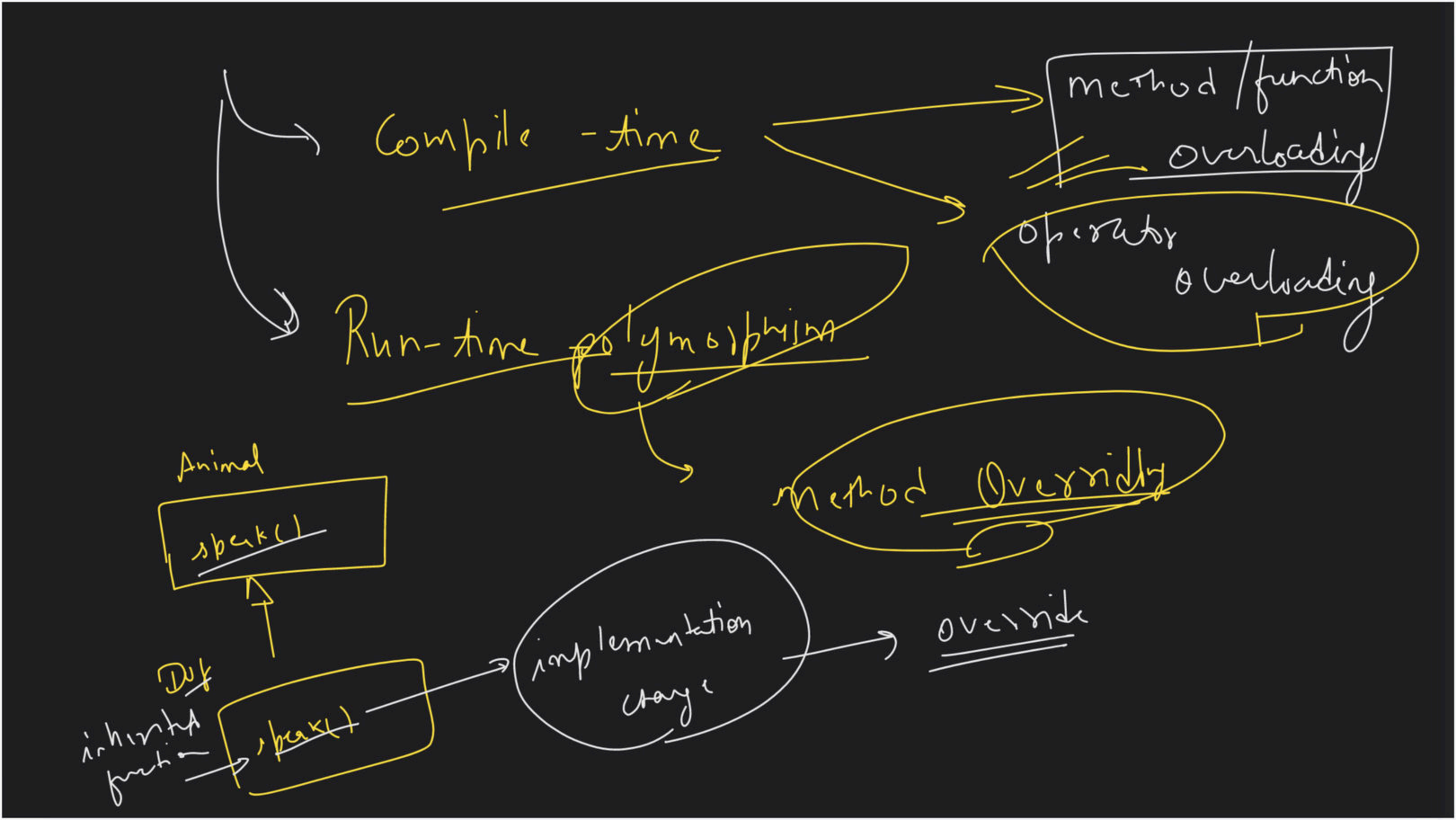
Polymorphism :-

existing in many forms

many

forms




Compile - time
method / function overloading
operator overloading
Run-time polymorphism
method Overriding
Animal
speak()
Dog
inherits
speak()
implementation change
override

→ method overloading

Math

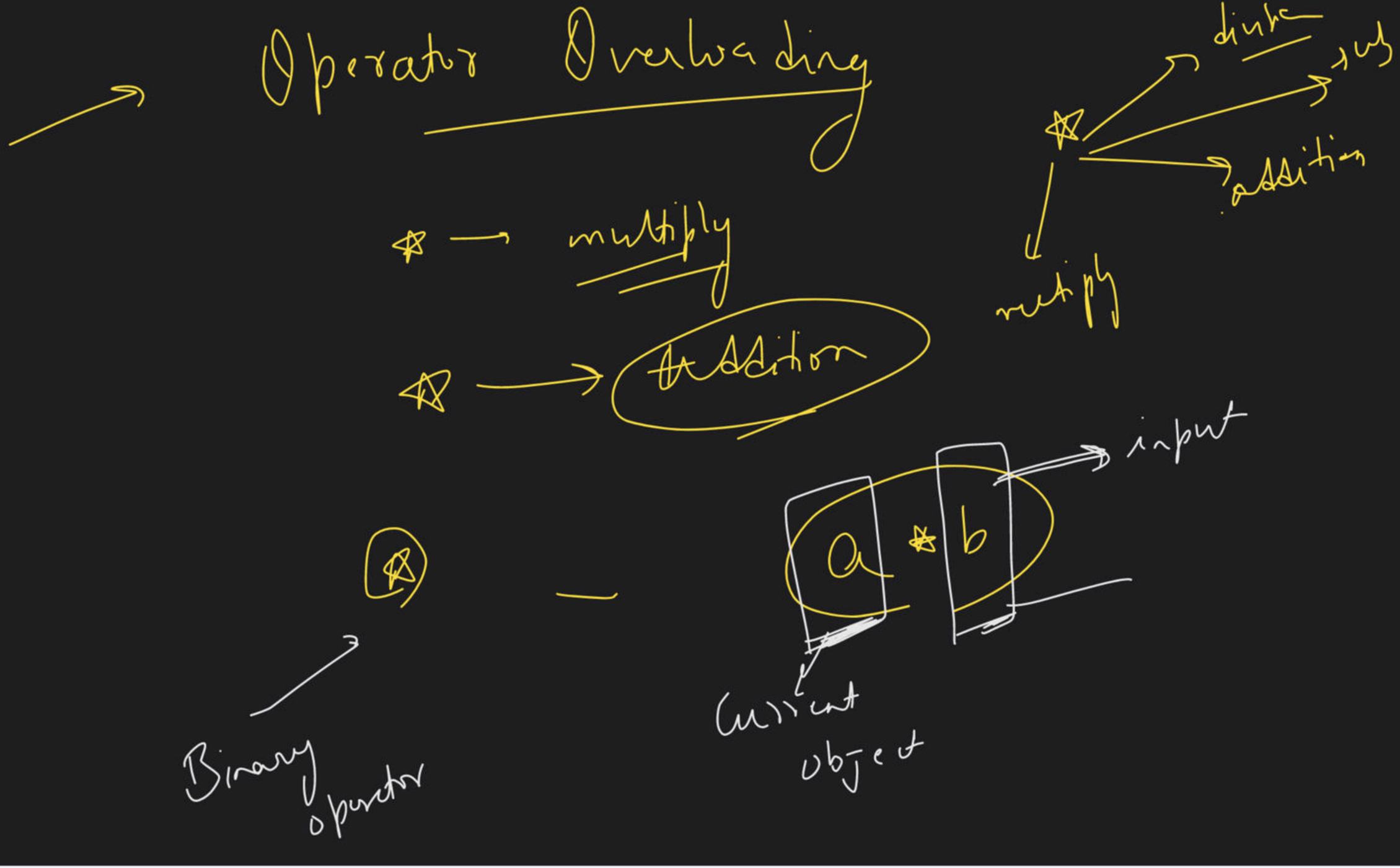
→ add (a, b)

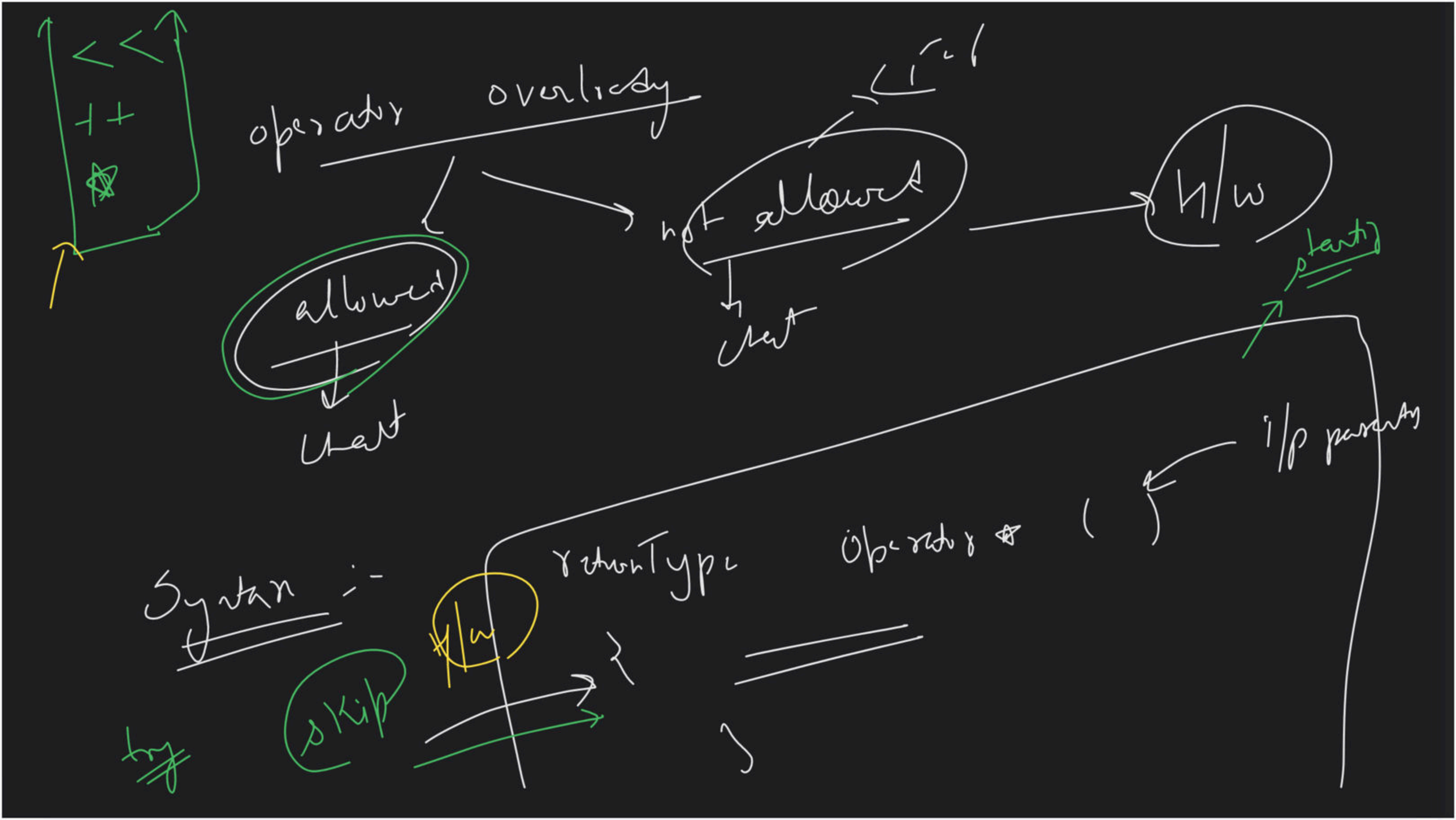
function

same name → (2) or more

Exist Kr skte → Overloading

Operator Overloading




++
operator overloading
allowed
not allowed
H/W
start
Syntax :-
returnType operator* ( )
i/p parameter
try
skip
H/W
{
}

Betrayed

12 - 13 hrs

Video

Abstraction :-

How to achieve → Access modifiers

Implementation Hiding

Only show essential stuff

→ Car

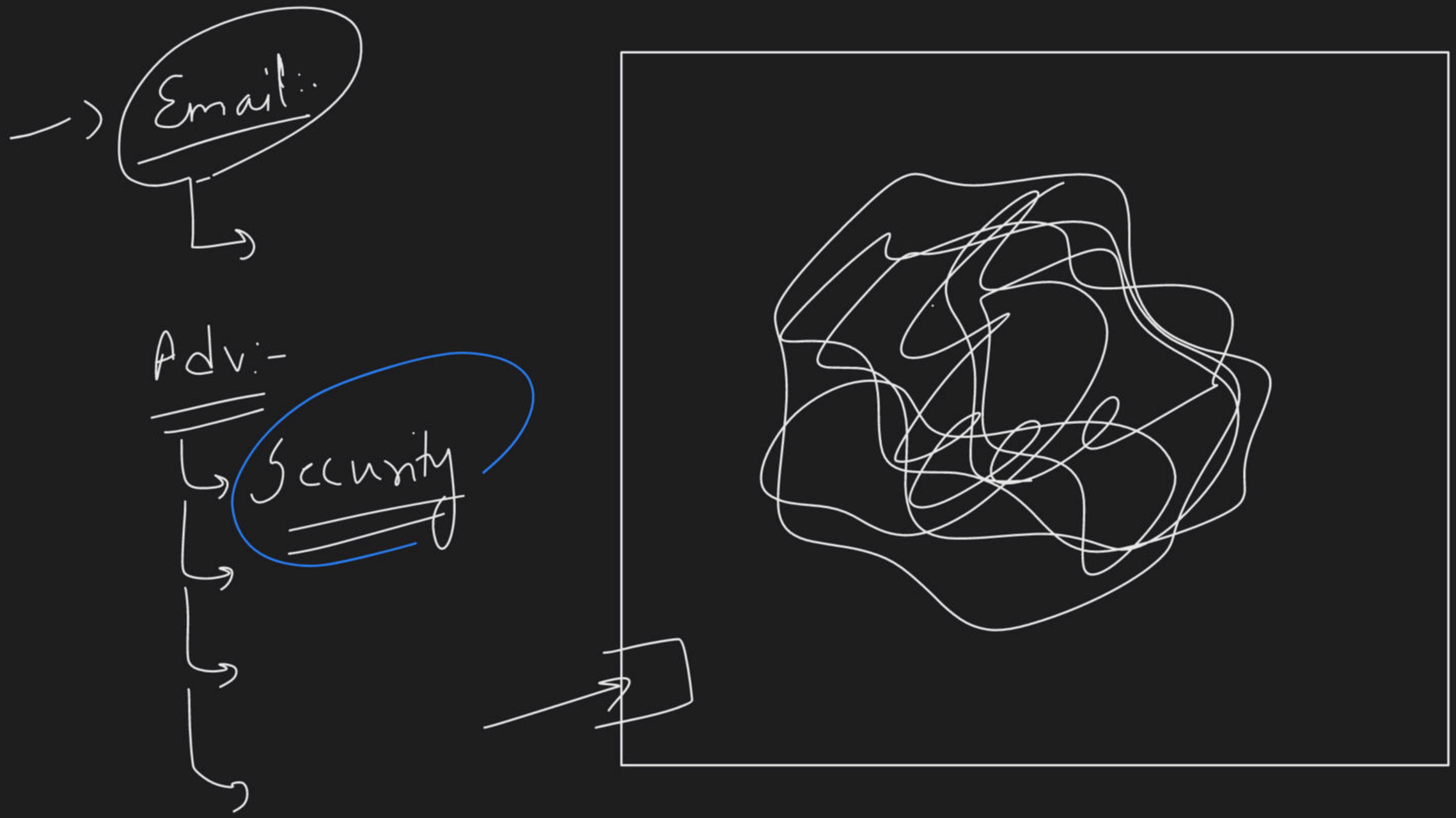
gear
clutch
accelerator
horn
4 - 5

Engine

stroke

12

ATM


Email:
Adv:-
Security

Abstraction → Implementation Problem

VS

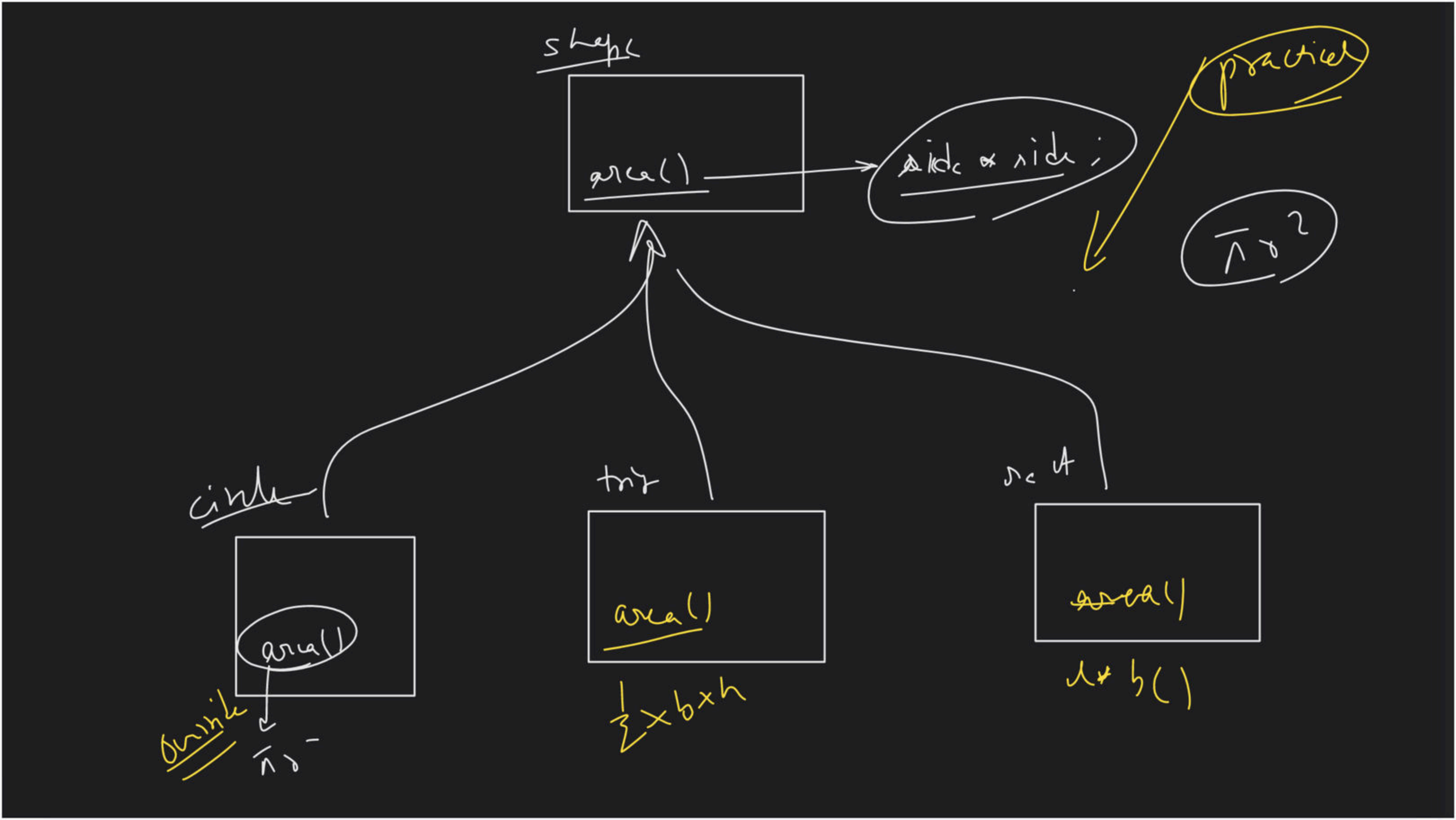
Stack Overflow

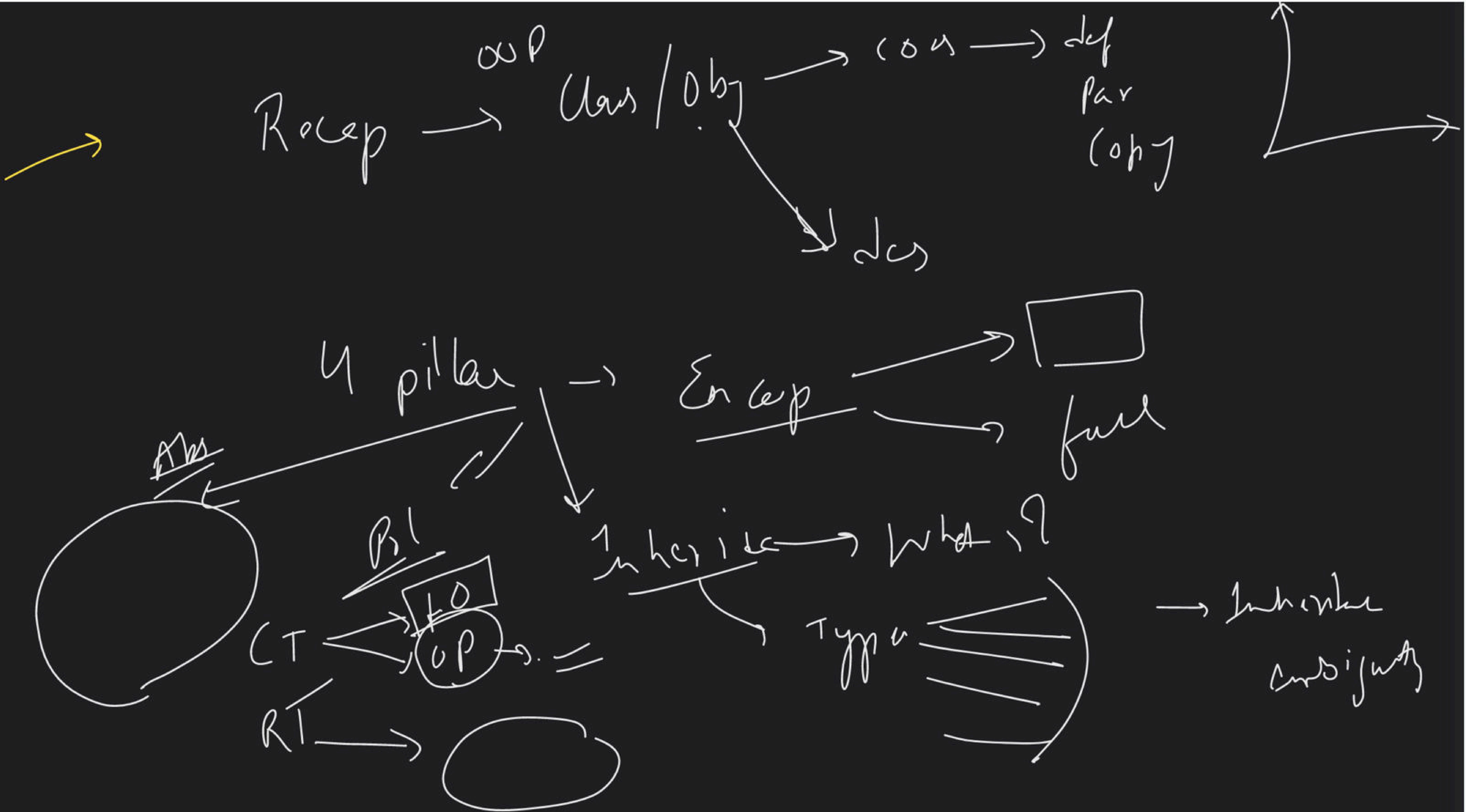
Encapsulation

data Hiding / Info Hiding

Polymorphism → Inheritance → Multiple

Diamond Problem


shape
area()
practical
circle
area()
override
tri
area()
1/2 x b x h
rectangle
area()
l * b()


OOP
Recap
Class / Obj
def
Par
Copy
4 pillar
Encap
Abs
Pol
CT
RT
OP
Inherita
What is?
Type