



OOPs - III

Foundation Course on Data Structures & Algorithm - Part I



→ Operator (II)

Overloading :- (I)

Compile-time

Polymorphism

method / function
Overloading

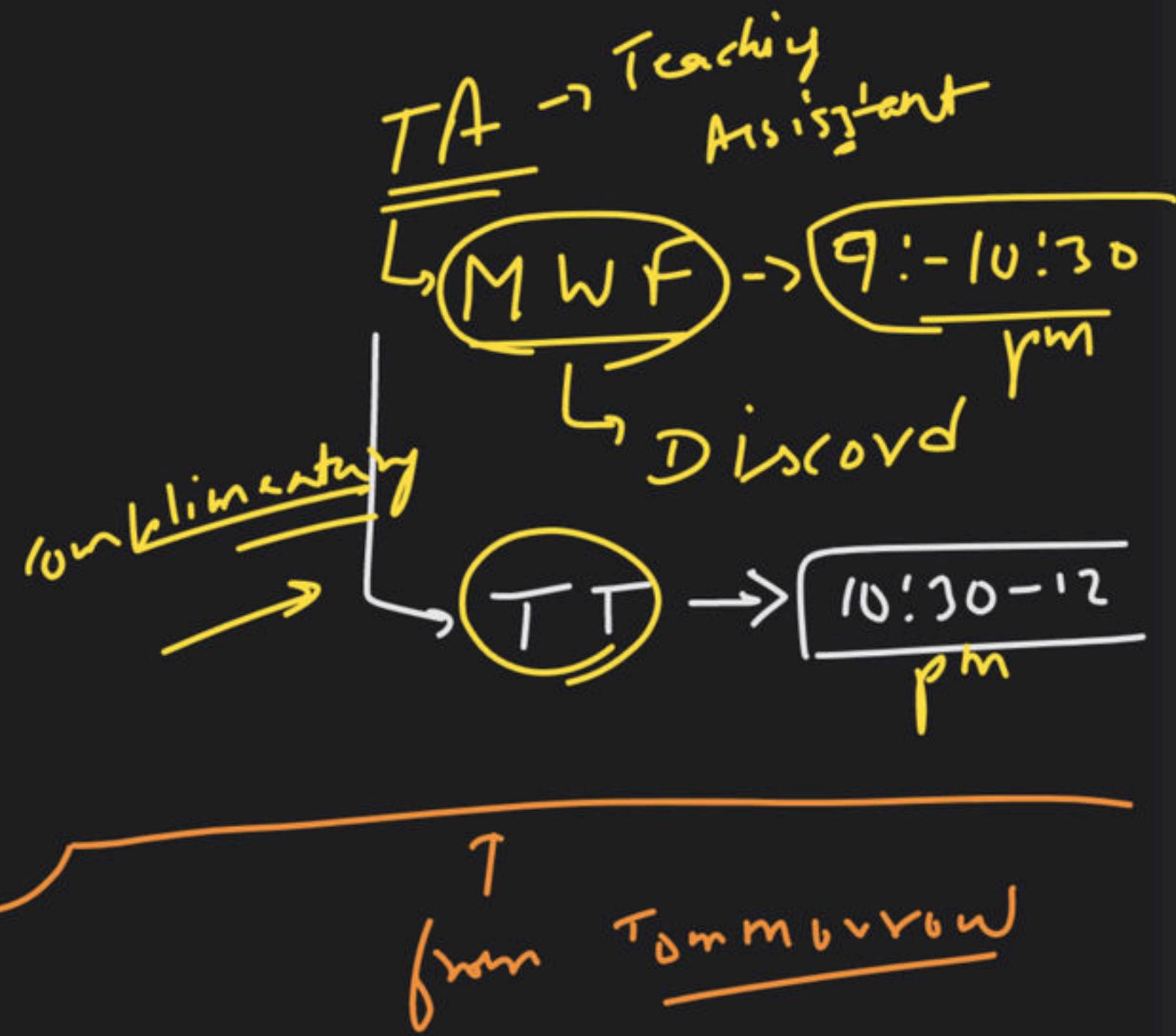
What → ?
class {

add (- , -)

ass (- , -)
ass (- , -)

wt (- , -)

L)



Operator Over.

operator
 \oplus + & i

\oplus → addition

a + b

print → Or bhaiya kya haal hai

{
→ unary -, ++ . why?
→ binary → +
→ >> outside class
why?

72 14

Syntax:

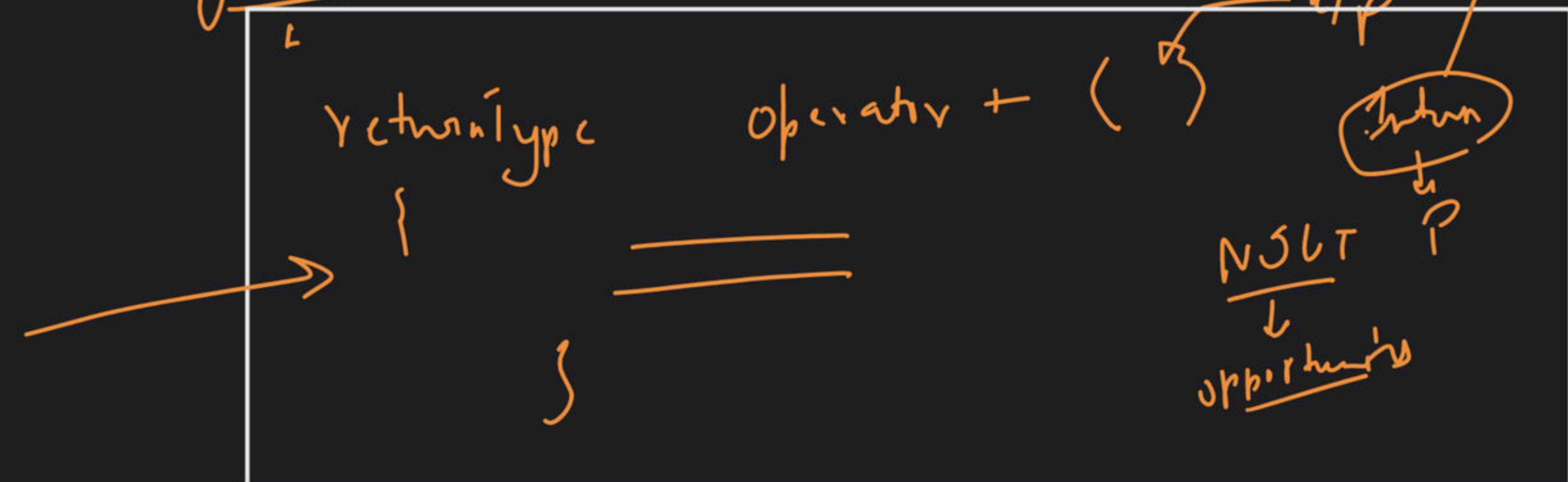
Stack Overflow → Laptops

operator + (

return
tip

NST

operator



why?

arg - changes

\rightarrow
 \rightarrow

① \rightarrow $H/S \rightarrow$ operators which cannot

be overloaded

↑

why?
P
BASIC

R A J A

SUNDAY

operator

member function

\rightarrow unary \rightarrow single operand \rightarrow $++$

class

{
}
}

\rightarrow binary \rightarrow double \rightarrow $(+)$

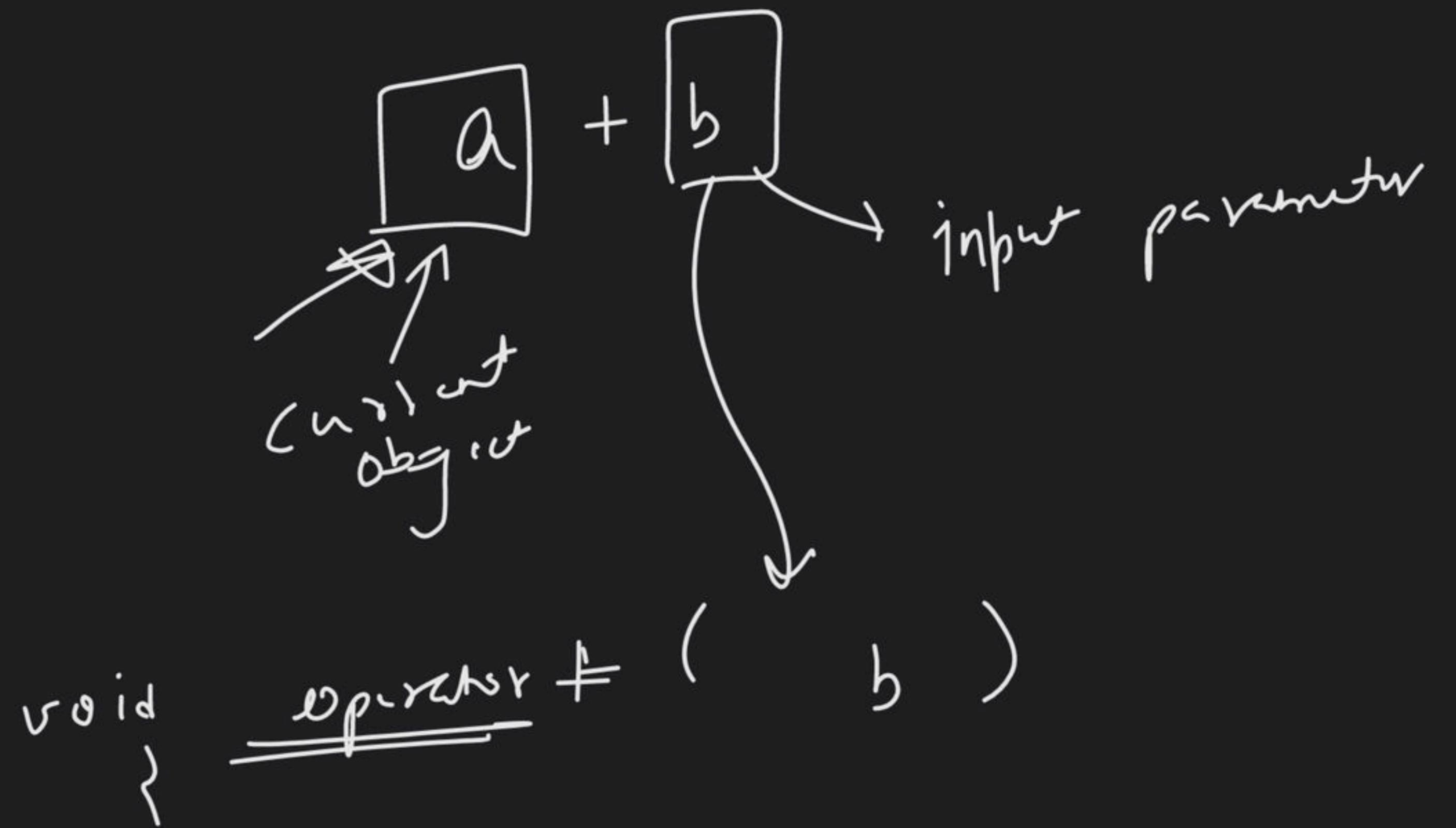
\rightarrow $<<$ () \rightarrow left/right shift

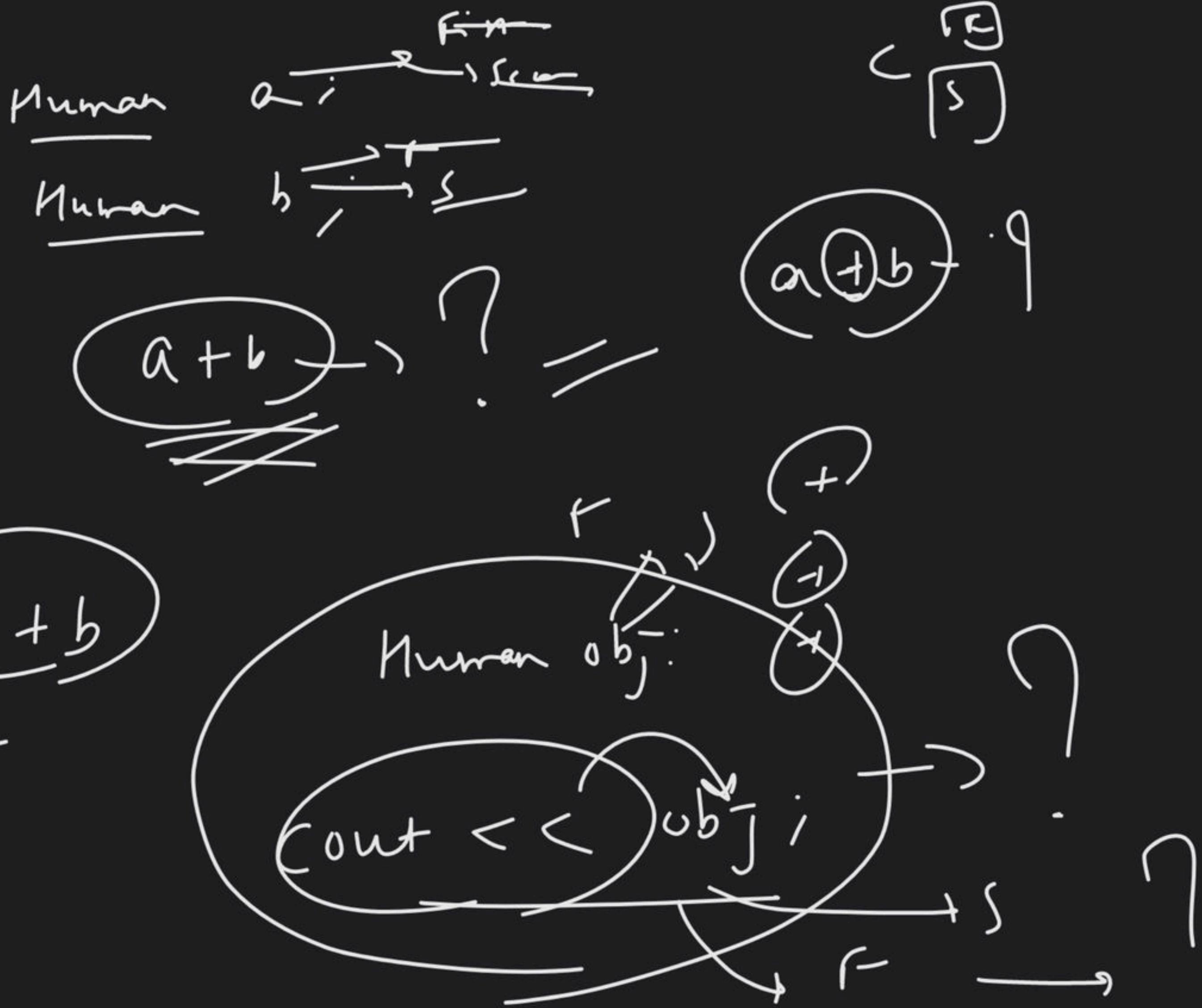
non-member function \rightarrow

outside class
why \rightarrow

$<<$

$>>$



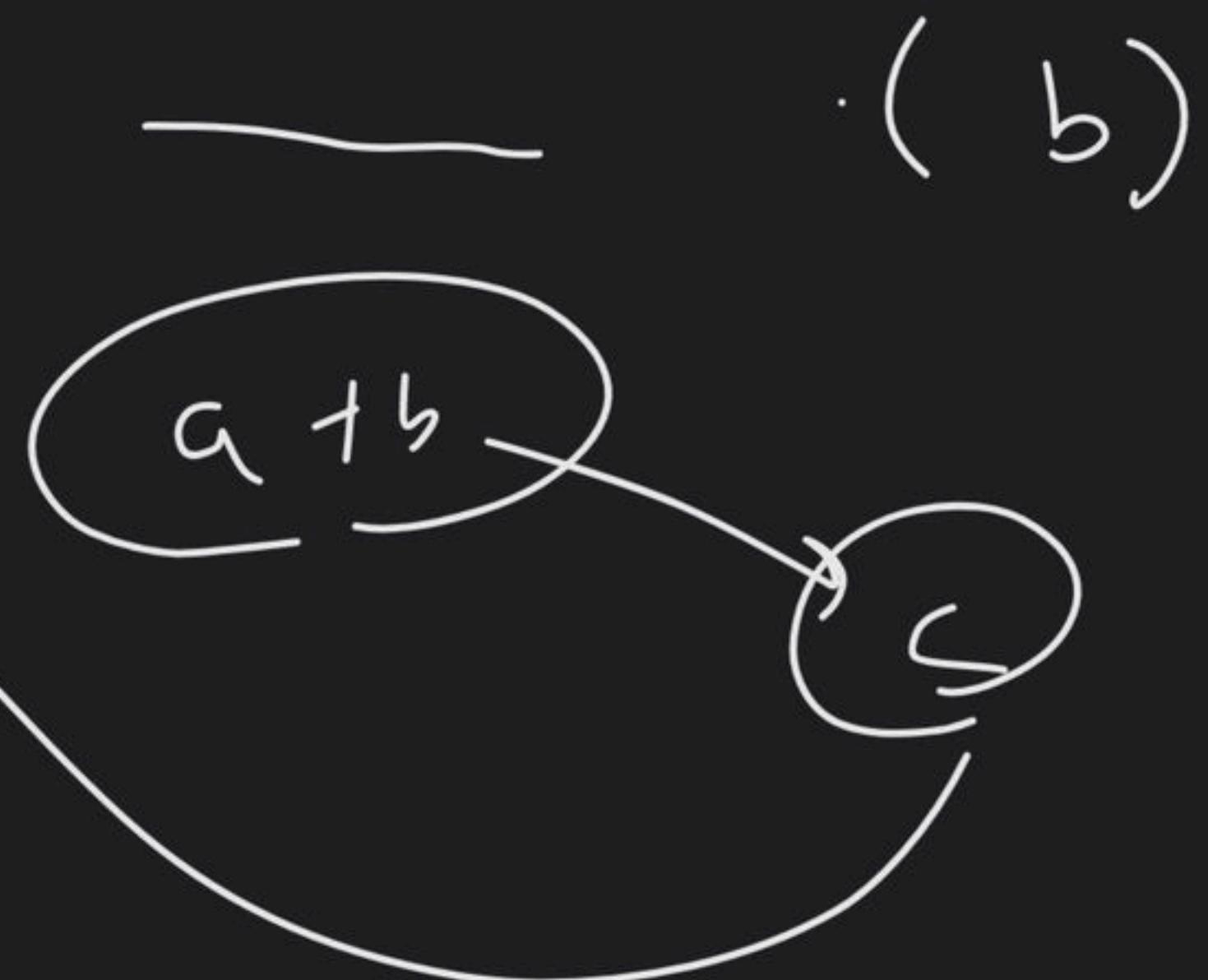


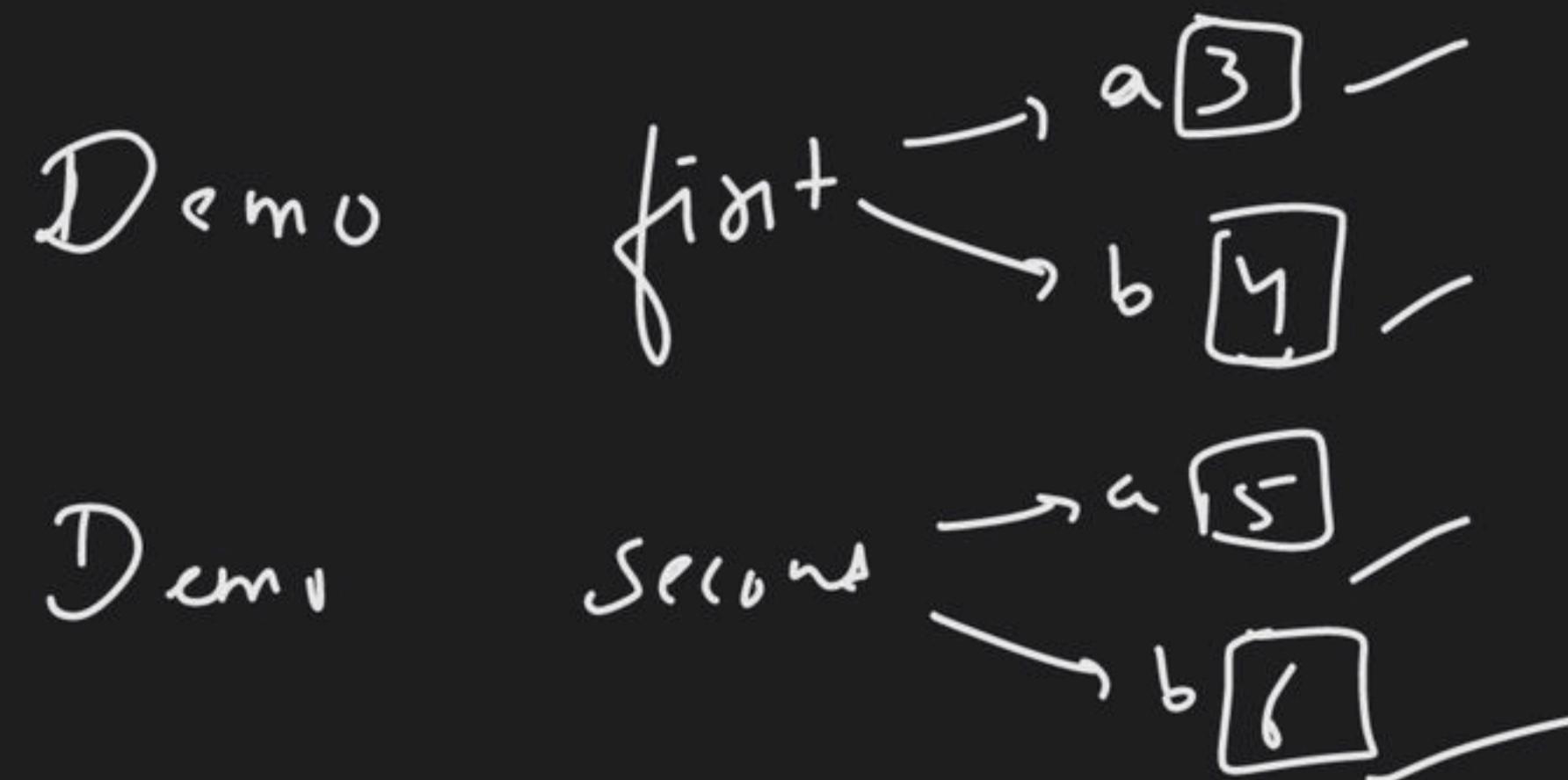
Binary Operator

$a+b$

Count << Obj

Demo





first + second
jtry → 3416

Demok

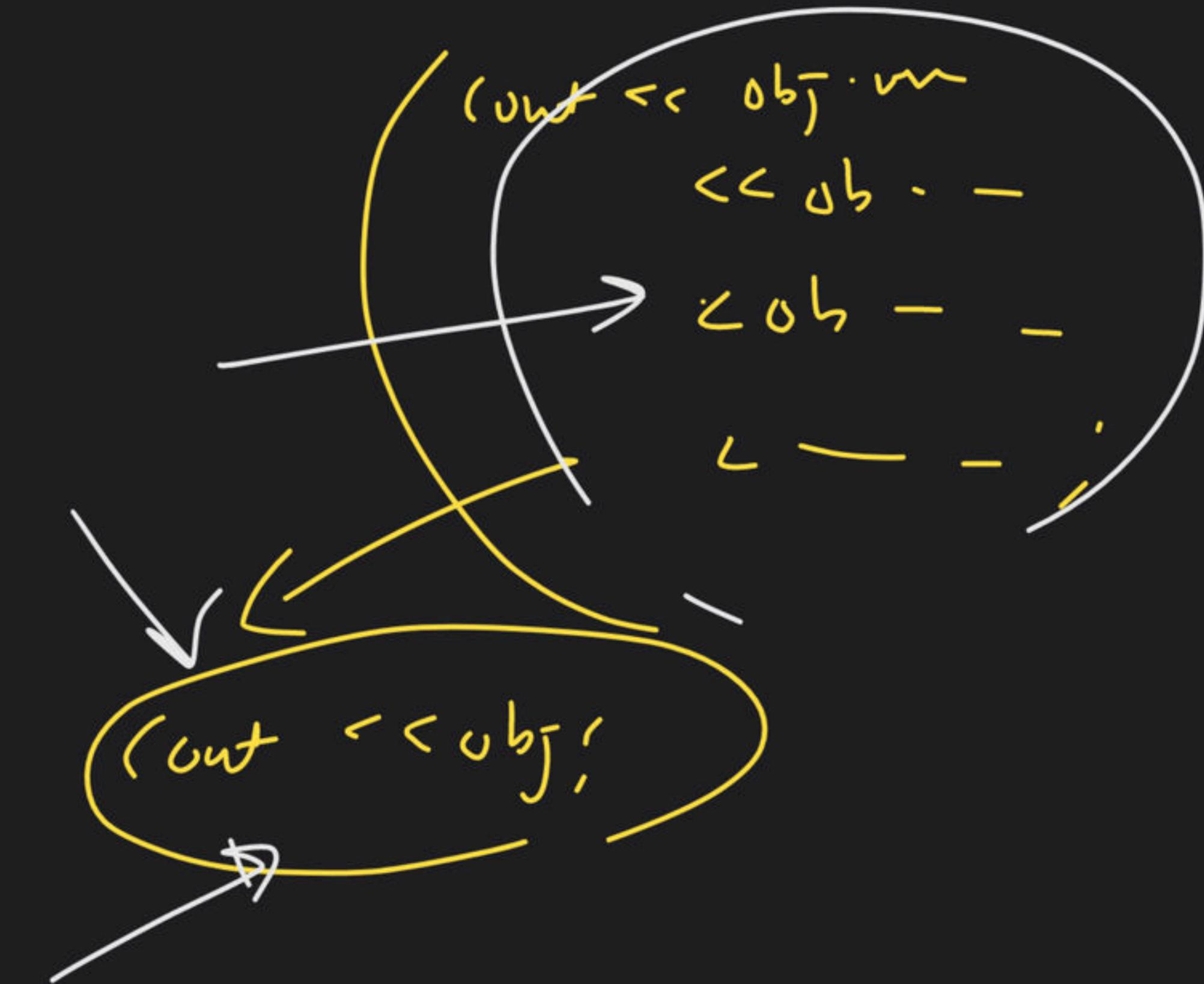
Demok

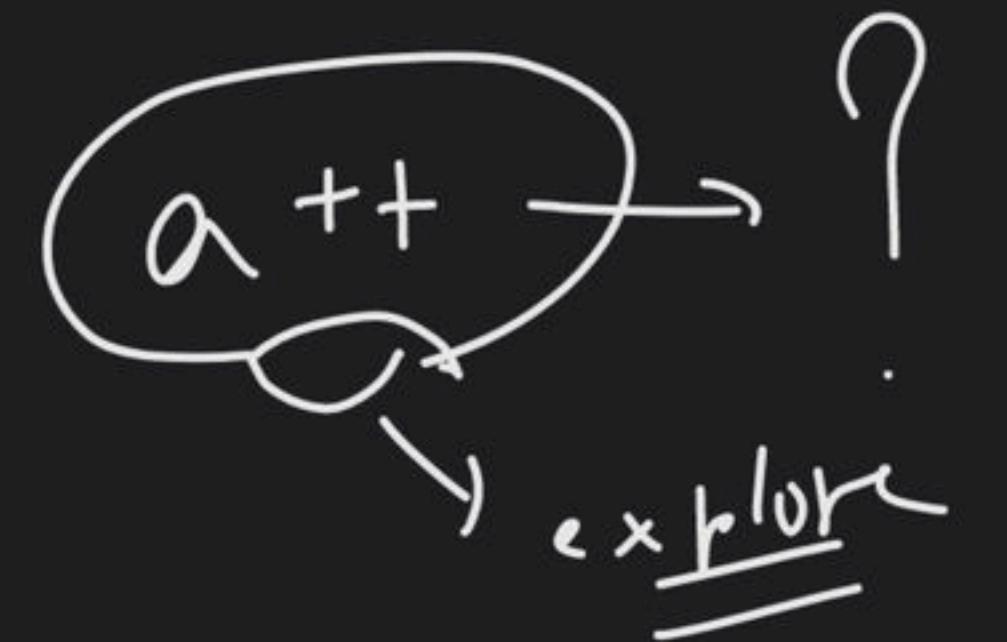
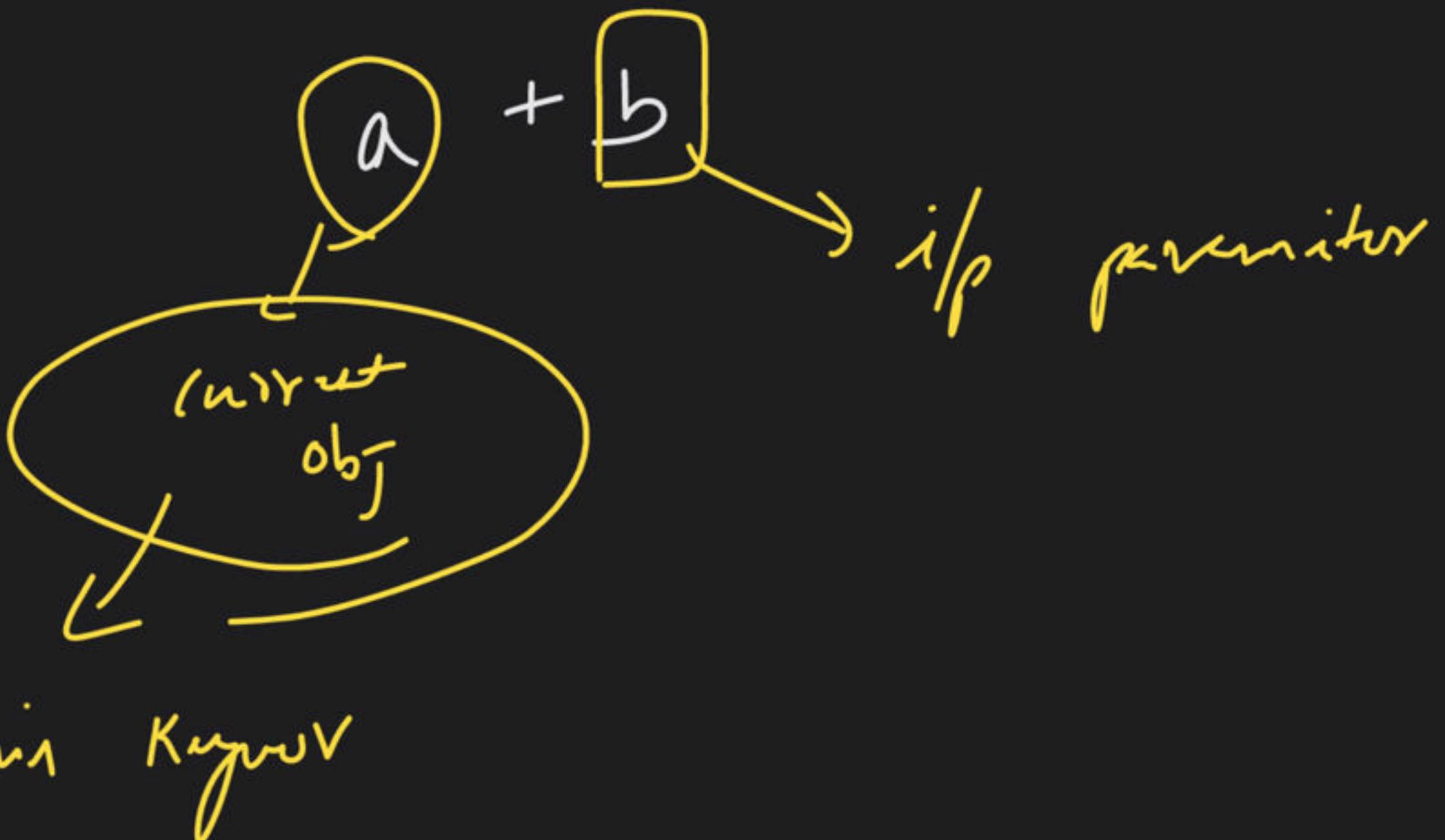
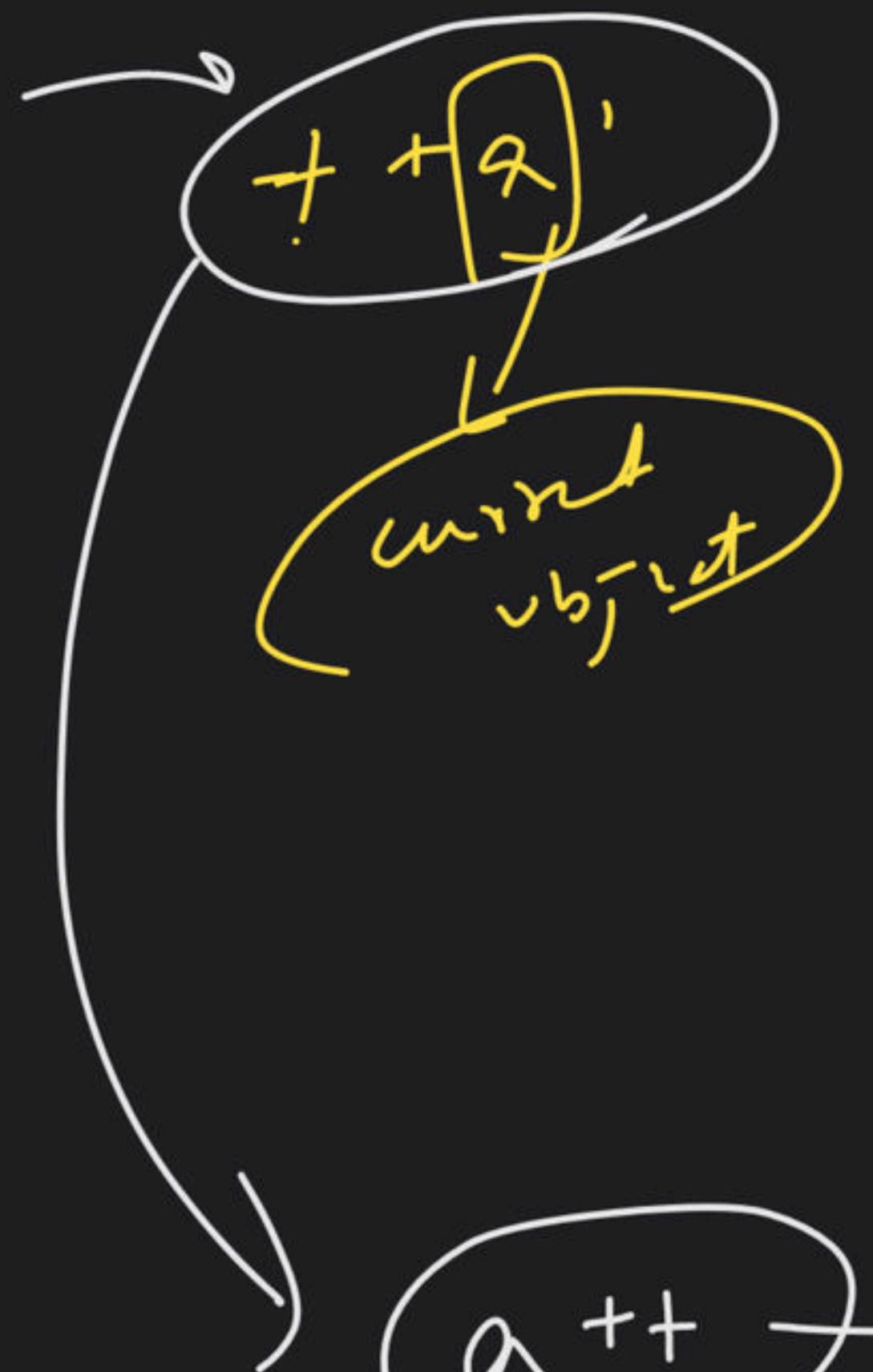
(+ D)
c9YDY

string operator + (Demok temp)
{
strn) ^ = " "
→ . pwh-back (_)

} return ;

```
class  
{  
    + user  
    - id  
    phon
```







Demo



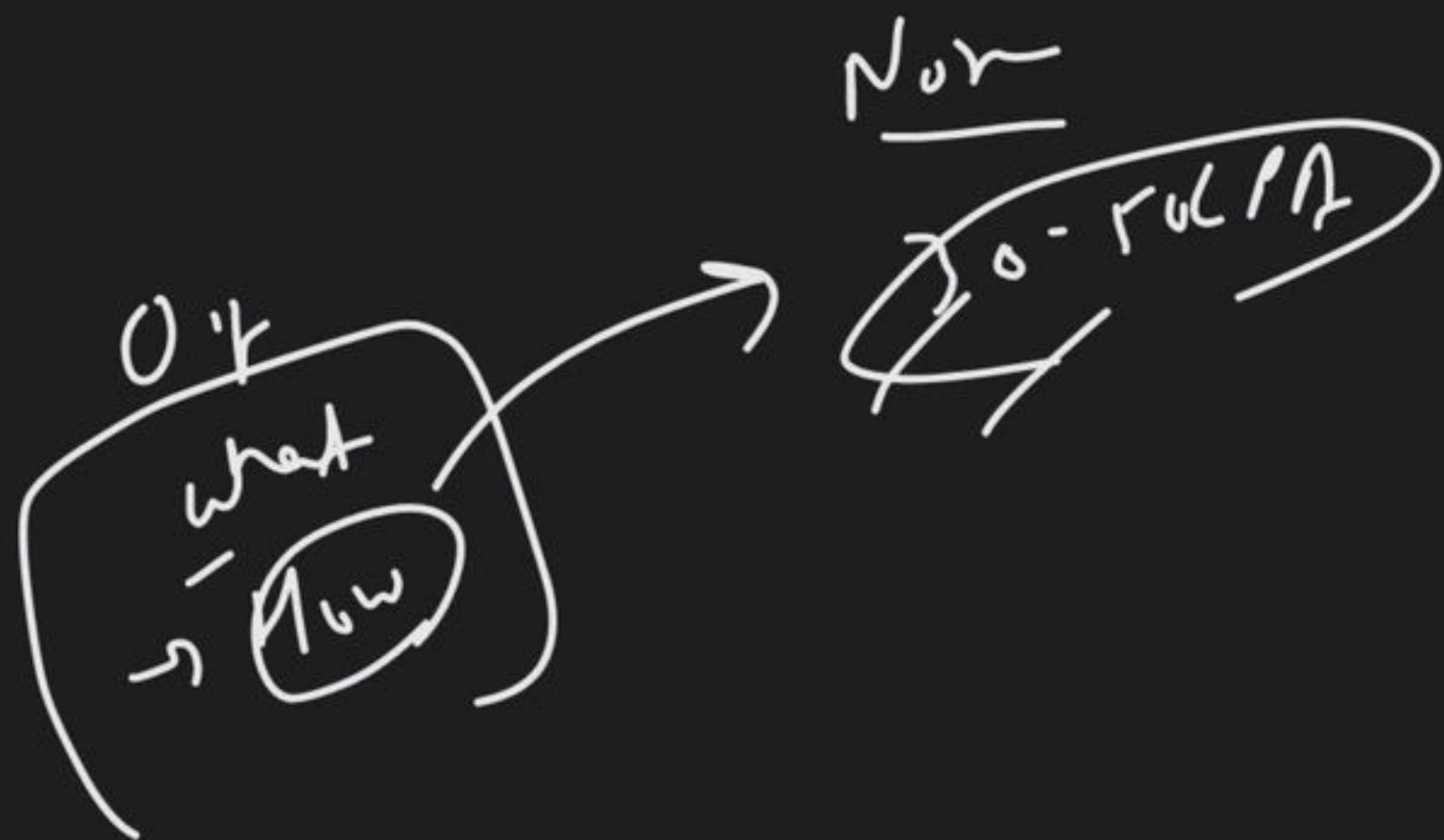
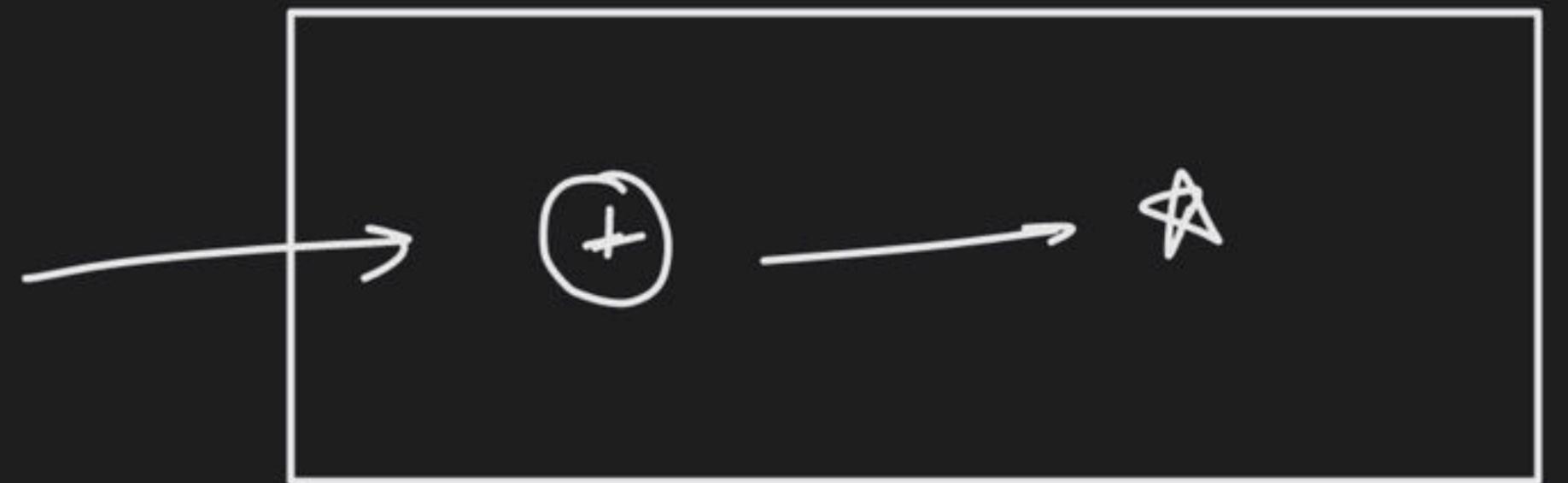
Demo obj;

cout << obj;

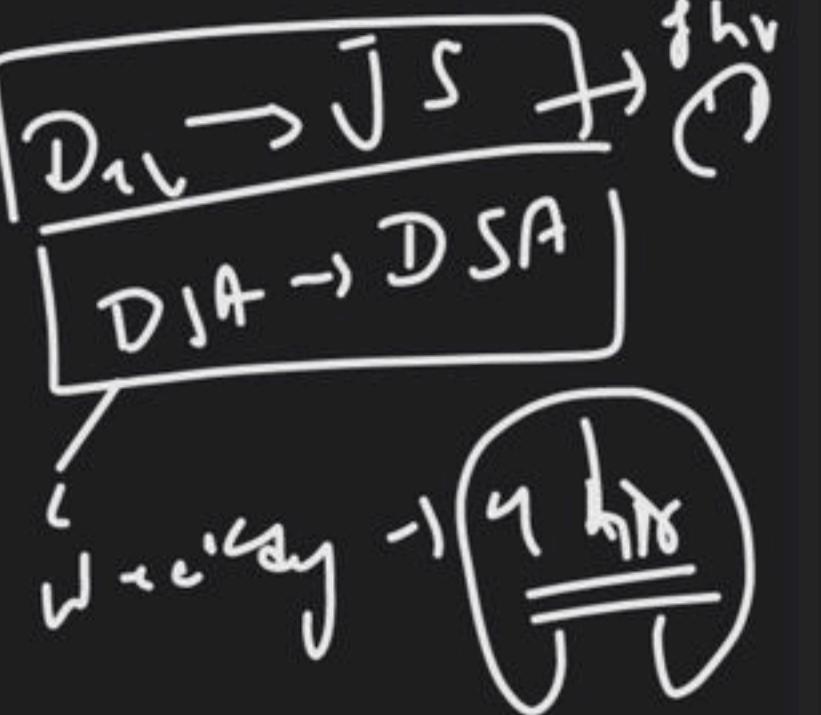
$a + b$
if 1/p row
arr
obj

count << obj

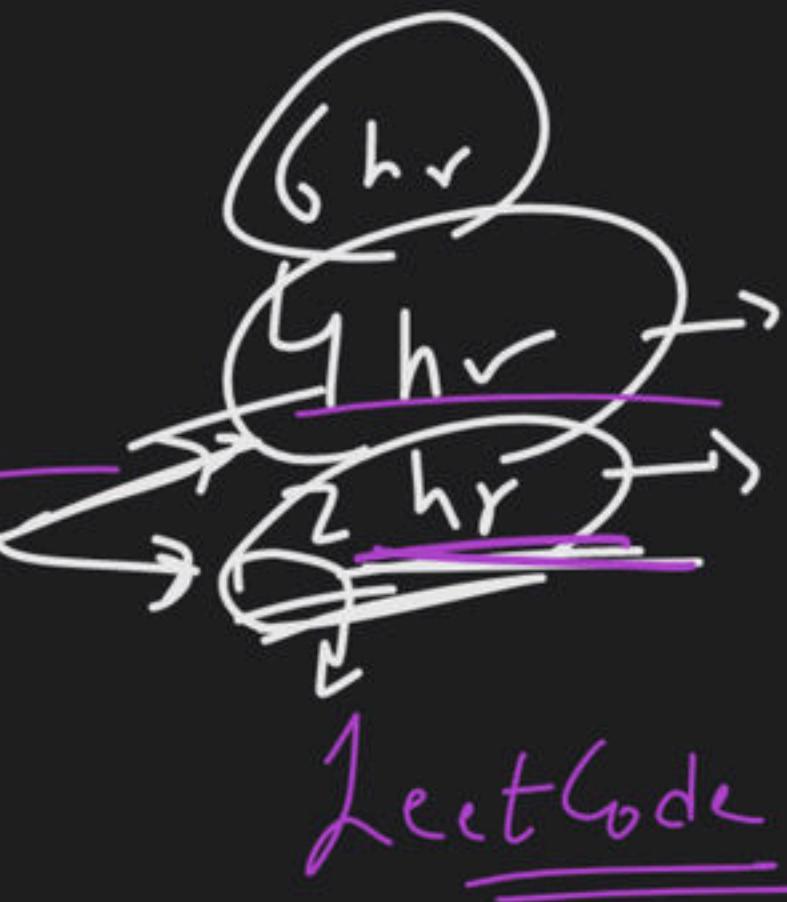
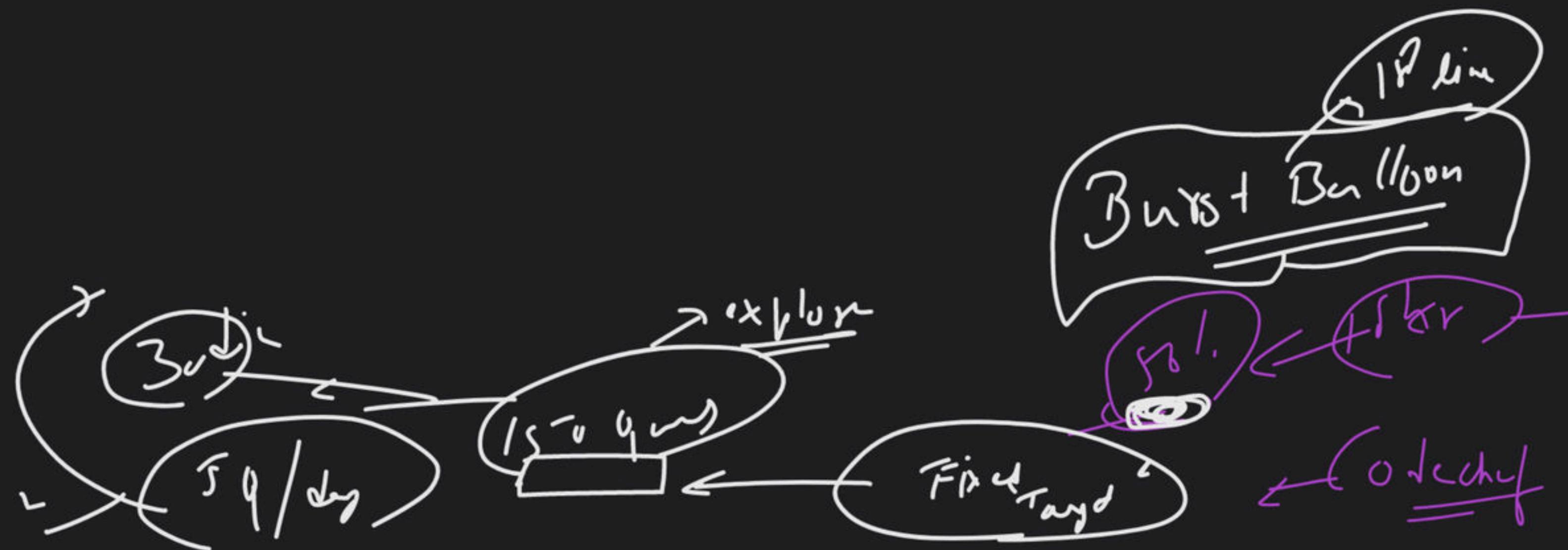
while
class index

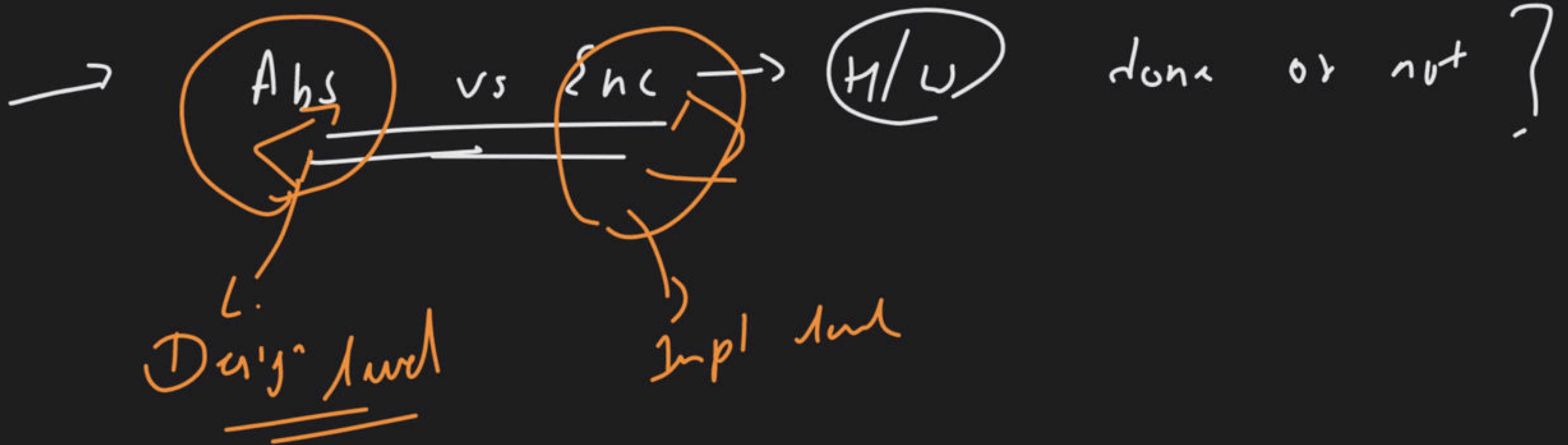


→ Abstraction vs Encapsulation → ↗



By Jas

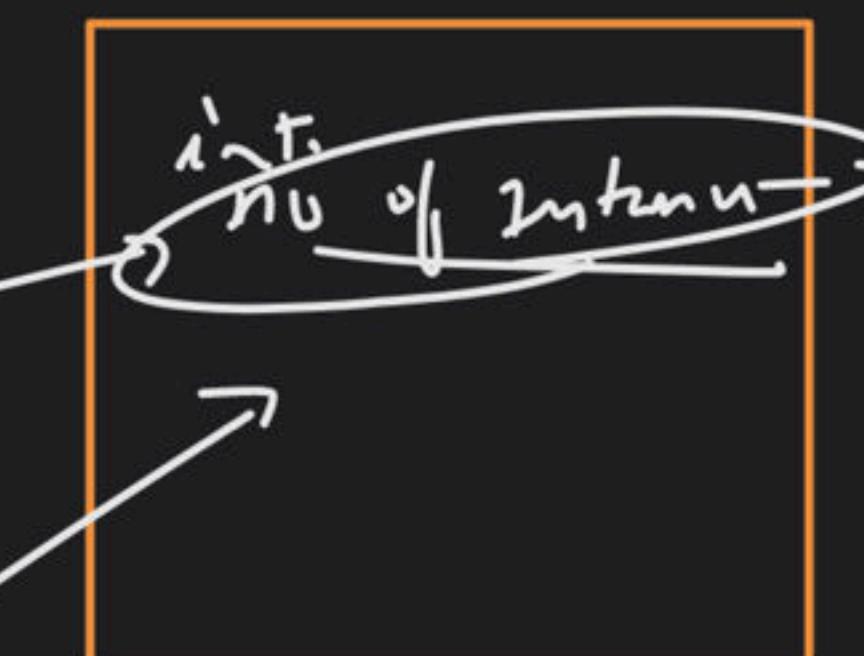




~~Static Keyword :-~~

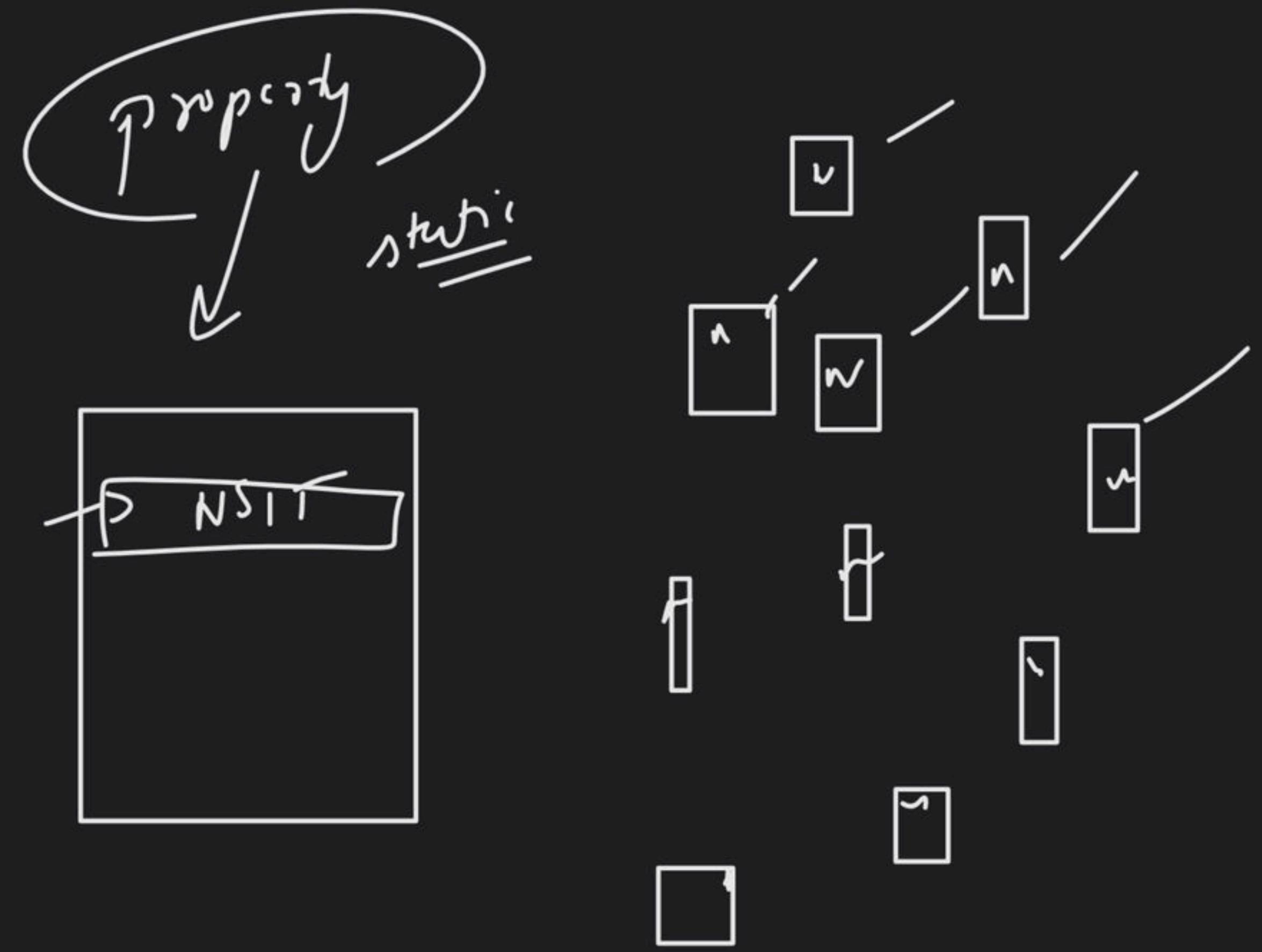


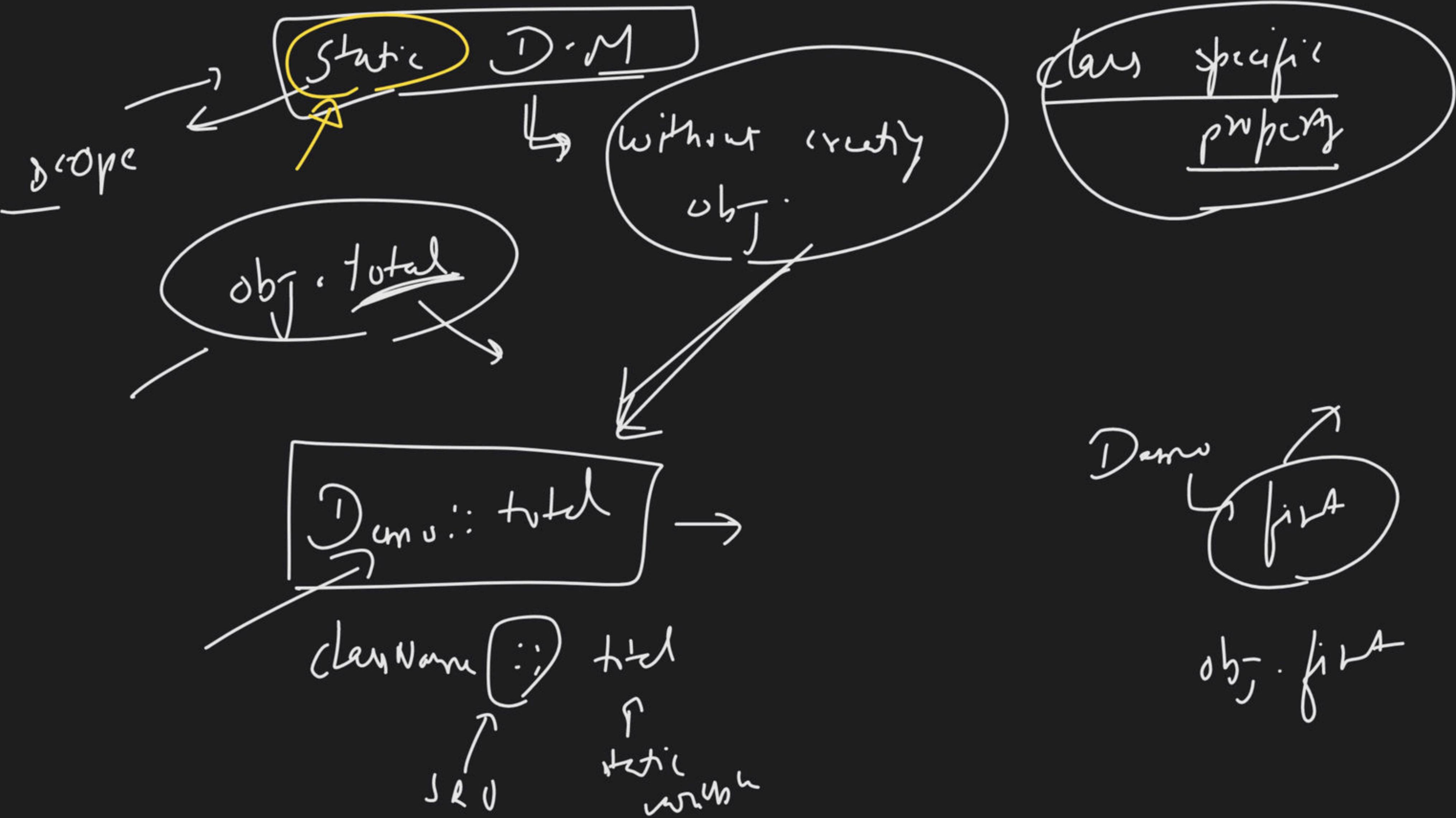
~~no. of Obj.~~ → OBT
→ CLM! → no. of



~~static int a;~~







Variable → parts



Memory

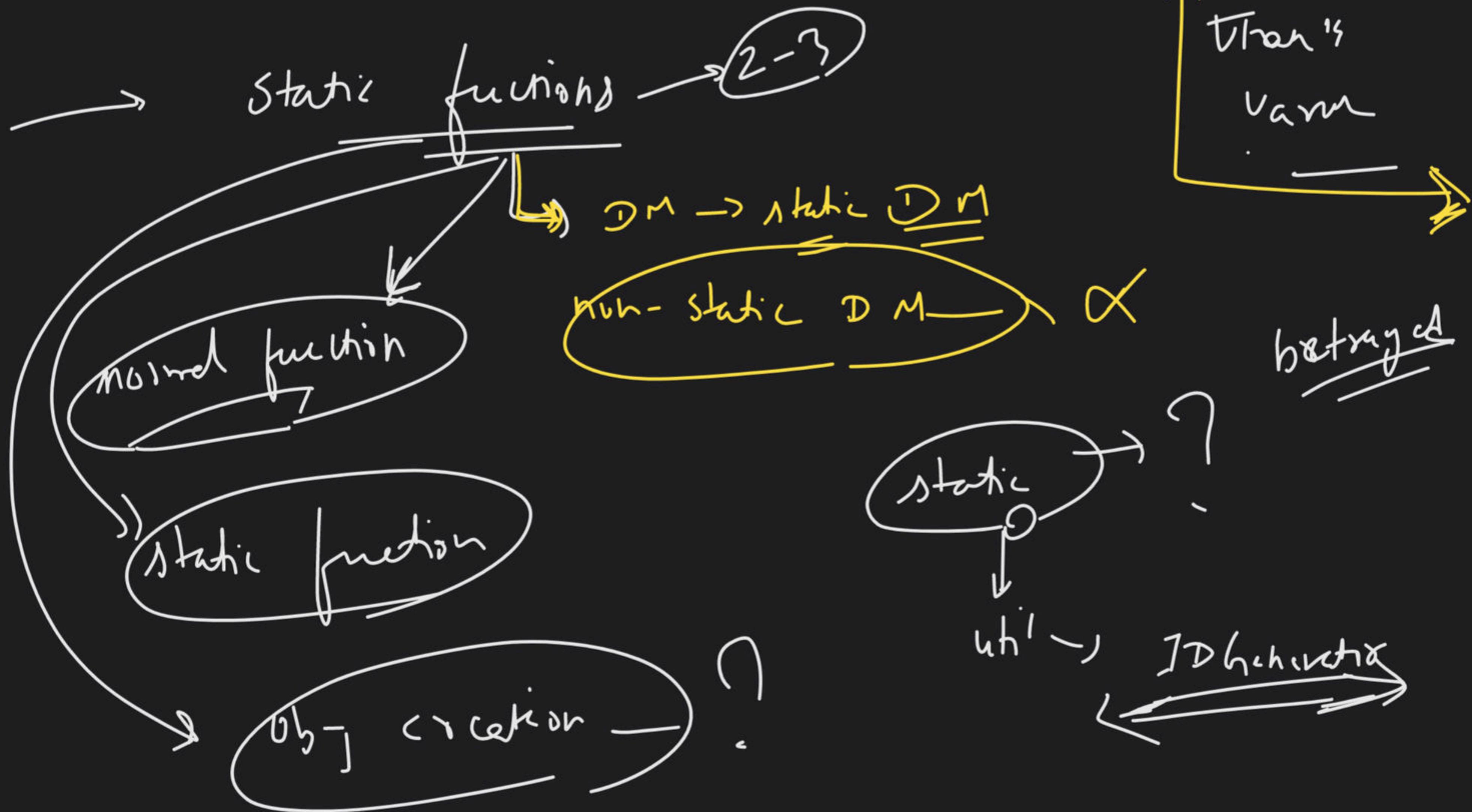
code section

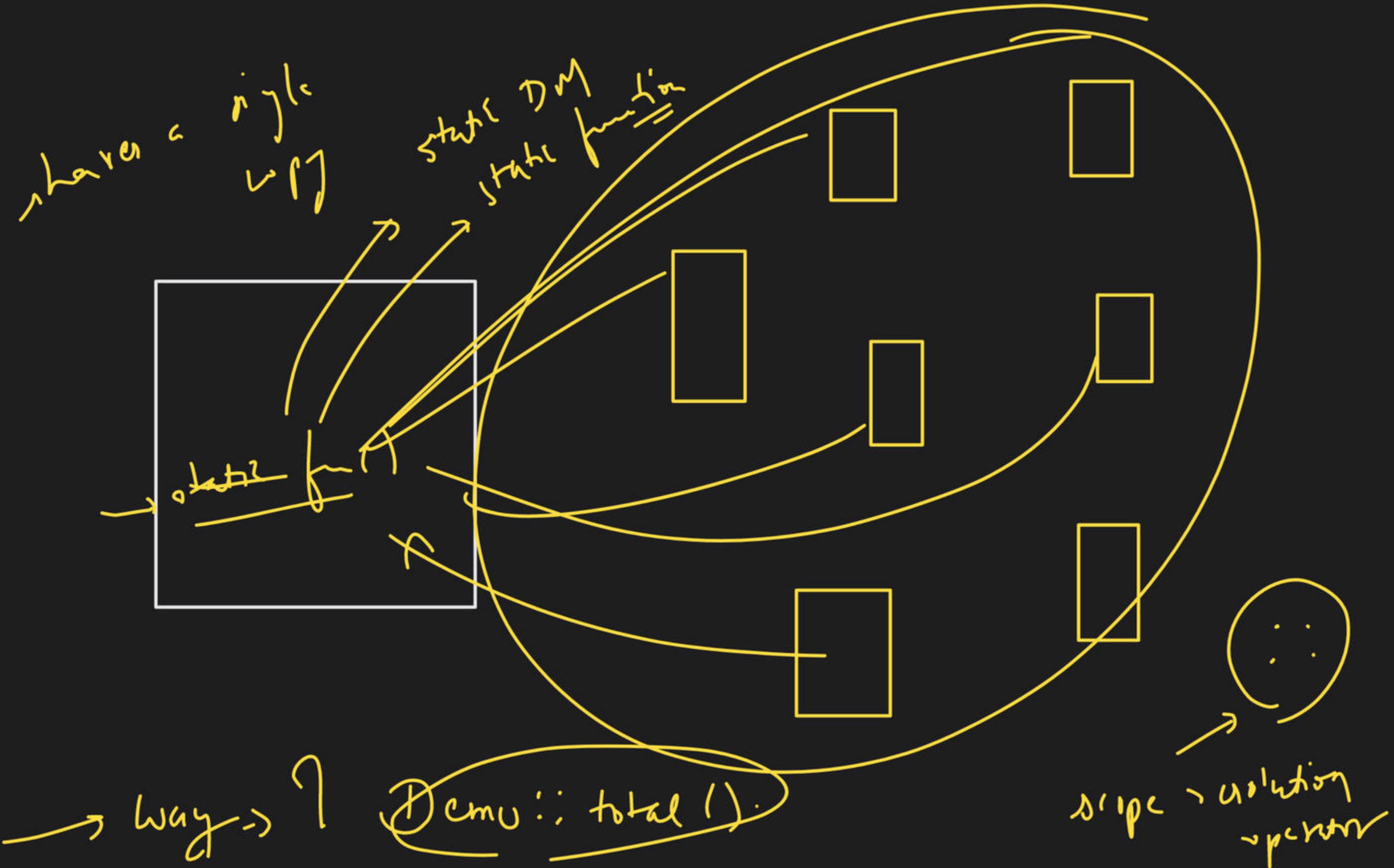
data section

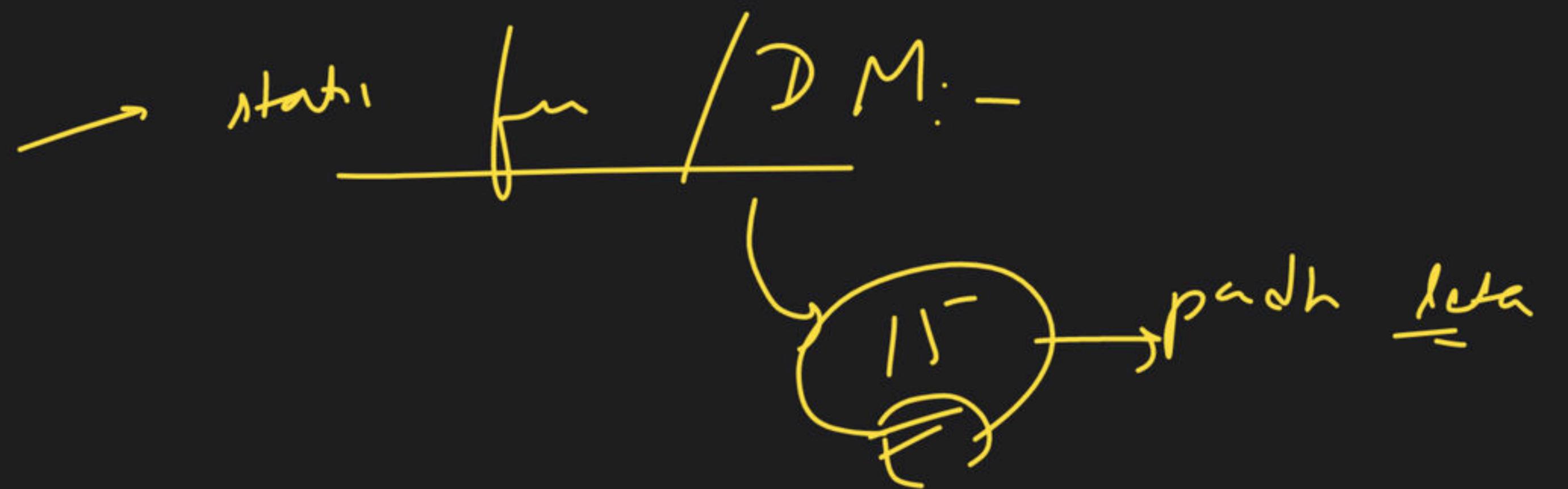
data section

static / global

C. verify

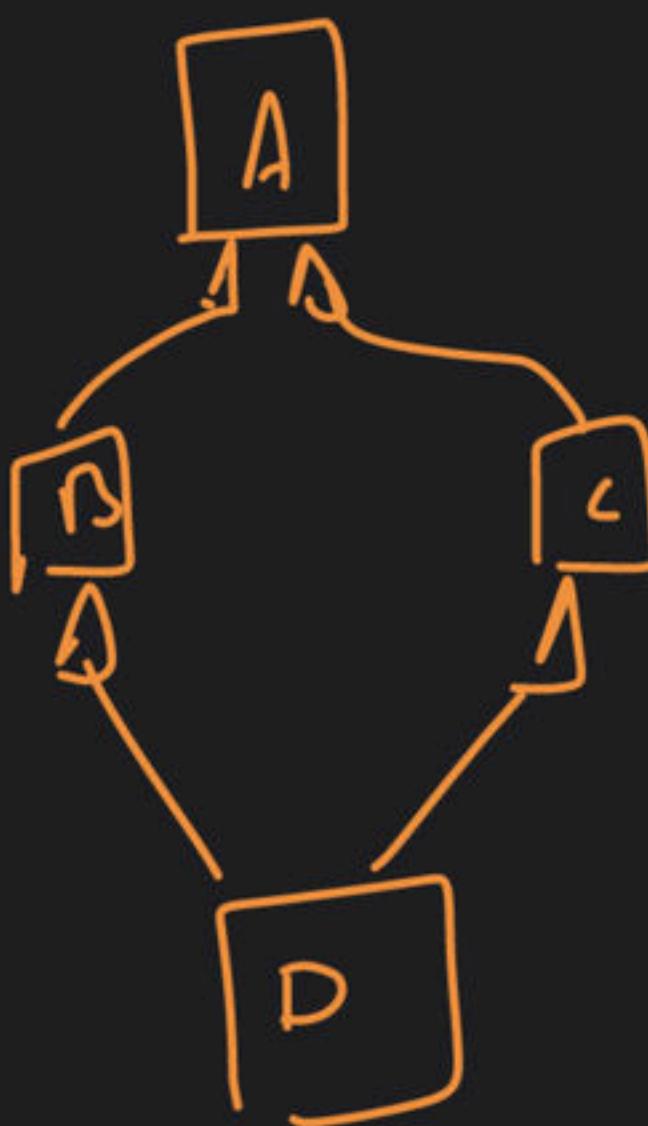






→ Diamond Problem → ?

A diagram showing a horizontal line with a yellow arrow pointing to it from the left. Below the arrow is the word "Diamond". To the right of the word is the text "Problem → ?". Below the text are two parallel horizontal lines.

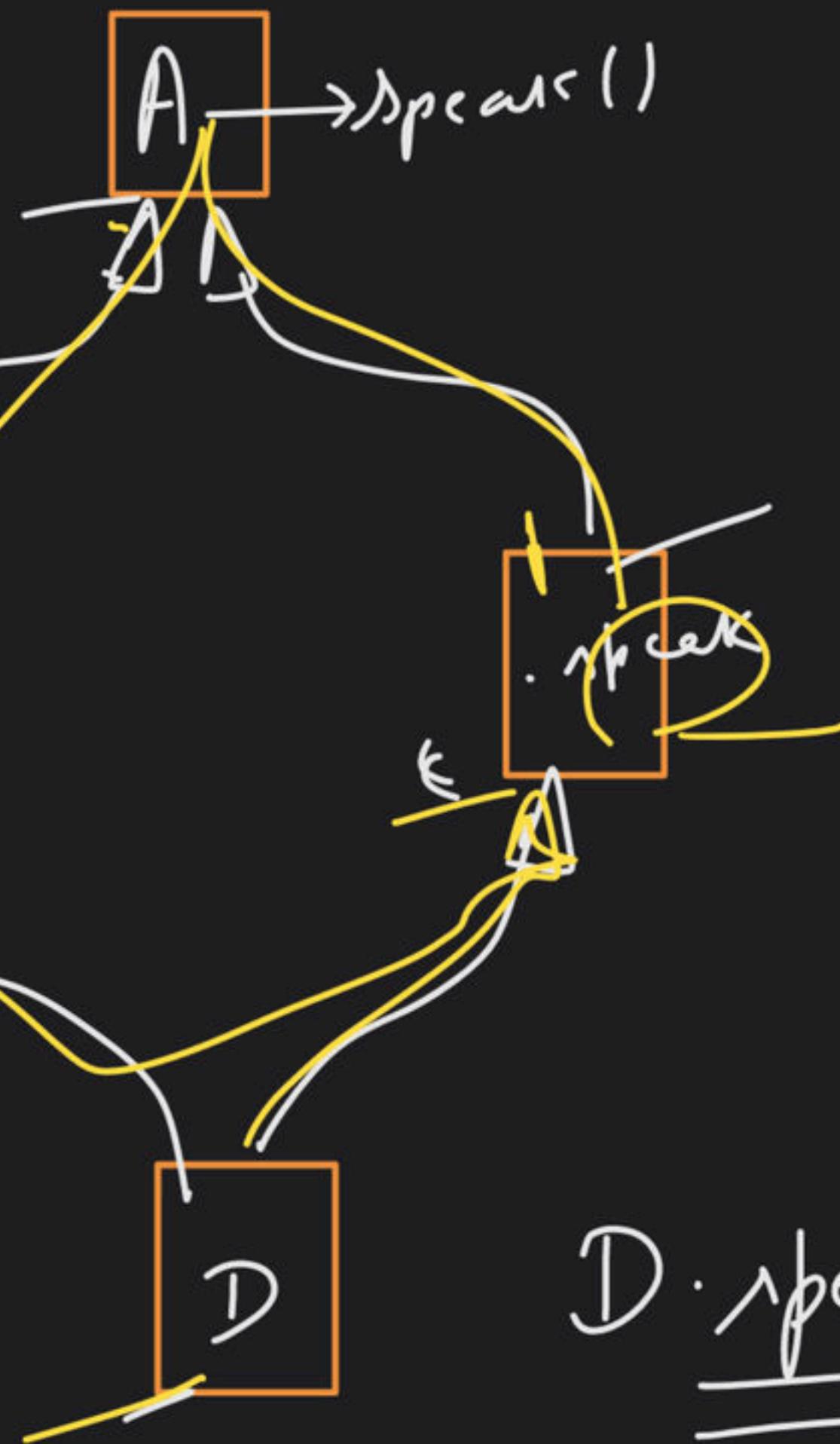


```
class A  
{  
    ...  
};
```

```
class B : virtual public A  
{  
    ...  
};
```

```
class C : virtual public B  
{  
    ...  
};
```

```
class D : public B, public C  
{  
    ...  
};
```



solve \Rightarrow
solve this

\dots

$B::speak()$

$C::speak()$

$D::speak()$

\therefore / Virtual

→

abstract class

class {

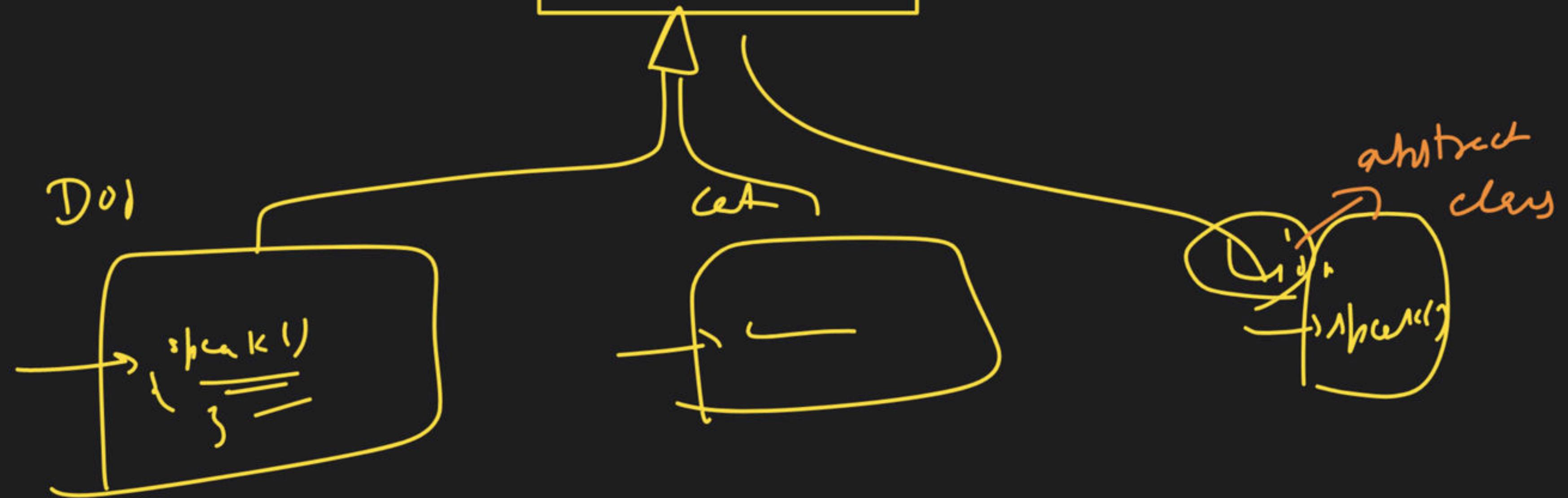
attribute | plus
virtual funct.

virtual int fun() = 0;

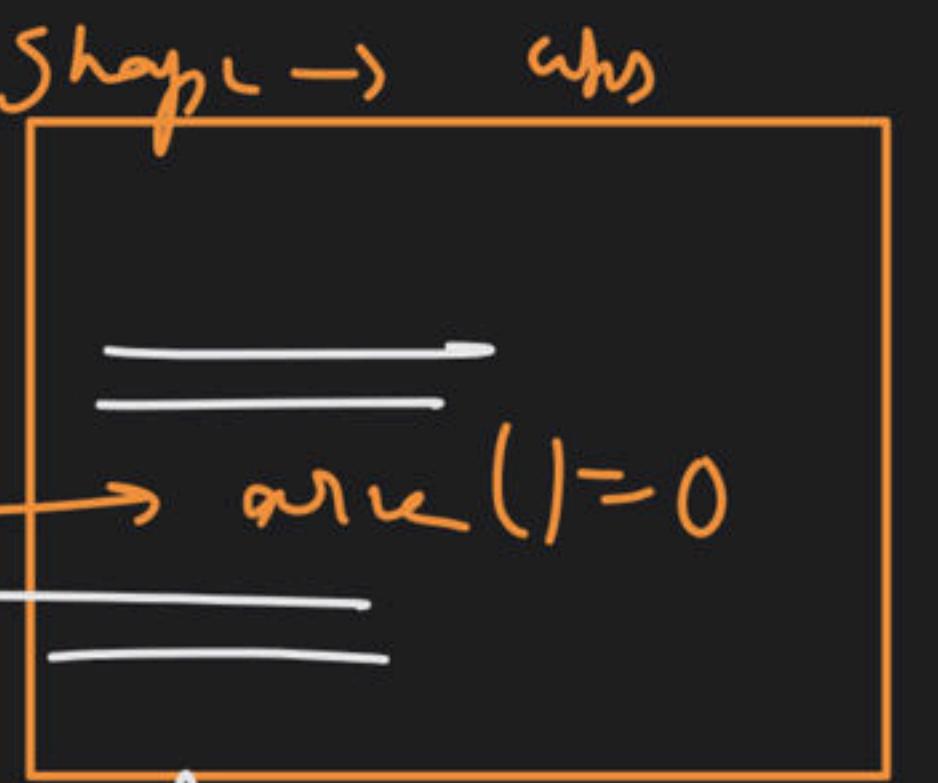
}

shear() -

Animal (abstract class)

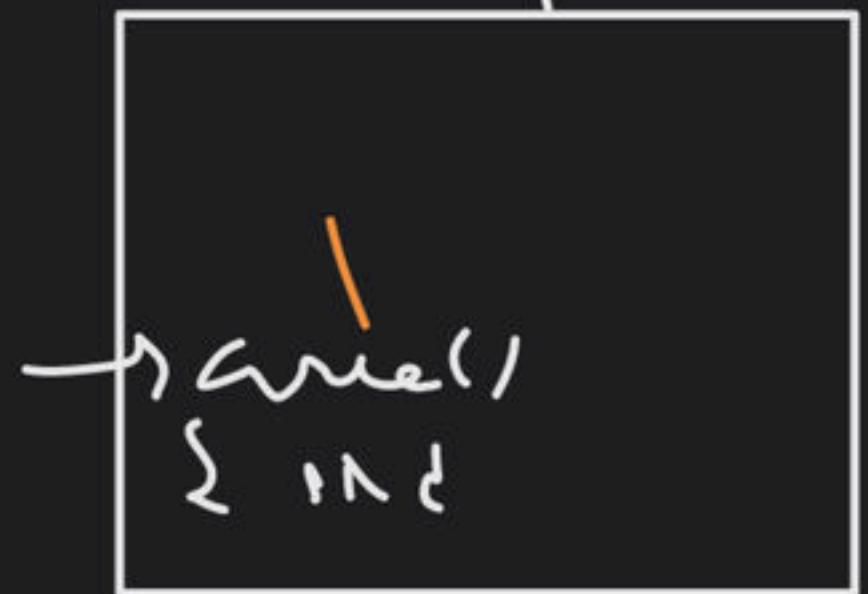


~~why \rightarrow ?~~

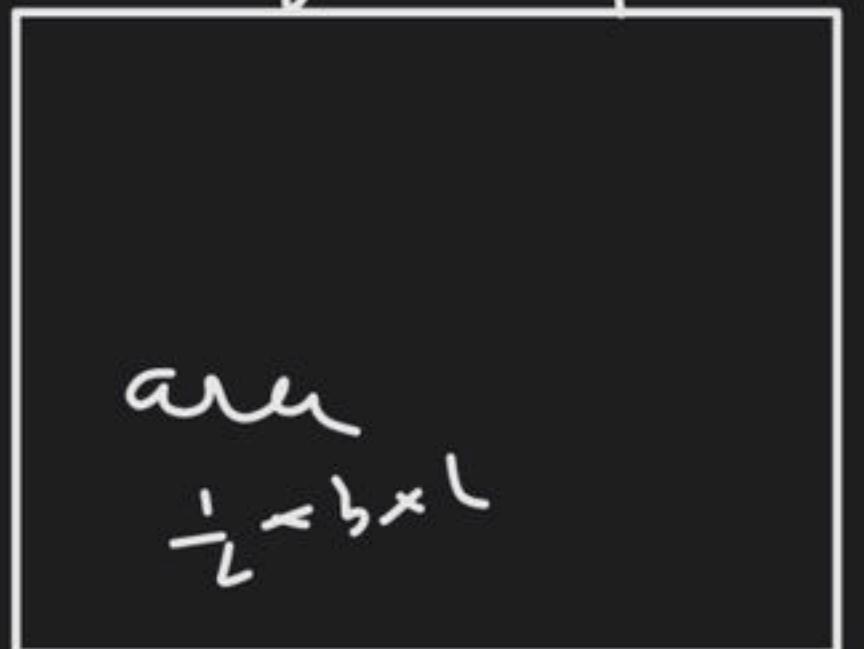


Real world
example

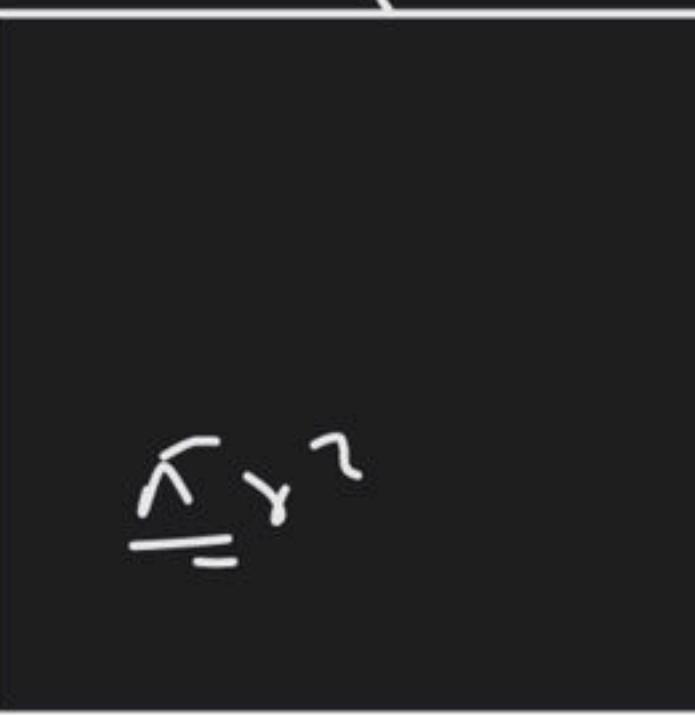
square



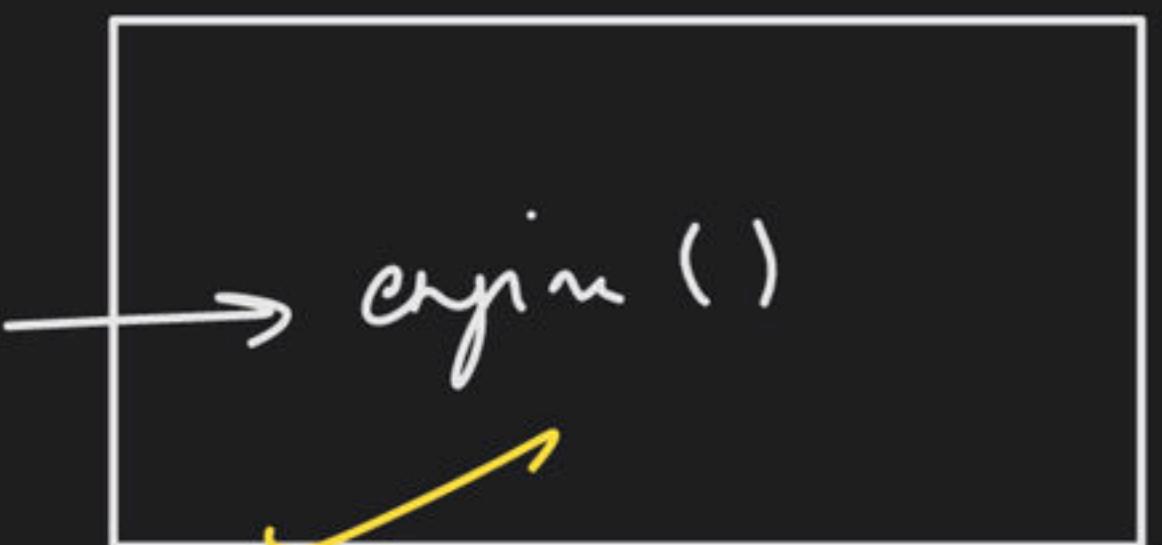
triangle



circle



car → control

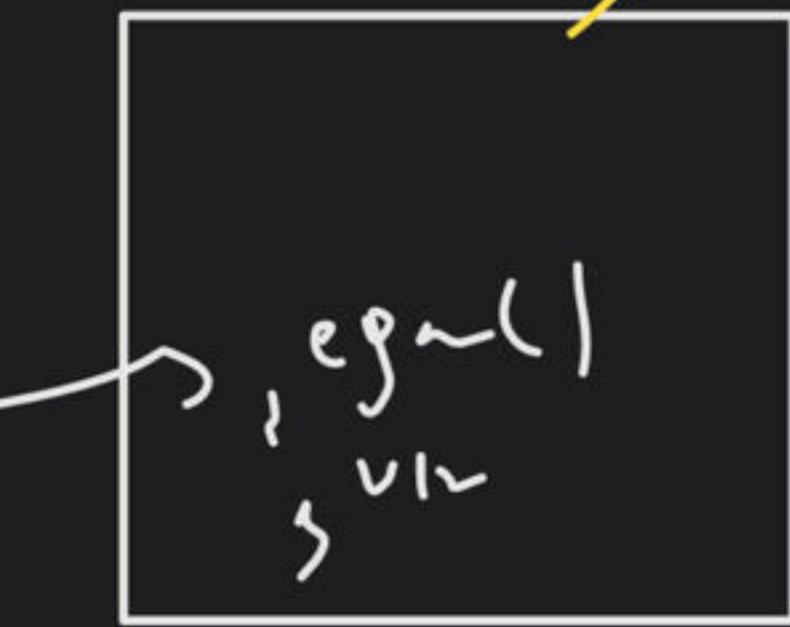


plug - play hai

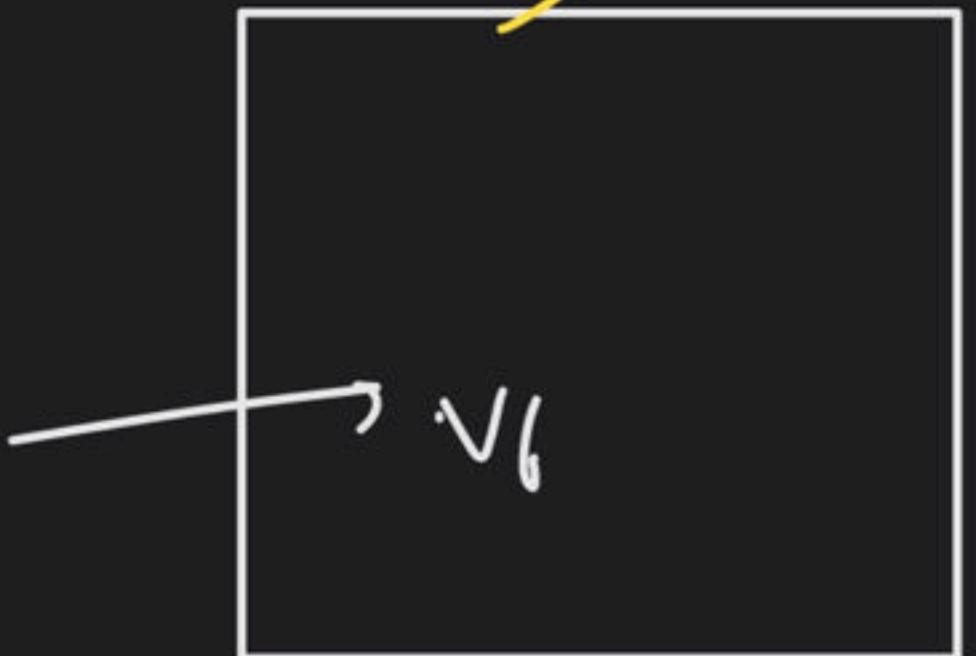
VOPS concept

VOPS Design

shov



hovel



ghetto



↗ Pillars → Yes - 

5% →

Virtual

function

class Base {

speak()

}

① Base class

pointer

Base * bptr;

define d;

bptr = & d;

class derived : public Base

bptr->speak();

speak()

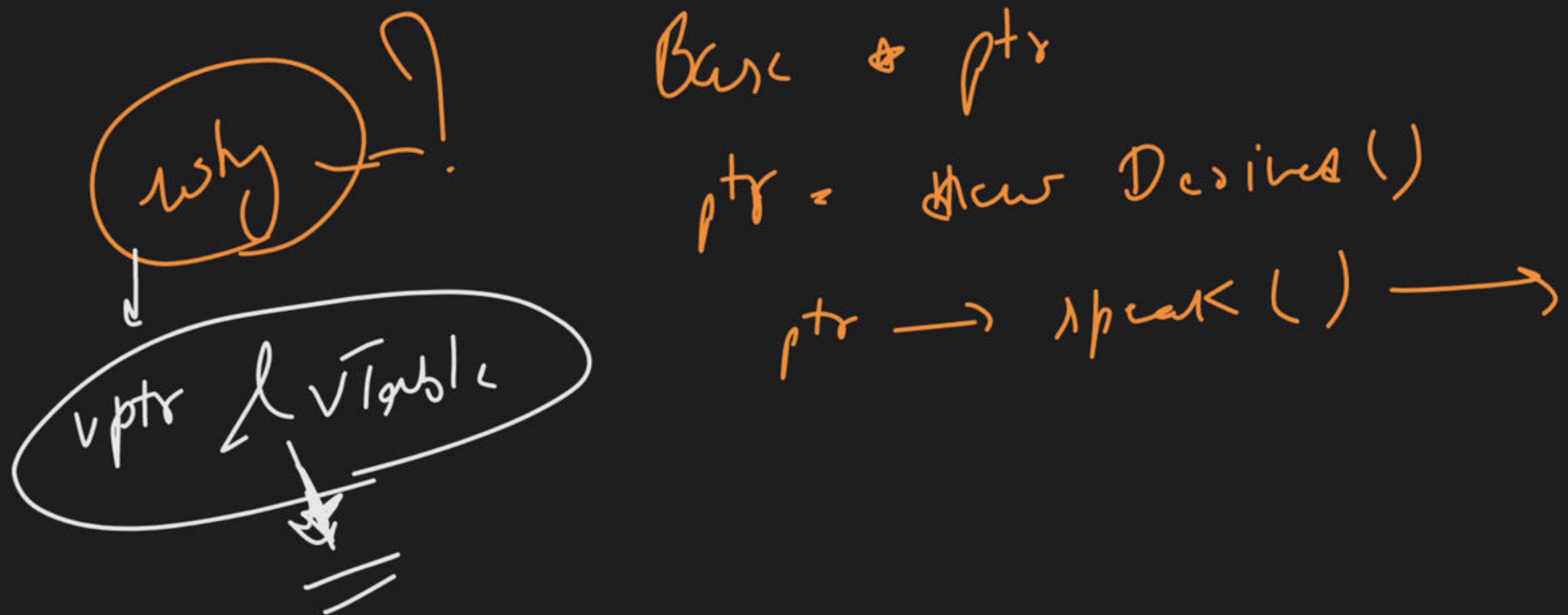
}

① Late Binding

② Early Binding

explore
join

→ Virtual function is used in Base class
to ensure that function is overridden.



viva
friend function

private
protchs
A
Lover
Babbar
Danau
Lovers

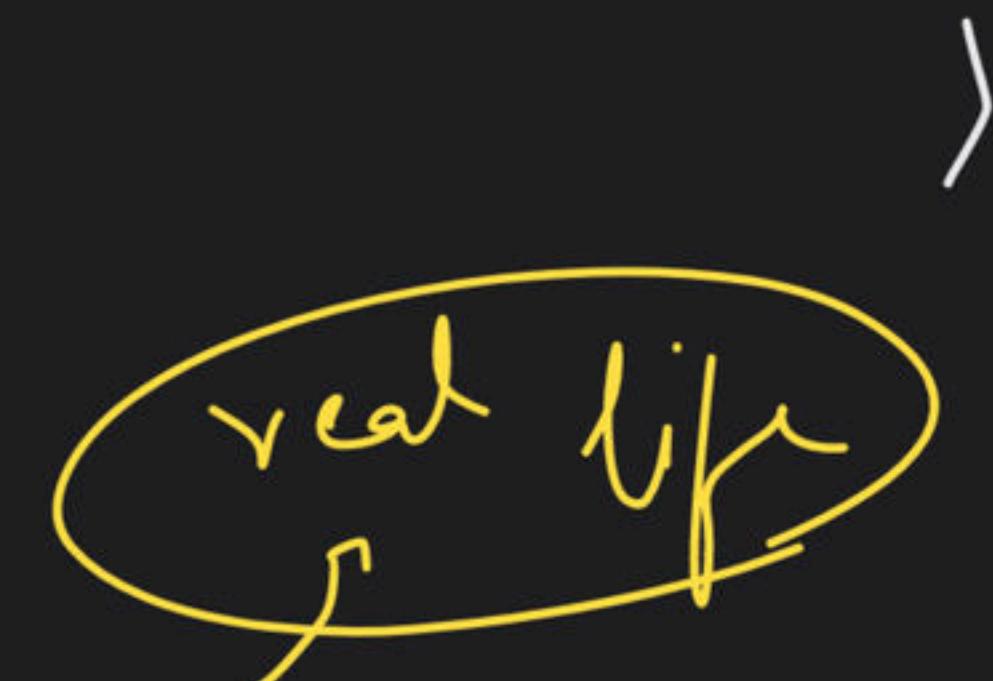
class A
{
private:
int a;
}
friend class B;

viva
friend class

class B
{
show f1
{
cout << A.a;
}
};

```
class A  
{  
    private:  
        int a;  
    friend class B;  
}
```

```
class D  
{  
public:  
    show(A& b)  
    {  
        cout << b.a;  
    }  
}
```



friend

function

class A → lone blau
{ lurk

pointer
punkt
acwrd

friend void show();

}

void show();

chudat
Daten

16-17

16th → 17th class

Intervia Corp BC
US
DBMS
OOPS
C/C++

friend
chudat
Daten

Class A

1

3

dass β

, how()

3

3

class A

{

 friend show()

}

 show()
 {
 }
 }

class B
{

 friend show();
}

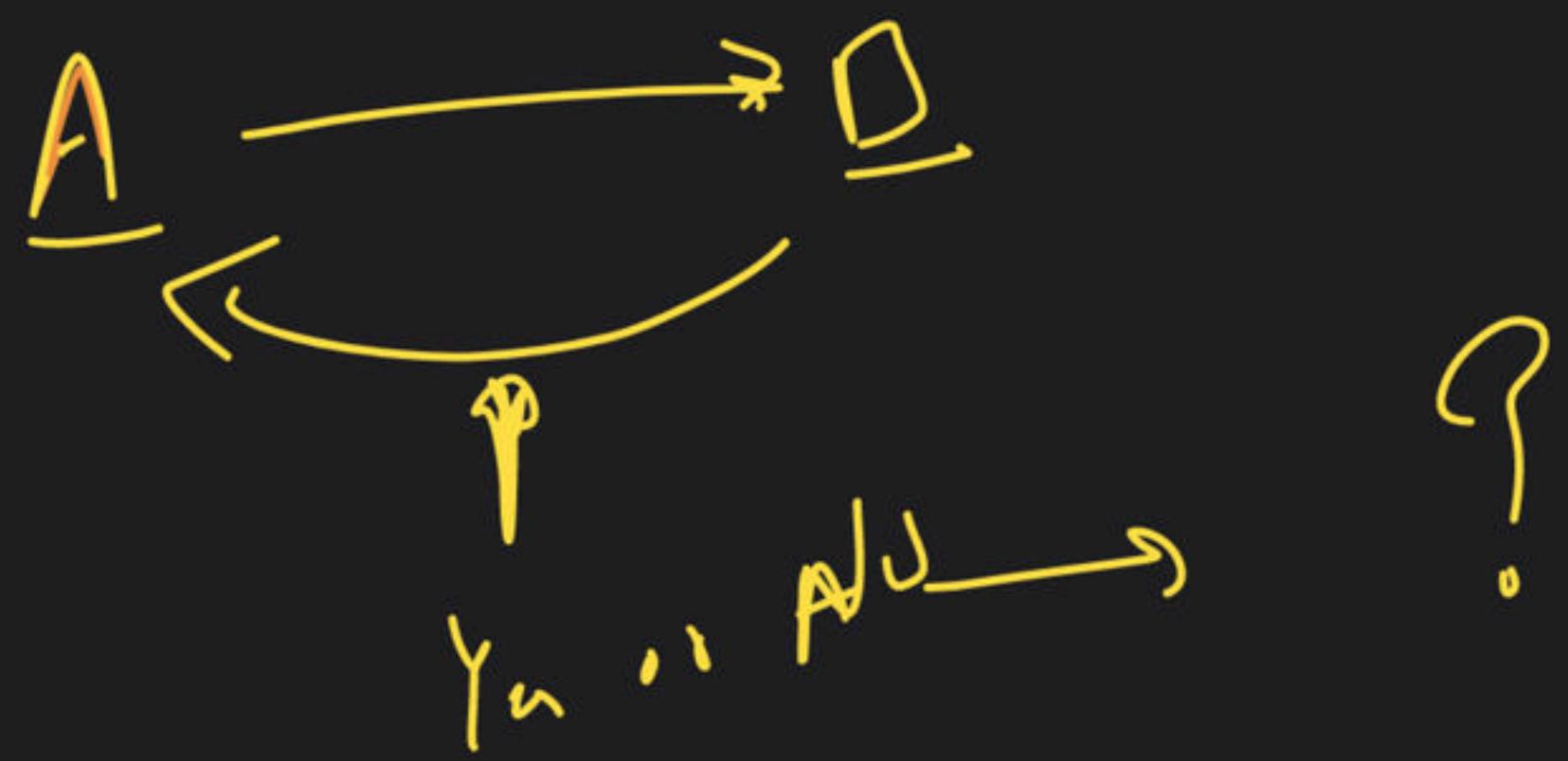
→

check

class A
{

friend class B
}

class D
{



class A

{

class B

{

class C

{

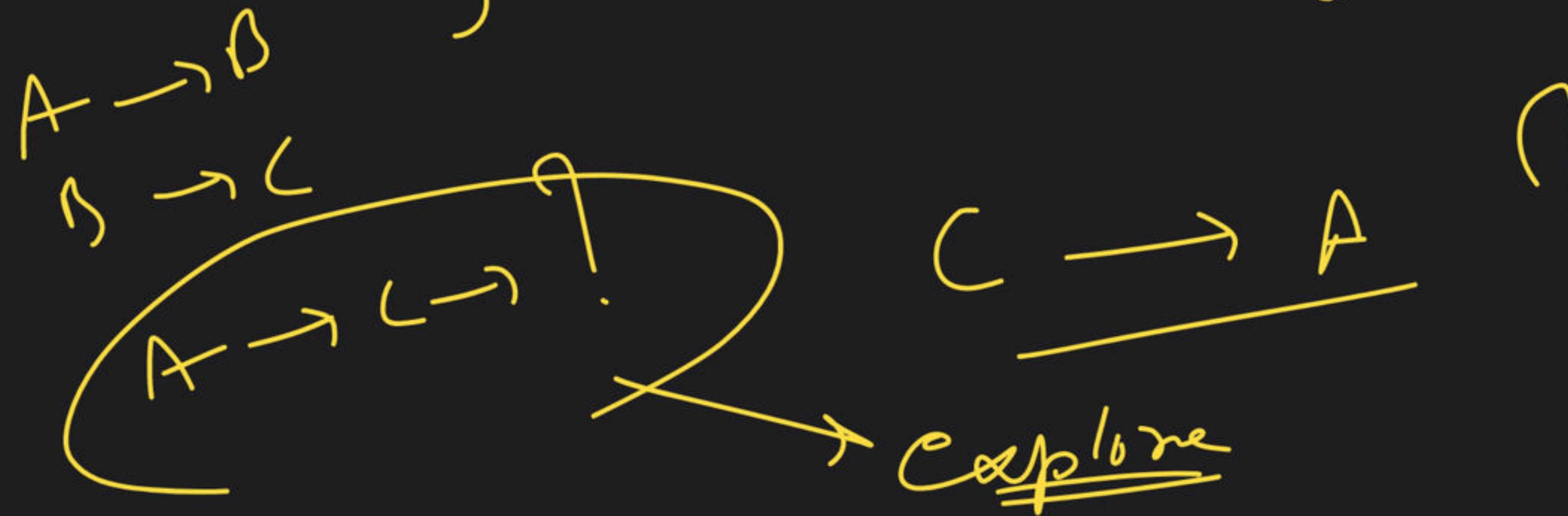
friends class B

}

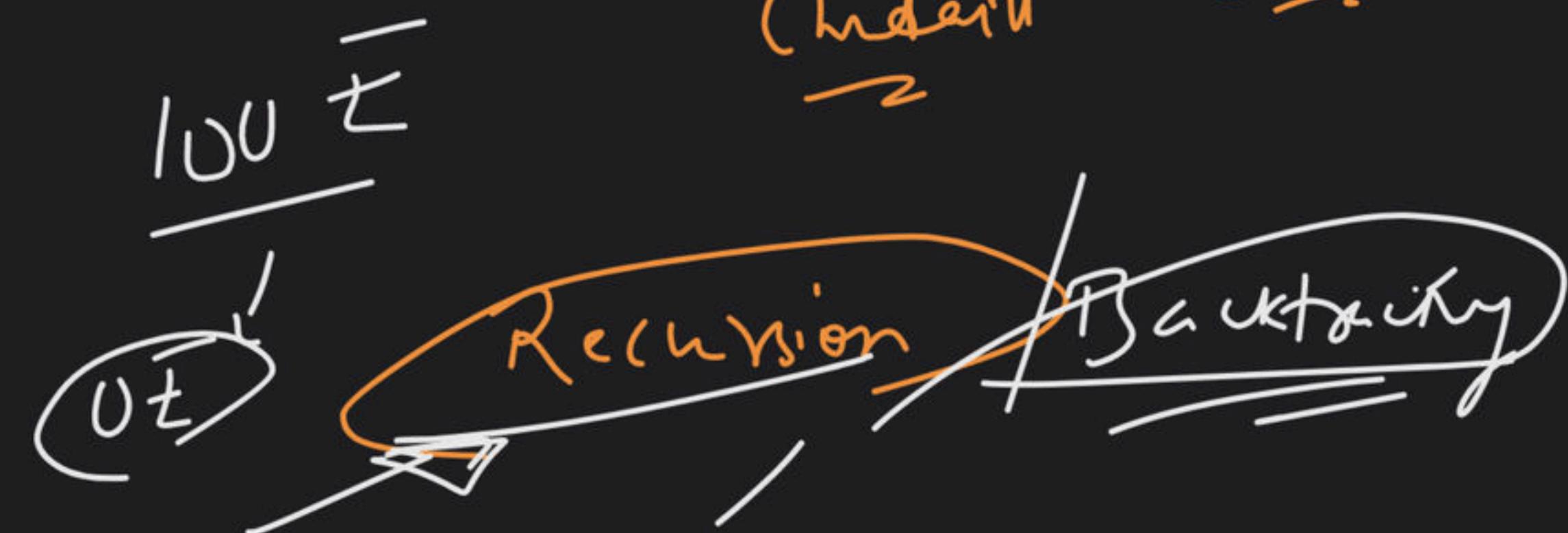
friends class C

}

}



- 0.0
- virtual class
- virtual fn
- m - class
- pm virtual fn
- global / static fn
- friend fn
- friend fn class



friend / chardik / Daron

DSA Series

CompU

Chardik

Daron

Backtracking

→ Blockchain

→ Rec

→ Web Dev

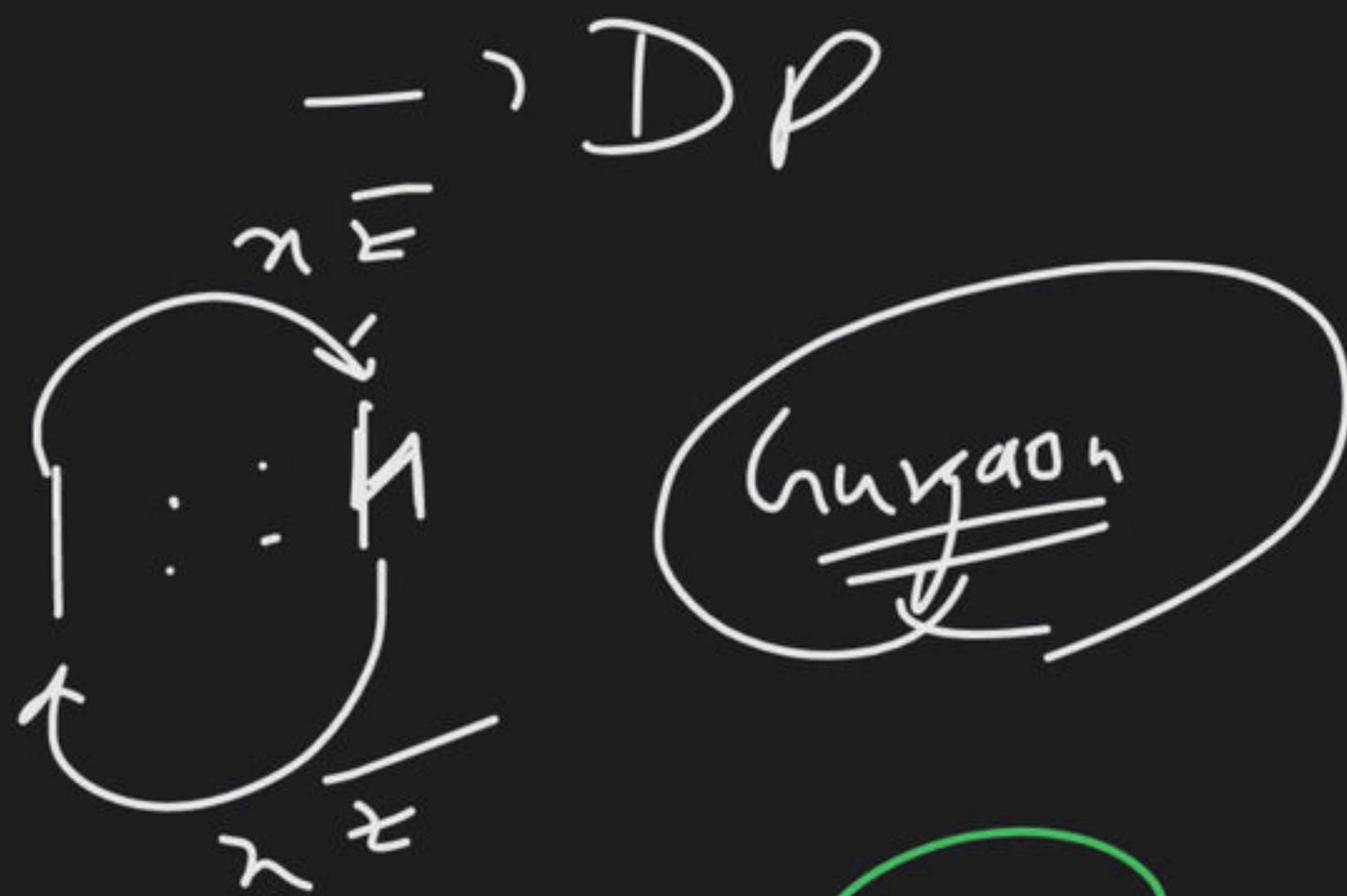
1.5 hr

2 hr

→ Linked List

$y_i \rightarrow$ playlist

TRUL → repeat
Karo



25%

False - met
Karo

25%

met
Karo

