



OOPs Class-1

Special class



DOP :-

what -> ?

programming

technique



OBJECT :

Entity

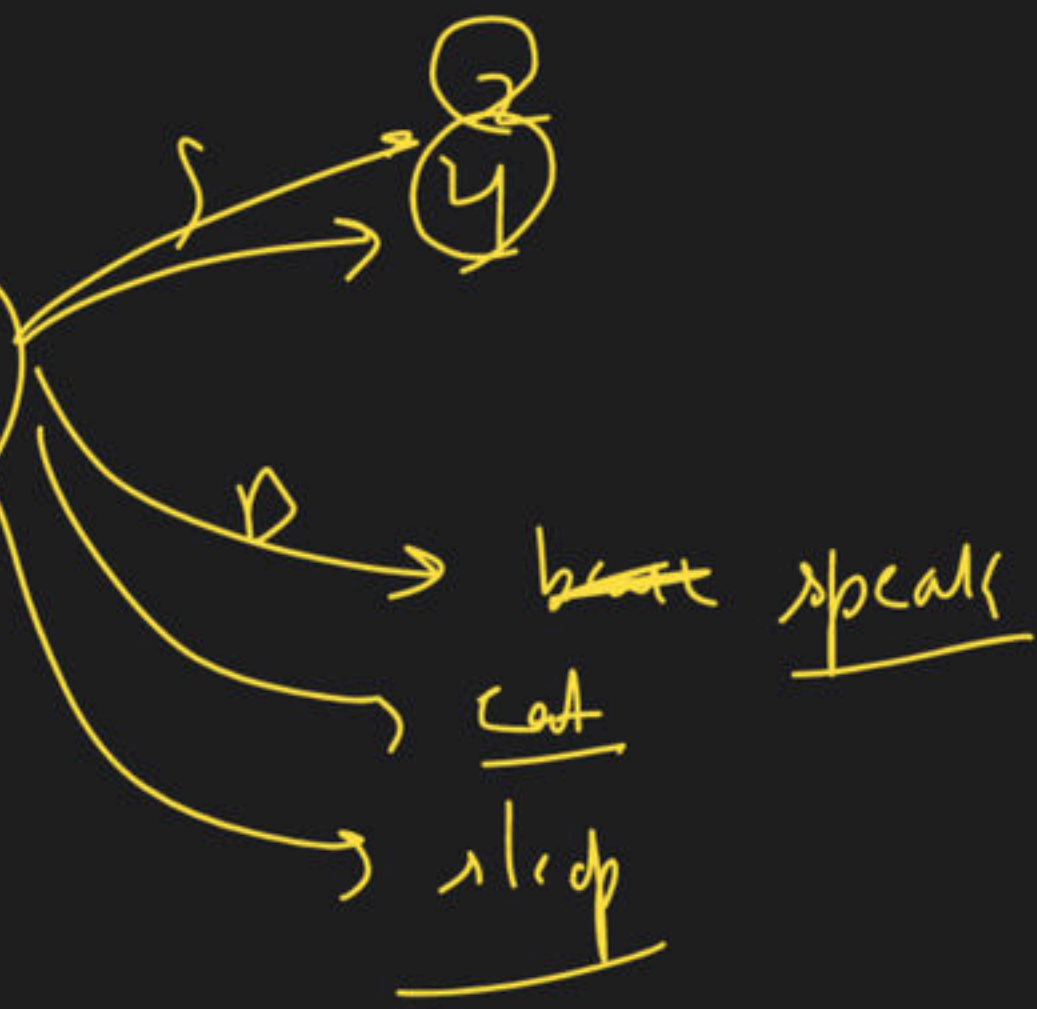
- State / property
- Behaviour / method

Why ?



How

code



→ class → How?

```
int num;  
string str;  
bool abc;
```

Custom datatype → Ramesh
User defined datatype
int
bool
string

Animal
name ✓
int ✓
int ✓

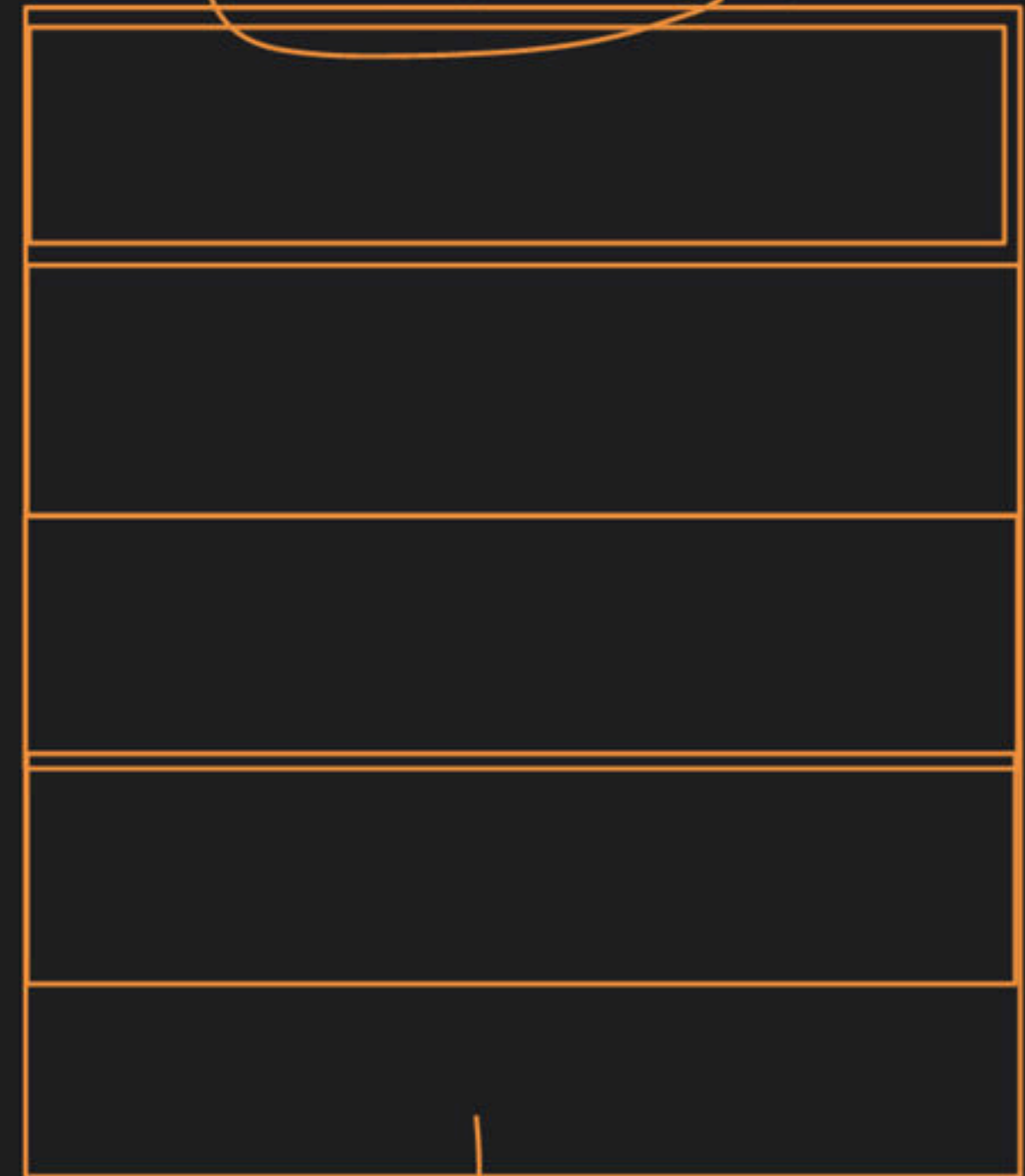
```
int    bool  
      string
```


Architect



Empty class
↓
0?
=

~~Real~~ ~~Real~~ Actual
Real



OBJECT → instance of class

n/w
↳

padding

↳

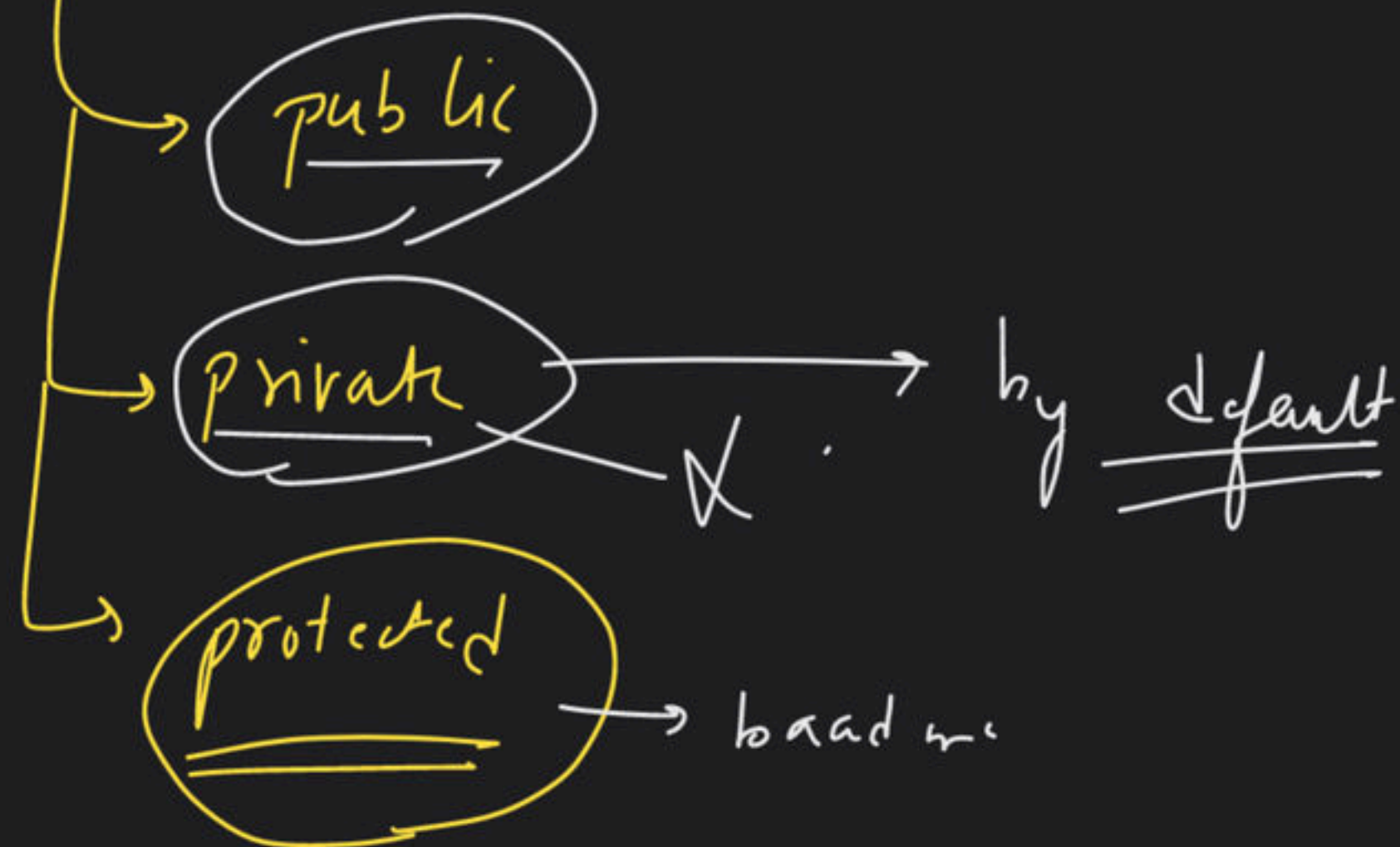
greedy alignment

OOP

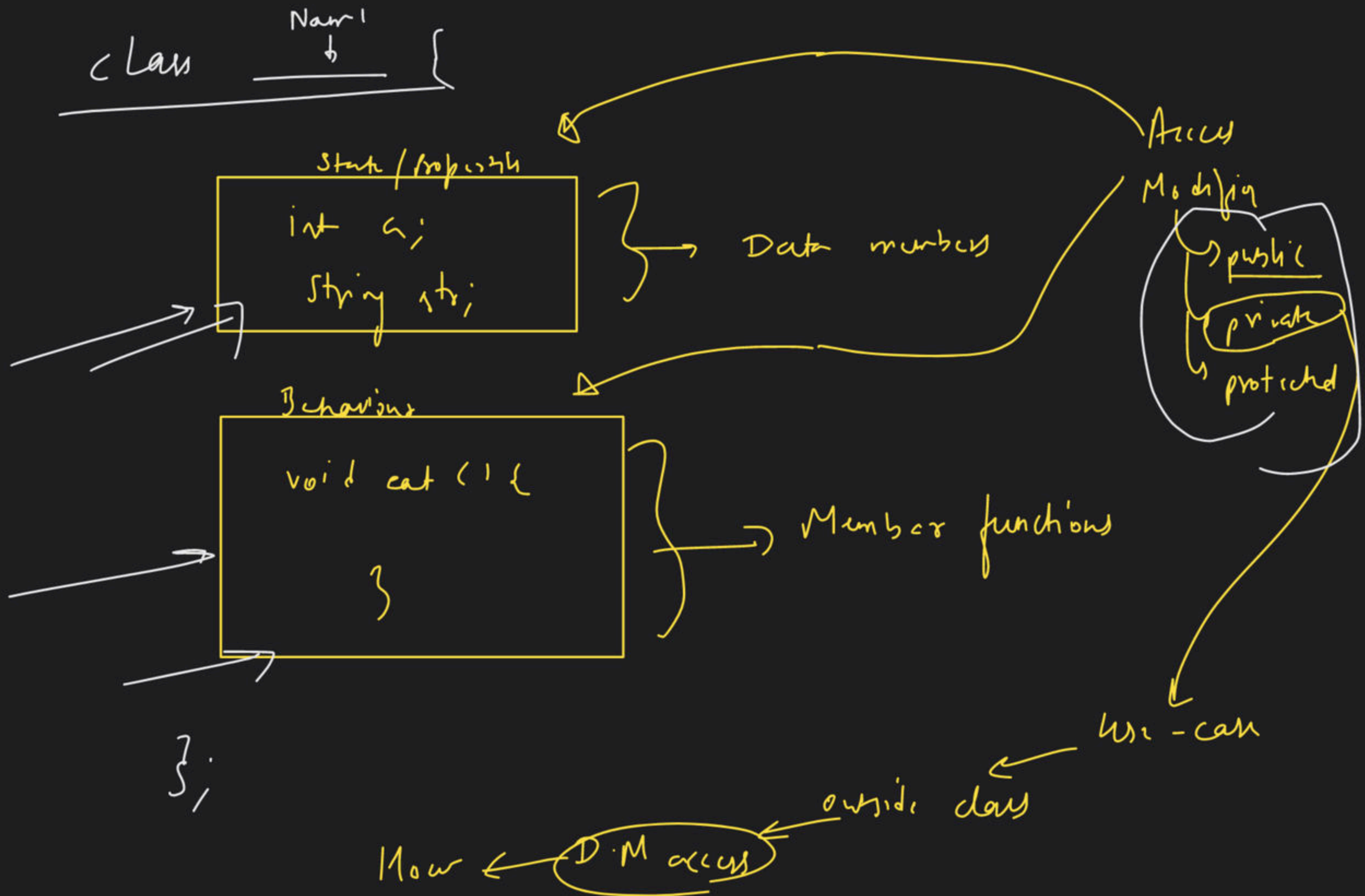
↳ object

↳ class

Access Modifier



age / name
↓
private



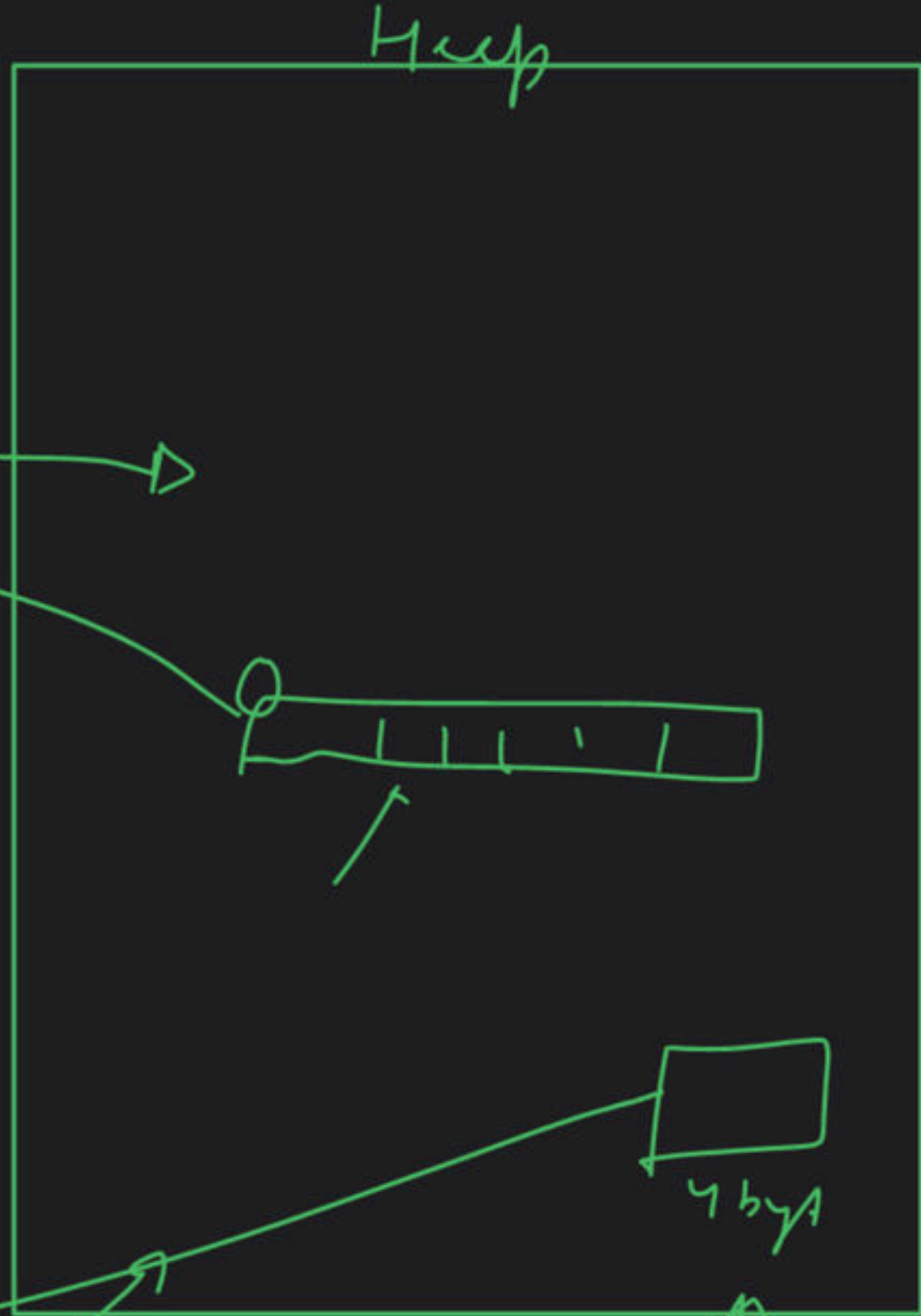
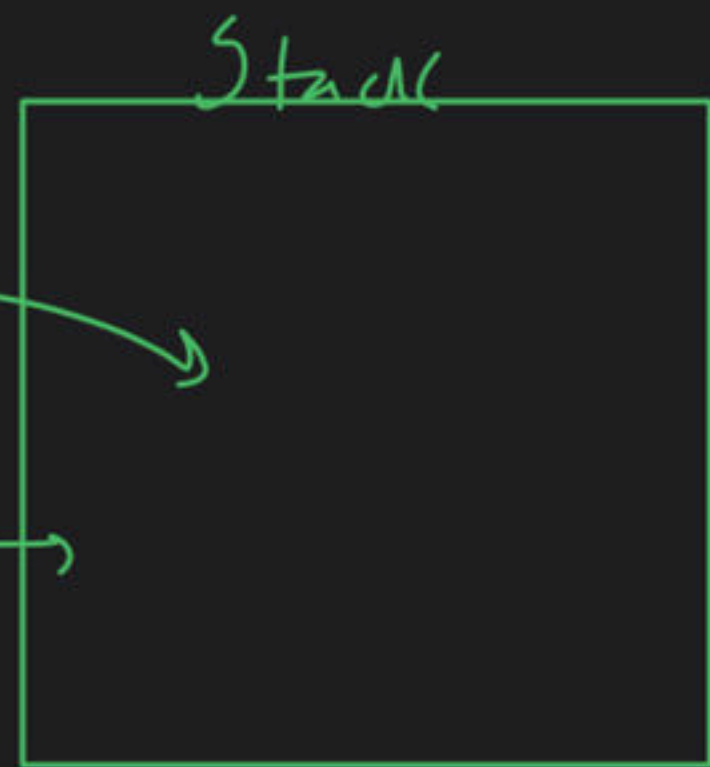


(100%) delete [] arr;
delete left[];
int* left = new int [len];

Dynamic Memory

Ex
 ↓
 new int
 new char
 new int
 new int[]
 address

int a
 string str
 int arr[10]



new

dynamic memory allocation

delete a

int* a = new int;

int* a = 5

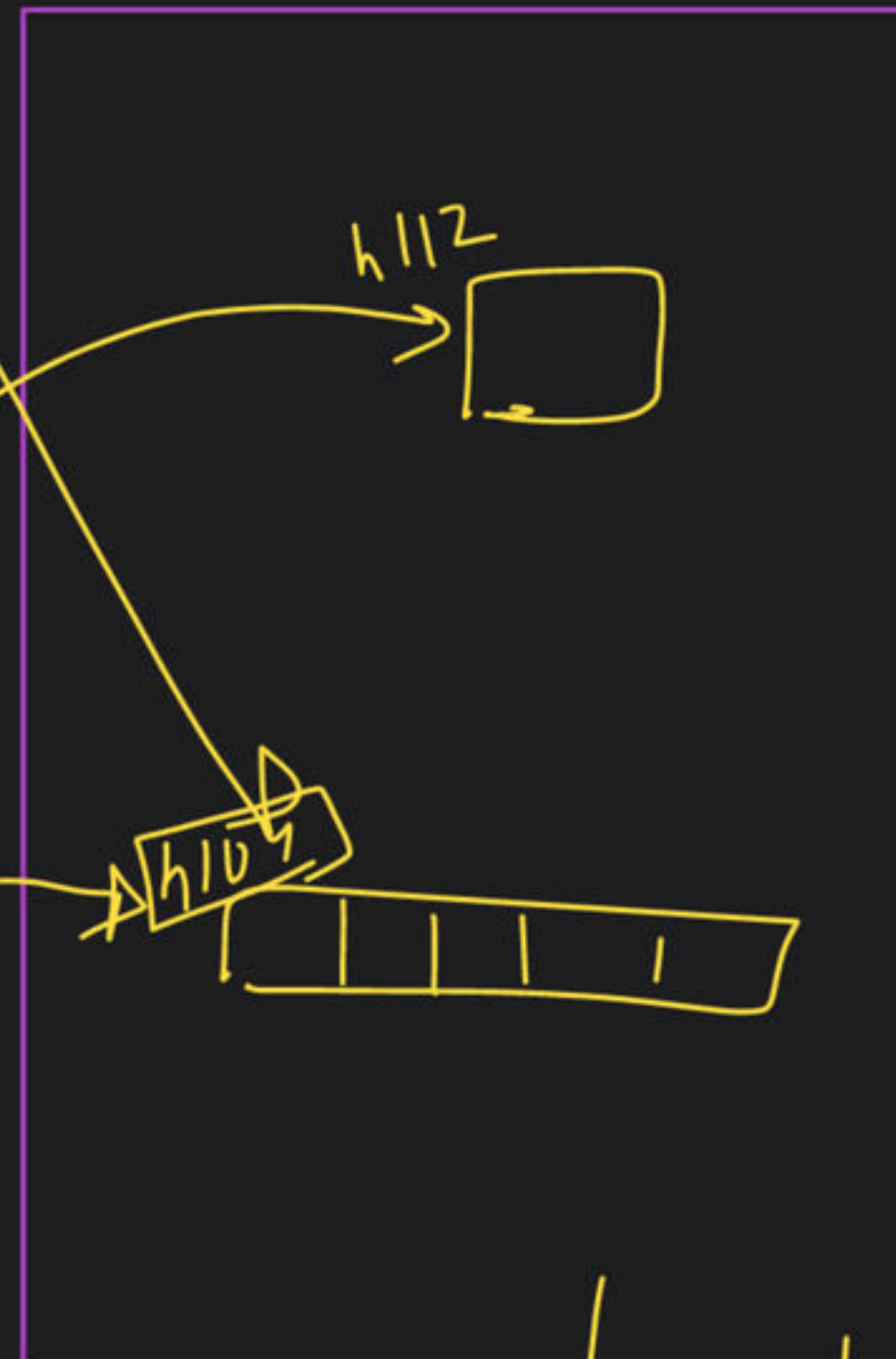
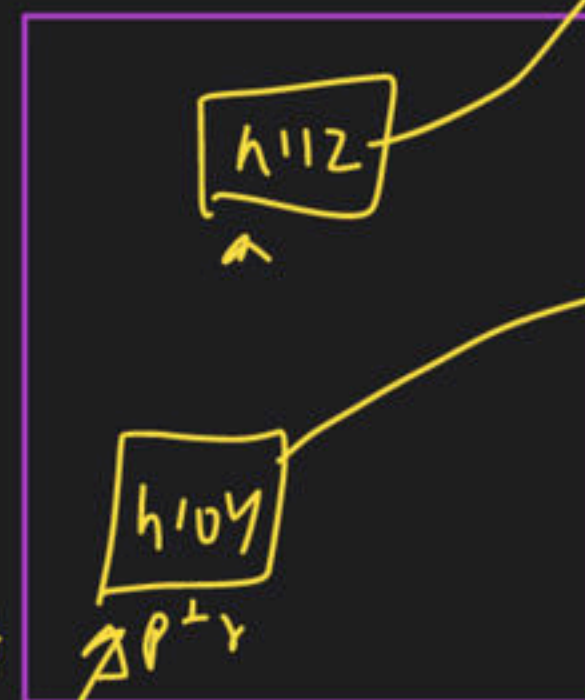
*a

→ Garbage Collector

→ Memory Leak

int* ptr = new int[5];

new int[5];



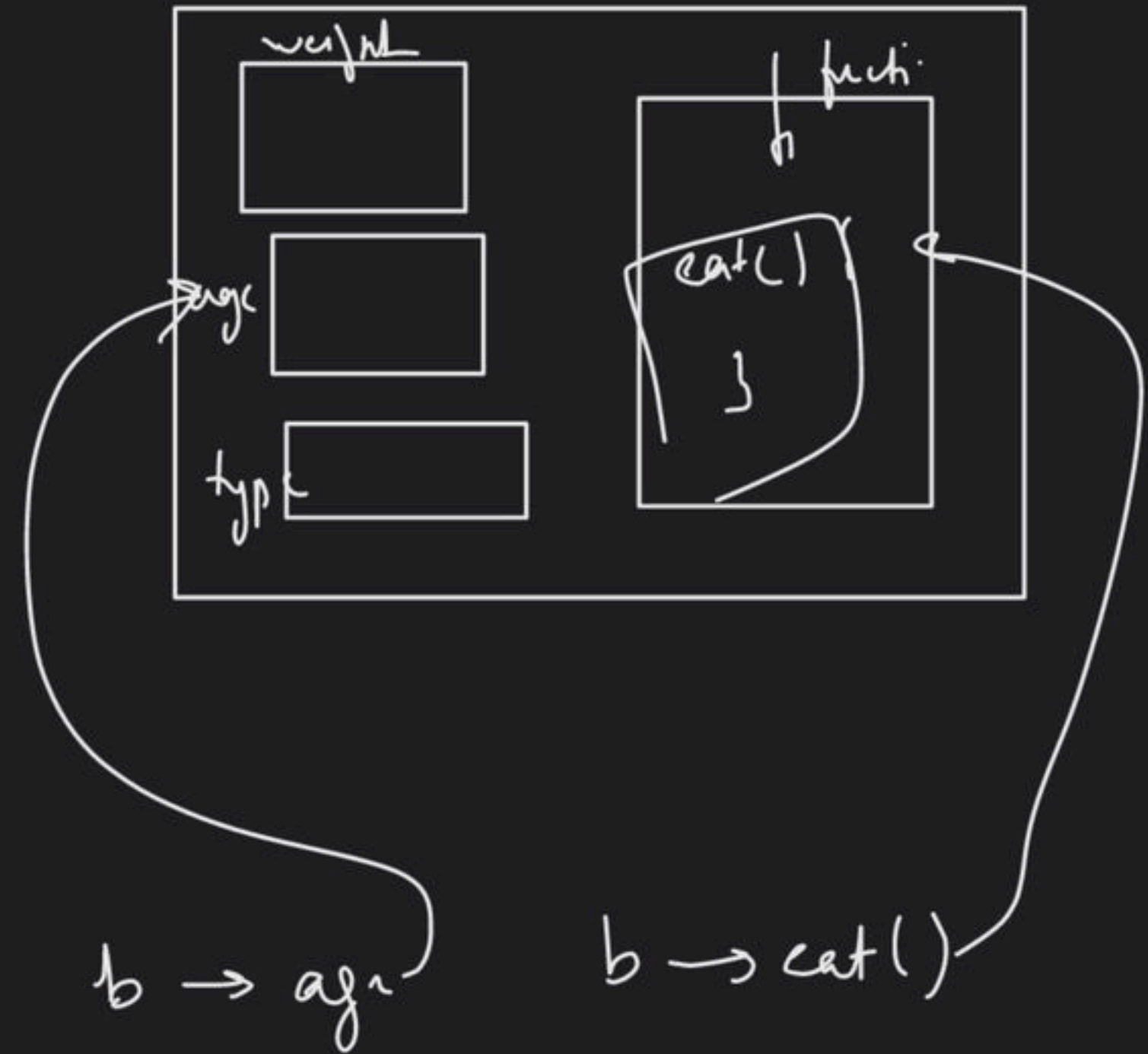
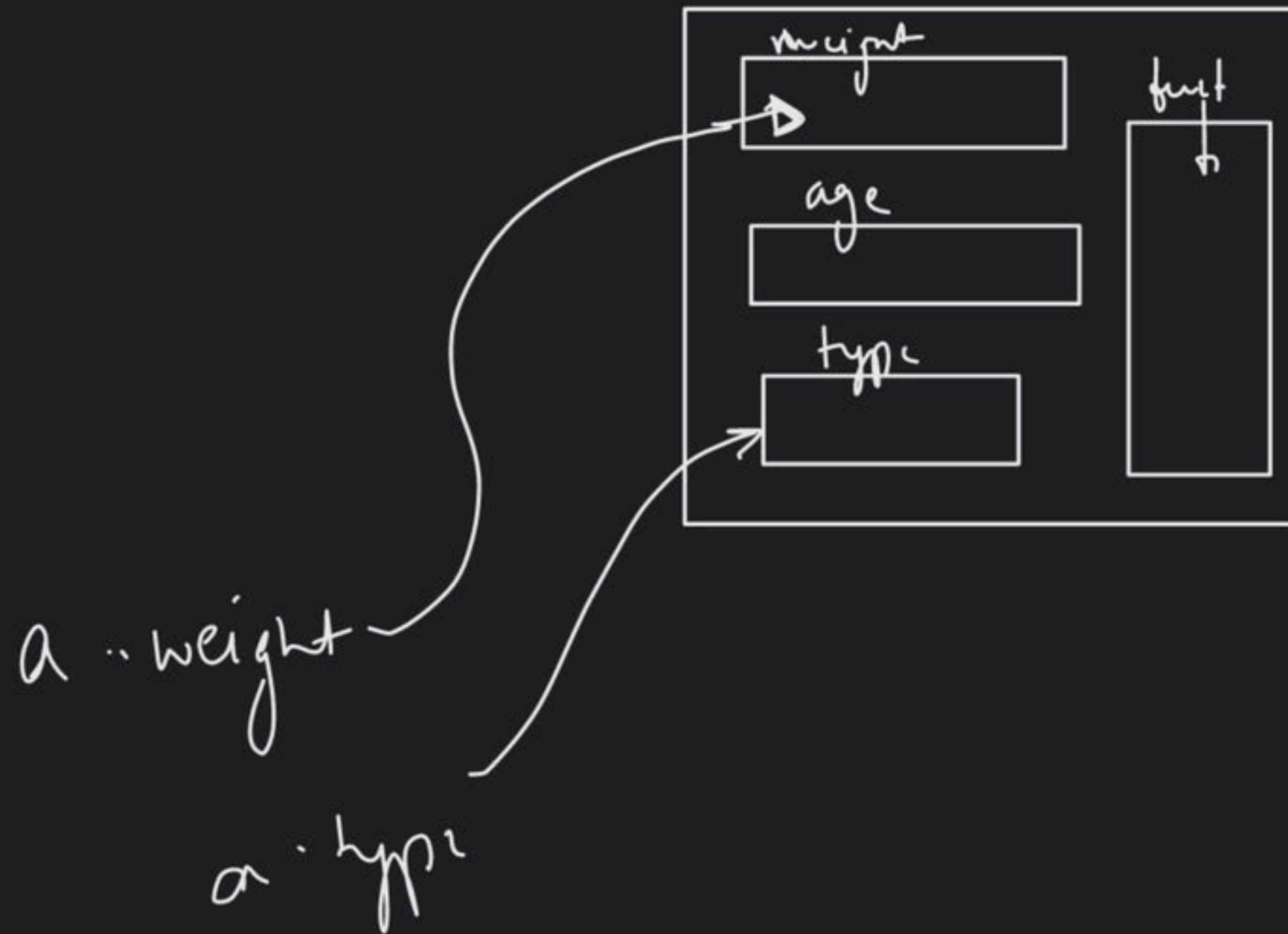
int* a = new int

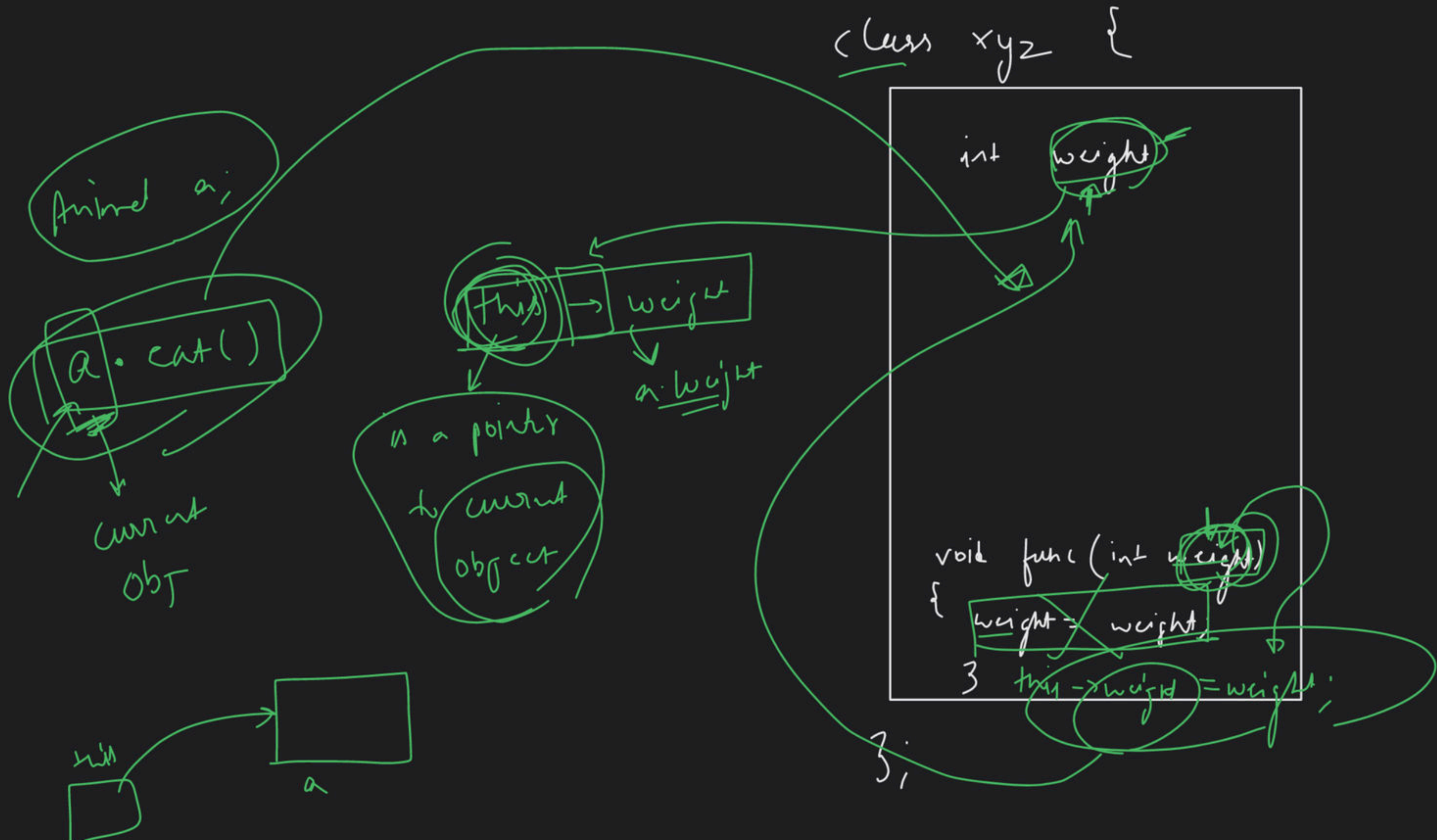
new int

allocation
↓
new keyword
de-allocation
↓
delete

Animal a;

Animal *b = new Animal;





Suruh → ayc;

or

(~~*~~ suruh) - ayc

(~~*~~ this) - ayc;

current object ki weight wali property

this → weight

Kaise

obj

this → weight = weight



class Animal

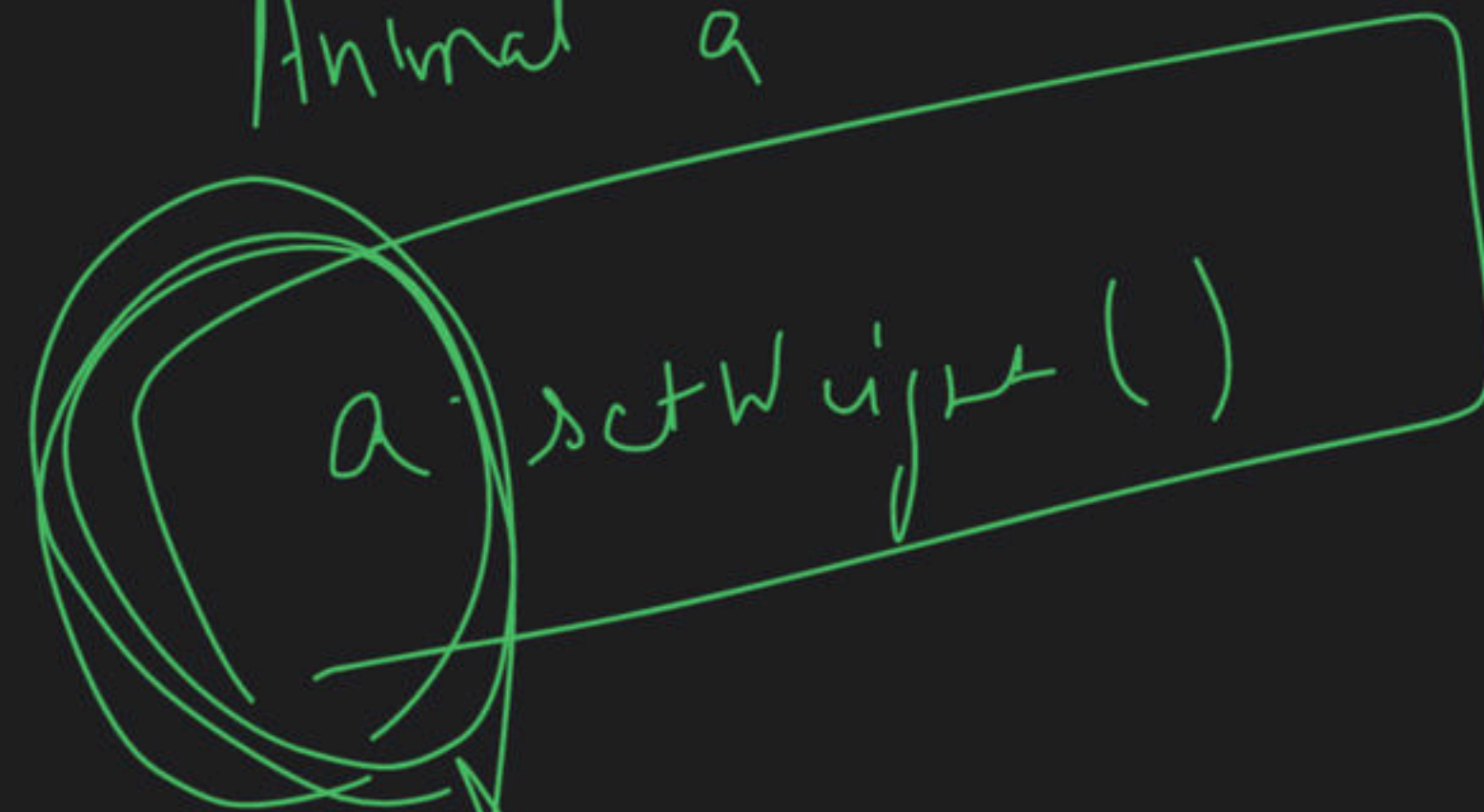
{

```
int age;  
int weight
```

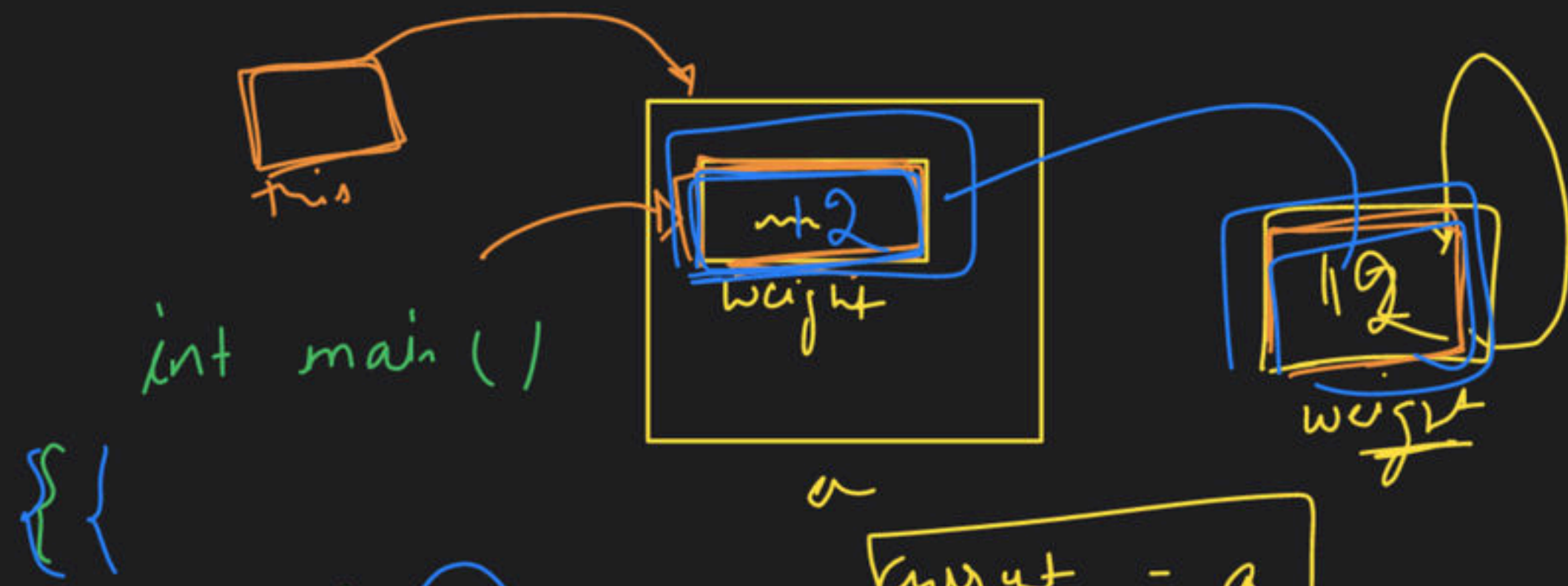
```
void setWeight(int weight)  
{  
    weight = weight;  
}
```

}

Animal a



current obj



```
int main()  
{  
}
```

Animal a

current object = a

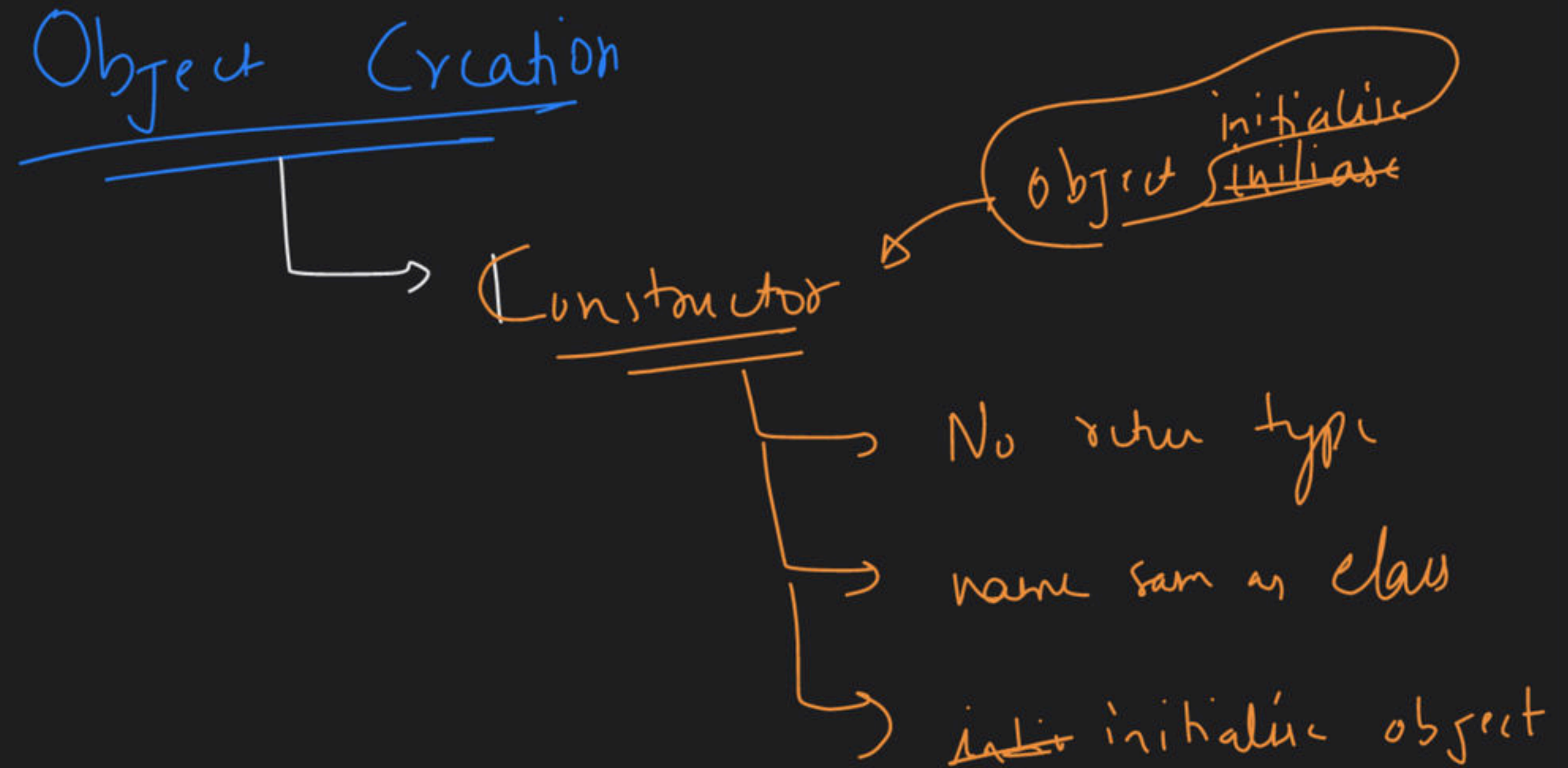
this pointer
is pointing
to a

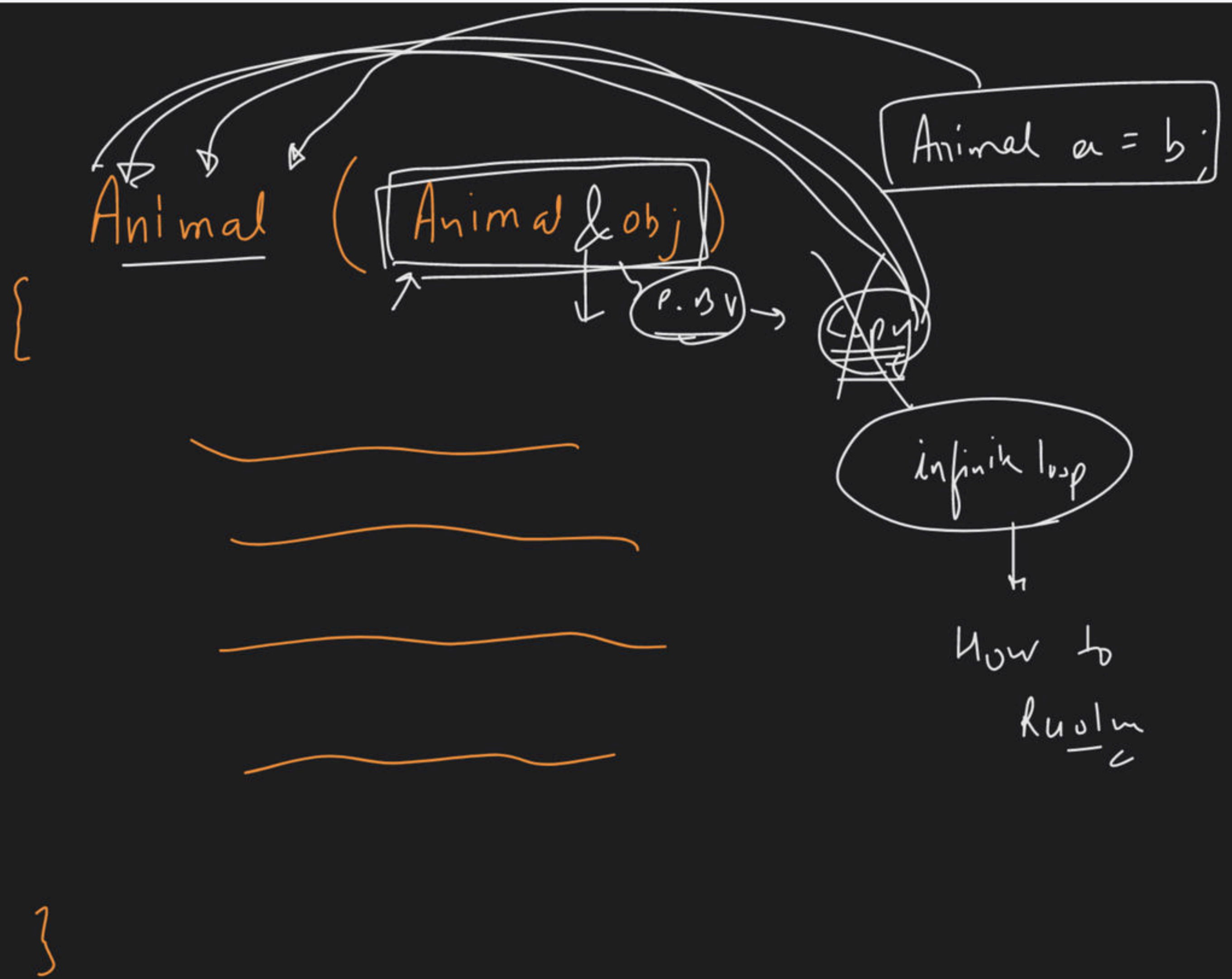
```
a->setWeight(2);  
}
```

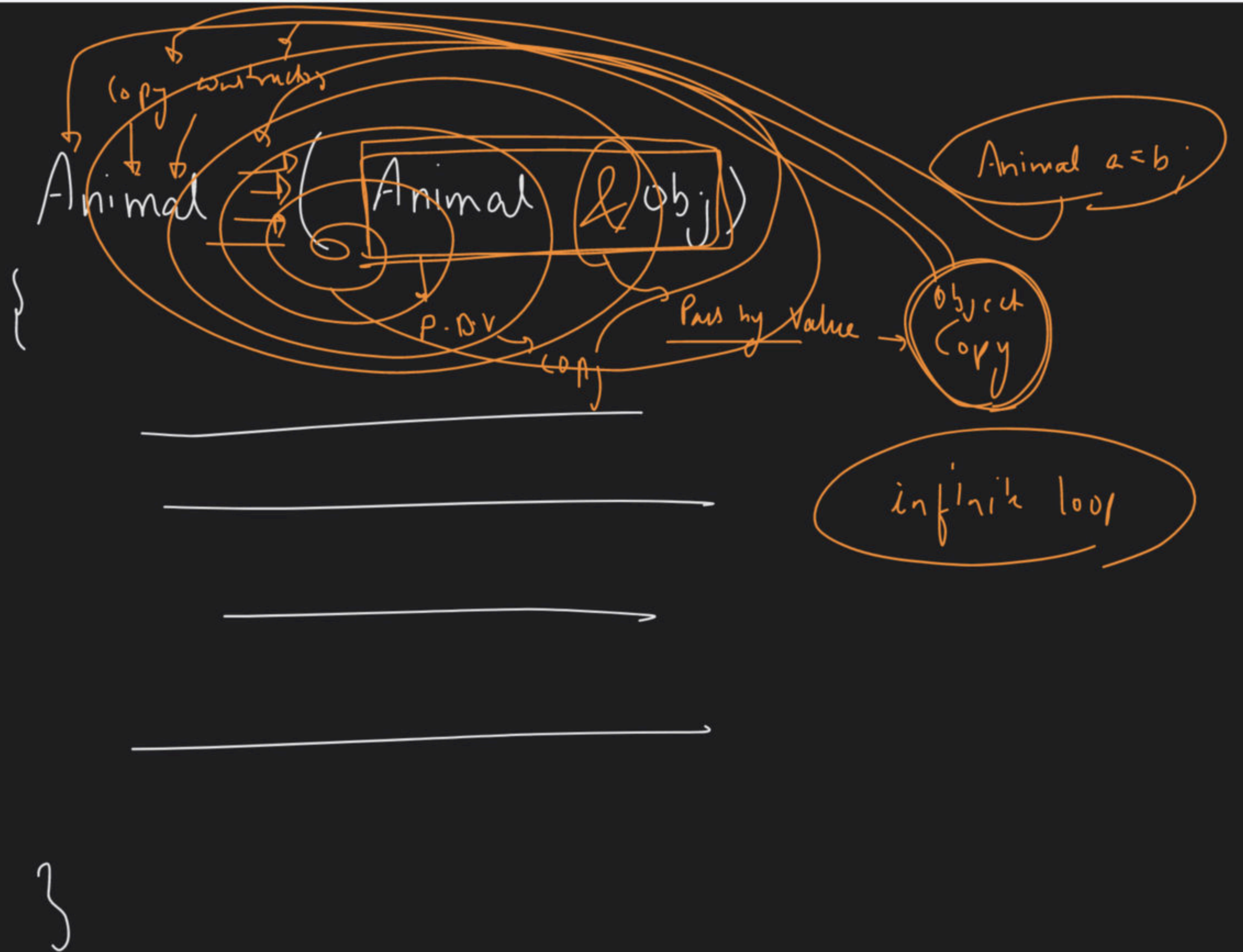
```
class Animal {  
    int weight;  
}
```

```
void setWeight(int weight)  
{  
    weight = weight;  
    this->weight = weight;  
}
```


Object Creation

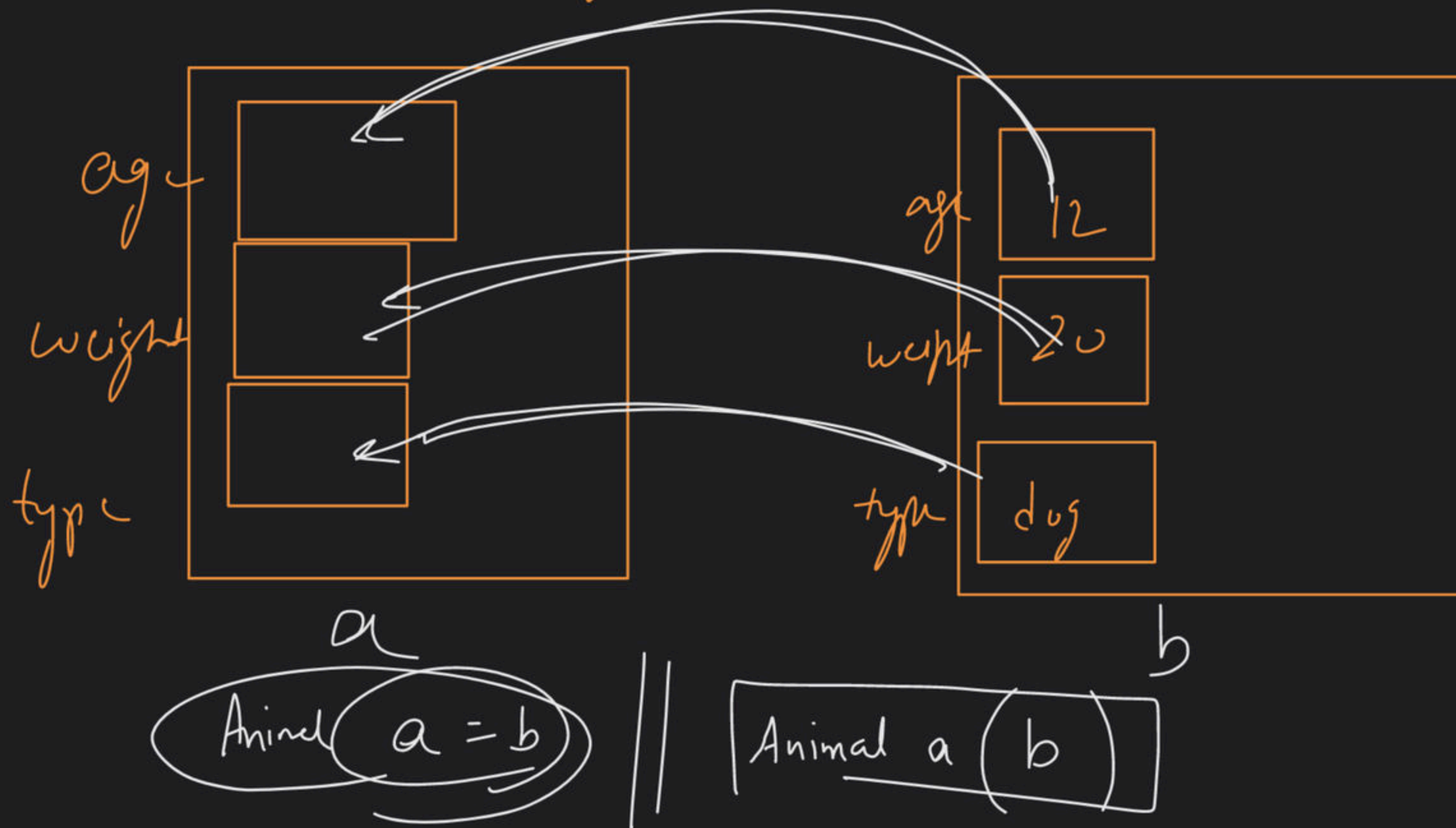




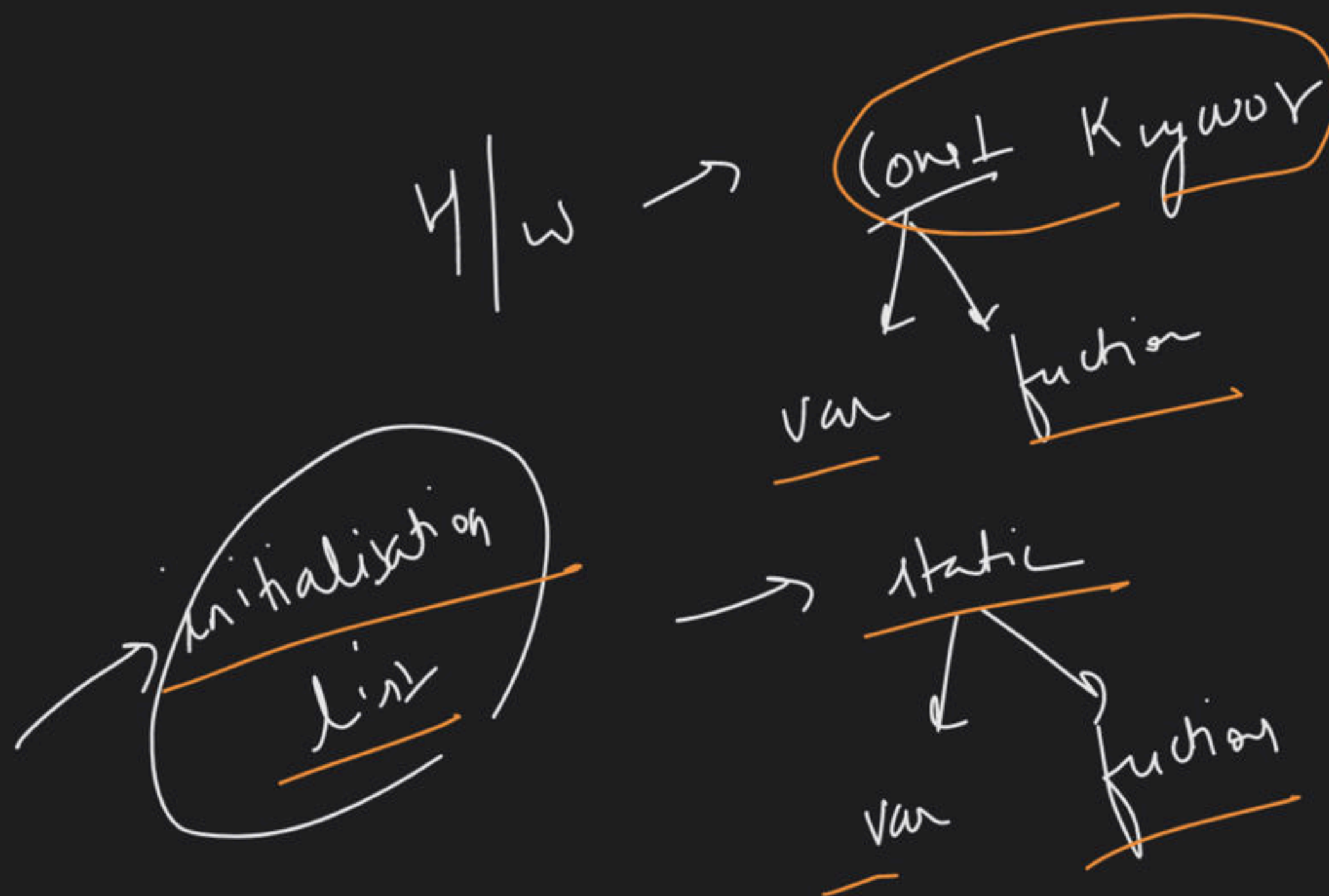
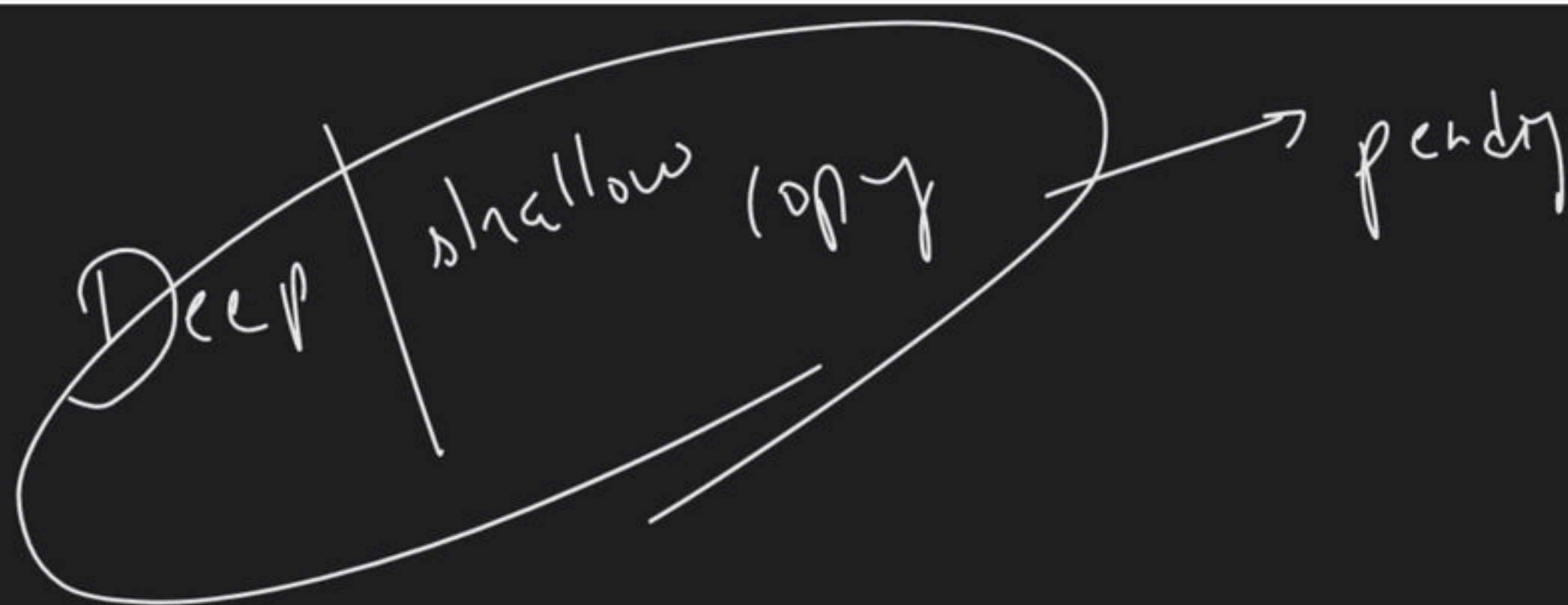


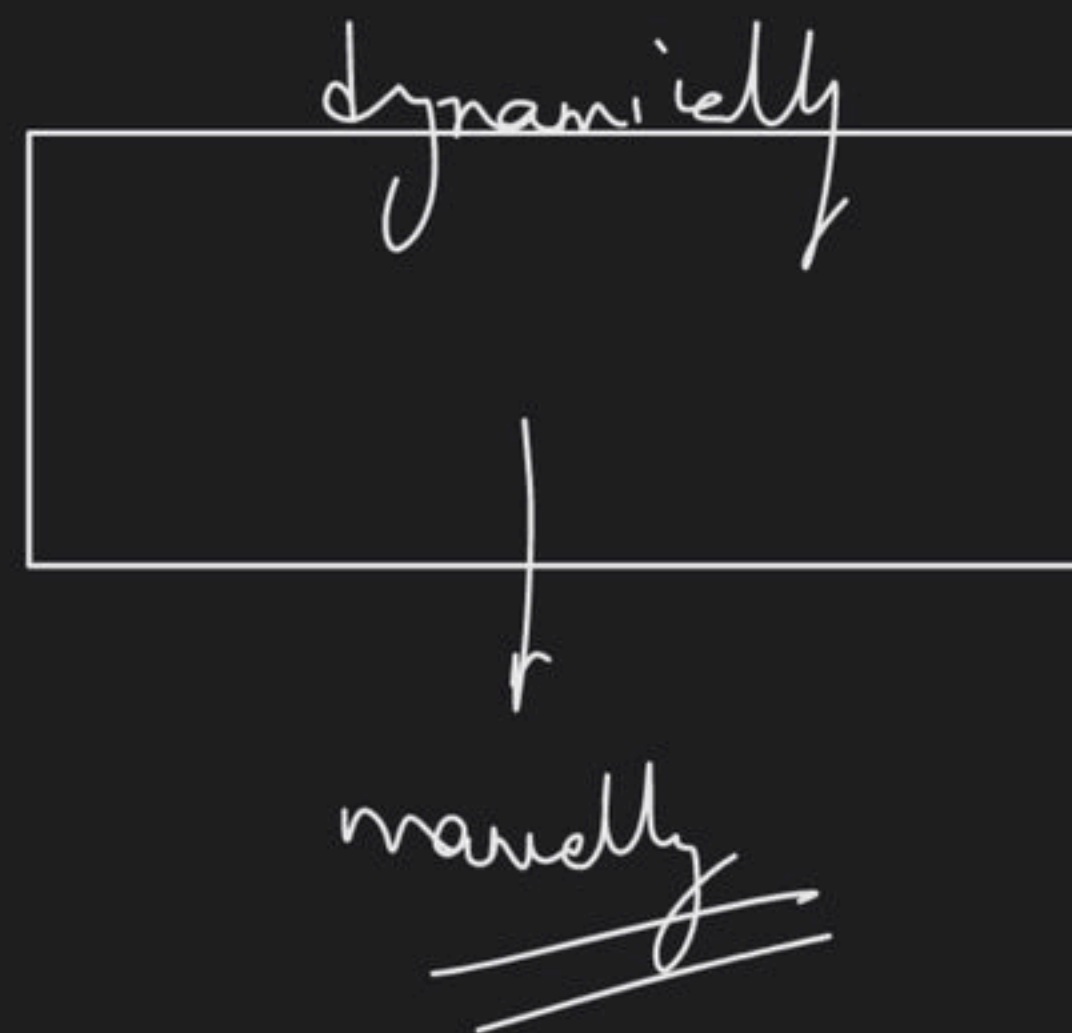
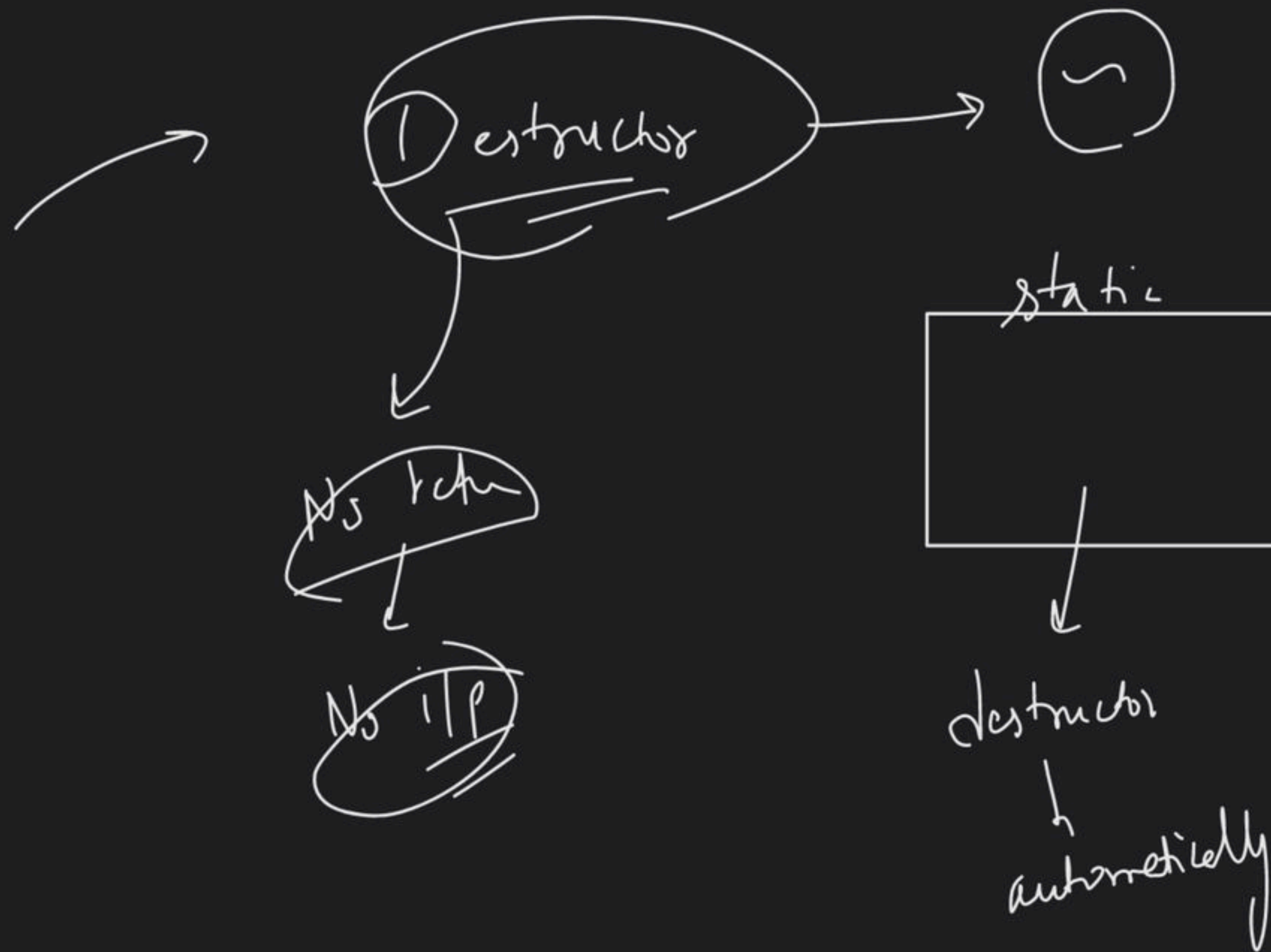
2 min
Break

Object copy



new





Basis

→ class → Empty
→ Non-Empty

→ Object → static
→ dynamic

→ Access modifier → public
→ private

→ getter & setter

→ ~~3~~ Object in Memory

→ Obj creation → constructor → default const.
→ parameterised const.
→ copy const.

→ Deep / shallow Copy → pending

padding
& ?
greedy
alignment

4 Pillars

→ Destructor
→ static & auto
→ dynamically
→ manually













