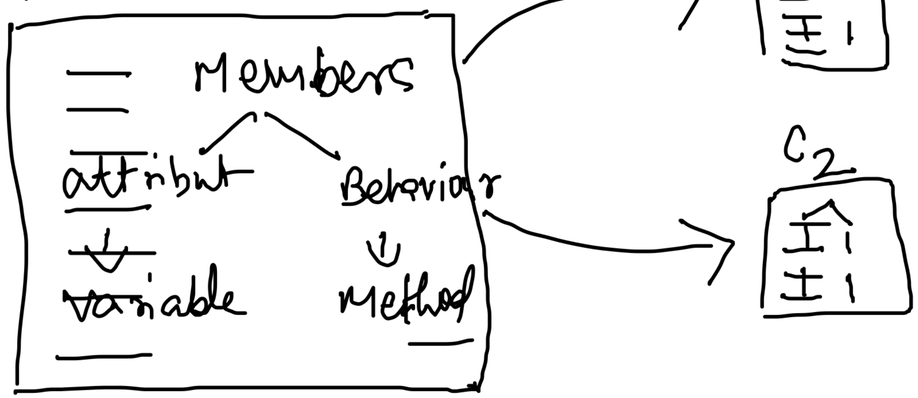
(1) In Python, Everything is an object.

Class - Blueprint

Class ABC




Employee

+ ename
+ esal
+ eid
} Attributes

+ Emp_detail()
+ Increment_sal()
} Behaviour

e1

+ ename = 'ABC'
+ esal = 10000
+ eid = 101

+ emp_detail()
+ Increment_sal()

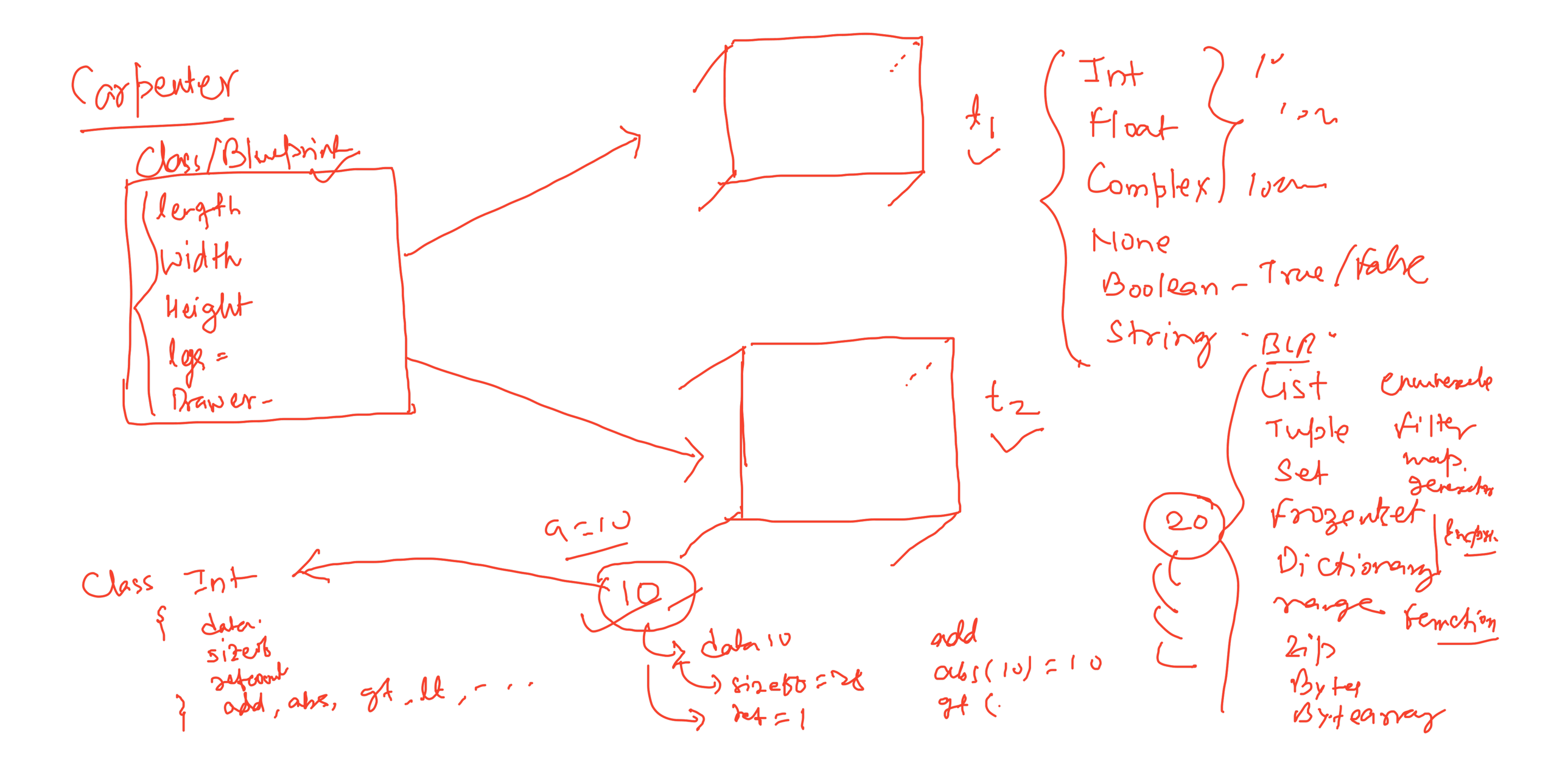
e2

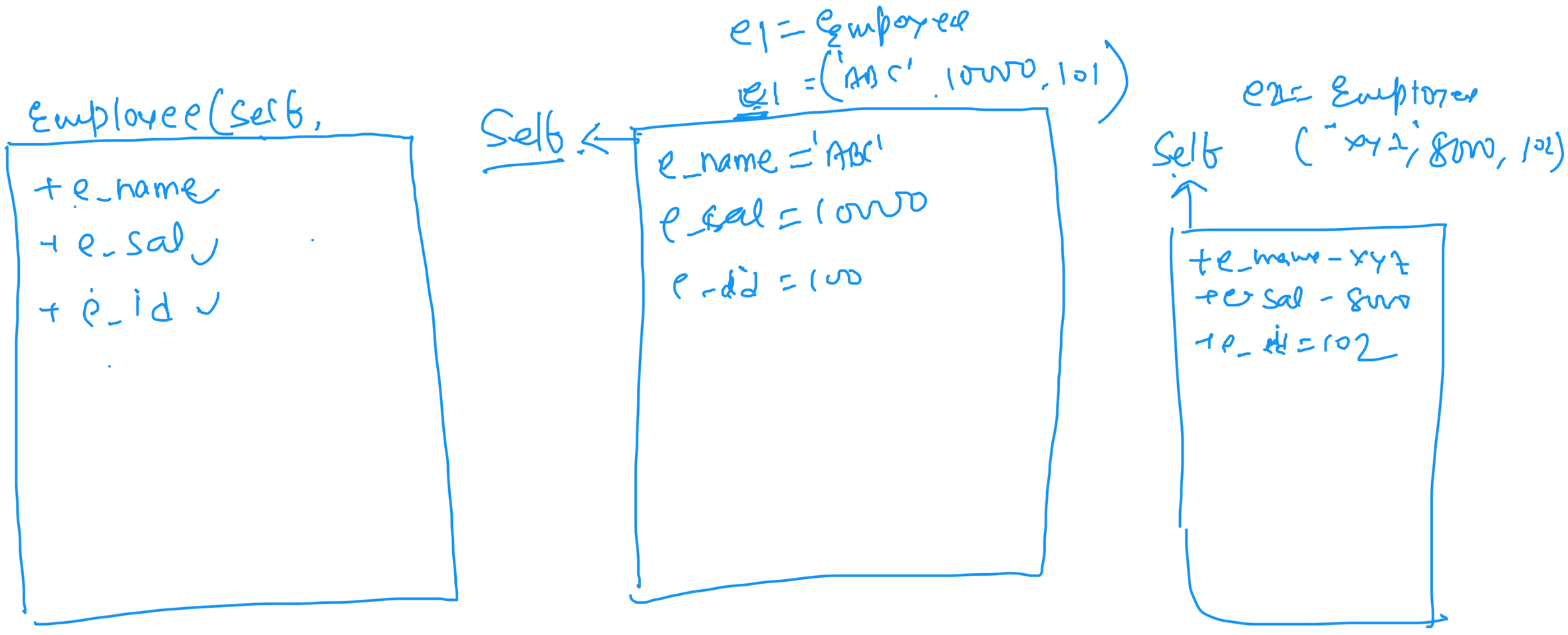
+ ename = 'xyz'
+ esal = 8000
+ eid = 102

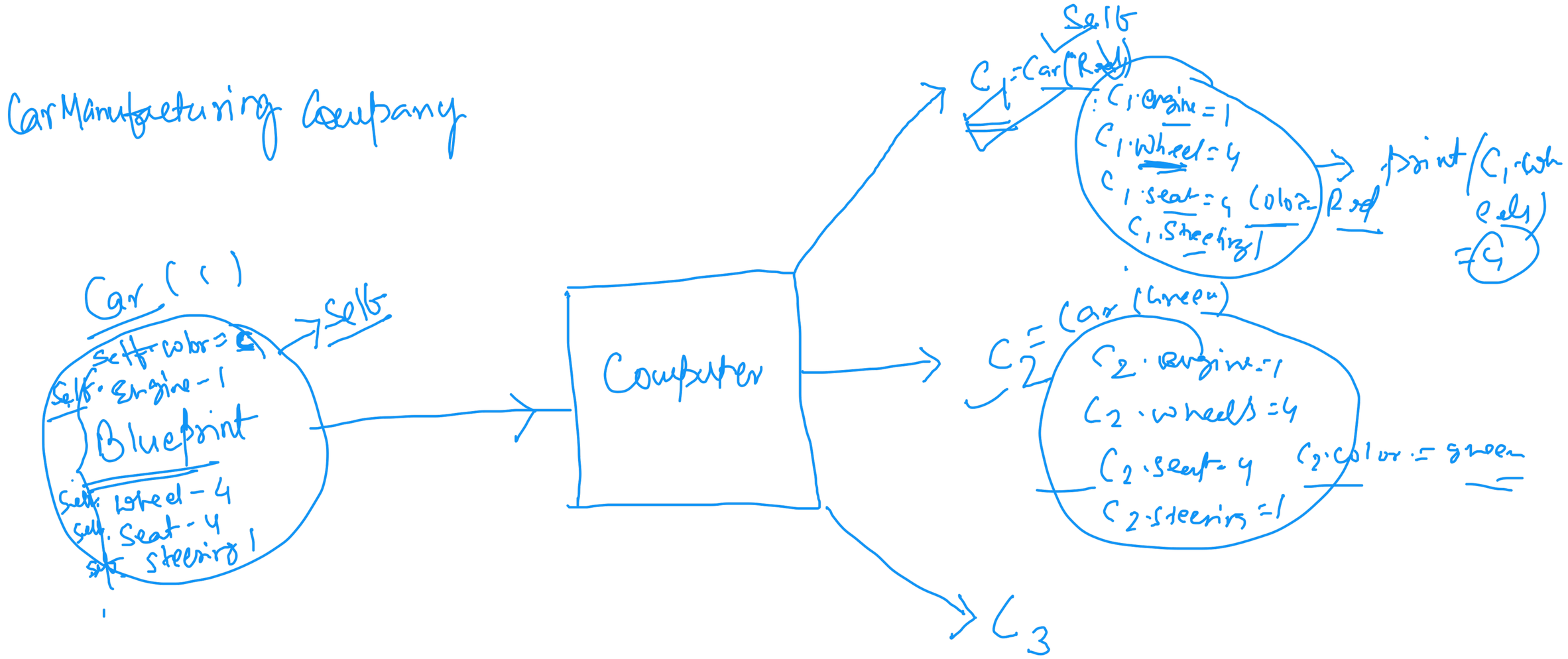
+ emp_detail()
+ Increment_sal()

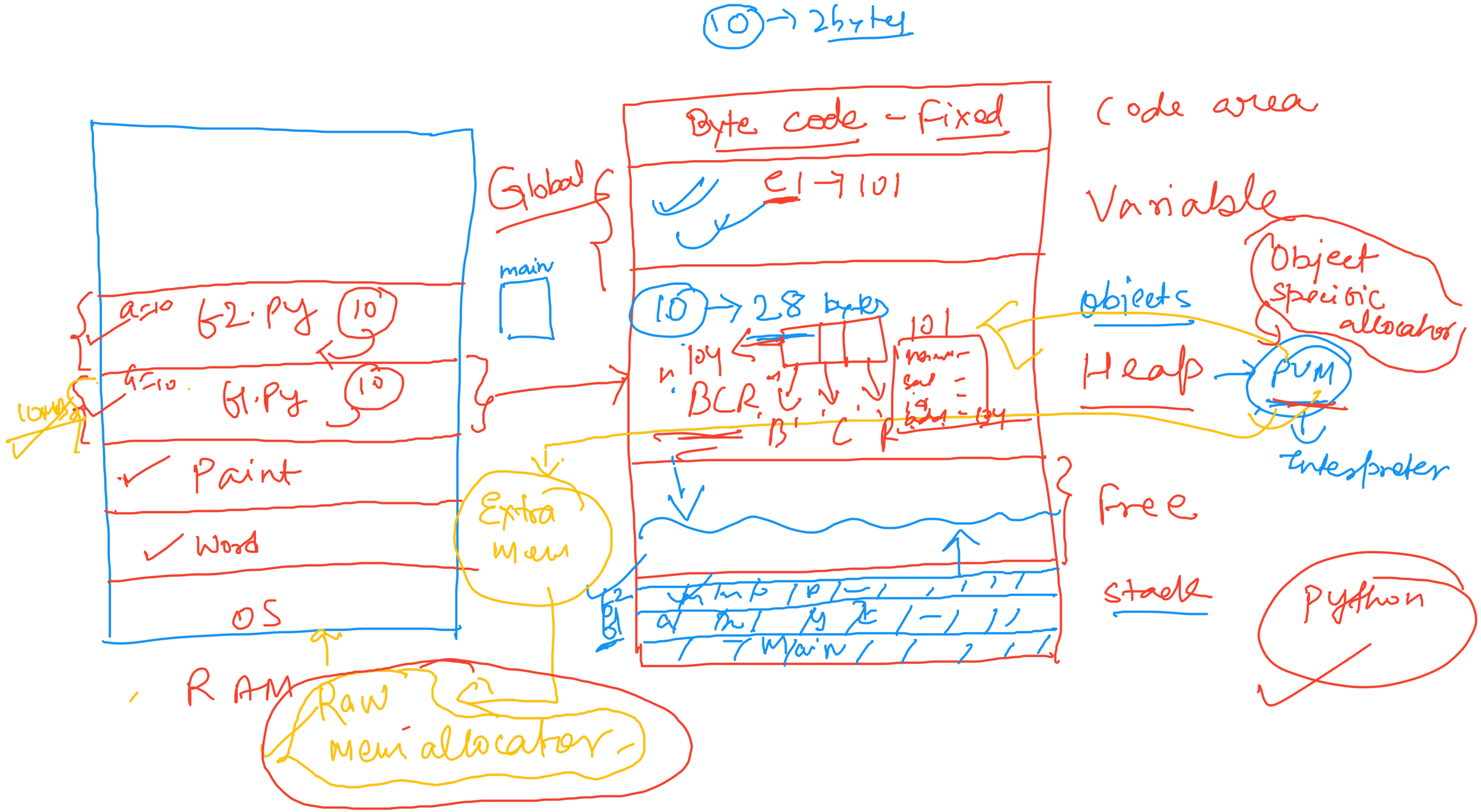
There Memory allocation will happen for object - Heap area

PVM - Python Virtual Machine


Carpenter
Class/Blueprint
length
width
Height
log =
Drawer -
t1
t2
Int
float
Complex
None
Boolean - True/False
String - "BLR"
List
Tuple
Set
Frozenset
Dictionary
range
zip
Byte
Bytearray
enumerate
filter
map
generator
Expression
function
20
a=10
10
Class Int
{ data
sizeof
refcount
add, abs, gt, lt, ...
}
data 10
sizeof = 28
ref = 1
add
abs(10) = 10
gt (


Employee(self,
+e_name
+e_sal
+e_id
e1 = Employee
e1 = ('ABC', 10000, 101)
self
e_name = 'ABC'
e_sal = 10000
e_id = 101
e2 = Employee
self ('xyz', 8000, 102)
+e_name = xyz
+e_sal = 8000
+e_id = 102


Car Manufacturing Company
Car ( )
self.color = c
self.Engine = 1
Blueprint
self.Wheel = 4
self.Seat = 4
self.steering = 1
self
Computer
C1 = Car(Red)
self
C1.engine = 1
C1.Wheel = 4
C1.seat = 4
C1.Steering = 1
C1.color = Red
print(C1.wheels) = 4
C2 = Car(Green)
C2.engine = 1
C2.wheels = 4
C2.seat = 4
C2.steering = 1
C2.color = green
C3


10 → 2 bytes
Byte code - Fixed
Code area
Global
c1 → 101
Variable
main
10 → 28 bytes
objects
Object specific allocator
a=10
b2.py
10
b=10
b1.py
10
Heap
PVM
Interpreter
BCR
Paint
Word
OS
Extra mem
Free
stack
Python
RAM
Raw mem allocator

Global → Any name defined outside the function is called Global identifier

Local → Any name defined inside the function is called local ~~variable~~ identifier

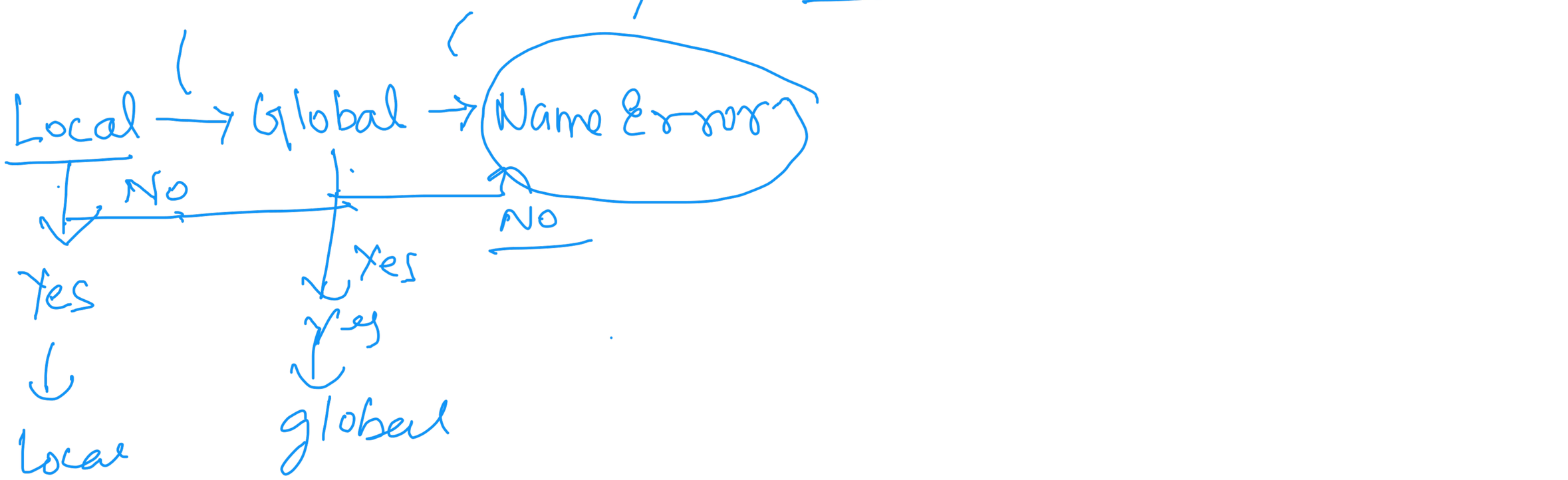
Variable / constants ⊂ Identifiers

↓

Any name

↓

{Variable, constant, Function, Class, Objects}




function Definition → def —

Function Call → fun name() → function will be executed

```
a = 10
b = 20
c = a+b
print(c)
```

10 + 20

101 + 102

103 →

a → 101

b → 102

c → 103

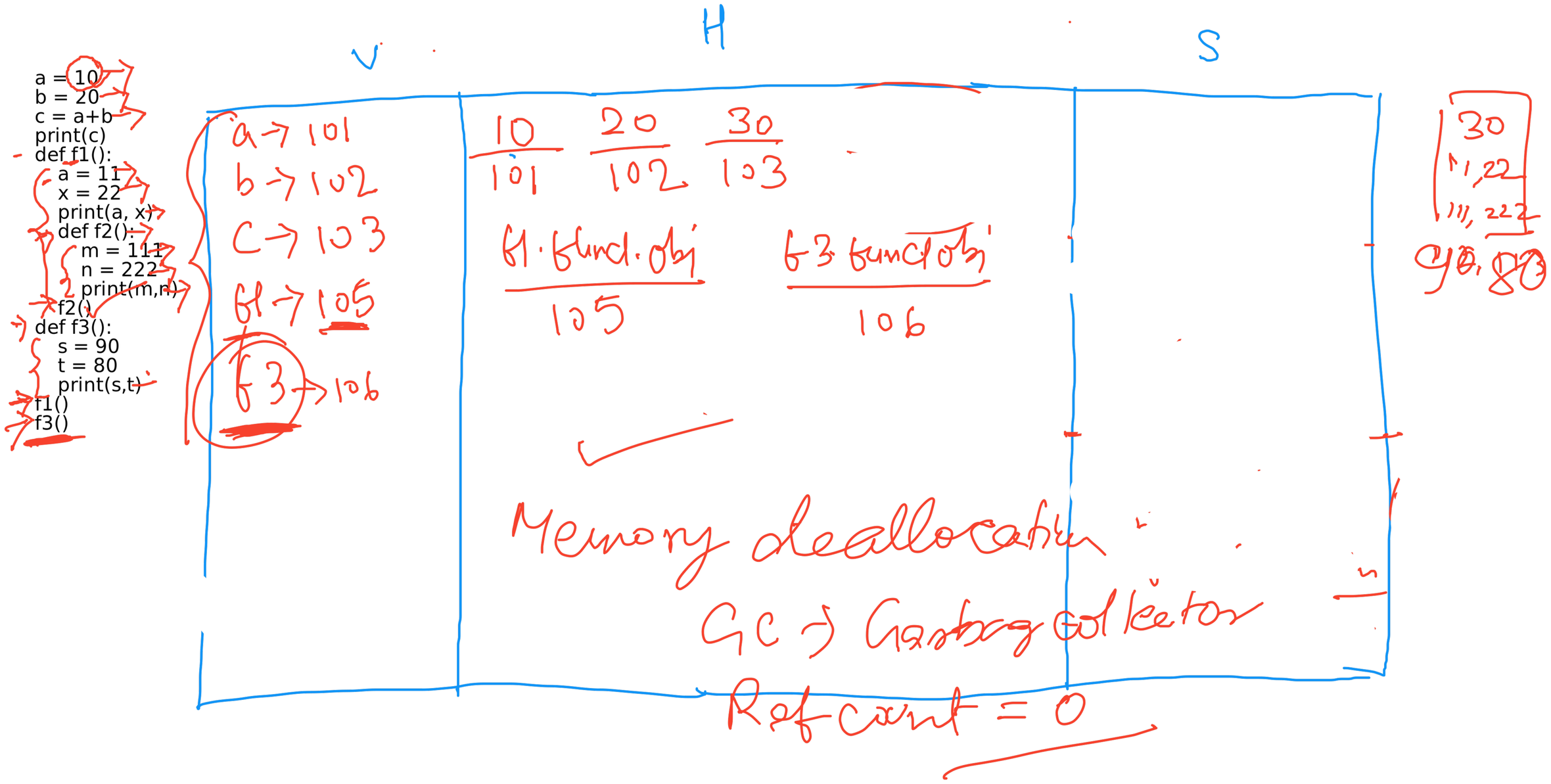
| 10 | 20 | 30 |
|---|---|---|
| 101 | ~~102~~ | 103 |

30

U

If

S


V
H
S
a = 10
b = 20
c = a+b
print(c)
def f1():
a = 11
x = 22
print(a, x)
def f2():
m = 111
n = 222
print(m,n)
f2()
def f3():
s = 90
t = 80
print(s,t)
f1()
f3()
a → 101
b → 102
C → 103
f1 → 105
f3 → 106
10
101
20
102
30
103
f1 · func1 · obj
105
f3 func1 obj
106
30
11, 22
111, 222
90, 80
Memory deallocation
GC → Garbage collector
Ref count = 0