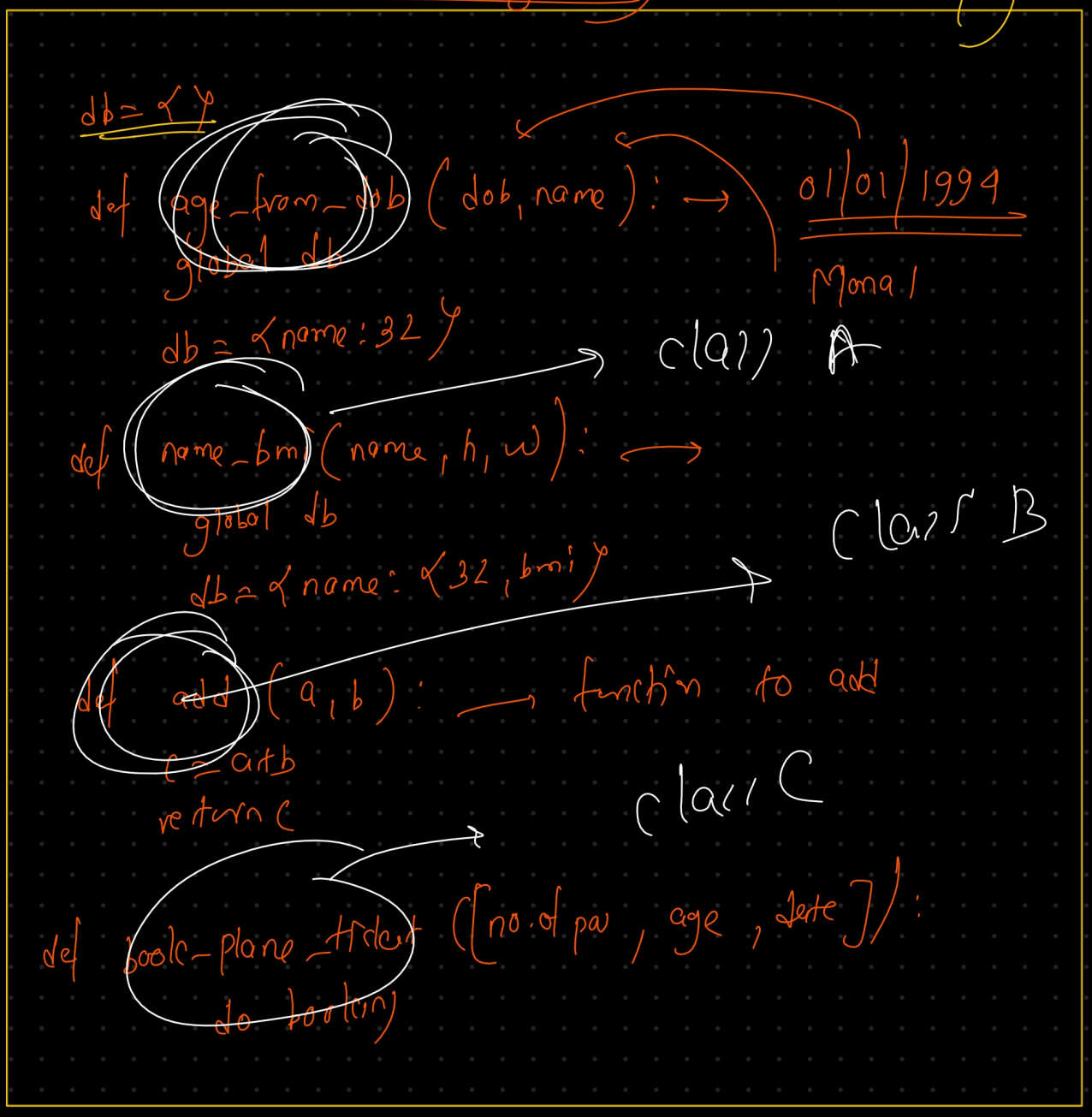
27-04-2025

Agenda

→ OOP
→ Module
→ Package

OOP - object oriented programming

test.py



Problem:

(1) Code becomes unmanagable.

(2) data is not in standard format → things will start to break
→ modify data

Solution:

→ scale
→ manyable
→ debug

← OOPs →

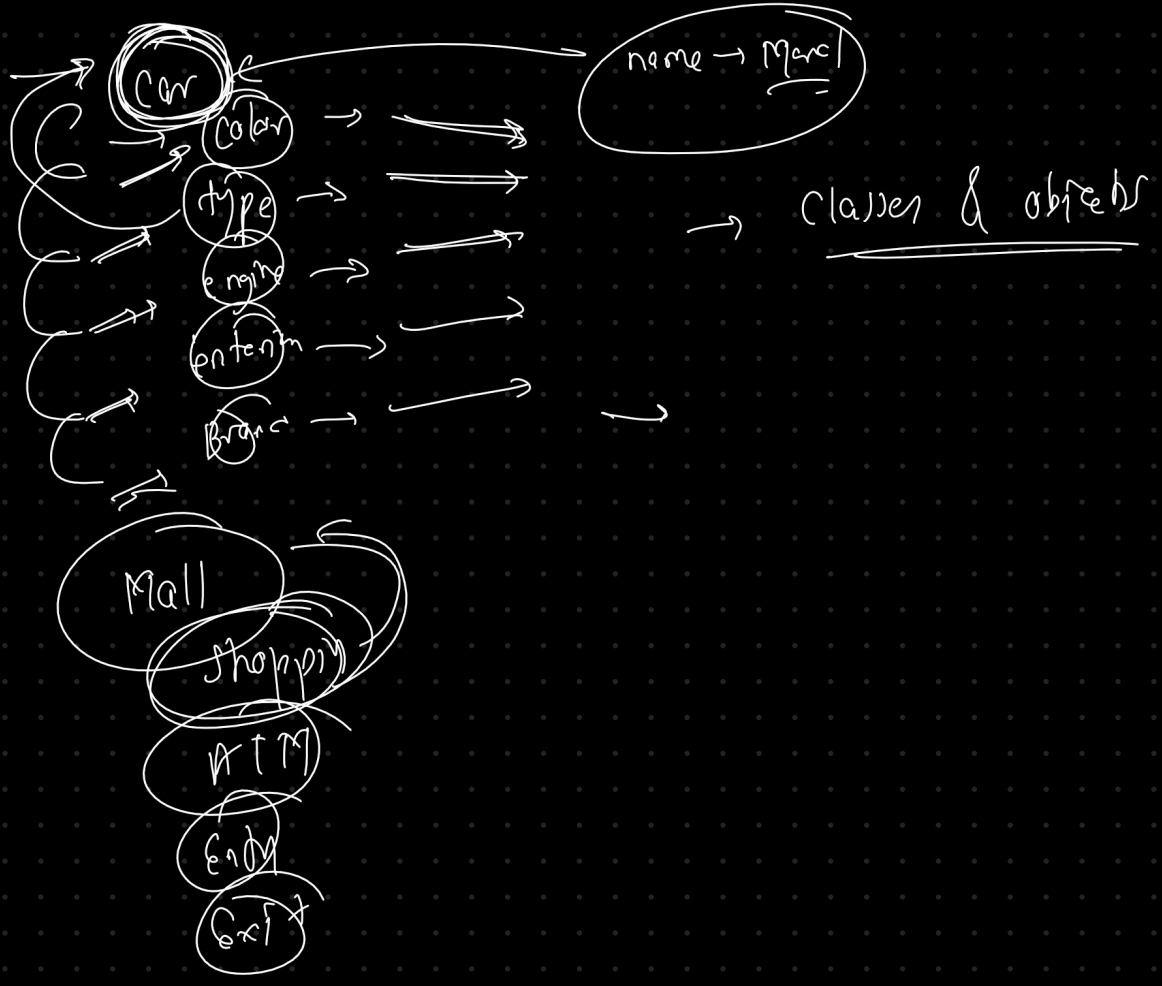
CAR

problem statement: To buy a car. more

→ — color → red → paint color
→ — Engine → 80hp
→ — Type - Sedan, SUV, Other → Sedan
→ — Brand
— — Interior color

→ lots of functions
— maintain the variable

bill



P.J → class / function

(1) Add 2 number → function

(2) calculator → class
- → add
- → sub
- → div

(3) Attendance → function

(4) School →
- → Att
- → Attend
- → Exam
- → Teach
- → Team

def add

class Add

Example :

restart →

enter → control

→ user

→ wallpaper

→ Date

→ Time

set

set Mobile Date

class SetMobileData ( )

def __init__(self):

data

initialize something

→ Construction

→ Date

→ user_name

→ other details

factory → start →

→ open office

→ open cash register

→ start machine

→ check raw material

start

starting

creat

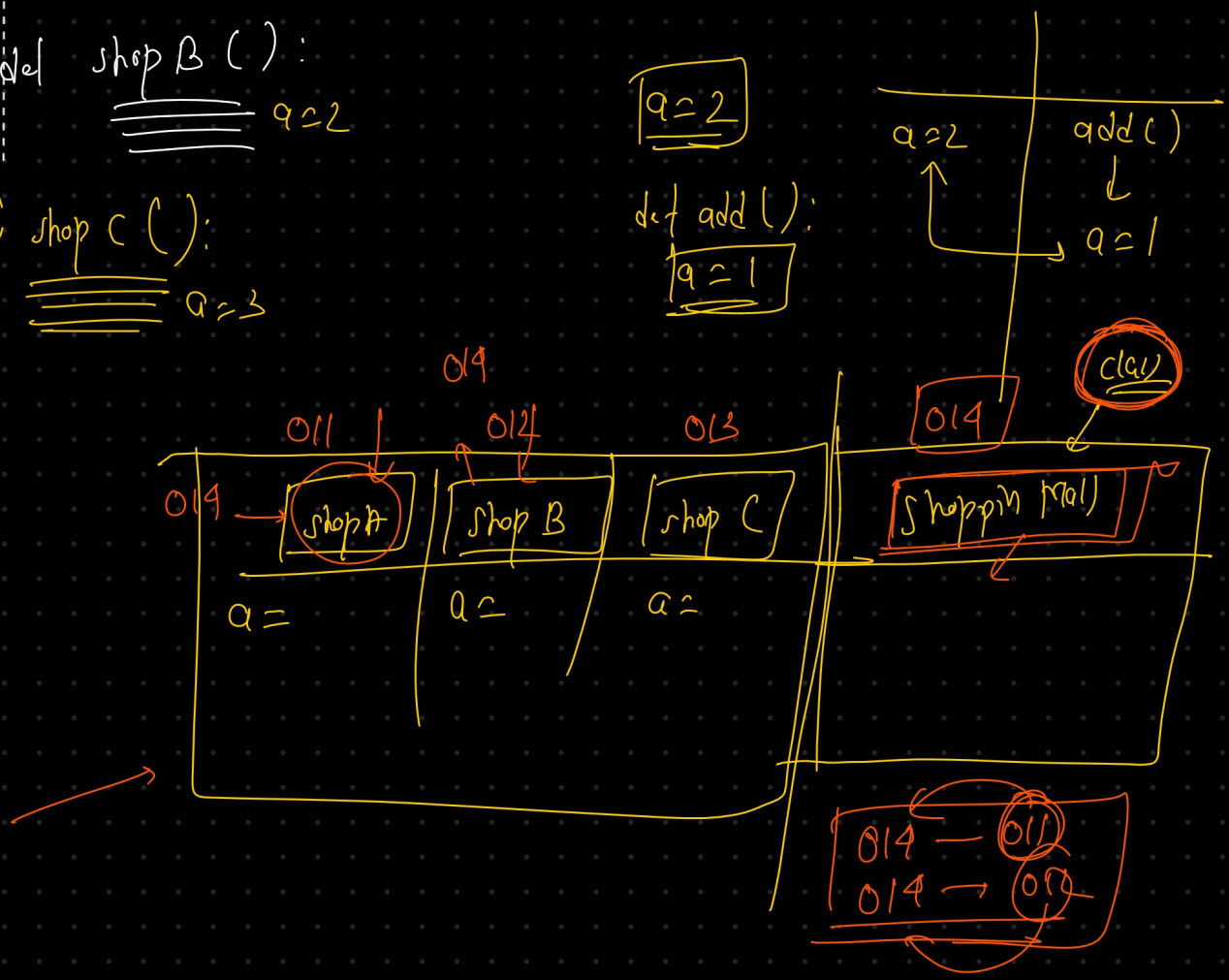
```
class ShoppingMall():
    def shopA():
        a=1
    def shopB():
        a=2
    def shopC():
        a=3
```

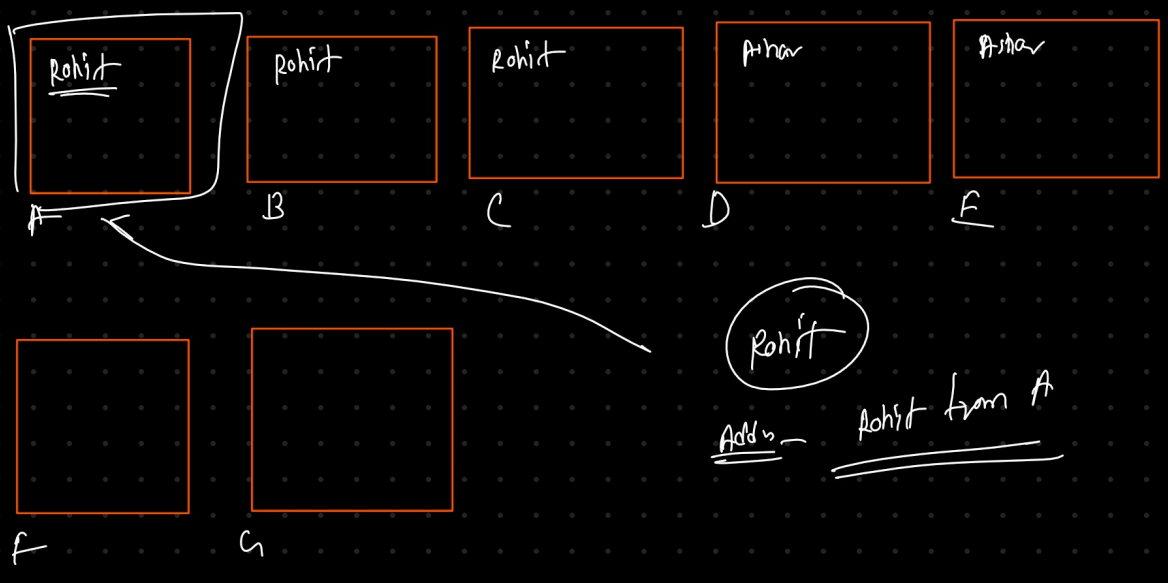
inside shopping

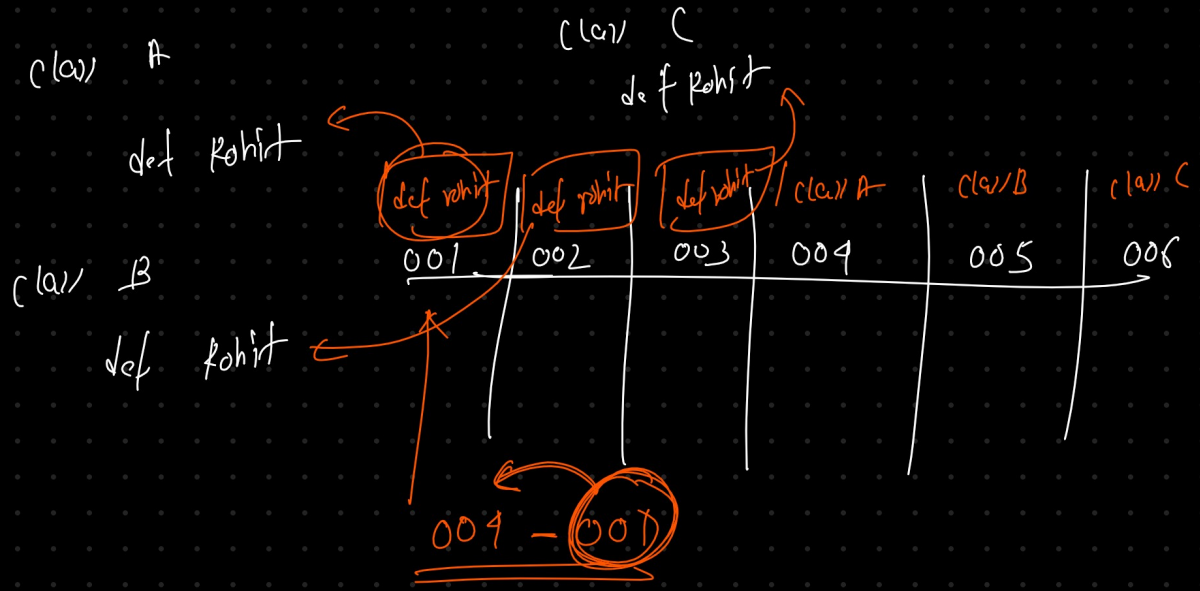
A & B

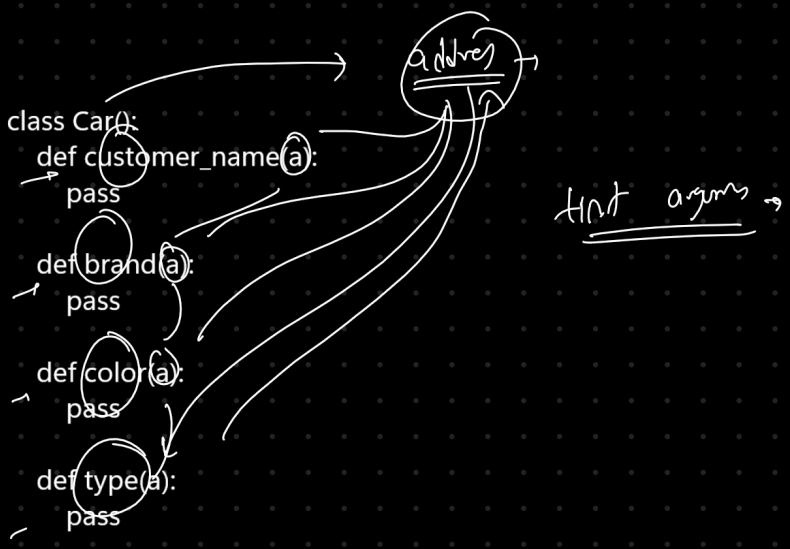
a=2

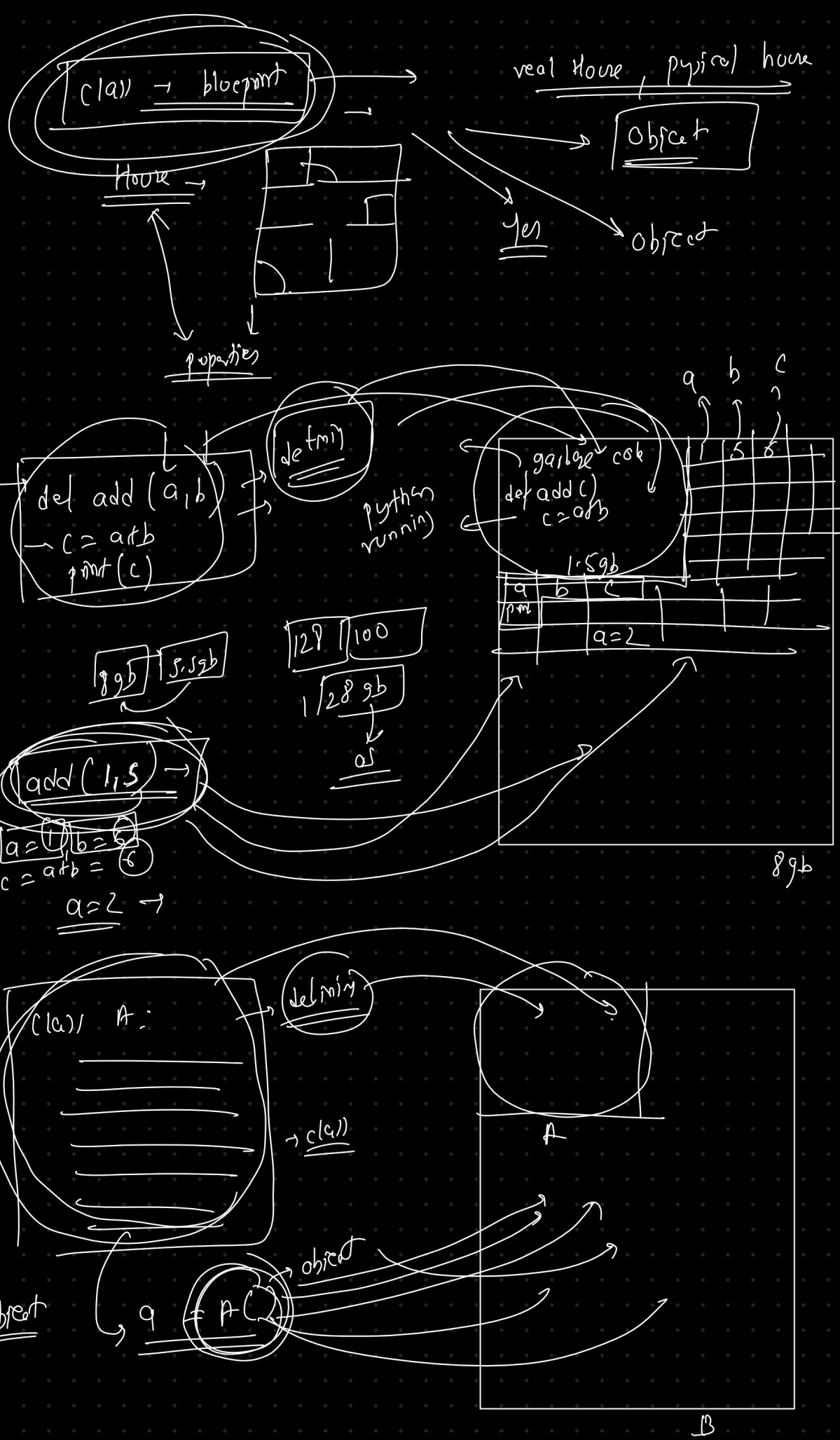
```
def add():
    a=1
```

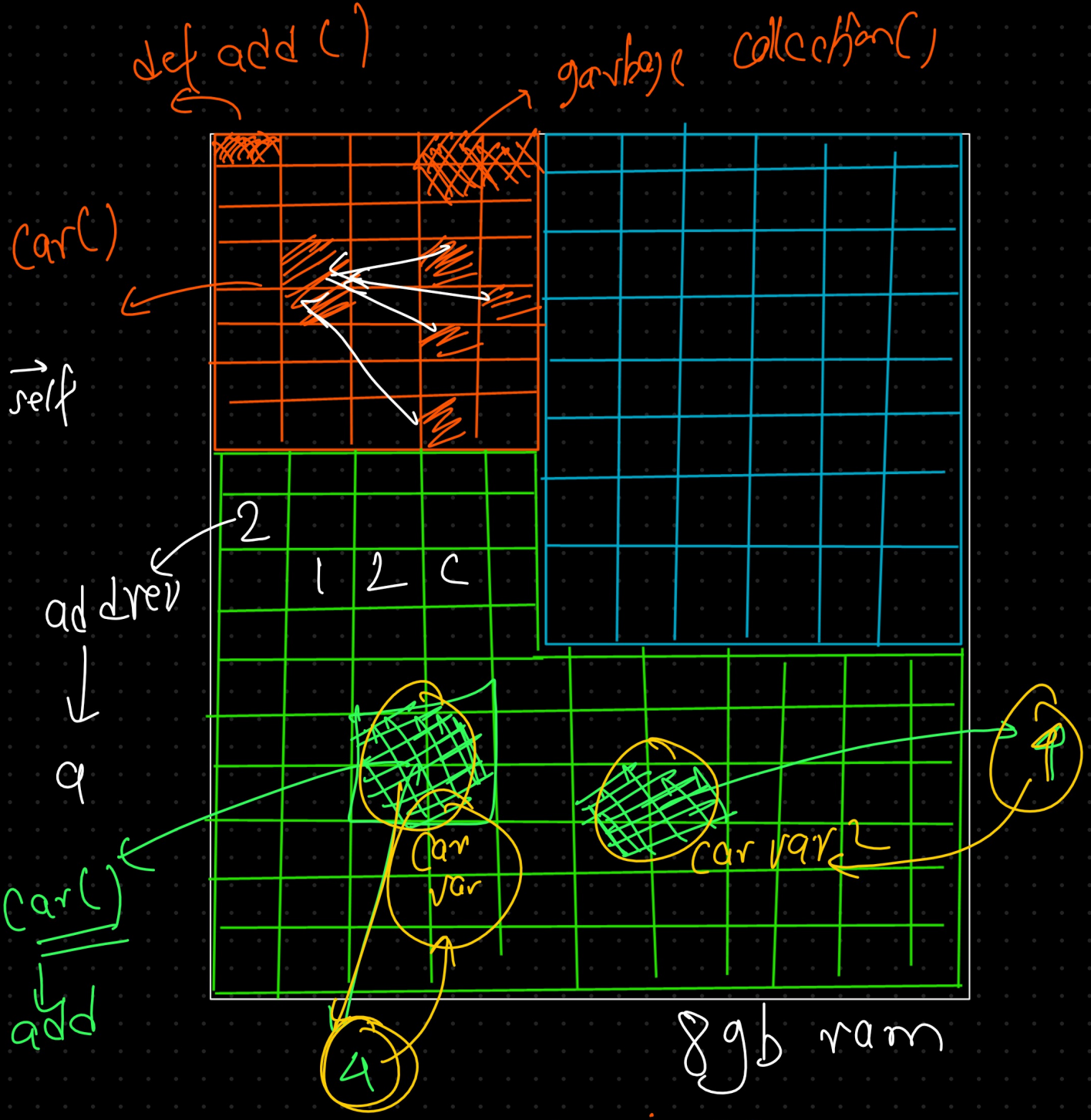





class A
def Rohit
class B
def Rohit
class C
def Rohit
def rohit
def rohit
def rohit
class A
class B
class C
001
002
003
004
005
006
004 - 001


address
class Car():
def customer_name(a):
pass
def brand(a):
pass
def color(a):
pass
def type(a):
pass
Hint args


class → blueprint
House →
properties
real House, pysical house
Object
Yes
object
def add(a,b)
c = a+b
print(c)
def init()
garbage coll
def add()
c = a+b
python running
a b c
5 6
1.5gb
a b c
print
a=2
128 100
128 gb
as
8gb 5.5gb
add(1,5) →
a=1 b=5
c = a+b = 6
a=2 →
8gb
class A:
def init()
→ call
A
object
a = A()
B

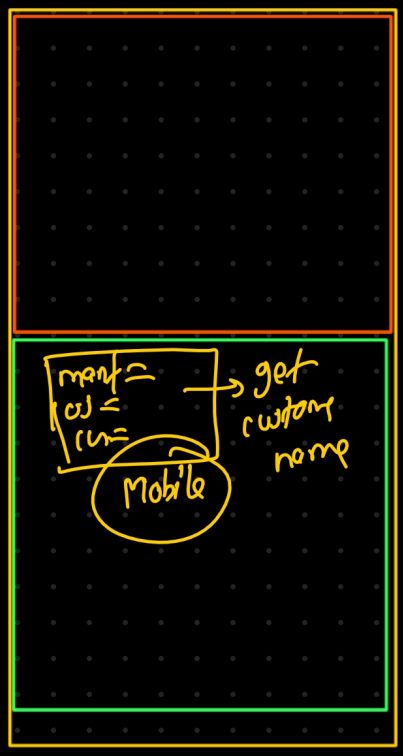


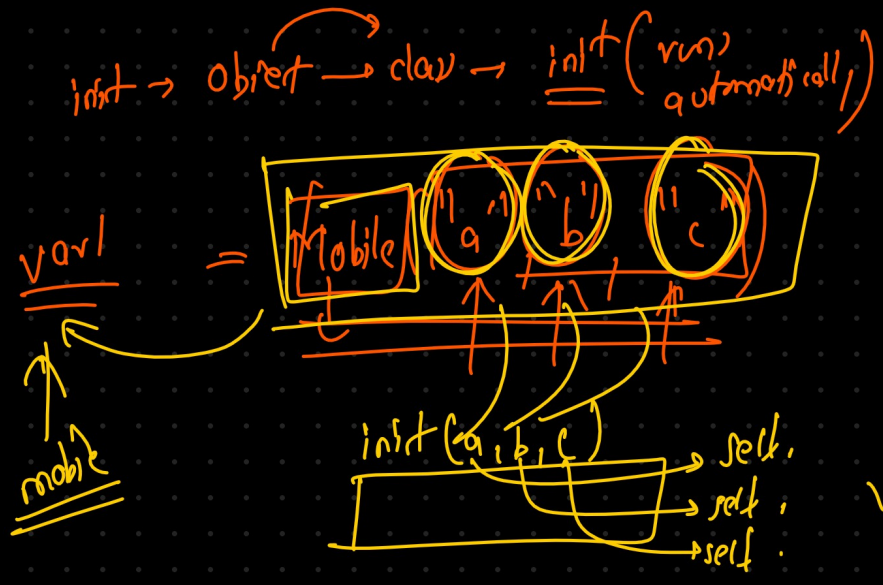
→ python running on this space

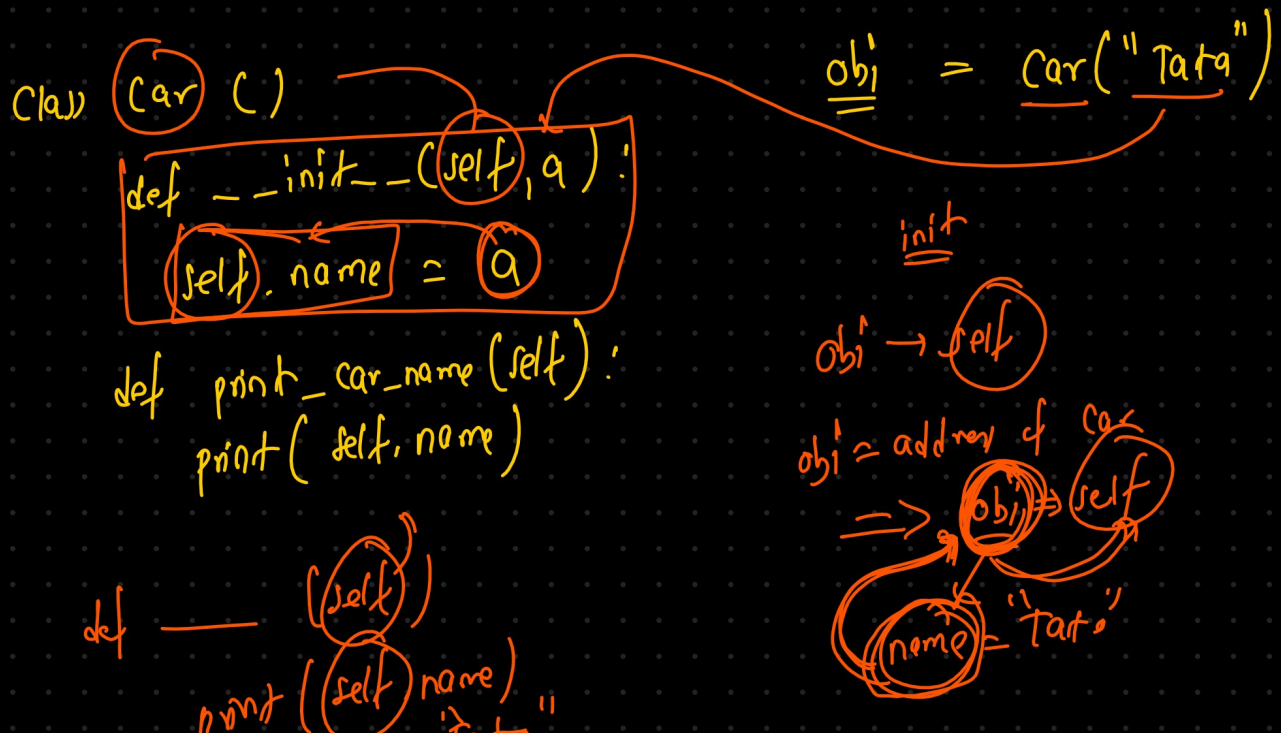
→ memory unused

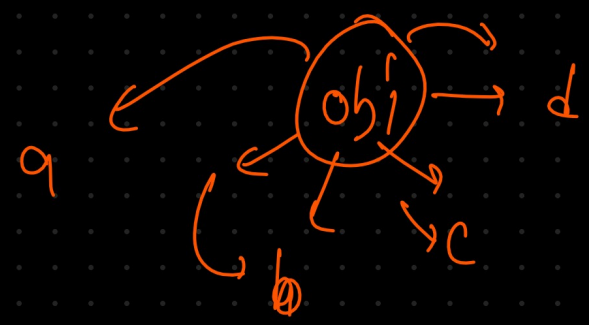
→ memeory used by OS

```
class Mobile():

    def __init__(self, manufacturer_name, os_version, customer_name):
        self.manufacturer_name = manufacturer_name
        self.os_version = os_version
        self.customer_name = customer_name

    def get_customer_name(self):
        print(self.customer_name)
```









def add (a,b): → function

c = a+b → variable

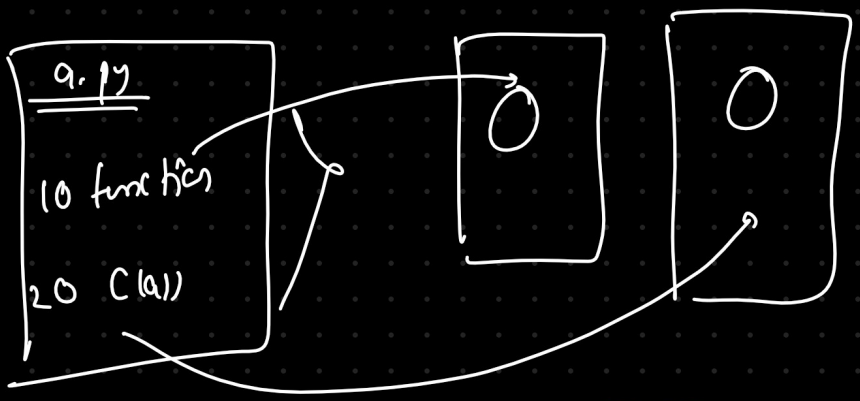
class
↳ attributes ← variable
↳ methods ← function

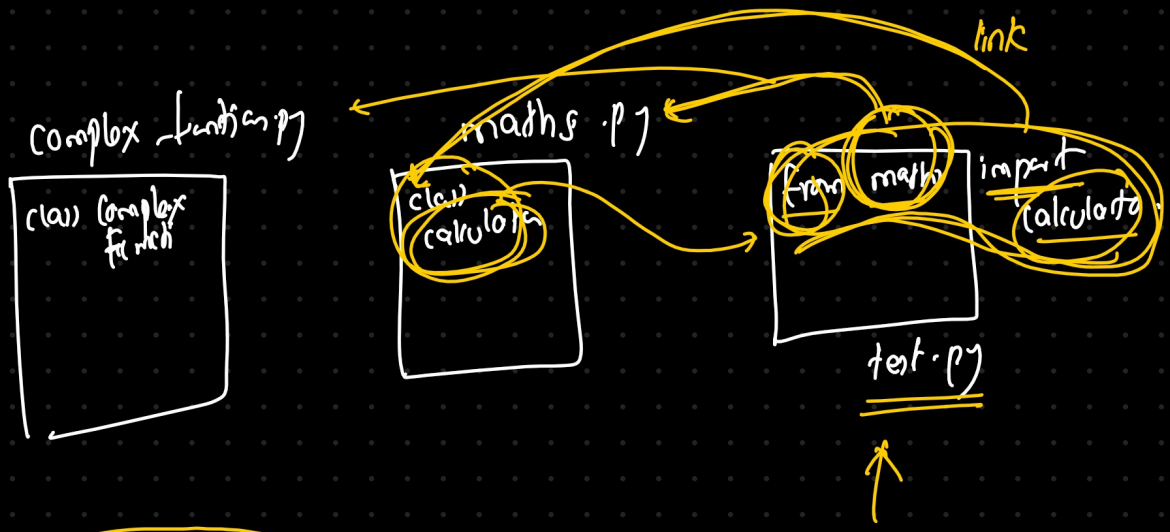
__init__ → constructor

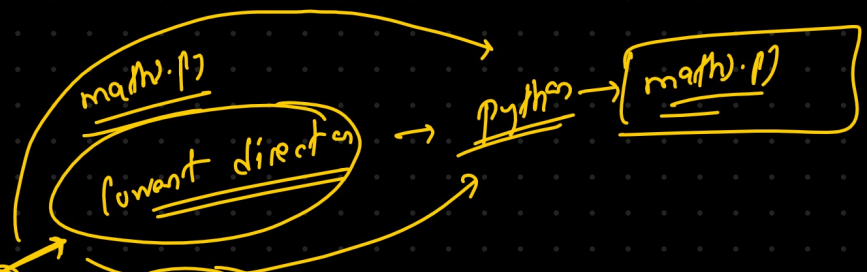
object / instance = car()

---

python module








```
from maths import Calculator
from complex_function import ComplexFunction
```

Assignment:

(1) you will ask user login or signup ?

→ if logic :

user_name & password → check

→ post ("Hey, how is everyone") →

→ posted.

→ see_all_post() →

if signup

user, password → store

You need to create signup and login page for a social media Platform-:
1. You have to ask user whether they want to login or signup
2. If user says login the we have to ask for username and password and then we have to validate user.
3. If user says sign up you need to ask username and password and store it.
4. Once user logged in then you need to ask user to Post-:
a. User can post on the platform
b. User can see all the posts done by user