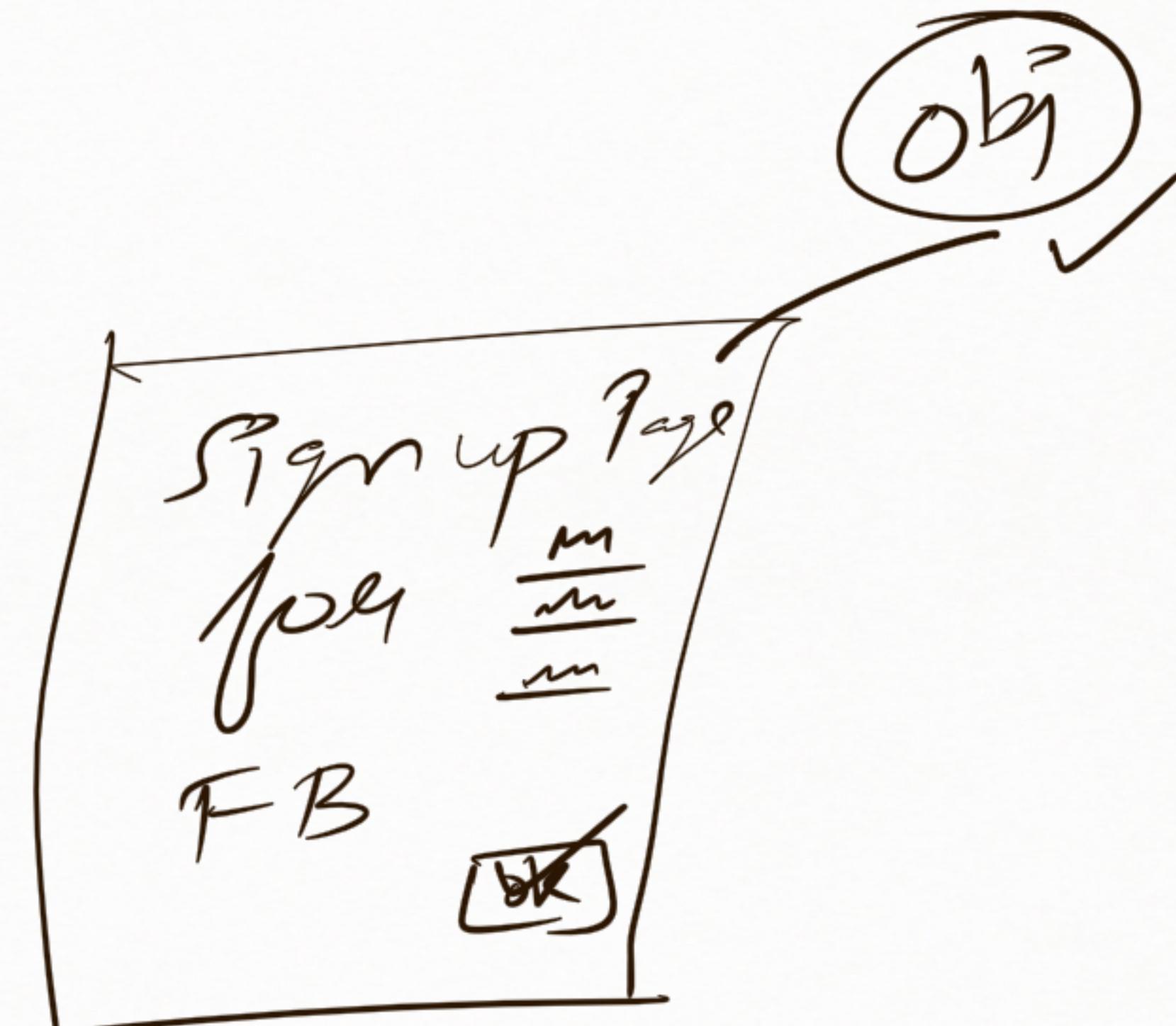
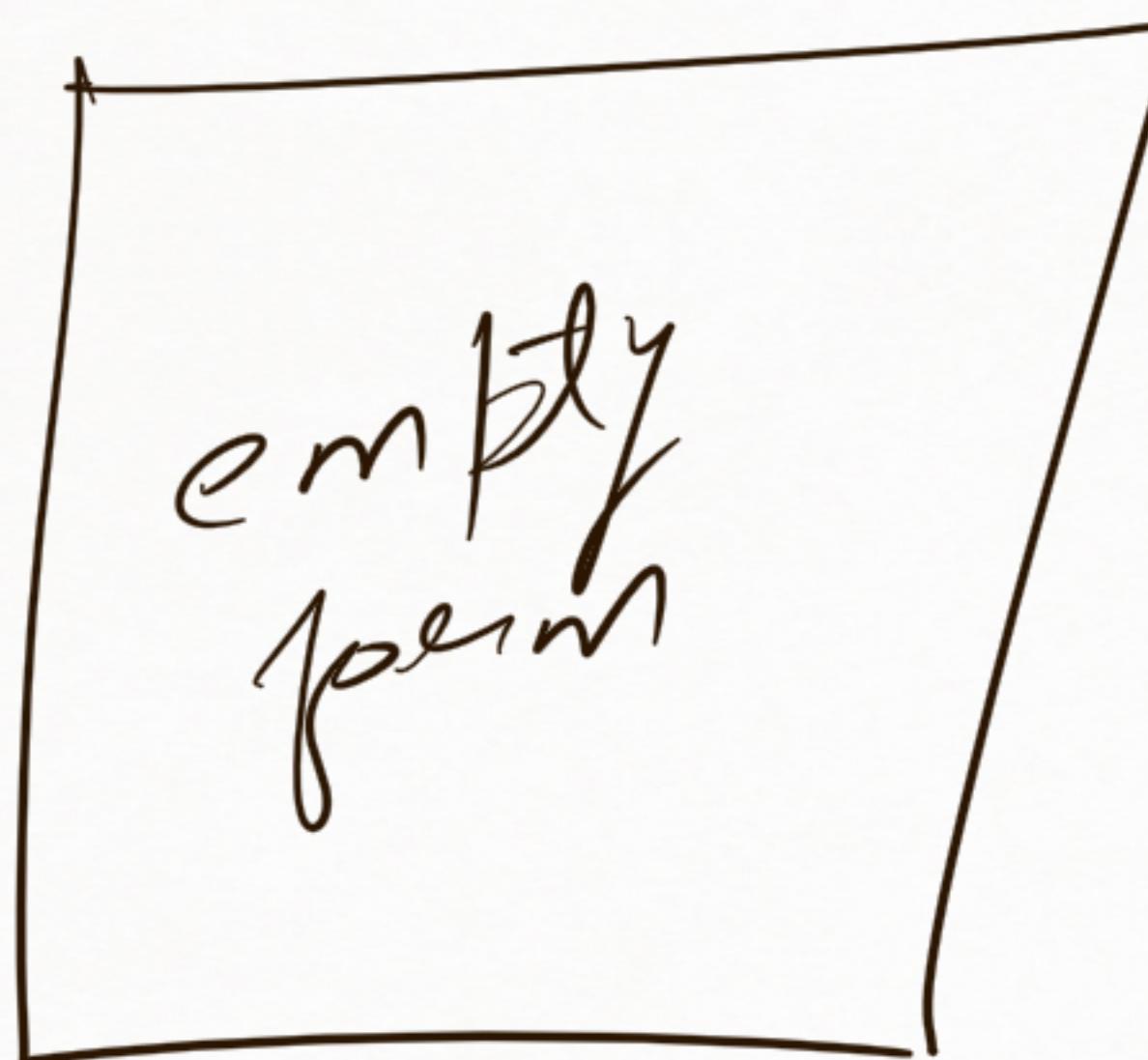


Class is a Blueprint

- Variables ✓
- Methods ✓



In [1]:

# OBJECT ORIENTED PROGRAMMING

In [ ]:

```
class Person:  
    #  
    # /method.  
    def __init__(self, name, gender):  
        print("Family is born with baby", gender)  
        self.NAME = name  
        self.GENDER = gender  
        self.friends = []  
  
    def eat(self):  
        print(self.NAME, "is eating")  
  
    def make_friends(self, friend):  
        self.friends.append(friend)  
  
object variables:=> NAME, friends is [], GENDER
```

When we Obj  
create this  
of class

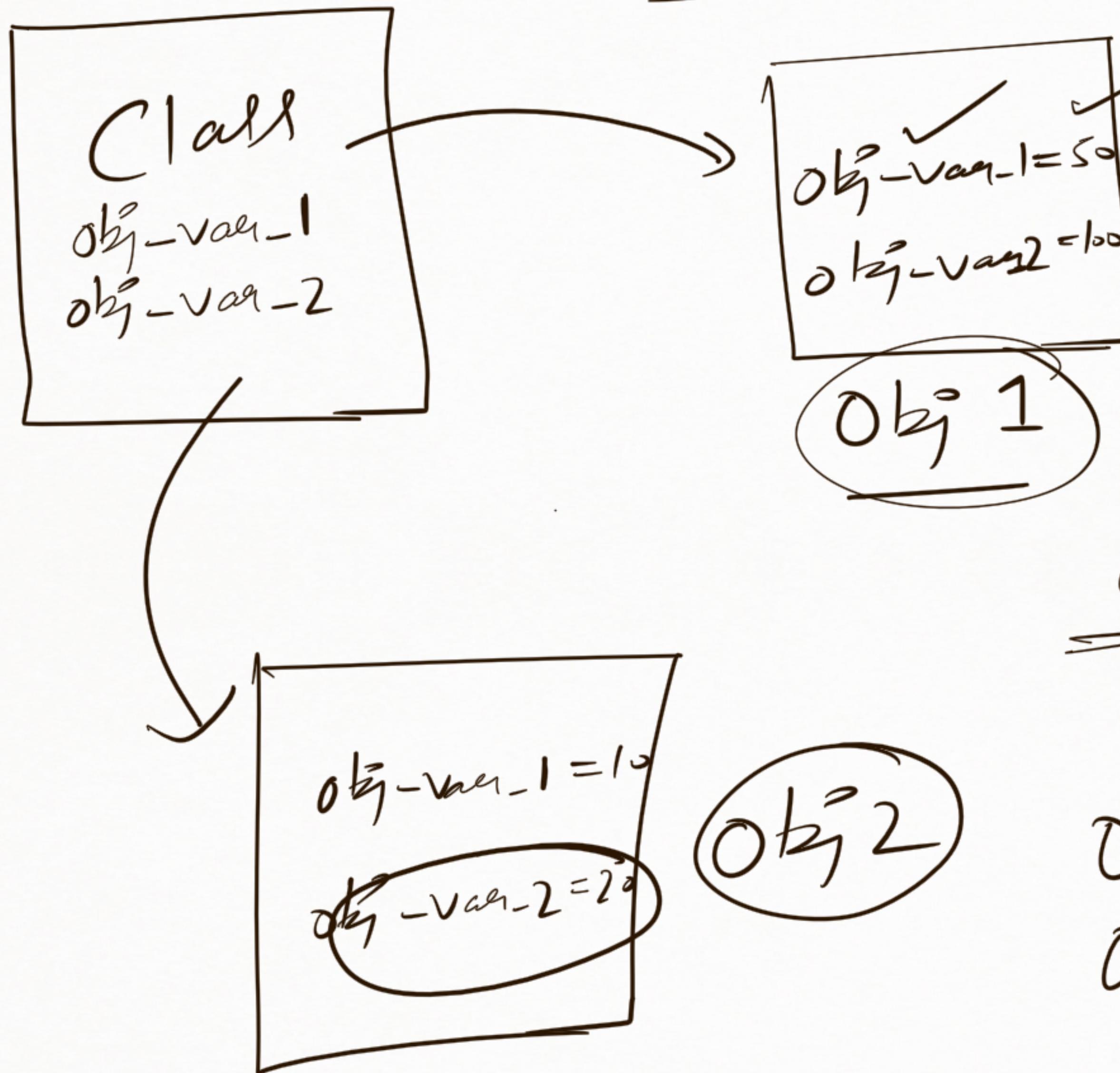
Self

Object

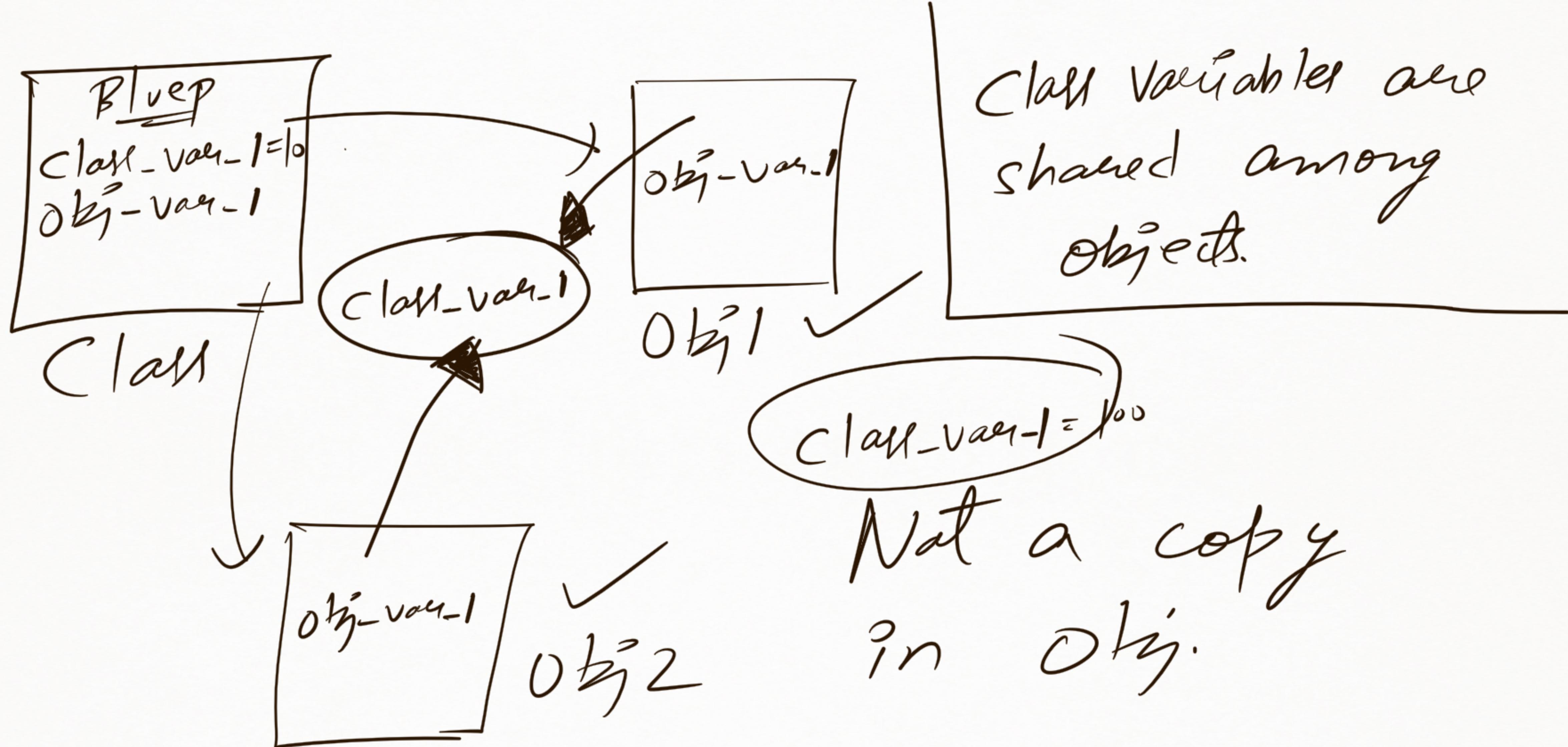
we can  
call these  
methods  
obj with obj  
Name =

Ankit = Person("Ankit", "M")

Object You have <sup>? it's</sup>  
own  
copy .



Obj 1 • obj-var-1 // 50  
Obj 2 • obj-var-2 // 20  
Obj 2 • obj-var-1 // 10  
Obj 1 • obj-var-2 // 100



```
In [52]: class Person:
    SPECIES = "homosapiens"
    def __init__(self):
        print("Family is born with baby", "gender")
        self.NAME = "name"
        self.GENDER = "gender"
        self.friends = []
    def eat(self):
        print("self.NAME", "is eating")
    def make_friends(self, friend):
        self.friends.append(friend)
    def setName(self, name):
        self.NAME = name
    def getName(self):
        return self.NAME
    @classmethod
    def changeSpecies(cls):
        cls.SPECIES = 'HOMOSAPIENS_PLUS_PLUS'
```

Obj  
Methods

Class Val

