Lambda functions

- Inline function (Anonymous function)
- We don't need def keyword
- logical expression only

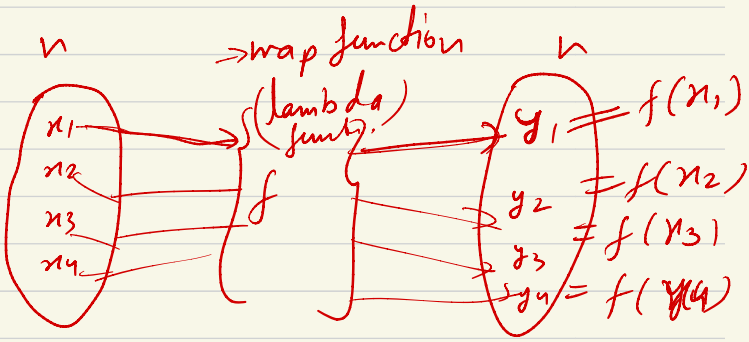
```
def function_name(a,b):
    ===
    ===
    ===
```

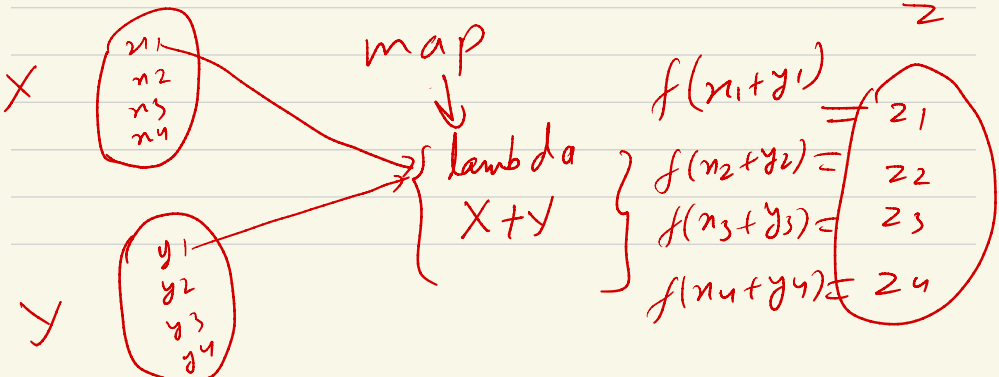
} User defined function

Syntax

lambda argument : expression

map()





Reduce()

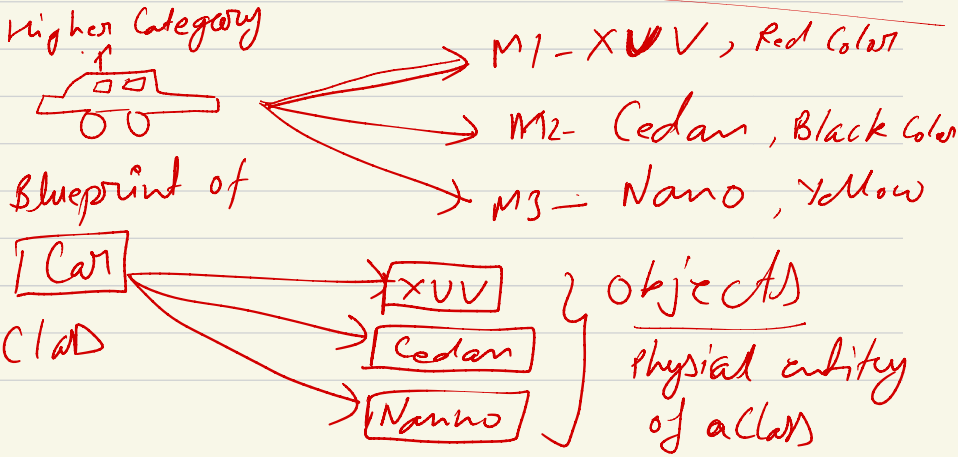
- At the end we will one final value (not a iterable)
- It just reduces the value given in a iterable to single value using some expression

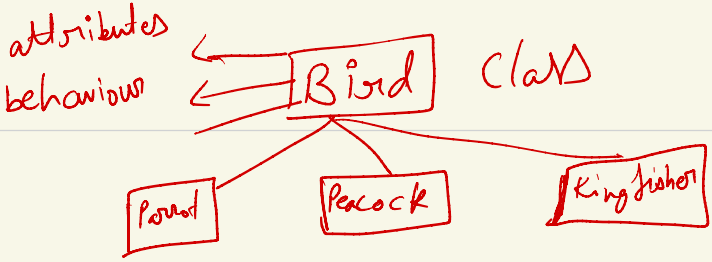
List Comprehension

Syntax: oldList

```
newList = [expression(x) for x in oldList if check ]
           (1)             (2)            (3)
```

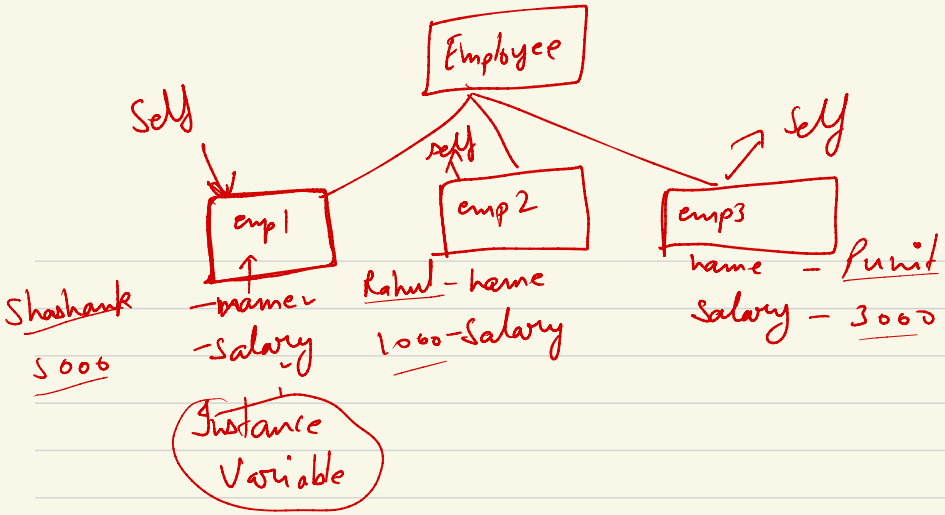
OOPs in Python (object oriented programming)





attributes → Stores values for a object

Behaviour → methods which shows the properties of a particular object



class Employee:

default → def __init__(self, name, salary): ← reference to the current object

self.empname = name

self.emp_salary = salary

def displayEmpInfo(self):

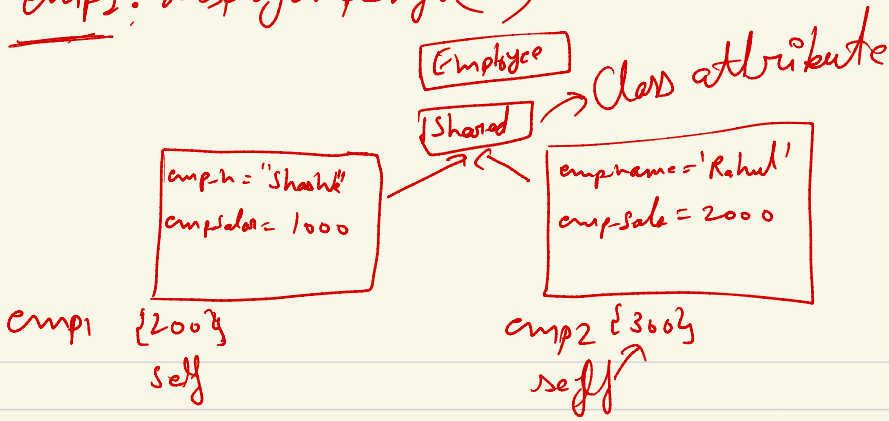
print(self.emp_name, self.emp_salary)

emp1 = Employee("Shashank", 1000)

x = 10 emp2 = Employee("Rahul", 2000)

y = 5

emp1.displayEmpInfo()



emp1.emp_name } 200.emp_name

emp2.emp_name } 300.emp_name

Employee → empCount = 0

emp1 = Employee() Count = 1

emp2 = Employee() Count = 2