
G
O
D
O
O
O
O
D

OOPs

Python

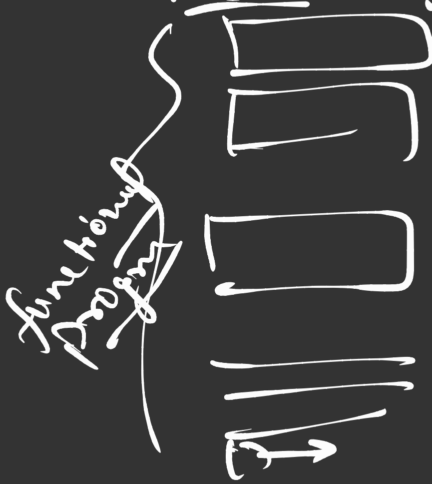
Programming lang.

01010001

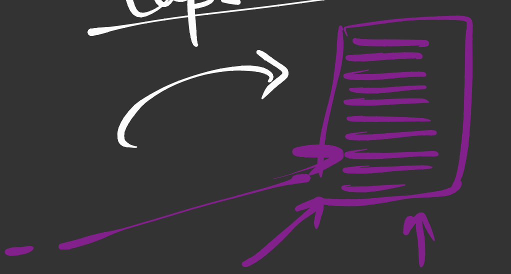
language

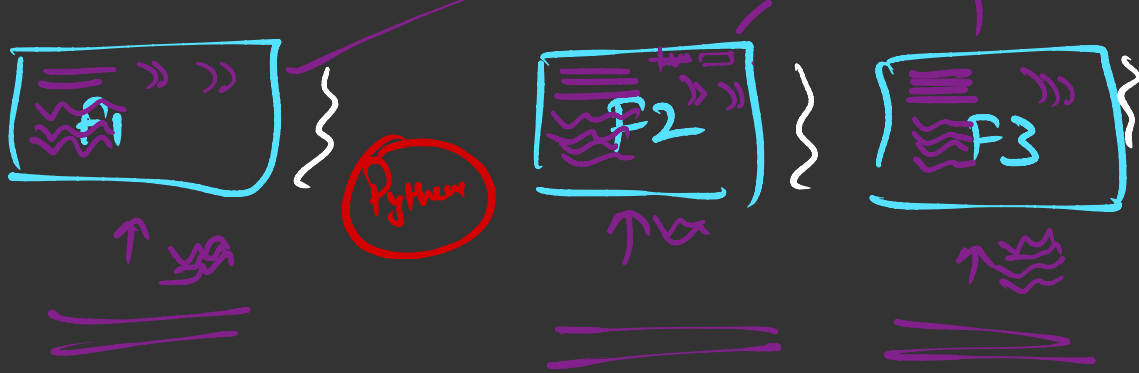
What hap in bw you learned?

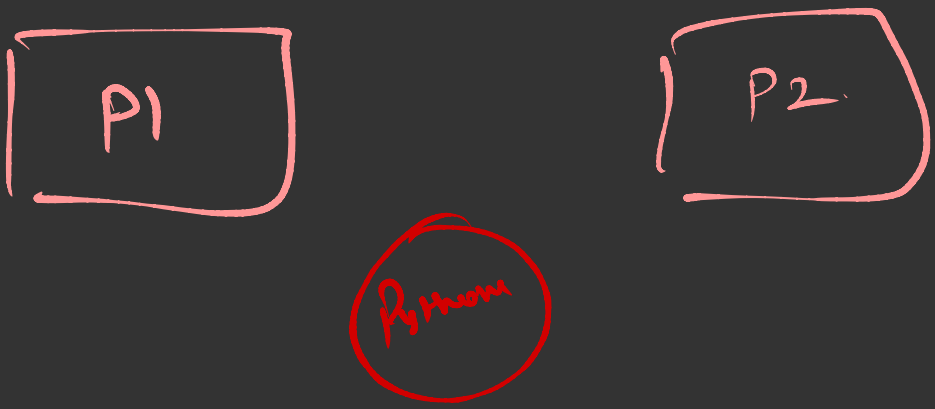
basic → function → Object Oriented Programming



Capsulation







Encapsulation

↳ pillar of OOPs

OOPs

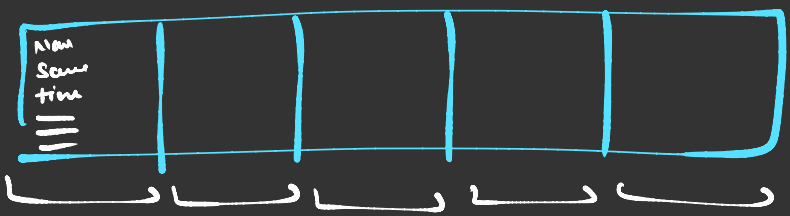
1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism ✓✓

l = list()
l.append()
l.pop()
l.sort()
l. ...

Leaderboard

1 : Nm : Sc
2 : ___ : ___
3 : ___ : ___

list

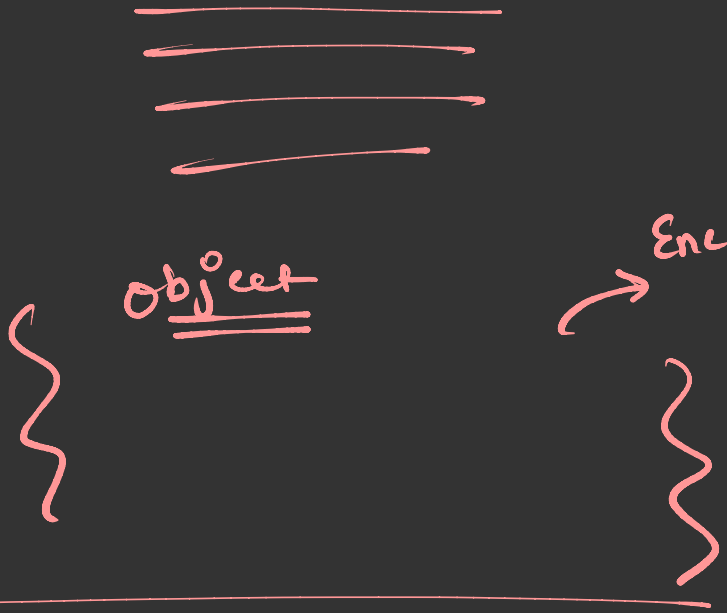


l.sort()

We want to build own DS.

.add(_____)

.update(_____)



↳ [ ]

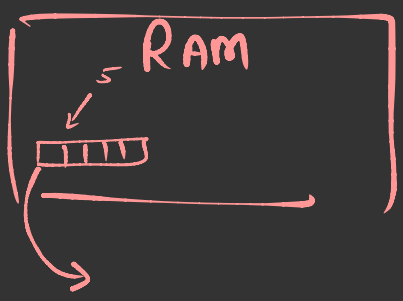
list()

[1, 2, 3]

list(1, 2, 3)

Simplified Version of Real Coding

l.index(-)




```
int* li = new int[5];
li[0] = 1;
li[1] =
```

Hardware

```
for(int i=0; i < n; i++){
    if(arr[i] == val){
        return i;
    }
} else{
    return -1
}
```

Abstract

Encapsulation

sum(5,2) # 7

sum(10,20) # 30

sum (100,50) # 150

sum (100,50,50) # 200

sum(1,1,1,...1) # 10

---

functions

1, 2, 3

```
def sum(a,b){
    return a+b
}
def sum(a,b,c){
    return a+b+c
}
```

function overloading

↓

Polymorphism

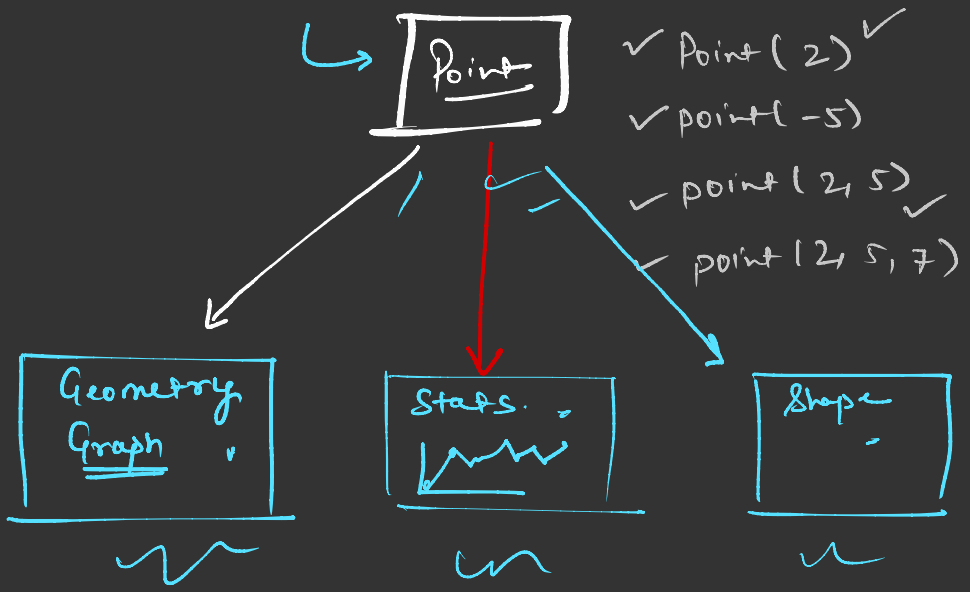
3

---

2 + 3 = 5

[1,2,3] + [4,5] = [1,2,3,4,5]

"Add" + "Vaibhav" = AddVaibhav



① Data Member : variables inside a class / object.

② function → methods.

```
def prime():
    ...
    ...
    ...
```

list class.

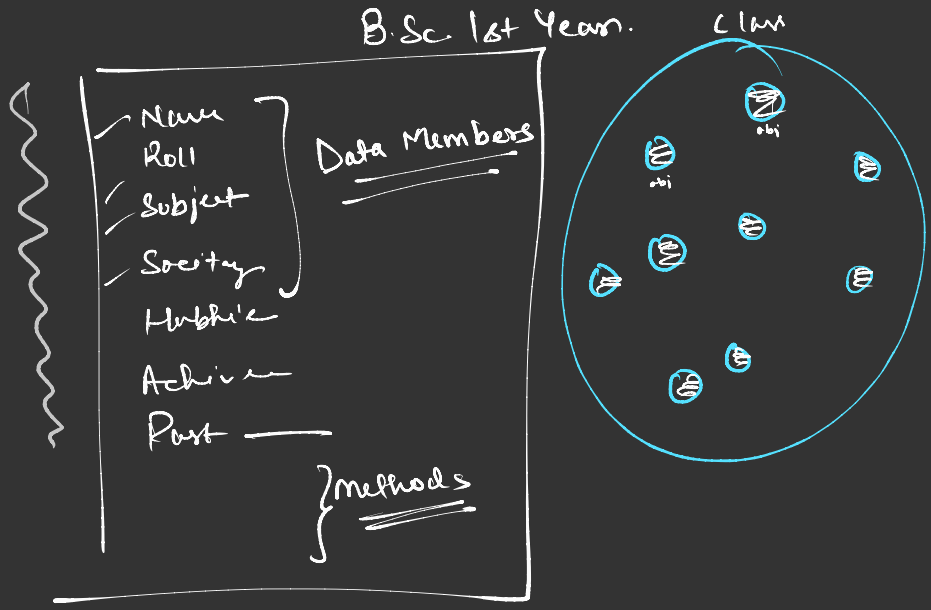
```
l.append(prime(5))
```

→ my func

→ function

↳ method

Class → Generic



Constructor

↳ initializer }

Object Creation ke time par.

= P1.distance(P3)

= P1.distance(P3)

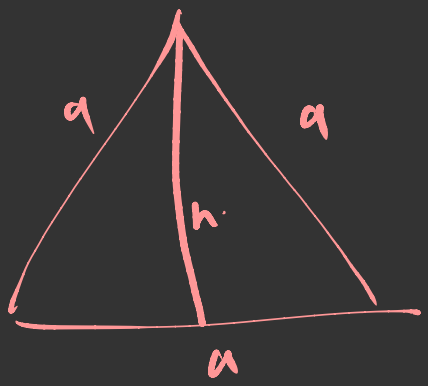
↖

l.append(s)

P1.distance(P2)

(x_1, y_1) (x_2, y_2)

$$\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$



retu pr
Area

(2* Self.area()) /
Self.a

$$\frac{1}{2} \times a \times h = \text{area}$$

$$h = \frac{2 \cdot \text{area}}{a}$$