OOPS (Object Oriented Programming System)

Monday, July 14, 2025 7:15 PM

Loop
1. Code repetation(-)

Functions:
1. Code repeatation (-)
2. Code reusability(+)

OOP:
1. Code repeatation(-)
2. Code reusability(+)
3. Proper structure
4. Code Flexibility
5. Work on real time entities
6. Collaboration

OOP is a programming that deals with class and object.

Class: It is a blueprint which consists of the features and functionality of the real time entity.

Object: It is the instance(copy) of the class.

- For 1 class, there can be n number of objects.
- Data present in class will be automatically present in it's objects.

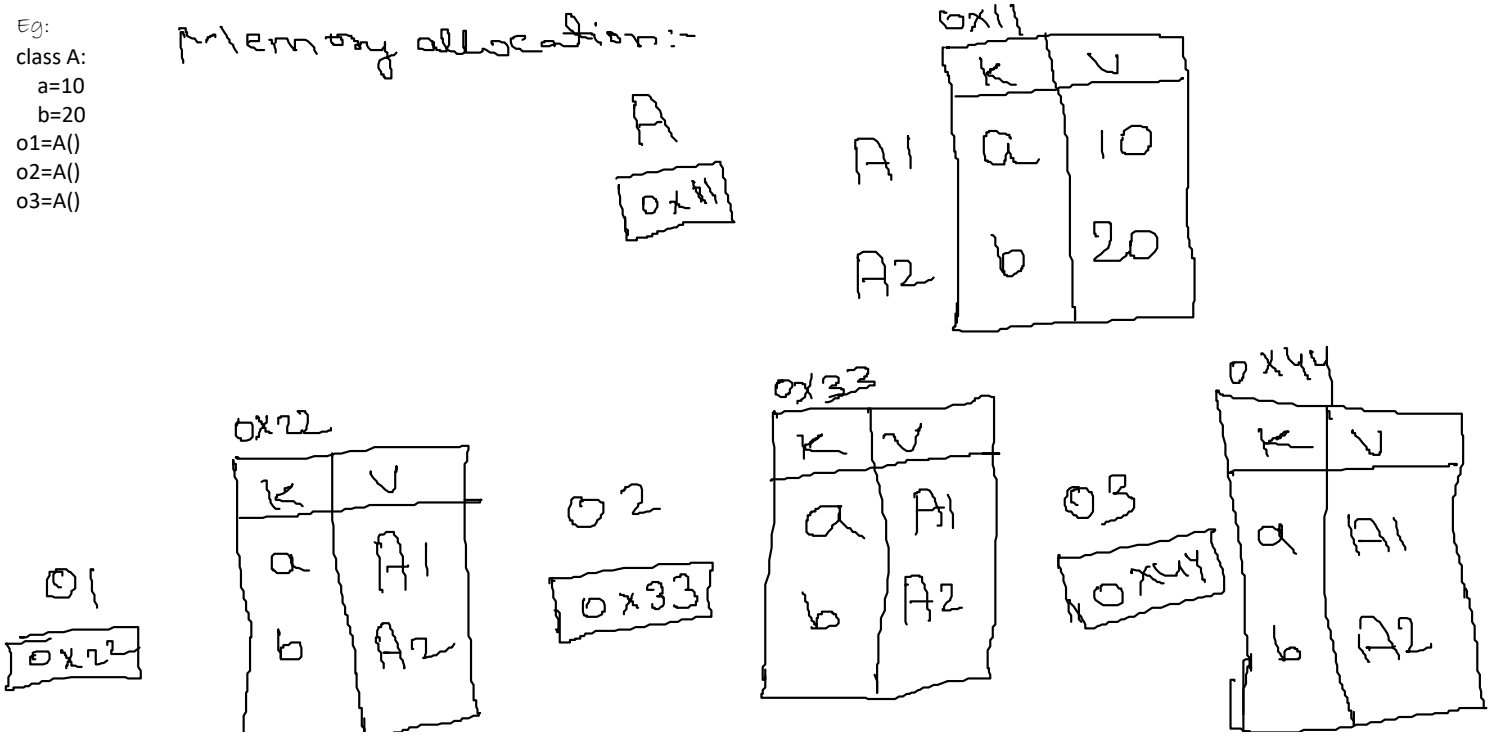
Class Creation:
class Cname:
    Properties
    Functionalities

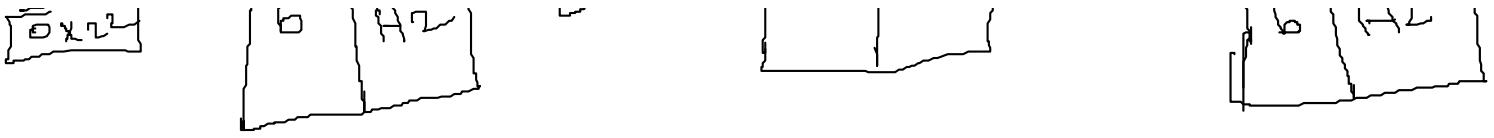
Object Creation:
Obj_name=Cname(arguments) where arguments is optional

Eg:
```
class Demo:
   pass
ob1=Demo()
ob2=Demo()
```

- A function written inside the class is known as method.

Eg:
```
class A:
   a=10
   b=20
o1=A()
o2=A()
o3=A()
```





For extracting the data:

Classname.propertyname
Objname.propertyname

For modifying the data:
Cname.pname=new_value
Objname.pname=new_value

Eg:

```
class A:
   a=10
   b=20
o1=A()
o2=A()
o3=A()
#print(A.a)
#print(o1.b)
#A.b=30
#print(A.b)
o2.a=15
print(o2.a)
print(A.a)
print(o1.a)
```

Types of Properties/States

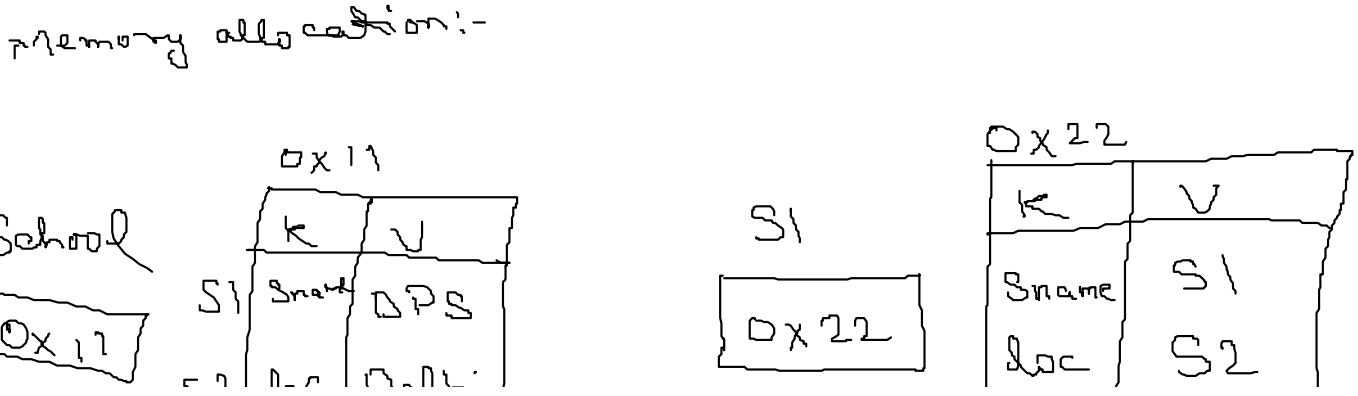
It is of 2 types:

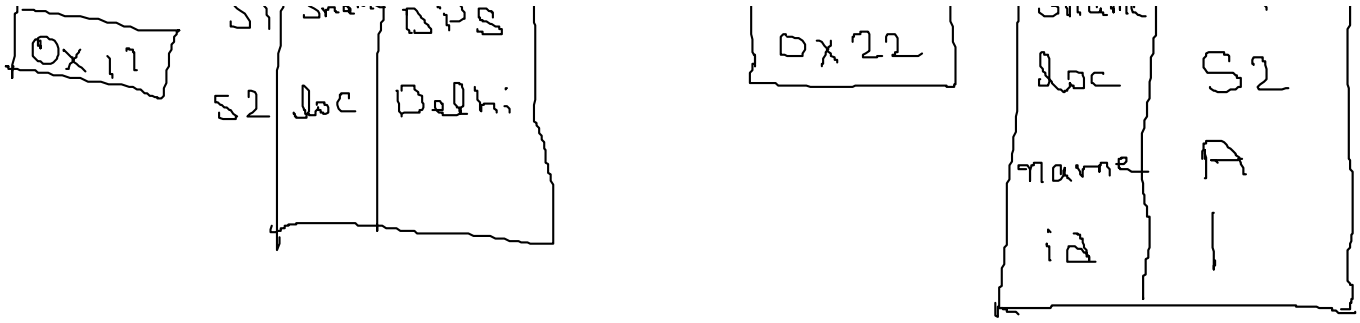
i. Static/Generic/Class property
ii. Specific/Object property

i. Static Property: These are the members of the class which are common for all the objects.
   Eg: Principal, uniform, email id, loc etc.
ii. Specific Property: These are the members which are different for all the objects.
   Eg: Student name, mob_no, address etc.

Eg:

```
class School:
   sname='DPS'
   loc='Delhi'
s1=School()
s1.name='A'
s1.id=1
print(s1.sname)
```





- __init__/Constructor/Initializer:

- It is used to initialize the members of the object.
- __init__ is used for creation
- It is automatically called in the object creation.

According to industry, self is used to store the address of the object.

Syntax:

```
Class Cname:
        Properties
        Def __init__(self,var1,var,var3…..var n):
                Self.var1=var1
                Self.var2=var2
                Self.var3=var3
                Self.varn=varn
Obj=Cname(val1,val2,val3…..val n)
```