

Memory allocation for class creation :-

- As soon as control see a **class** keyword, it will create a dictionary inside the memory, We can call that one as a class dictionary.
- It consists of key layer & value layer. Address will be given to key layer and that will get stored with respect to class name.
- All the properties and functionalities will get stored into class dictionary in the form of key & value, the reference address will be given to each & every key.

Memory allocation for object creation :-

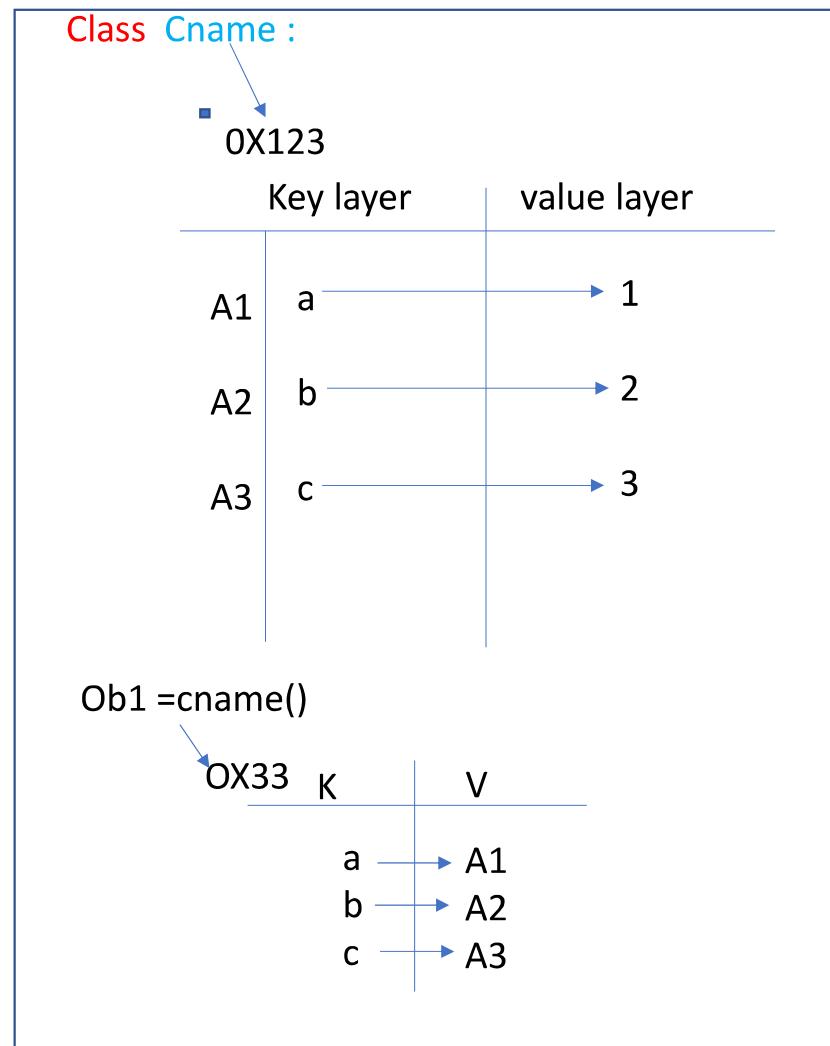
- As soon as control see an object creation process,it will create dictionary inside a memory ,that is object instance dictionary
- Which consist of key value pair,address will be given & that will get stored with respect to object name.
- All the properties of class dictionary will get stored into object dictionary in the form of key & value.
- Control will check whether the object dictionary is having “`__init__`” method or not ,if the method exist then it will get invoke by default.
- All the properties of object will get stored into object dictionary

MEMORY ALLOCATION FOR CLASS OBJECT CREATION

Class Point:

```
a = 1  
b = 2  
c = 3
```

```
>>>Ob1= Point()  
>>>ob2= Point()
```



Types of states/properties/members

- In class we can store the two types of states
- 1.generic /static/class members
- 2.specific/object members.
- 1.Generic states:-
 - Generic members are the member which will be common for each & every object.
 - Ex: if we consider bank as a class,it will be having n number of customers as an objects,
 - In this case bank name,main branch,location,CEO name& so on will be common for each & every object

- 2.specific or object members:-

- These are the members which will be different for each and every object created for a class.
- Ex: if we consider bank as class & customers as on object then customer.name,account No,phno,address,emailed and so on will be different for each and every object

```
>>> class Bank:
```

```
    Bname = 'SBI'  
    MBL = 'Bangalore'  
    CEO = 'yash'
```

Class members

```
>>> c1 = Bank()      #object creation
```

```
>>> c1.name = 'steve'  
>>> c1.phno = '88994543500'  
>>> c1.gmail = 'steve@1976.com'  
>>> c1.bal = 10000  
>>> c1.addrs = 'mysore'
```

Object members

```
>>> c2 = Bank()      #object creation
```

```
>>> c2.name = 'jon'  
>>> c2.phno = 9037255  
>>> c2.gmail = 'jon@134gmail.com'  
>>> c2.bal = 45000  
>>> c2.addrs = 'dehli'
```

Constructor method/initialization method (__init__)

- __init__ is method is used to initialize the members of object.
- It is not required to call init method outside the class (in the process of object creation,init method will get invoke by default)
- For __init__ method we need to pass ‘self’ as on argument to store the address of on object, we can pass ony other variable to store the address of on object but according to industrial standard passing self is mandatory.
- In the process of object creation no need to pass the address for self by default self will take the address of an object to which we are initializing the members.

- class Bank:

```
bname = 'SBI'  
mbl = 'bangalore'  
ceo = 'rocky'  
def __init__(self,name,phno,amount,gmail):  
    self._name = name    #c1.name = 'steve'  
    self._phno = phno  
    self.amount = amount  
    self.gmail = gmail
```

- c1 = Bank('steve','9900445589',10000,'steve@gmail.com')
- c2 = Bank('jon','33444667788',30000,'jon@gmail.com')