

```
In [ ]: # Object oriented programming
# -----
# class<-->object
# |      |
# type   value

# class - codeblock (or) blueprint of an object
# object - runtime entity

class classname:
    <class attribute1>
    <class attribute2>
    ..
    <class attributeN>

classname.attribute
classname.newattribute
```

```
In [4]: class Emp:
        name='Mr.X'
        eid=1234
        print(Emp.name)
        print(Emp.eid)
        Emp.edept='sales' # newattribute
        print(Emp.edept)
```

```
Mr.X
1234
sales
```

```
In [5]: # obj=Emp()
# ---
# |__real entity / real instance
obj1=Emp()
obj2=Emp()
print(obj1,obj2)
print(obj1.name,obj1.eid,obj1.edept)
```

```
<__main__.Emp object at 0x0000000005038C40> <__main__.Emp object at 0x000000000
50382E0>
```

```
In [ ]: +-----+
| <--> | | [] [] |
|      | |      |
+-----+

|      |      |      |      |
B1      B2      B3      ...      Bn <==object
|      |      |      |
1st     2nd     3rd     nth block <== address
obj1    obj2    obj3    ..
```

```
In [8]: class Emp:
        ename='defaultname'
        eid=0
e1=Emp()
e2=Emp()
print(Emp.ename,Emp.eid)
print(e1.ename,e1.eid)
print(e2.ename,e2.eid)
```

```
defaultname 0
defaultname 0
defaultname 0
```

```
In [11]: Emp.ename='UserA'
Emp.eid=101
print(Emp.ename,Emp.eid)
print(e1.ename,e1.eid)
print(e2.ename,e2.eid)
```

```
UserA 101
UserA 101
UserA 101
```

```
In [13]: e1.ename='PY-User'
e1.eid='P123'
print(Emp.ename,Emp.eid)
print("=>",e1.ename,e1.eid)
print(e2.ename,e2.eid)
```

```
UserA 101
=> PY-User P123
UserA 101
```

```
In [14]: Emp.ename='USERB'
Emp.eid=102
print(Emp.ename,Emp.eid)
print("=>",e1.ename,e1.eid)
print(e2.ename,e2.eid)
```

```
USERB 102
=> PY-User P123
USERB 102
```

```
In [15]: e2.ename='JAVA-user'
e2.eid='J555'
print(Emp.ename,Emp.eid)
print("=>",e1.ename,e1.eid)
print(e2.ename,e2.eid)
```

```
USERB 102
=> PY-User P123
JAVA-user J555
```

```
In [17]: Emp.ename='USERC'
Emp.eid=103
print(e1.ename,e1.eid)
print(e2.ename,e2.eid)
print(Emp.ename,Emp.eid)
```

```
PY-User P123
JAVA-user J555
USERC 103
```

```
In [19]: e3=Emp()
print(e3.ename,e3.eid)
```

```
USERC 103
```

```
In [ ]: # classname -> Filesysinfo
#         fstype='ext'
#         fmount='/'
#         |   |   |   |
#         obj1obj2obj3 obj4
#         |   |   |   |
#
```

```
In [21]: class Filesysinfo:
          fstype='ext'
          fmount="/"

          obj1=Filesysinfo()
          obj1.fstype='ext4'
          obj1.fmount="/D1"

          obj2=Filesysinfo()
          obj2.fstype="xfs"
          obj2.fmount="/D2"

          obj3=Filesysinfo()
          obj3.fstype="btrfs"
          obj3.fmount="/D3"
          print("filesystem type:{}\tmountpoint{}".format(obj1.fstype,obj1.fmount))
          print("filesystem type:{}\tmountpoint{}".format(obj2.fstype,obj2.fmount))
          print("filesystem type:{}\tmountpoint{}".format(obj3.fstype,obj3.fmount))
          print("")
          print(Filesysinfo.fstype,Filesysinfo.fmount)
```

```
filesystem type:ext4      mountpoint/D1
filesystem type:xfs       mountpoint/D2
filesystem type:btrfs     mountpoint/D3
```

```
ext /
```

```
In [22]: # in python everything is a class-->object
          a=10
          b=2.455
          c=True
          d='data'
```

```
Out[22]: int
```

```
In [28]: class Box:
        pass
        # empty class
obj1=Box()
obj2=Box()
print(type(obj1))
print(type(obj2))

obj1.NAME='USER-A'
obj1.eid=123

obj2.Place='city'
obj2.NAME='USER-B'
print(obj1.NAME,obj1.eid)
print(obj2.Place,obj2.NAME)
print(Box.NAME)
```

```
<class '__main__.Box'>
<class '__main__.Box'>
USER-A 123
city USER-B
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-28-6ad153939a97> in <module>
      14 print(obj1.NAME,obj1.eid)
      15 print(obj2.Place,obj2.NAME)
----> 16 print(Box.NAME)
```

**AttributeError:** type object 'Box' has no attribute 'NAME'

```
In [ ]: # function -> procedure style code -> function() //function call
        #
        # method -> object oriented style code -> object.function() //method call
        #
L=[]
L.append("D1") # method call
del(L) # functioncall
```

```
In [31]: s='data'
s.upper()
# s.append() ->AttributeError: 'str' object has no attribute 'append'
# class str:                                class list:
#     def upper():                            def append():
#
#
```

Out[31]: 'DATA'

```
In [33]: def f1():
          print("Hello")
          f1()

          def f2():
              print("Welcome")

          # f2(10) # f2() takes 0 positional arguments but 1 was given - Error
```

Hello

```
In [37]: class Box:
          var=100
          def f1(self):
              print('Hello')

          # Box
          # |__ var f1 <attributes>
          #
          obj=Box()
          obj.var
          # obj.f1() TypeError: f1() takes 0 positional arguments but 1 was given
          #
          # obj.f1() -->f1(obj)
          # obj1.f1(100,200,300) -->f1(obj1,100,200,300)
          obj.f1()
```

Hello

```
In [38]: def fx(a1):
          print(type(a1),a1)

          fx(10)
          fx(1.45)
          fx('abc')
          fx([])
          fx({})
```

```
<class 'int'> 10
<class 'float'> 1.45
<class 'str'> abc
<class 'list'> []
<class 'dict'> {}
```

```
In [39]: class Box:
          def f1(self):
              print("self:{}".format(self))
obj1=Box()
obj2=Box()
obj3=Box()
print(obj1,obj2,obj3)
print("")
obj1.f1() # f1(obj1)
print("")
obj2.f1() # f1(obj2)
print("")
obj3.f1() # f1(obj3)
```

```
<__main__.Box object at 0x000000007979EB0> <__main__.Box object at 0x000000007979F70> <__main__.Box object at 0x000000006FBCFD0>
```

```
self:<__main__.Box object at 0x000000007979EB0>
```

```
self:<__main__.Box object at 0x000000007979F70>
```

```
self:<__main__.Box object at 0x000000006FBCFD0>
```

```
In [42]: class Box:
          var=100
          def f1(self):
              self.var=200
obj1=Box()
print(obj1.var) ## (A)
obj1.f1() # f1(obj1)
print(obj1.var) ## (B)
print(Box.var) ## (C)
```

```
100
```

```
200
```

```
100
```

```
In [ ]: class Filesysinfo:
    fstype='ext'
    fmount="/"

    obj1=Filesysinfo()
    obj1.fstype='ext4'
    obj1.fmount="/D1"

    obj2=Filesysinfo()
    obj2.fstype="xfs"
    obj2.fmount="/D2"

    obj3=Filesysinfo()
    obj3.fstype="btrfs"
    obj3.fmount="/D3"
    print("filesystem type:{}\tmountpoint{}".format(obj1.fstype,obj1.fmount))
    print("filesystem type:{}\tmountpoint{}".format(obj2.fstype,obj2.fmount))
    print("filesystem type:{}\tmountpoint{}".format(obj3.fstype,obj3.fmount))
    print("")
    print(Filesysinfo.fstype,Filesysinfo.fmount)
```

```
In [70]: class Filesysinfo:
    fstype='ext'
    fmount="/"
    def f1(self,a1,a2):
        self.fstype=a1
        self.fmount=a2
    def f2(self):
        print("File System Type:{}\t MountPoint:{}".format(self.fstype,self.fmount))
    def f3(self,a1):
        self.fstype=a1

    obj1=Filesysinfo()
    obj1.f1("ext4","/D1") # f1(obj1,ext4,/D1)

    obj2=Filesysinfo()
    obj2.f1("xfs","/D2") # f1(obj2,xfs,/D2)

    obj3=Filesysinfo()
    obj3.f1("btrfs","/D3") # f1(obj3,btrfs,/D3)

    obj1.f2()
    obj2.f2()
    obj3.f2()
    print("")
    obj4=Filesysinfo()
    obj4.f2()
```

```
File System Type:ext4      MountPoint:/D1
File System Type:xfs       MountPoint:/D2
File System Type:btrfs     MountPoint:/D3
```

```
File System Type:ext      MountPoint:/
```



```
In [49]: obj4=Filesysinfo()
obj4.f1("ocfs2", "/D3")
obj4.f2()
obj4.f3('VFAT')
obj4.f2() # updated result
```

```
File System Type:ocfs2   MountPoint:/D3
File System Type:VFAT   MountPoint:/D3
```

```
In [52]: class Box:
        var=100
        def f1(self):
            print(self.var)

obj=Box()
obj.f1()
```

```
100
```

```
In [60]: class Box:
        fname='p1.log'
        __password='abc333' # user defined private attribute

Box.fname
obj=Box()
obj.fname
#Box.__password ->Error
# obj.__password ->Error
```

```
Out[60]: 'p1.log'
```

```
In [64]: class Box:
        __fname="p1.log"
        __password="abc123"
        def f1(self):
            print(self.__fname,self.__password)
        def f2(self,a1):
            self.__password=a1

obj=Box()
obj.f1()
obj.f2("XYZ1234")
obj.f1()
```

```
p1.log abc123
p1.log XYZ1234
```

```
In [ ]: # Constructor ->method() ->initialization
        # __init__()
```

```
In [ ]: class Box:
        def f1(self):
            print("non-constructor")
```

```
In [67]: class Filesysinfo:
        def f1(self,a1,a2):
            self.fstype=a1
            self.fmount=a2
        def f2(self):
            print(self.fstype,self.fmount)

obj1=Filesysinfo()
obj1.f1("xfs", "/D1")
obj1.f2()

obj2=Filesysinfo()
# obj2.f2() # AttributeError
```

xfs /D1

```
In [ ]: # class DBI:
        #     def connect():
        #         (1)
        #     def method2():
        #     def method3():
        #     def method4():
        #     def method5():

        # obj=DBI()
        # obj.method4() ->Attribute Error
```

```
In [74]: class Filesysinfo:
        def __init__(self,a1,a2):
            self.fstype=a1
            self.fmount=a2
        def f1(self):
            print("{}\t{}".format(self.fstype,self.fmount))
obj1=Filesysinfo("xfs", "/D1")
obj1.f1()
obj2=Filesysinfo("ext4", "/D2")
obj2.f1()
print("-->",obj1.fstype,obj2.fstype)
```

xfs        /D1  
ext4       /D2  
--> xfs ext4

```
In [76]: class Filesysinfo:
          def __init__(self,a1,a2):
              self.__fstype=a1
              self.__fmount=a2
          def f1(self):
              print("{}\t{}".format(self.__fstype,self.__fmount))
obj1=Filesysinfo("xfs","/D1")
obj1.f1()
obj2=Filesysinfo("ext4","/D2")
obj2.f1()
#print("-->",obj1.__fstype) # Error
```

```
xfs      /D1
ext4     /D2
```

```
In [78]: a=10 # procedure code
          b=int(10) # oops style # obj=classname(args)
          print(a,b,type(a),type(b))
```

```
10 10 <class 'int'> <class 'int'>
```

```

In [ ]: >>> class Box:
...       var=100
...
>>> Box.var=200
>>>
>>> obj1=Box()
>>> # obj1.var ----->(A)
>>> obj1.var='Data1'
>>> obj2=Box()
>>> # obj2.var ----->(B)
>>> Box.var=300
>>> # obj1.var ----->(C)
>>> # obj2.var ----->(D)
>>> obj3=Box()
>>> # obj3.var ----->(E)
>>>
>>>
>>> a=10
>>> type(a)
<class 'int'>
>>> type(10)
<class 'int'>
>>> type(20)
<class 'int'>
>>> type(33)
<class 'int'>
>>> type(-1)
<class 'int'>
>>>
>>> # int -- class - type
>>> # |
>>> # -ve 0 +ve - object - value
>>> #
>>> class Box:
...     var=100
...
>>> obj1=Box()
>>> obj2=Box()
>>> obj3=Box()
>>>
>>> type(obj1)
<class '__main__.Box'>
>>> type(obj2)
<class '__main__.Box'>
>>> type(obj3)
<class '__main__.Box'>
>>>
>>> type(Box)
<class 'type'>
>>> id(obj1)
15381744
>>> hex(id(obj1))
'0xeab4f0'
>>> hex(id(obj2))
'0xeab570'
>>> hex(id(obj3))

```

```
'0xeab3d0'  
>>>  
>>> import cgi  
>>> cgi.FieldStorage  
<class 'cgi.FieldStorage'>  
>>>
```