Data Science & AI
&
NIC - Param

Python-For Data Science

OOPs

Lecture No.- 02

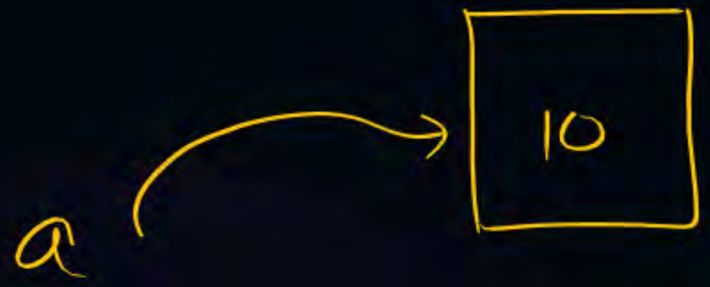
Recap of Previous Lecture

Topic Object-Oriented Programming Part -01

Topics to be Covered

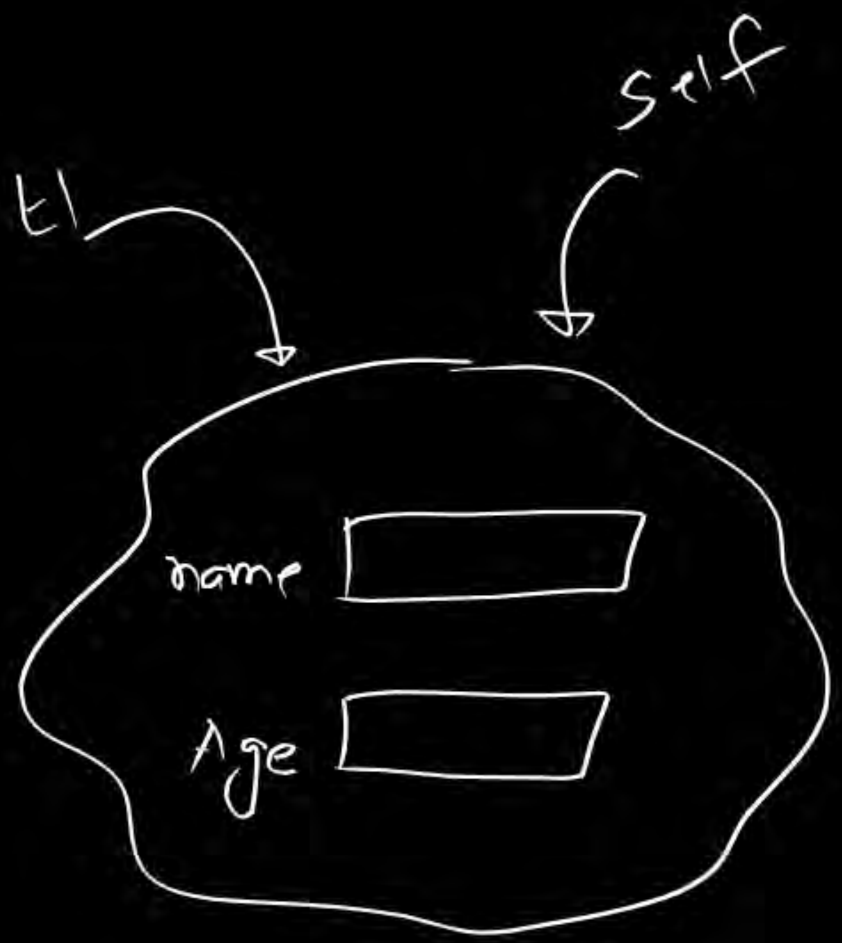
Topic Object-Oriented Programming Part -02

Topic : Object-Oriented Programming



```
class Teacher :

    def __init__(self):

            self.name = 'Pankaj'
            self.Age  = 41

    def  display(self):
            print( self.name )
            print (self . Age)

t1= Teacher( )
t2
```

To declare & initialize
object (Instance)
variable



```
class Teaches :

    def __init__(self):
        self.name = 'Pankaj'
        self.Age  = 41
    def display(self):
      ✓ print(self.name)
      ✓ print(self.Age)
t1= Teacher()  ✓
t2 = Teacher()  ✓
t1.display()   self
```

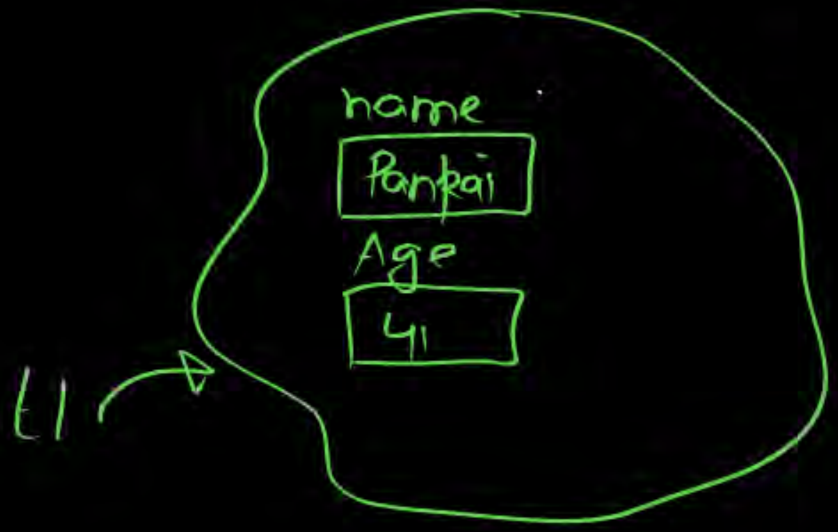
functions

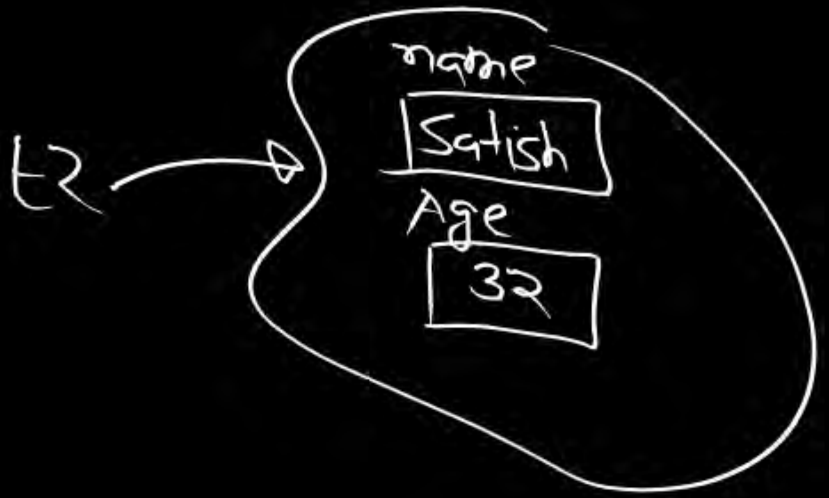
add(a,b) | add(10,20)

Ram.playing()
Shyam.playing()

```
class Teaches :

    def __init__(self):

        self.name = 'Pankaj'
        self.Age  = 41

    def display(self):
      ✓ print(self.name)
      ✓ print(self.Age)

t1 = Teacher()    ✓          } ⇒
t2 = Teacher()    ✓          }
t1.display()    self         }
```

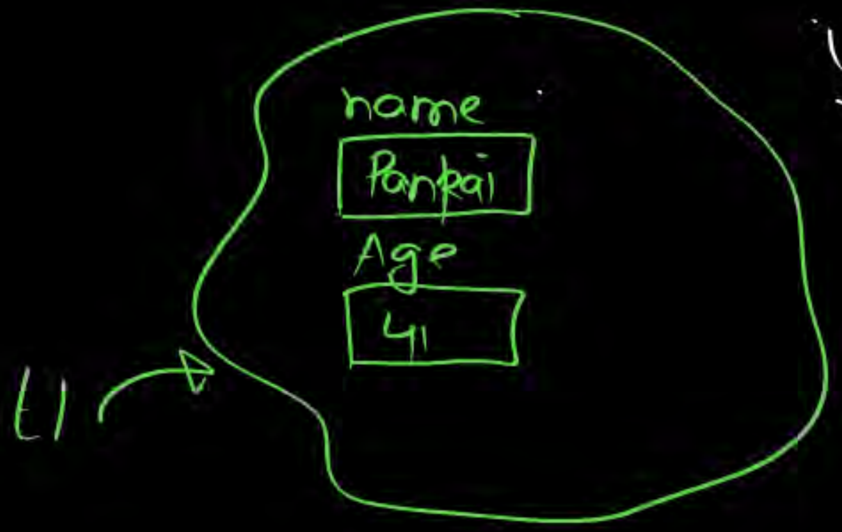
```
class Teacher :

    def __init__(self, name, Age)

        self.name = name
        self.Age  = Age
    def display(self):
      ✓ print(self.name)
      ✓ print(self.Age)

t1 = Teacher('Pankaj', 41)
t2 = Teacher('Satish', 32)
t3 = Teacher('Ankit', 34)
          ✓
```

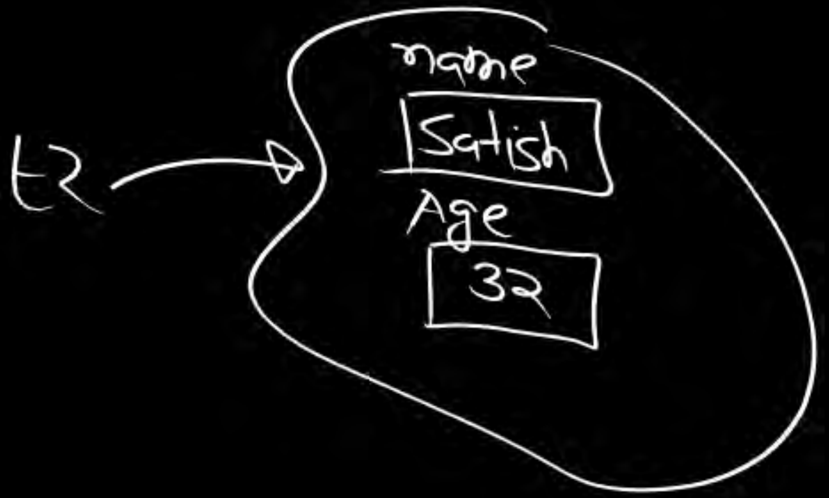




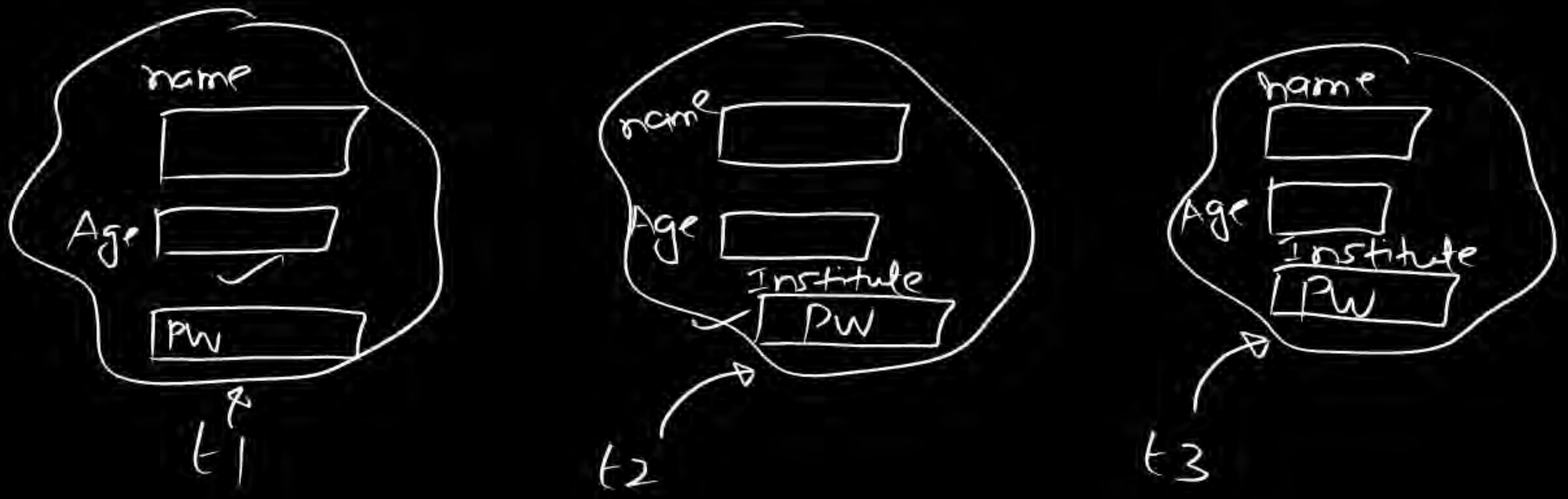
```
class Teacher :

    def __init__(self, Naam, Umar)

        self.name = Naam
        self.Age  = Umar

    def display(self):

        ✓ print(self.name)
        ✓ print(self.Age)

t1 = Teacher('Pankaj', 41)
t2 = Teacher('Satish', 32)
t3 = Teacher('Ankit', 34)
```

instance variable





name
Age
Institute



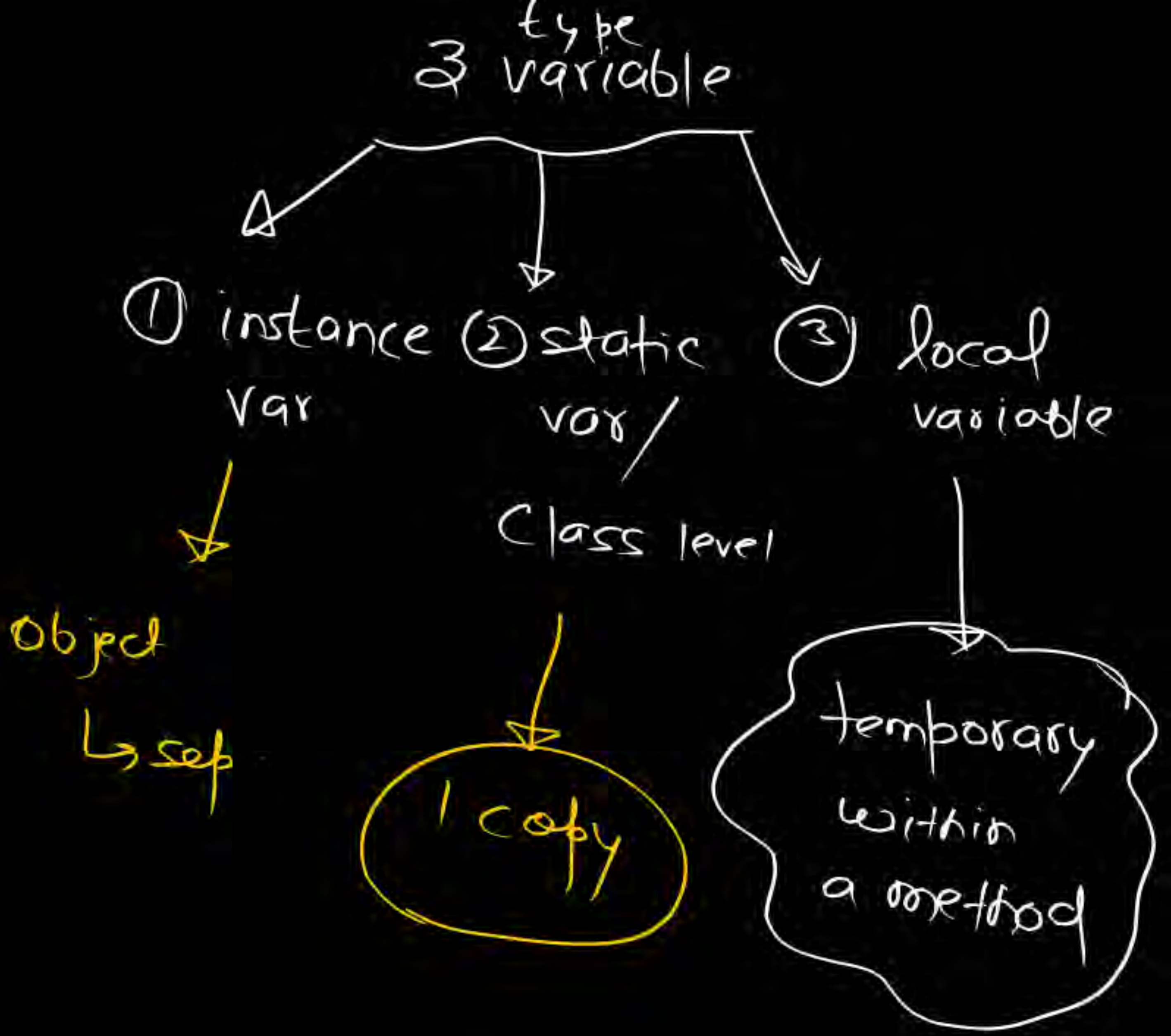
{ name
Age } different from Object to Object

variables
① instance variable

② static variables (class level)
Institute

the val. of variable is same for all the object
⇒ class level variables

```
class Teacher :

    Institute = 'PW'

    def __init__(self) :

        ---
        ---
        ---

    def display(self)

        ---
        ---
        ---
        ---
        ---
```



3 methods

① instance method

```
display(self) :
    print(self.name)
    print(self.Age)
```

② class method

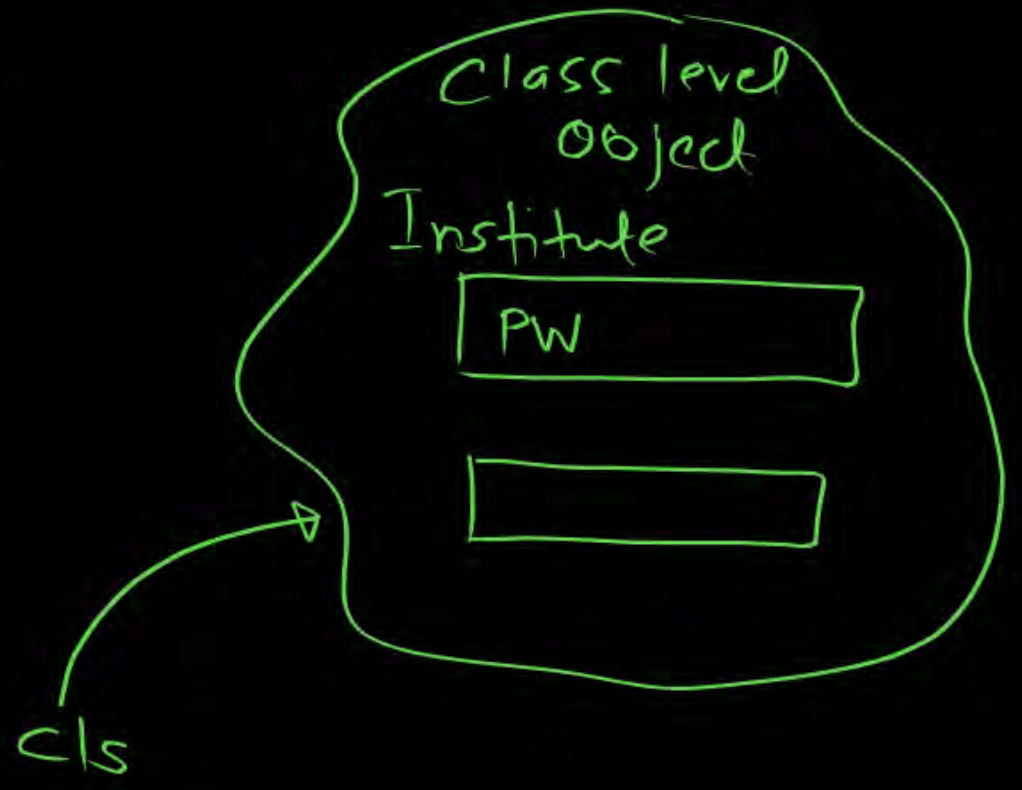
inside any method

↓

only class level variable (only static variable)

```
@classmethod
def  display_classinfo(cls ) :

        print( cls. Institute)
```

what is cls ?



static method

```
@staticmethod
def f1(a,b,c):

    return a*2 + b*2 + c*2
```

No instance var

No class variable

general purpose

C videos
→ 16 videos

oops concepts

- Object
- class
- refrence variable
- self
- variables type
  - instance
  - class (static) variable
  - local
- method type
  - instance method — self
  - class method — cls — { only accessing class level variable (static var) }
  - static method

t.me/PWpankajsir

OOPS
data structure

Day 22 oops part2

In [11]:
```
class Teacher:
    def __init__(self):
        self.name='pankaj'
        self.age=41
        print(id(self))
        print("constructor executes")
    def display(self):
        print(self.name)
        print(self.age)
t1=Teacher() #__init__() executes automatically
print(id(t1))
t2=Teacher()
print(id(t2))
```

```
2884054587344
constructor executes
2884054587344
2884054579600
constructor executes
2884054579600
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[11], line 14
     12 t2=Teacher()
     13 print(id(t2))
---> 14 print(self.name)

NameError: name 'self' is not defined
```

In [7]:
```
t1.display()
```

```
pankaj
41
```

In [6]:
```
t2.display()
```

```
pankaj
41
```

In [15]:
```
class Teacher:
    def __init__(self):
        self.name='pankaj'
        self.age=41
        print(id(self))
        print("constructor executes")
    def display(self):
        print(self.name)
        print(self.age)
```

In [16]:
```
t1=Teacher()
t2=Teacher()
t3=Teacher()
```

```
2884066809808
constructor executes
2884059818576
constructor executes
2884054587344
constructor executes
```

In [18]:
```
t1.display()
```
```
pankaj
41
```

In [19]:
```
t2.display()
```
```
pankaj
41
```

In [20]:
```
t3.display()
```
```
pankaj
41
```

In [21]:
```
class Teacher :
    institute="Pw"
    def __init__(self):
        self.name='pankaj'
        self.age=41
```

In [22]:
```
 print(Teacher.institute) #this is better classname.static_variable
```
```
Pw
```

In [23]:
```
t=Teacher()
print(t.institute)
```
```
Pw
```

In [24]:
```
class Teacher:
    @classmethod
    def fun(cls):
        print(id(cls)) #class level object refrence
print(id(Teacher))
```
```
2884035680912
```

In [25]:
```
#fun===> class method
Teacher.fun()
```
```
2884035680912
```

In [ ]:

THANK - YOU