14/6

Types of Variables :-

(i) Static / Class Variable

(ii) Non-Static / Instance Variable

→ Static / Class Variable :- It is a ~~def~~ Variable which define for the common object if one object is changed all will be changed.

It is nothing but Common Variable

* Example let in banking Sector the Common Variable will be Bank name for that bank application. So now the program is written as follows

```
class Bank:
        bank_name = "Andhra bank"  # Static Variable
        def display (self):
            print( Bank.bank_name) # classname is mandatory to access
        def update (self):
            Bank.bank_name = "union bank"
eg
rajesh = Bank()
Suresh = Bank()

rajesh.display()                  rajesh.display()
Suresh.display()                  Suresh.display()
                                  rajesh.update()
o/p: Andhra Bank                  rajesh.display()
     Andra Bank                   Suresh.display()
                                  o/p: Andhra Bank
                                       Andhra Bank
                                       Union Bank
                                       Union Bank
```

→ Non - Static / Instance Variable :-

If the data is not Common we use these type of variables.

Here we represent in form of self.***

Ex: class Bank:

```
        def load(self):
            Self.address = "hyd"
            Self.phno = *****
        def display(self):
            print(self.address)
            print(self.phno)
        def update(self):
            self.add = "mumbai"
            Self.phno = 12345
rajesh = Bank()
Suresh = Bank()
rajesh.display()
```

o/p: Attribute Error

Note: When it is a instance variable the object creation/the data will come with object flow
So

```
rajesh = Bank()
Suresh = Bank()
rajesh.Load()
rajesh.display()
```

o/p
```
hyd
*****
```

Here we need to load Everytime before we call the method.

let $\overset{take}{\text{~~one~~}}$ more Examples

1. class Demo:

```
        def m1(self):
                print("I'm Super Star")

    o.Demo()
```

o/pt No o/p displayed

```
∴ o=Demo()
  o.m1()
```

o/p: I'm Super Star

2. class Demo:

```
        def init (self):
                print ("I'm Super Star")

    o.Demo()
```

o/pt "I'm Super Star

Note: Here _init_ is a Constructor. Constructor is a Special method which we doesn't need to call it. It is used to initialised instance variable

If the details are different we can change the past code of class Bank little bit as follows

```
    def _init_(self,address,phno)# Constructor
            self.address = address
            self.phno = phno
        ⋮
rajesh=Bank("hyd",2999, 1111)
Suresh=Bank("viz",888,2222)
```

```
rajesh.display()
Suresh.display()
o/pt 'hyd', 9999,1111
     'viz' 888,2222
```

```
rajesh.update()
rajesh.display()
suresh.display()
```

O/p:- mumbai
12345

viz
888

Global Variable :-

Defines outside the ~~funcki~~ Class

```
a=10
class Demo:
    def m1(self):
        print(a)
o=Demo()
o.m1()
```

O/p:- 10

local variable :

Defines inside the function/method

```
Class Demo:
    def m1(self):      [# if we want to make local a as global
        a=100           global a → keyword is used]
    def m2(self):
        print(a)
O=Demo()
O.m1()
O.m2()
```

O/p:- 100

Error