Example:

```
class Customer:
    def __init__(self):
        self.__accno=None
        self.__cname=None
        self.__balance=None
    def  set_accno(self,a):
        self.__accno=a
    def set_cname(self,c):
        self.__cname=c
    def set_bal(self,b):
        self.__balance=b
    def deposit(self,a):
        self.__balance=self.__balance+a
    def withdraw(self,a):
        if a<self.__balance:
            self.__balance=self.__balance-a
        else:
            print("Insuff Balance")
    def print_account(self):
        print(f'Account {self.__accno}')
        print(f'CustomerName {self.__cname}')
        print(f'Balance {self.__balance}')



cust1=Customer()
cust1.print_account()
cust1.set_accno(101)
cust1.set_bal(50000)
cust1.set_cname("naresh")
cust1.print_account()
cust1.deposit(5000)
cust1.print_account()
cust1.withdraw(2000)
cust1.print_account()
```

Output:

```
Account None
CustomerName None
Balance None
Account 101
CustomerName naresh
Balance 50000
Account 101
CustomerName naresh
Balance 55000
Account 101
CustomerName naresh
Balance 53000
```

Example:

```
class Queue:
    def __init__(self):
        self.__q=[]
    def insert(self,value):
        self.__q.append(value)
    def remove(self):
        if len(self.__q)==0:
            return "Queue is Empty"
        else:
            value=self.__q[0]
            del self.__q[0]
            return value

q1=Queue()
q1.insert(10)
q1.insert(20)
q1.insert(30)
value1=q1.remove()
value2=q1.remove()
value3=q1.remove()
value4=q1.remove()
print(value1,value2,value3,value4,sep="\n")
```

Output:

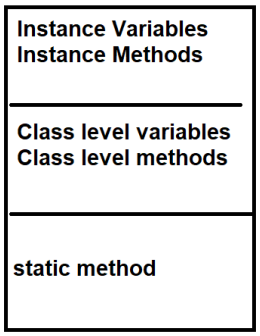
```
10
20
30
Queue is Empty
```

Example:

```
class Player:
    def __init__(self,n,s):
        self.__name=n
        self.__score=s
    def get_name(self):
        return self.__name
    def get_score(self):
        return self.__score



n=int(input("Enter How Players?"))
playerList=[]
for i in range(n):
    name=input("Enter Player Name ")
    score=int(input("Enter Player Score "))
    p=Player(name,score)
    playerList.append(p)

for p in playerList:
    print(f'{p.get_name()}--> {p.get_score()}')
```

Output:

```
Enter How Players?2
Enter Player Name aaa
Enter Player Score 60
Enter Player Name bbb
Enter Player Score 50
aaa--> 60
bbb--> 50
```

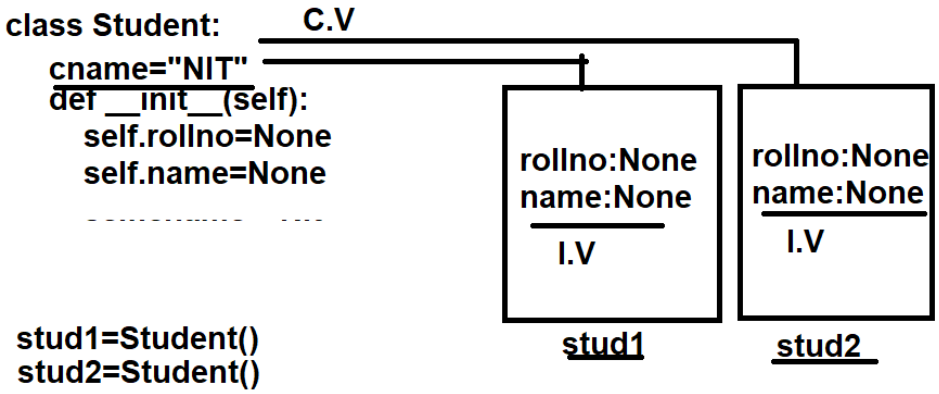


Class level variable
A variable declared/created inside class and outside method is called class level variable.
Class level variable is global variable, which is global to more than one object.

Syntax:

```
class <class-name>:
        <class-level-variable>;
        <class-level-variable>;
        def instance-method(self):
                self.<instance-variable>
                self.<instance-variable>
```

Class level variables bind with class name; these variables can be accessed without creating object.



Inside the class, class level variables are bind with class name.

class A:

```
    x=100 # class level variable
    def __init__(self):
        self.y=200



print(A.x)
obj1=A()
print(obj1.y)
```

Output:

```
100
200
```

Class level variable memory is allocated only once.

Example:

```
class Product:
    count=0 # C.V
    def __init__(self):
        print("Product is Created")
        Product.count=Product.count+1




print(Product.count)
p1=Product()
p2=Product()
p3=Product()
print(Product.count)
```

Output:

```
0
Product is Created
Product is Created
Product is Created
3
```

Example:

```
class Account:
    __minBal=5000
    def __init__(self,a,c,b):
        self.__accno=a
        self.__cname=c
        self.__bal=b
    def withdraw(self,a):
        if (self.__bal-a)<Account.__minBal:
            print("Insuff Balance")
        else:
            self.__bal=self.__bal-a
    def print_account(self):
        print(f'Account {self.__accno}')
        print(f'CustomerName {self.__cname}')
        print(f'Balance {self.__bal}')

cust1=Account(101,"naresh",50000)
cust2=Account(102,"suresh",70000)
cust1.withdraw(40000)
cust1.print_account()
cust2.print_account()
```

Output:

```
Account 101
CustomerName naresh
Balance 10000
Account 102
CustomerName suresh
Balance 70000
```

Class level method

A method defined inside class with first argument as **“cls”** is called class level method.

This method is bind with class name and it can be called without creating object.

Syntax:

```
class <class-name>:
        class-level-variable
        class-level-variable
        def instance-method-name(self,..):
                self.instance_variable
                self.instance-variable
        def class-method-name(cls,…):
                cls.class-level-variable
                cls.class-level-variable
                class-name.class-level-variable
```