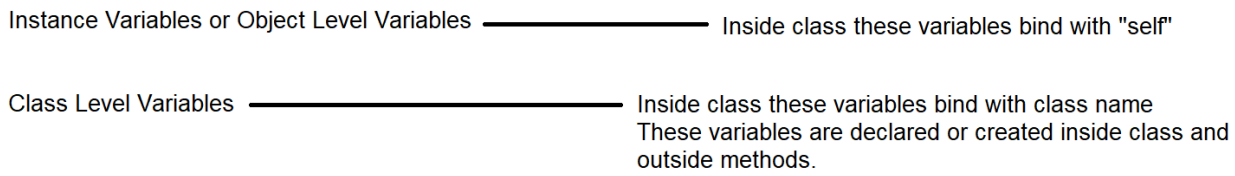
Class Level Variables

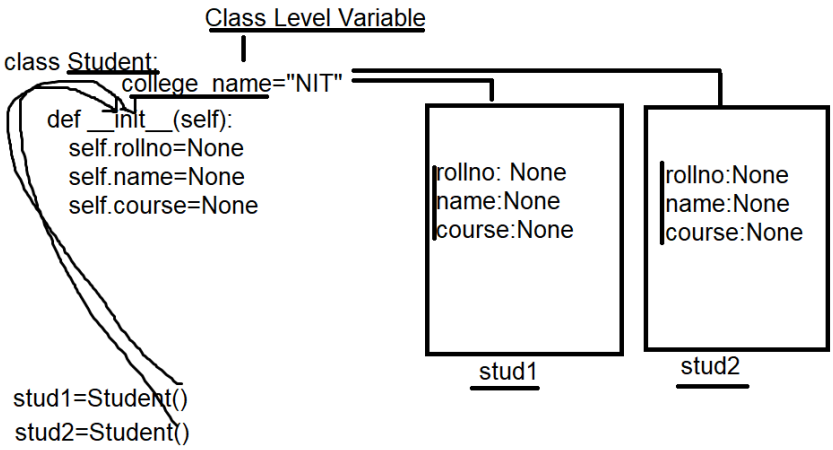
A variable declared inside class outside the methods is called class level variable. These variables bind with class name. These variables can be used without creating object.



Syntax:

```
class <class-name>:
        <class-level-variable>=<value>
        <class-level-variable>=<value>
        def <instance-method>(self,param,param,param):
                <self>.<instance-variable>=<value>
                <self>.<instance-variable>=<value>
```

Class level variables are global variables, which are global to one or more than object.



Class level variables define common property.
Class level variables outside the class or inside the class access with class name or object name.
Class level variables cannot modify or update with object name.
Updating or modifying is done with class name.

Example:

```
class A:
    y=200
    def __init__(self):
        self.x=100


obj1=A()
obj2=A()
obj3=A()
print(obj1.x)
print(obj2.x)
print(obj3.x)
print(obj1.y)
print(obj2.y)
```

```
print(obj3.y)
A.y=400
print(A.y)
print(obj1.y)
print(obj2.y)
print(obj3.y)
```

Output:

```
100
100
100
200
200
200
400
400
400
400
```

Example:

```
class Account:
    min_balance=10000 # C.L.V
    def __init__(self,a,c,b):
        self.__accno=a # I.V
        self.__cname=c # I.V
        self.__balance=b # I.V
    def print_account(self):
        print(f'''AccountNo {self.__accno}
CustomerName {self.__cname}
Balance {self.__balance}''')
    def deposit(self,t):
        self.__balance=self.__balance+t
    def withdraw(self,t):
        if self.__balance-t<Account.min_balance:
```

```
            print("insuff funds")
        else:
            self.__balance=self.__balance-t


acc1=Account(101,"naresh",50000)
acc2=Account(102,"ramesh",45000)
acc1.print_account()
acc2.print_account()
acc1.deposit(10000)
acc1.print_account()
acc2.withdraw(5000)
acc2.print_account()
```

Output

AccountNo 101
CustomerName naresh
Balance 50000
AccountNo 102
CustomerName ramesh
Balance 45000
AccountNo 101
CustomerName naresh
Balance 60000
AccountNo 102
CustomerName ramesh
Balance 40000

What is difference between instance variables and class level variables?

Instance variables	**Class level variables**
Inside class instance variables are created and accessed with "self"	Inside class, class level variables are created within class and

and outside the class bind with object name	outside method. These variables inside class bind with “self” or class name. outside the class bind with class name
These variables memory is allocated within object context.	These variables memory is allocated class context.
Memory is allocated for every object	Memory is allocated only once.

Example:

```
class A:
    __x=100 # C.L.V
    def __init__(self):
        self.__y=200 # O.L.V


print(A.__x)
obj1=A()
print(obj1.__y)
```

Output

```
Traceback (most recent call last):
  File "C:\Users\nit\PycharmProjects\project1\test21.py", line 7, in
<module>
    print(A.__x)
          ^^^^^
AttributeError: type object 'A' has no attribute '__x'
```

Class Level Method

A method defined inside class with first parameter as "cls" is called class level method. To change or transform method to class level, it is decorated with @classmethod decorator.

Syntax:

```
class <class-name>:
        class-level-variable=<value>
        class-level-variable=<value>
        @classmethod
        def <method-name>(cls,<param>,<param>,…):
                statement-1
                statement-2
        def <method-name>(self,<param>,<param>,…):
                statement-1
                statement-2
```

1. Class level method is bind with class name, this method can be called or invoked without creating object
2. Class level method is access only class level variables but cannot access instance variables
3. Class level methods define class level operation.

Example:

```
class A:
    def m1(self):
        print("instance method")
    @classmethod
    def m2(cls):
        print("class method")
```

```
A.m2()
obj1=A()
obj1.m1()
```

Output:

class method
instance method

Example:

```
class Product:
    count=0 # C.L.V
    def __init__(self):
        self.__product_id=None
        self.__product_name=None
        Product.count=Product.count+1
    @classmethod
    def getProductCount(cls): # C.L.M
        return cls.count



a=Product.getProductCount()
print(f'Product Count is {a}')
p1=Product()
p2=Product()
a=Product.getProductCount()
print(f'Product Count is {a}')
```

Output

Product Count is 0
Product Count is 2

What is difference between instance method and class level method?

Instance method	**Class method**
A method defined inside class with first parameter as “self” is called instance method	A method defined inside class with first parameter as “cls” is called class level method. It is has to be decorated with @classmethod decorator
Instance method is able access instance variables and class level variables.	Class method is able access only class level variables but cannot access instance variables
This method is bind with object name	This method is bind with class name
This method is required object to invoke	This method can be invoked without creating object
This method defines object level operations.	This method define class level operations