Python Programming

RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

PYTHON'S OBJECT OROENTED PROGRAMMING - 5

Guido Van Rossum

Agenda:

1. Instance Variables vs Static Variables
2. Local Variables
3. Mini Bank Application Development

1. Instance Variables vs Static Variables

Instance Variables:

1. These are Object level variables.

2. For every object, a separate copy will be created.

3. By using one object reference, if we are trying to perform any changes to the instance variables, then the changes won't be reflected to the remaining objects, because for every object a separate copy of instance variables will be there.

Static Variables:

1. These are Class level variables.

2. A single copy will be created at class level and shared by all objects of that class.

3. If we perform any change to the static variable, then those changes will be reflected to all objects, because a single copy of static variable will be maintained at class level.

Demo Program:

```
class Test:
    a = 10
    def __init__(self):
        self.b = 20

t1 = Test()
t2 = Test()
Test.a = 888
t1.b = 999
print('t1:',t1.a,t1.b)      # 888 999
print('t2:',t2.a,t2.b)      # 888 20
```

```
t1: 888 999
t2: 888 20
```

2. Local Variables

Key Features:

- Sometimes to meet temporary requirements of programmer, we can declare variables inside a method directly, such type of method level variables are called local variables or temporary variables.
- Local variables will be created at the time of method execution and destroyed once method completes.
- Local variables of a method cannot be accessed from outside of method.

Demo Program 1:

```
class Test:
    @staticmethod
    def average(list1):
        result = sum(list1)/len(list1)
        print("The average value :",result)

list1=[10,20,30,40]
Test.average(list1)
```

```
The average value : 25.0
```

Demo Program 2:

```
class Test:
    @staticmethod
    def average(list1):
        result = sum(list1)//len(list1)
        print("The average value :",result)

list1=[10,20,30,40]
Test.average(list1)
```

```
The average value : 25
```

Demo Program 3:

```
class Test:
    @staticmethod
    def average(list1):
        result = sum(list1)//len(list1)
        print("The average value :",result)

    @staticmethod
    def wish(name):
        for i in range(10):                    # here, 'i' is a local variable
            print("Good Evening :",name)

list1=[10,20,30,40]
Test.average(list1)
Test.wish('Karthikeya')
```

```
The average value : 25
Good Evening : Karthikeya
Good Evening : Karthikeya
Good Evening : Karthikeya
Good Evening : Karthikeya
Good Evening : Karthikeya
Good Evening : Karthikeya
Good Evening : Karthikeya
Good Evening : Karthikeya
Good Evening : Karthikeya
Good Evening : Karthikeya
```

Demo Program 4:

```
class Test:
    @staticmethod
    def average(list1):
        result = sum(list1)//len(list1)
        print("The average value :",result)

    @staticmethod
    def wish(name):
        for i in range(10): # 'i' is a local variable used to hold iteration number
            print("Good Evening :",i, name)

list1=[10,20,30,40]
Test.average(list1)
Test.wish('Karthikeya')
```

```
The average value : 25
Good Evening : 0 Karthikeya
Good Evening : 1 Karthikeya
Good Evening : 2 Karthikeya
Good Evening : 3 Karthikeya
Good Evening : 4 Karthikeya
Good Evening : 5 Karthikeya
Good Evening : 6 Karthikeya
Good Evening : 7 Karthikeya
Good Evening : 8 Karthikeya
Good Evening : 9 Karthikeya
```

Demo Program 5:

```
class Test:
    def m1(self):
        a= 10
        print(a)

    def m2(self):
        print(a)

t = Test()
t.m1()
```

```
10
```

Demo Program 6:

```
class Test:
    def m1(self):
        a= 10
        print(a)

    def m2(self):
        print(a)

t = Test()
t.m1()
t.m2()
```

```
10
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-12-434d32cbed63> in <module>
      9 t = Test()
     10 t.m1()
---> 11 t.m2()

<ipython-input-12-434d32cbed63> in m2(self)
      5
      6     def m2(self):
----> 7         print(a)
      8
      9 t = Test()

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

3. Mini Bank Application Development

So far, we covered instance, static and local variables. With all these things let we develop a small bank application. So that, we know that how the concepts will require to use in the applications.

```python
class customer:
    '''This Class developed by Karthi and describes
       bank operations'''
    bankName = 'RGMBANK'

    def __init__(self,name,balance = 0.0):   # balance is default argument
        self.name=name
        self.balance=balance

    def deposit(self,amt):
        self.balance=self.balance+amt
        print('Balance after deposit:',self.balance)

    def withdraw(self,amt):
        if amt>self.balance:
            print('Insufficient Funds..cannot perform this operation')
        else:
            self.balance=self.balance-amt
            print('Balance after withdraw:',self.balance)

print('Welcome to',customer.bankName)
name=input('Enter Your Name:')
c = customer(name)
while True:
    print('d-Deposit \nw-Withdraw \ne-exit')
    option=input('Choose your option:')
    if option=='d' or option=='D':
        amt=float(input('Enter amount:'))
        c.deposit(amt)
    elif option=='w' or option=='W':
        amt=float(input('Enter amount:'))
        c.withdraw(amt)
    elif option=='e' or option=='E':
        print('Thanks for Banking')
        break
    else:
        print('Invalid option..Plz choose valid option')
```

```
Welcome to RGMBANK
Enter Your Name:Karthikeya
d-Deposit
w-Withdraw
e-exit
Choose your option:d
Enter amount:10000
Balance after deposit: 10000.0
d-Deposit
w-Withdraw
e-exit
Choose your option:d
Enter amount:25000
Balance after deposit: 35000.0
d-Deposit
w-Withdraw
e-exit
Choose your option:w
Enter amount:5000
Balance after withdraw: 30000.0
d-Deposit
w-Withdraw
e-exit
Choose your option:w
Enter amount:1
Balance after withdraw: 29999.0
d-Deposit
w-Withdraw
e-exit
Choose your option:w
Enter amount:30000
Insufficient Funds..cannot perform this operation
d-Deposit
w-Withdraw
e-exit
Choose your option:e
Thanks for Banking
```

Any question?

If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

Thank You