



2. Method Overloading:

If 2 methods having same name but different type of arguments then those methods are said to be overloaded methods.

Eg: m1(int a)
m1(double d)

But in Python Method overloading is not possible.

If we are trying to declare multiple methods with same name and different number of arguments then Python will always consider only last method.

Demo Program:

```
1) class Test:
2)     def m1(self):
3)         print('no-arg method')
4)     def m1(self,a):
5)         print('one-arg method')
6)     def m1(self,a,b):
7)         print('two-arg method')
8)
9) t=Test()
10) #t.m1()
11) #t.m1(10)
12) t.m1(10,20)
```

Output: two-arg method

In the above program python will consider only last method.

How we can handle overloaded method requirements in Python:

Most of the times, if method with variable number of arguments required then we can handle with default arguments or with variable number of argument methods.

Demo Program with Default Arguments:

```
1) class Test:
2)     def sum(self,a=None,b=None,c=None):
3)         if a!=None and b!= None and c!= None:
4)             print('The Sum of 3 Numbers:',a+b+c)
5)         elif a!=None and b!= None:
6)             print('The Sum of 2 Numbers:',a+b)
7)         else:
8)             print('Please provide 2 or 3 arguments')
9)
10) t=Test()
```