Working with looping statements

1. while loop
2. for loop

while condition:

    statements

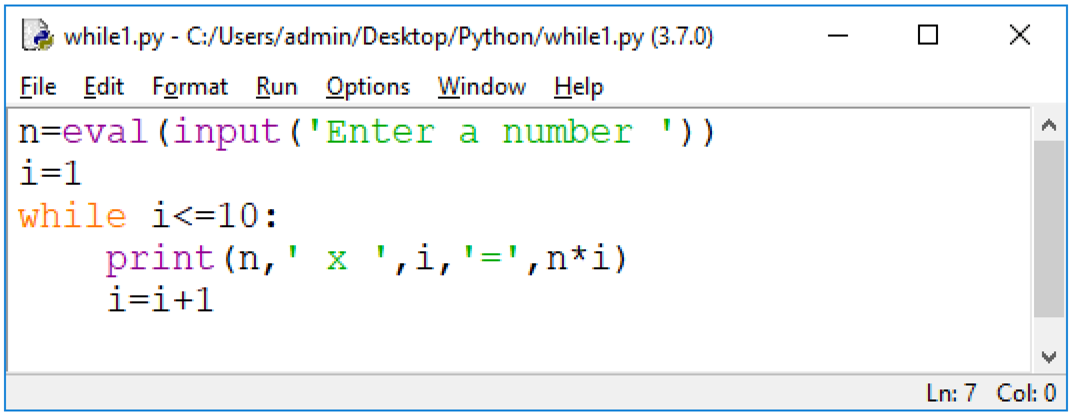
for *variable* in range(*[lboud], ubound, [step]*):

    statement

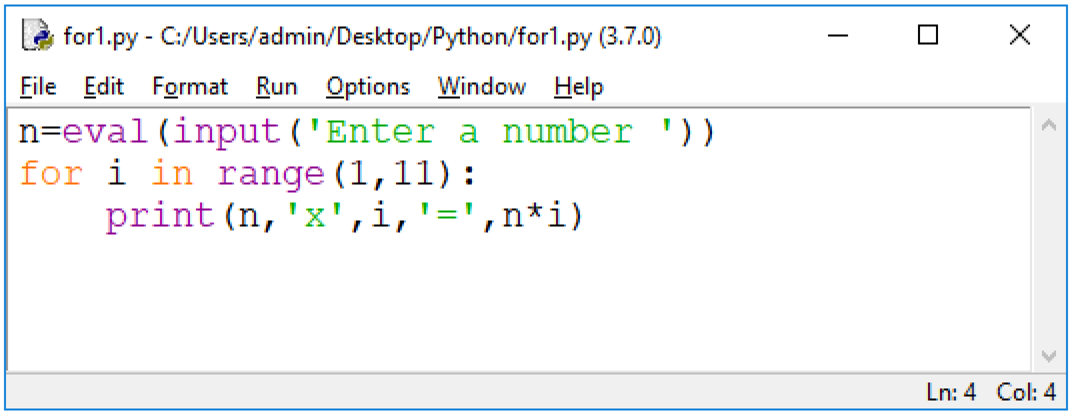
Note: ubound is not included

Example

WAP to input a number and show table of that number using while loop and for loop.


while1.py - C:/Users/admin/Desktop/Python/while1.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 7 Col: 0

```
n=eval(input('Enter a number '))
i=1
while i<=10:
    print(n,' x ',i,'=',n*i)
    i=i+1
```

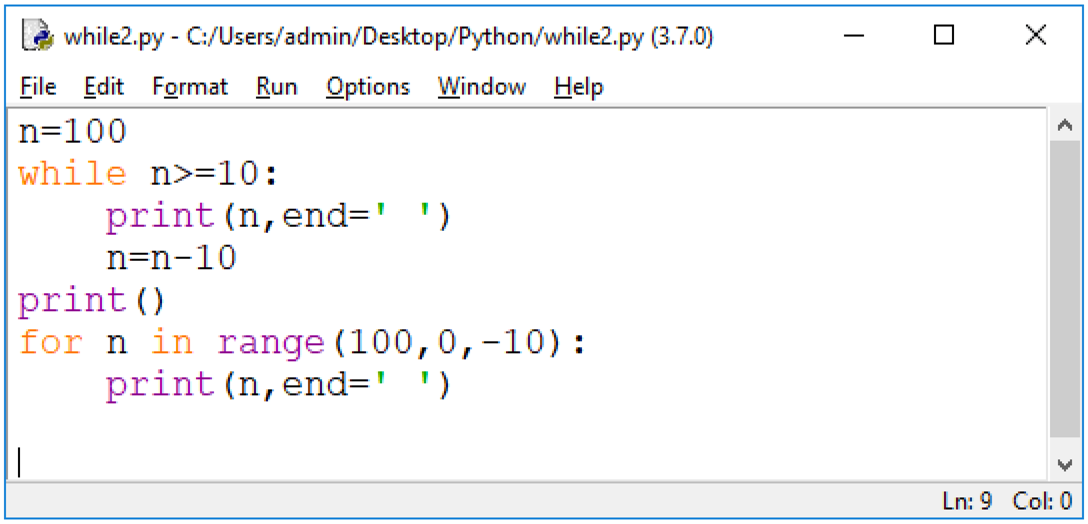

for1.py - C:/Users/admin/Desktop/Python/for1.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 4 Col: 4

```
n=eval(input('Enter a number '))
for i in range(1,11):
    print(n,'x',i,'=',n*i)
```

Example

WAP to print the following using while loop and for loop.

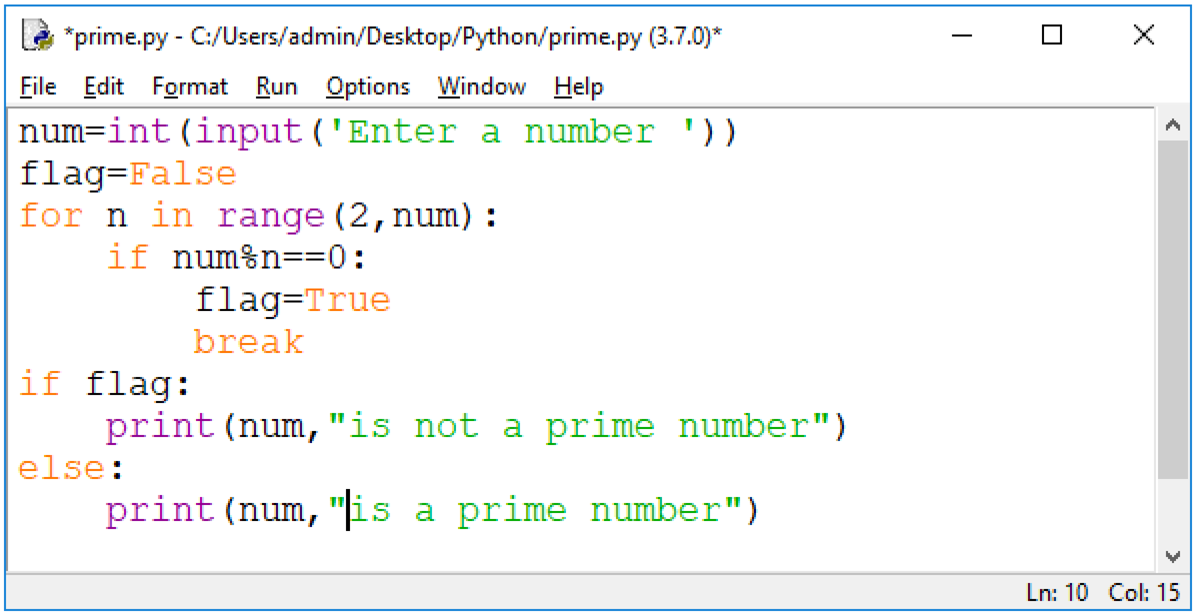
100 90 80… 10


while2.py - C:/Users/admin/Desktop/Python/while2.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 9 Col: 0

```
n=100
while n>=10:
    print(n,end=' ')
    n=n-10
print()
for n in range(100,0,-10):
    print(n,end=' ')
```

Class assignment

WAP to input a number and check it to be prime no.


prime.py - C:/Users/admin/Desktop/Python/prime.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 10 Col: 15

```
num=int(input('Enter a number '))
flag=False
for n in range(2,num):
    if num%n==0:
        flag=True
        break
if flag:
    print(num,"is not a prime number")
else:
    print(num,"is a prime number")
```

Creating and using the functions

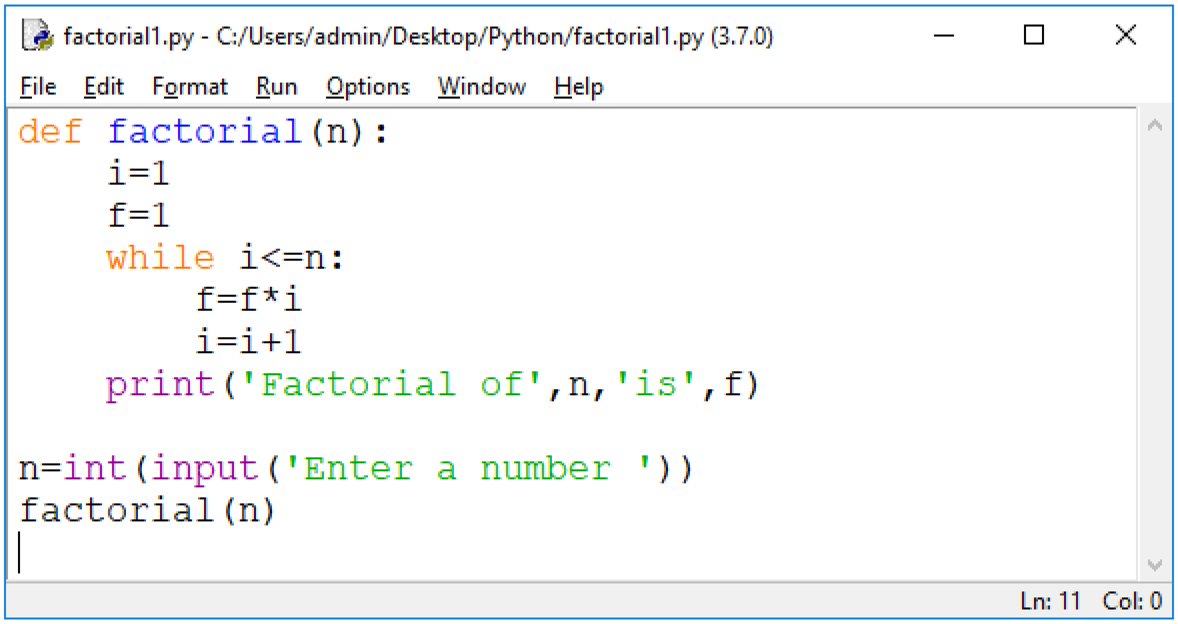
A set of statement given some name to avoid repetition of same statements again and again.

Use **def** keyword to define a function

Use **return** statement to return some value if required

Example

WAP having a function factorial() which takes a number and show the factorial of given number.

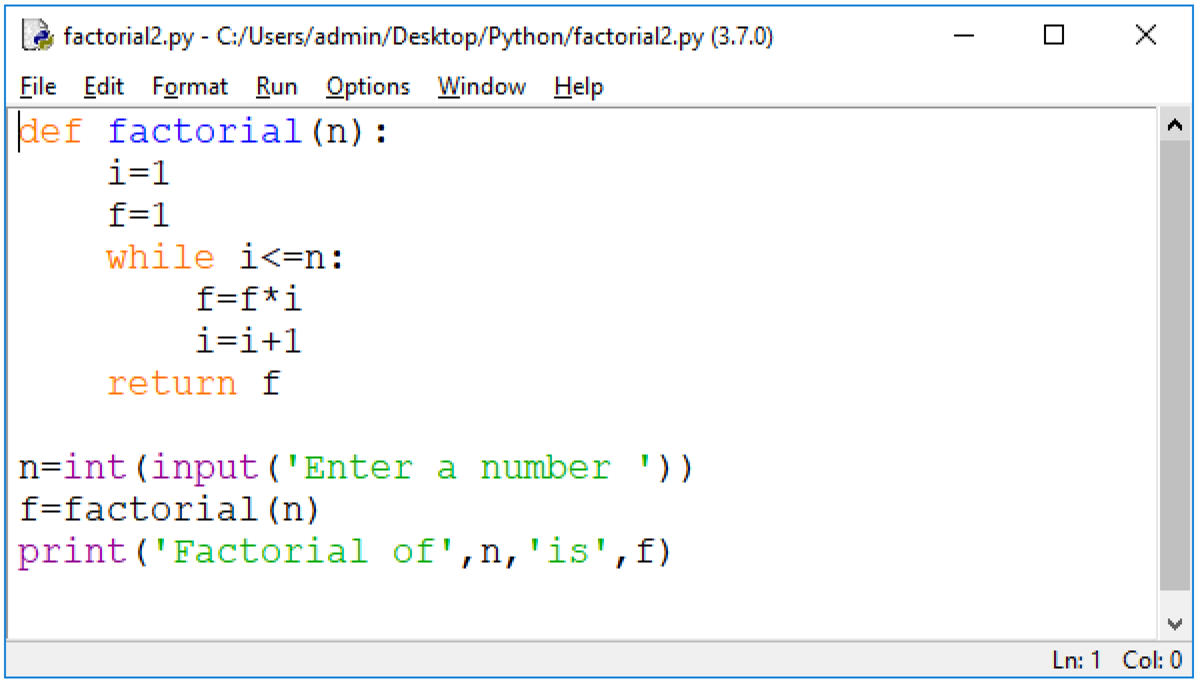

factorial1.py - C:/Users/admin/Desktop/Python/factorial1.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 11 Col: 0

```
def factorial(n):
    i=1
    f=1
    while i<=n:
        f=f*i
        i=i+1
    print('Factorial of',n,'is',f)

n=int(input('Enter a number '))
factorial(n)
```

Example

WAP having a function factorial() which takes a number and return the factorial of given number.

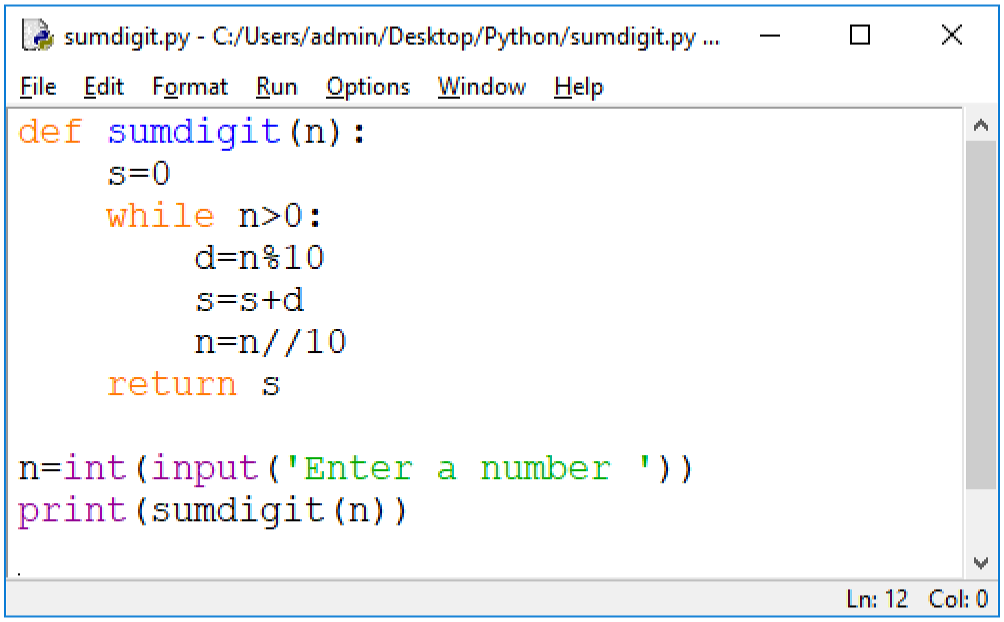

factorial2.py - C:/Users/admin/Desktop/Python/factorial2.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```
def factorial(n):
    i=1
    f=1
    while i<=n:
        f=f*i
        i=i+1
    return f

n=int(input('Enter a number '))
f=factorial(n)
print('Factorial of',n,'is',f)
```

Class assignment

WAP having a function which takes a number and returns sum of its digits. Input a number and call the function sumdigit().


sumdigit.py - C:/Users/admin/Desktop/Python/sumdigit.py ...
File Edit Format Run Options Window Help
Ln: 12 Col: 0

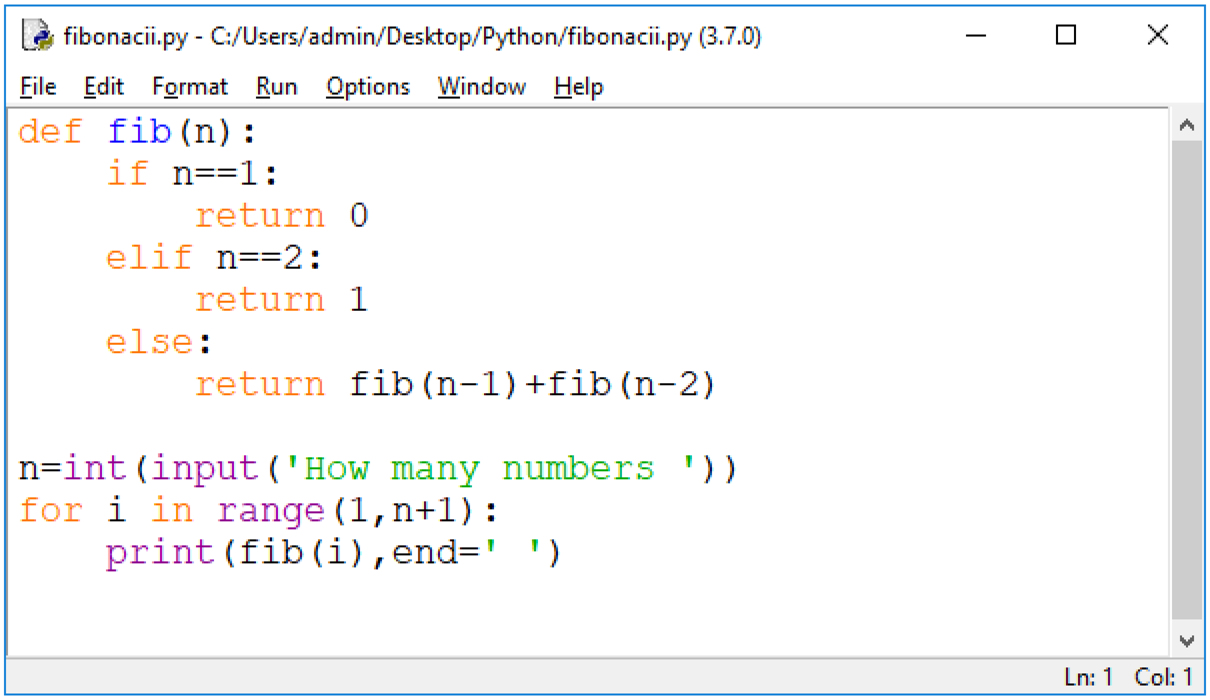
```
def sumdigit(n):
    s=0
    while n>0:
        d=n%10
        s=s+d
        n=n//10
    return s

n=int(input('Enter a number '))
print(sumdigit(n))
```

Using recursion

When a process repeats itself is called as recursive process and the function which implements such process is called as recursive process.

1. Factorial of a number
2. Fibonacci series
    a. 0 1 1 2 3 5 ....


fibonacii.py - C:/Users/admin/Desktop/Python/fibonacii.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 1 Col: 1

```python
def fib(n):
    if n==1:
        return 0
    elif n==2:
        return 1
    else:
        return fib(n-1)+fib(n-2)

n=int(input('How many numbers '))
for i in range(1,n+1):
    print(fib(i),end='  ')
```

Working with classes

A class is used to categorized different functions.

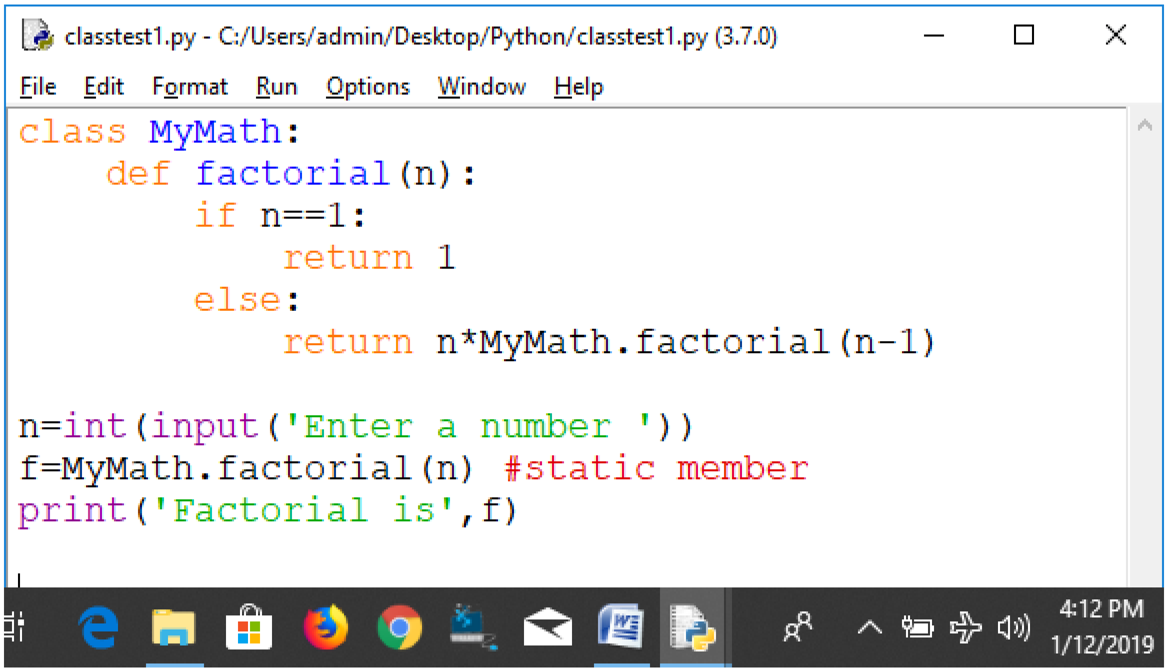
A class can have two types of functions

1. Static or class members
   a. Called with the class name
2. Non-static or instance member
   a. Called with some instance
   b. Use **self** keyword to hold the data related with instance

Use class keyword to define a class.

Example

Create a class MyMath having a function factorial() which takes a number and returns factorial of that number.

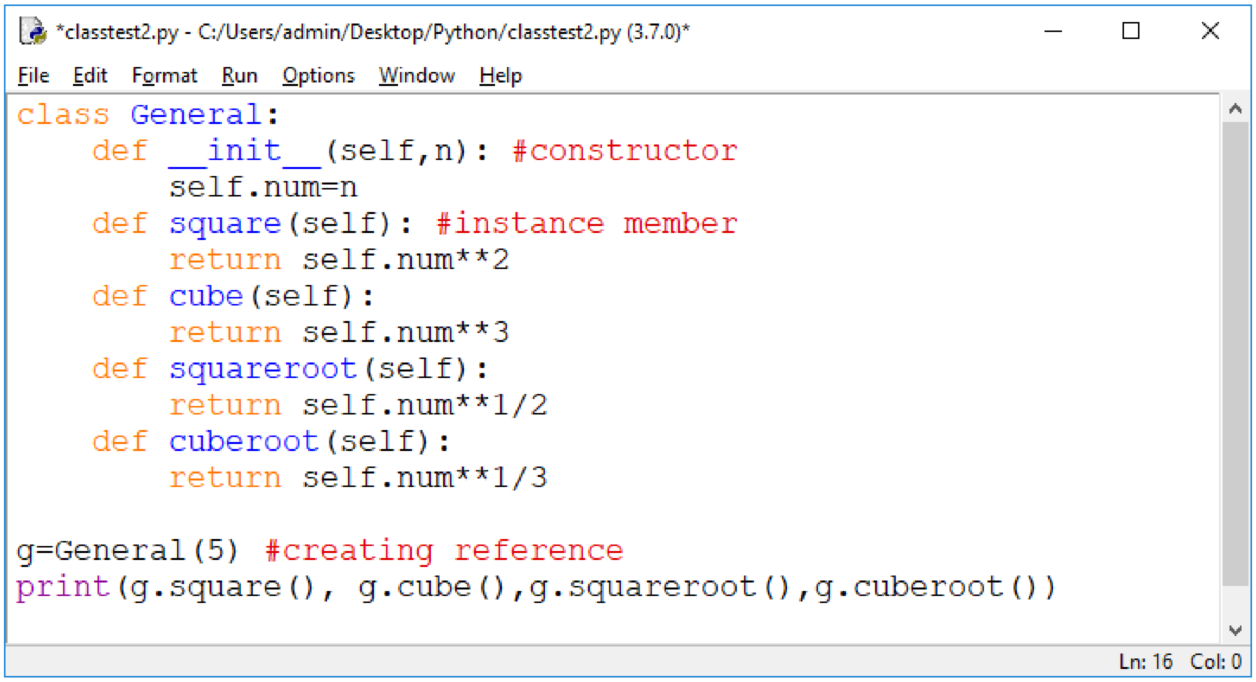

classtest1.py - C:/Users/admin/Desktop/Python/classtest1.py (3.7.0)
File Edit Format Run Options Window Help
4:12 PM
1/12/2019

```
class MyMath:
    def factorial(n):
        if n==1:
            return 1
        else:
            return n*MyMath.factorial(n-1)

n=int(input('Enter a number '))
f=MyMath.factorial(n) #static member
print('Factorial is',f)
```

Example

Create a class General having a data member as num. Create the functions square(), cube(), squareroot(), cuberoot() which works on num.


classtest2.py - C:/Users/admin/Desktop/Python/classtest2.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 16 Col: 0

```python
class General:
    def __init__(self,n): #constructor
        self.num=n
    def square(self): #instance member
        return self.num**2
    def cube(self):
        return self.num**3
    def squareroot(self):
        return self.num**1/2
    def cuberoot(self):
        return self.num**1/3

g=General(5) #creating reference
print(g.square(), g.cube(),g.squareroot(),g.cuberoot())
```