Name: Md.Abdullah
ID : IT-17015

Lab report no :05
lab report name : Introduction to Python
objectives:

1. Setup python environment for programing,
2. Learn the basics of python,
3. Create and run basic examples using python.

Theory:
Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.
Main Features of Python:

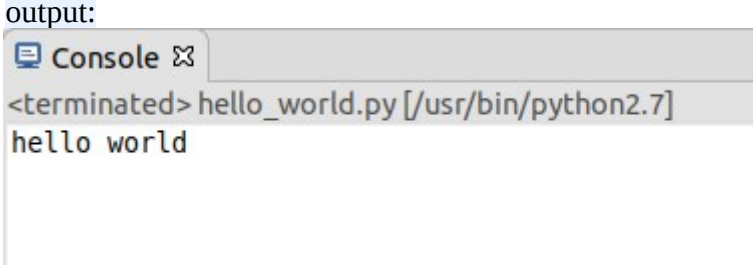
- ✔ Simple
- ✔ Easy to Learn
- ✔ Free and Open Source
- ✔ High-level Language
- ✔ Portable
- ✔ Multi-Plarform

Exercise 4.1.2: Write a Hello World program

Ans:

```
print('hello world')
```

output:

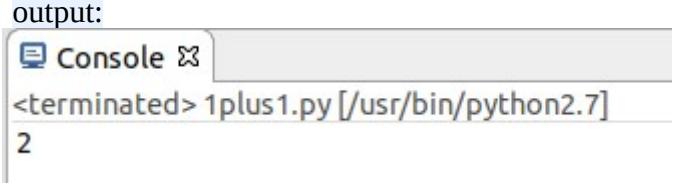


Exercise 4.1.3: Compute 1+1

Ans:

```
a=1+1
print(a)
```

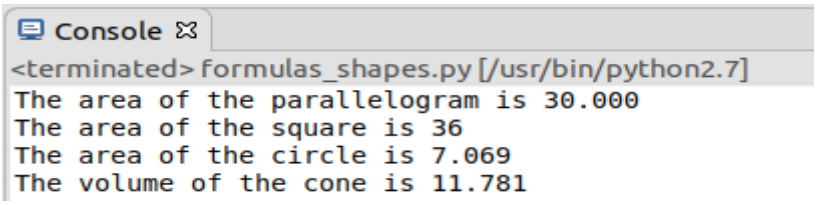
output:



Exercise 4.1.4: Type in program text

```
h = 5.0 # height
r = 1.5 # radius
b = 6.0 #width
area_parallelogram = h*b
print ('The area of the parallelogram is %.3f' % area_parallelogram)
area_square = b**2
print ('The area of the square is %g' % area_square)
area_circle = 3.1416*r**2
print ('The area of the circle is %.3f' % area_circle)
volume_cone = 1.0/3*3.1416*r**2*h
print ('The volume of the cone is %.3f' % volume_cone)
```

output :



Exercise 4.2.1: Verify the use of the following operator. Execute the example code in python script and provide the output.

Operator	Name	Explanation	Examples
+	Plus	Adds two objects	3 + 5 'a' + 'b'
-	Minus	Gives the subtraction of one number from the other; if the first operand is absent it is assumed to be zero.	-5.2 50 - 24
*	Multiply	Gives the multiplication of the two numbers or returns the string repeated that many times.	2 * 3 'la' * 3
**	Power	Returns x to the power of y	3 ** 4
/	Divide	Divide x by y	13 / 3
//	Divide and floor	Divide x by y and round the answer down to the nearest whole number	13 // 3 -13 // 3
%	Modulo	Returns the remainder of the division	13 % 3 -25.5 % 2.25
<<	Left shift	Shifts the bits of the number to the left by the number of bits specified. (Each number is represented in memory by bits or binary digits i.e. 0 and 1)	2 << 2
>>	Right shift	Shifts the bits of the number to the right by the number of bits specified.	11 >> 1
&	Bit-wise AND	Bit-wise AND of the numbers	5 & 3
|	Bit-wise OR	Bitwise OR of the numbers	5 | 3
^	Bit-wise XOR	Bitwise XOR of the numbers	5 ^ 3
~	Bit-wise invert	The bit-wise inversion of x is -(x+1)	~5
<	Less than	Returns whether x is less than y. All comparison operators return True or False.	5 < 3 3 < 5

>	Greater than	Returns whether x is greater than y	5 > 3
<=	Less than or equal to	Returns whether x is less than or equal to y	x = 3; y = 6; x <= y
>=	Greater than or equal to	Returns whether x is greater than or equal to y	x = 4; y = 3; x >= 3
==	Equal to	Compares if the objects are equal	x = 2; y = 2; x == y x = 'str'; y = 'stR'; x == y x = 'str'; y = 'str'; x == y
!=	Not equal to	Compares if the objects are not equal	x = 2; y = 3; x != y
not	Boolean NOT	If x is True, it returns False. If x is False, it returns True.	x = True; not x
and	Boolean AND	x and y returns False if x is False, else it returns evaluation of y	x = False; y = True; x and y
or	Boolean OR	If x is True, it returns True, else it returns evaluation of y	x = True; y = False; x or y

Ans:

plus (+) operator:

```
a=input('Enter 1st object:\n');
b=input('Enter 2nd object:\n');
plus=a+b
print 'plus:',plus
```

```
Console
<terminated> Plus.py [/usr/bin/python2.7]
Enter 1st object:
'a'
Enter 2nd object:
'b'
plus: ab
```

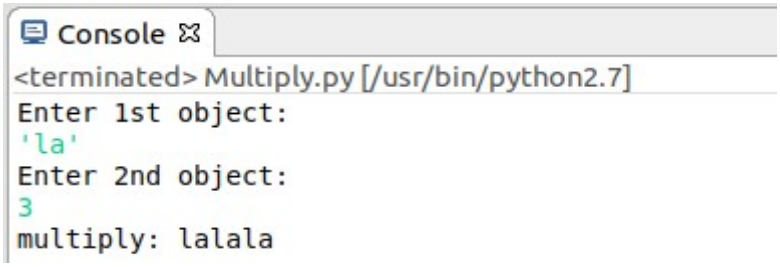
Minus (-) operator:

```
a=input('Enter 1st object:\n');
b=input('Enter 2nd object:\n');
minus=a-b
print 'minus:',minus
```

```
Console
<terminated> Minus.py [/usr/bin/python2.7]
Enter 1st object:
50
Enter 2nd object:
-24
minus: 74
```

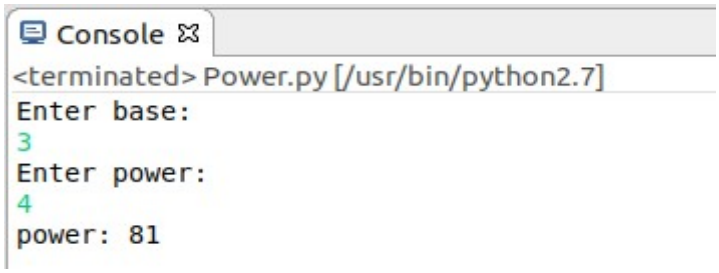
Multiply (*) operator:

```
a=input('Enter 1st object:\n');
b=input('Enter 2nd object:\n');
multiply=a*b
print 'multiply:',multiply
```



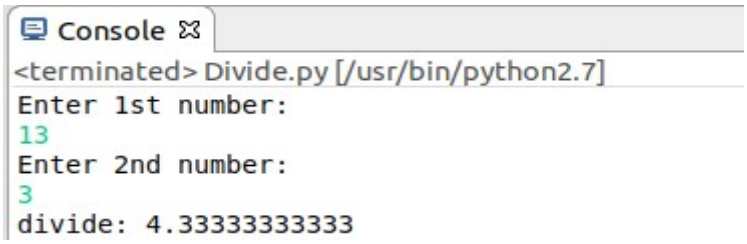
Power(**) operator:

```
a=input('Enter base:\n');
b=input('Enter power:\n');
power=a**b
print 'power:',power
```



Divide (/) operator:

```
a=float(input('Enter 1st number:\n'))
b=float(input('Enter 2nd number:\n'))
divide=a/b
print 'divide:',divide
```

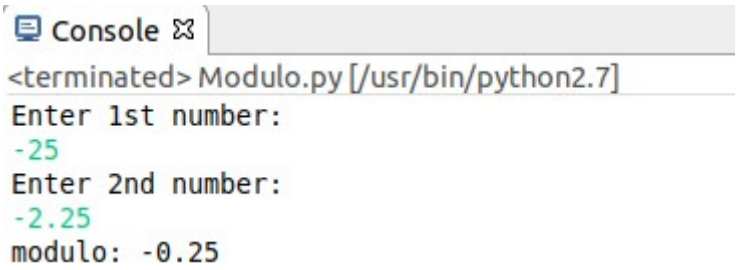


Divide and floor (//)operator:

```
a=float(input('Enter 1st number:\n'))
b=float(input('Enter 2nd number:\n'))
divide_and_flor=a//b
print 'divide_and_flor:',divide_and_flor
```

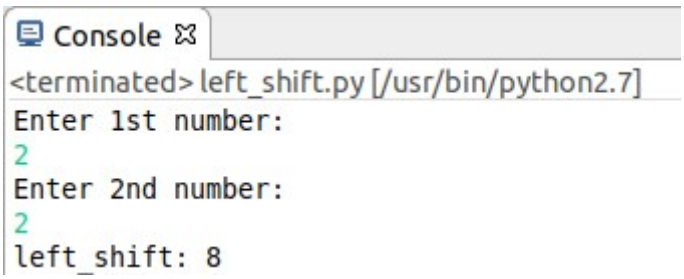
Modulo (%) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
modulo=a%b
print 'modulo:',modulo
```



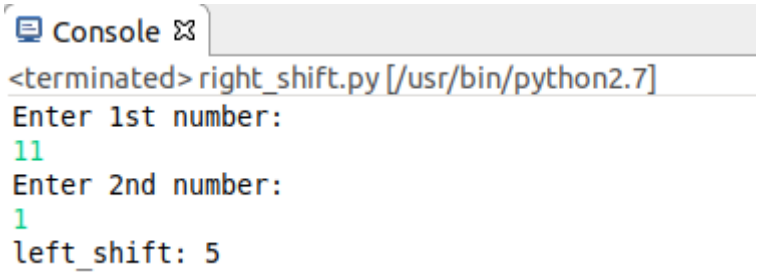
Left shift (<<) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
left_shift=a<<b
print 'left_shift:',left_shift
```



Right shift (>>) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
left_shift=a>>b
print 'left_shift:',left_shift
```

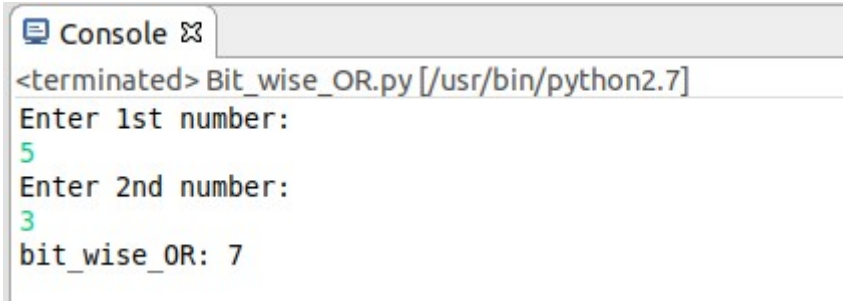


Bit-wise AND (&) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
bit_wise_AND=a&b
print 'bit_wise_AND:',bit_wise_AND
```

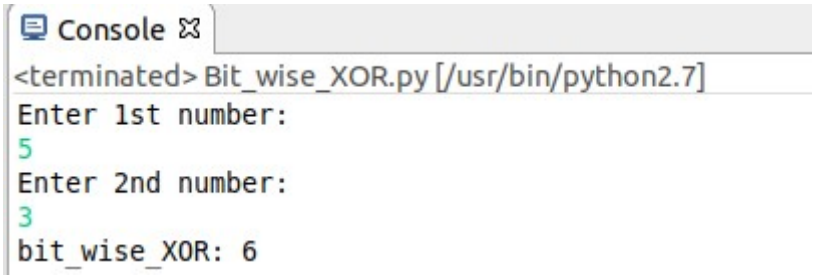
Bit-wise OR (|) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
bit_wise_OR=a|b
print 'bit_wise_OR:',bit_wise_OR
```



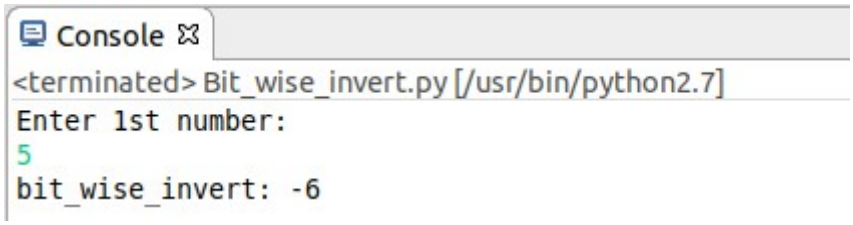
Bit-wise XOR (^) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
bit_wise_XOR=a^b
print 'bit_wise_XOR:',bit_wise_XOR
```



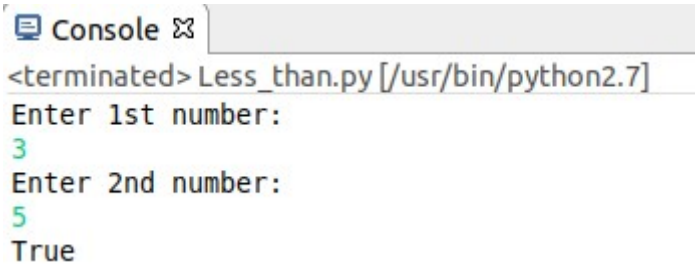
Bit-wise invert operator:

```
a=input('Enter 1st number:\n')
bit_wise_invert=-(a+1)
print 'bit_wise_invert:',bit_wise_invert
```



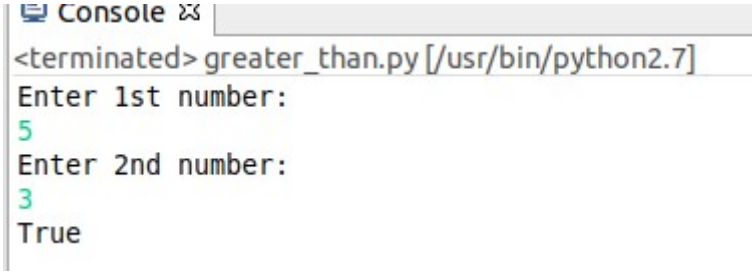
Less than (<)operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
if a<b:
    print True
else:
    print False
```



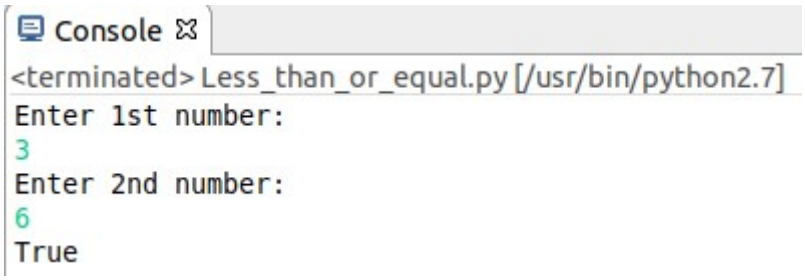
Greater than(>) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
if a>b:
    print True
else:
    print False
```



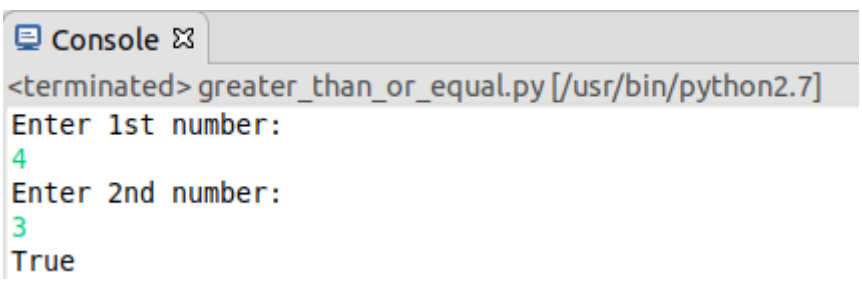
Less than or equal to(<=) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
if a<=b:
    print True
else:
    print False
```



Greater than or equal to (>=):

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
if a>=b:
    print True
else:
    print False
```

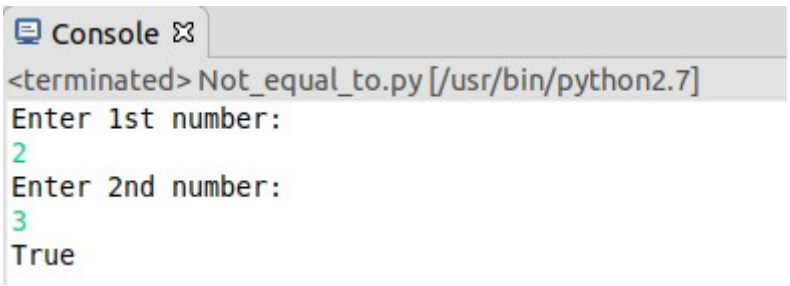


Equal to (==) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
if a==b:
    print True
else:
    print False
```

Not equal to(!=) operator:

```
a=input('Enter 1st number:\n')
b=input('Enter 2nd number:\n')
if a!=b:
    print True
else:
    print False
```



Boolean NOT(not) operator:

```
from operator import not_
a=True
print not True
```

Boolean AND(and) operator:

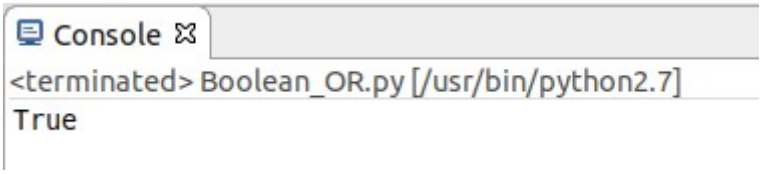
```
a=True
b=False

print  a and b
```

Boolean OR(or) operator:

```
a=True
b=False

print  a or b
```

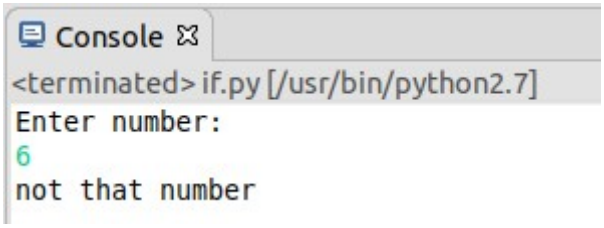


Exercise 4.2.2: The if statement

Create a program for taking a number from the user and check if it is the number that you have saved in the code

Ans:

```
a=input('Enter number:\n')
b=5;
if a==b:
   print a
else:
   print "not that number"
```

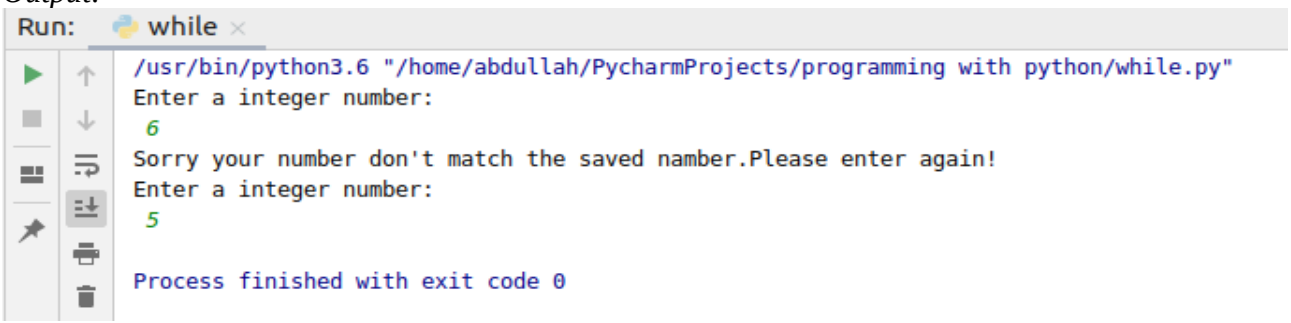


Exercise 4.2.3: The while Statement
Create a program for taking a number from the user and check if it is the number that you have saved in the code. The program run until the user will guess the number

program:

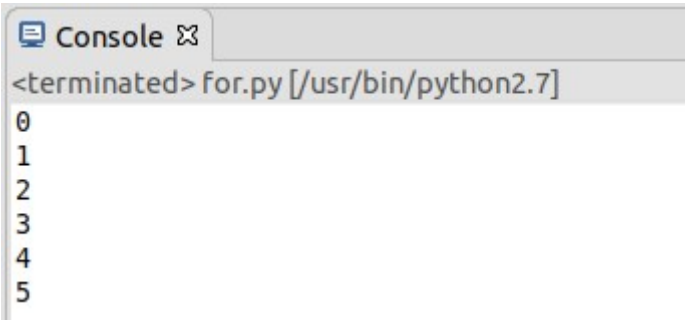
```
saved_namber=5
number=int(input('Enter a integer number:\n '))
while number !=saved_namber:
        print("Sorry your number don't match the saved namber.Please enter again!")
        number= int(input('Enter a integer number:\n '))
```

Output:



Exercise 4.2.4: The for Statement
Create a program for printing a sequence of numbers.
Ans:

```
for x in range(6):
    print(x)
```



Question 5.1: *Explain what is eclipse? And why we use it for programing on python?*

Ans:
Eclipse is an integrated development environment (IDE) for developing applications using the Java programming language and other programming languages such as C/C++, Python, PERL, Ruby etc.

We use eclipse for developing python modules.

Question 5.2: *Explain three main characteristics of python that you test in the lab?*

Ans:

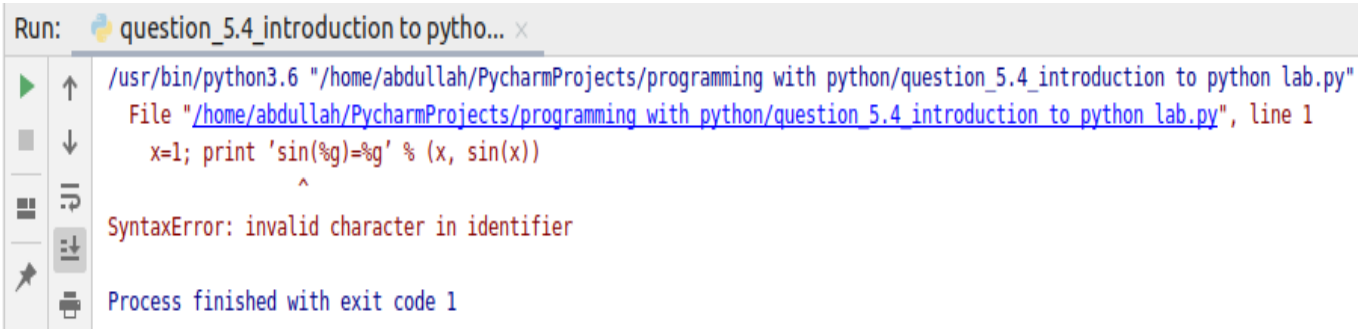
- Simple
- Easy to Learn
- *Free and Open Source*

Question 5.4: *Find error(s) in a program*
Suppose somebody has written a simple one-line program for computing sin(1):
x=1; print 'sin(%g)=%g' % (x, sin(x))
Create this program and try to run it. What is the problem? Which is the correct code?

Ans:

Program:
x=1; print 'sin(%g)=%g' % (x, sin(x))

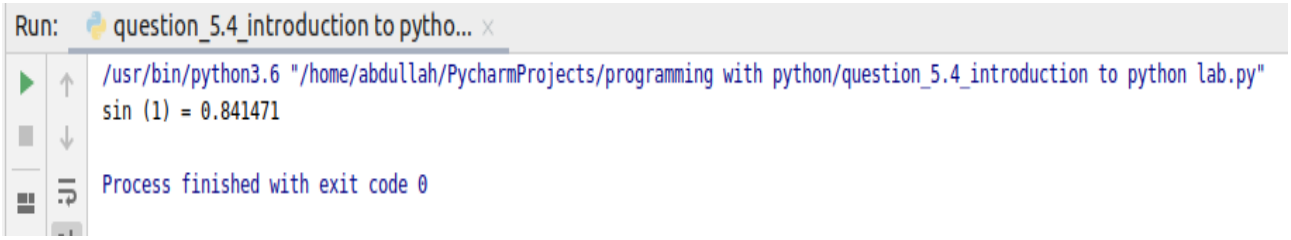
Problem:



Correct code:

```
import math as m
x=1
print("sin (%g) = %g"%(x,m.sin(x)))
```

Output:



Question 5.5: *Create a python program that combines at least 4 operators and one statement (if, while or for)*

Ans:

```
a=input('Enter number:\n')
b=5;
if a>b:
   print a-b
else:
   print a+b
```