Mawlana Bhashani Science & Technology University

Lab Report No : 01

Course Code : ICT3208

Course Title : Computer Network Lab

Report Name : Introduction to python

Submitted by

Name : Maskur Al Shal Sabil

Id: IT18021

3rd year 2nd Semester

Session 2017-2018

Dept of ICT

MBSTU

Submitted To

Nazrul Islam

Assistant Professor

Dept of ICT

MBSTU

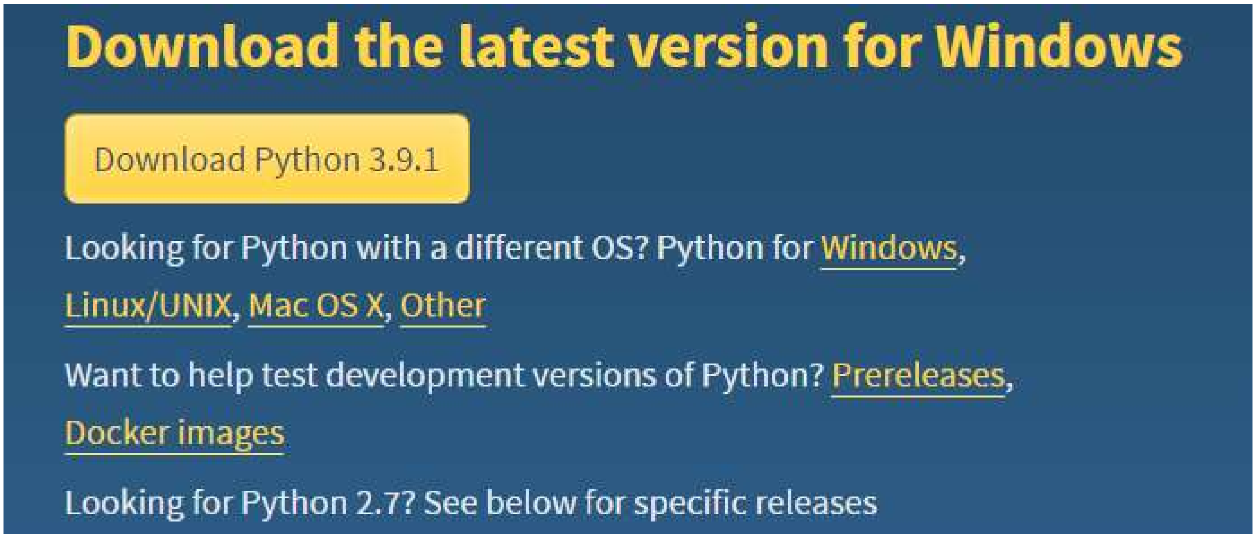
What is python?

Python is an easy , object oriented high level programming language with interpreted nature make it an ideal language for scripting and many other application development .

How to install ?

Step 1> Install python3 or python2 on our computer . Go to this site https://www.python.org/downloads/

And then download python in our computer .



Download the latest version for Windows

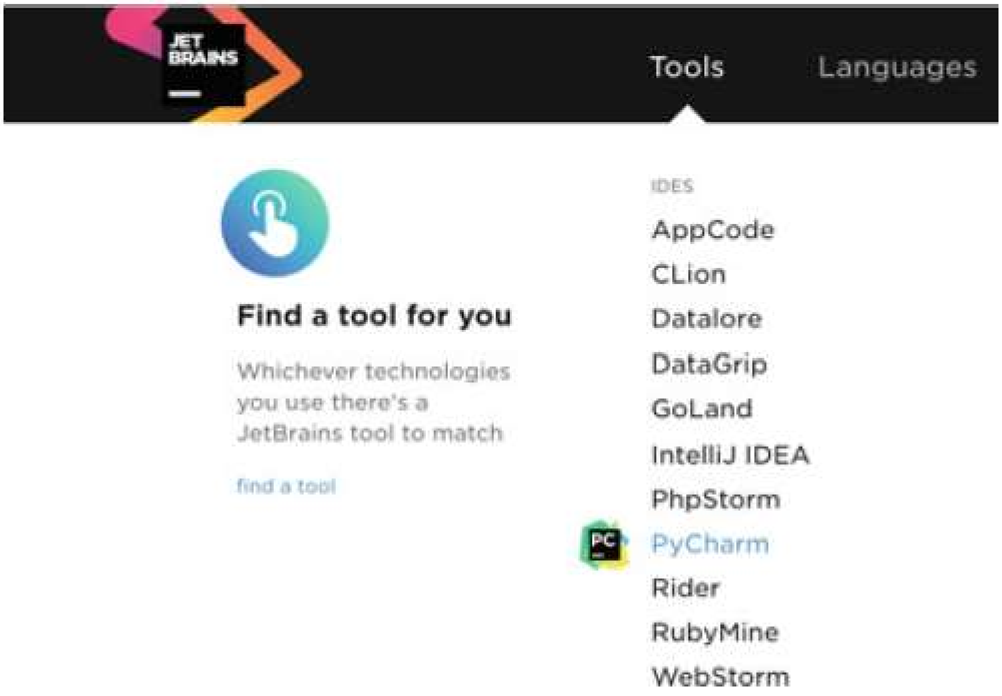
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Step2 -> To write code more easily we can use any editor like jupyter, pycharm etc. Here we will download pycharm IDE.

To download we need to visit : https://www.jetbrains.com/. Look for the menu heading ‘Tools’ and select that. We will see a long list of tools, which should include PyCharm.



Select this option.,

And select the 'DOWNLOAD NOW'. Make sure that we select the

operating system we use (there are options for Windows, Mac OS, and Linux).

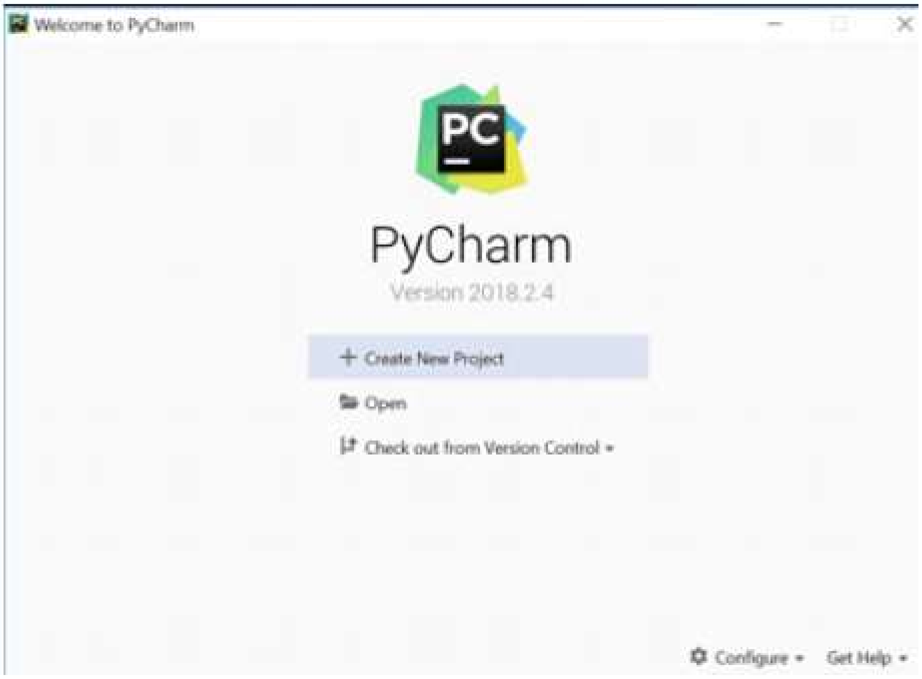
There are then two download options available: Professional and Community.

The Professional version is the charged for option, while the Community version is free.

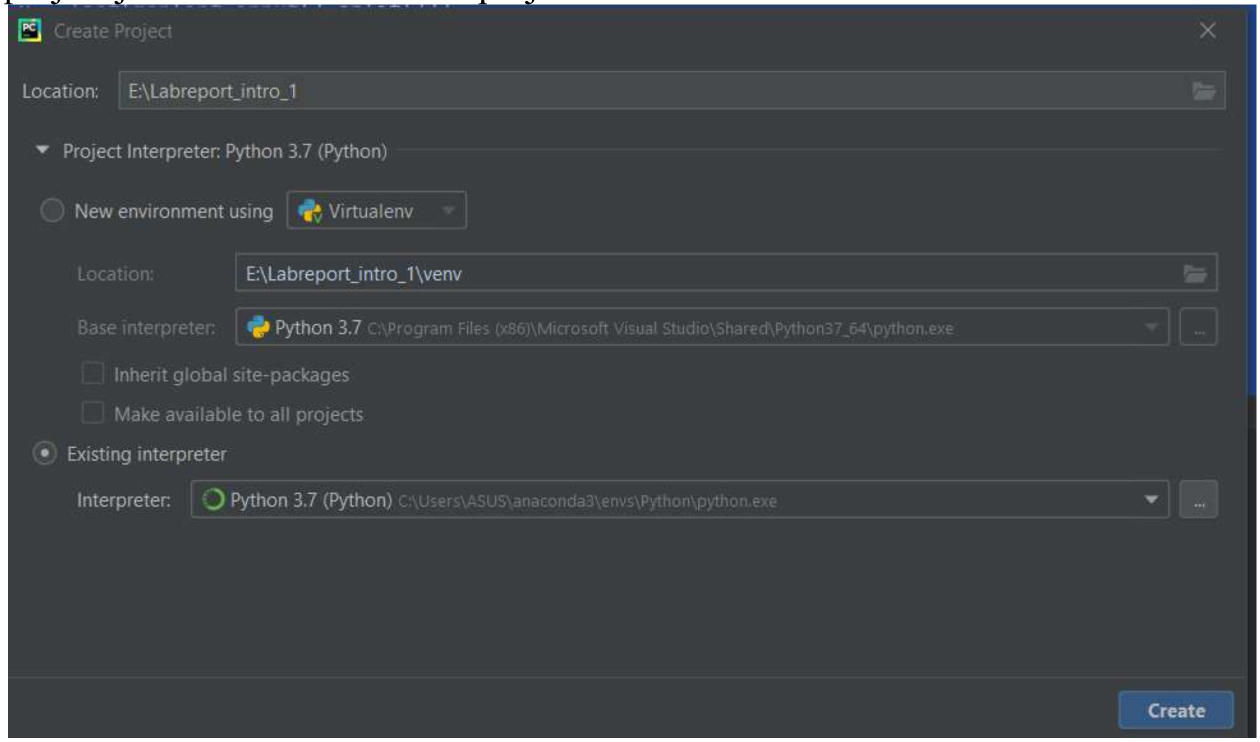
After completing download we need to run the installer . And the next setting up step ,.

Setting Up the IDE

We need to first start the PyCharm IDE. Once started, the first dialog shown to us asks if we want to import any settings we may have had for another version of PyCharm. At this point, select 'Do not import settings'.
Once we have completed this, click the 'Start PyCharm' option. It presented with the landing screen for PyCharm:



We can also change the light mode to dark mode. Now if we want to create a project just click on create new project .



Location: E:\Labreport_intro_1

Project Interpreter: Python 3.7 (Python)

○ New environment using Virtualenv

Location: E:\Labreport_intro_1\venv

Base interpreter: Python 3.7 C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe

□ Inherit global site-packages

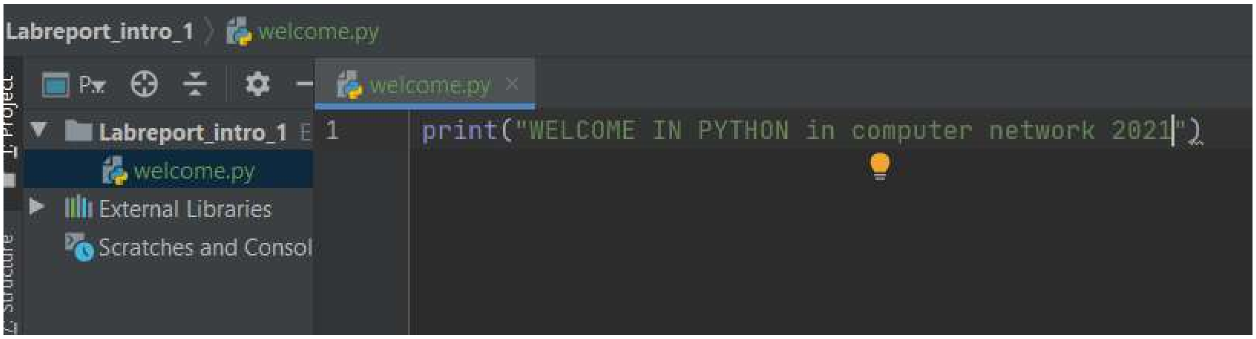
□ Make available to all projects

● Existing interpreter

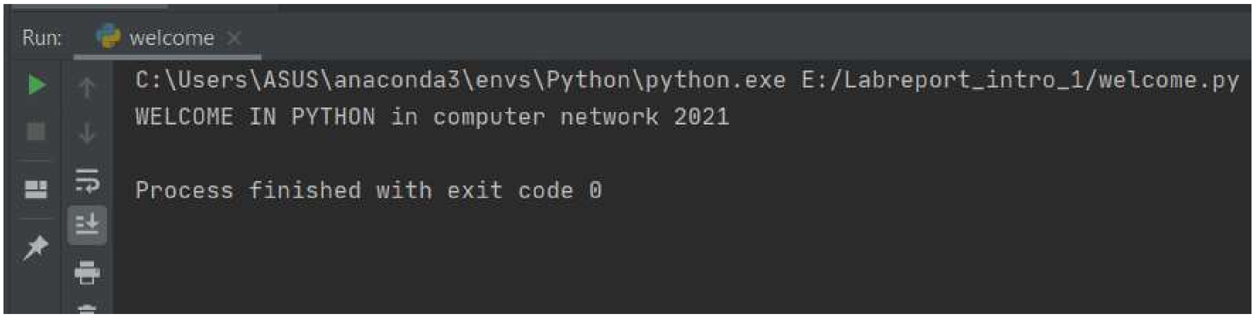
Interpreter: Python 3.7 (Python) C:\Users\ASUS\anaconda3\envs\Python\python.exe

And then we need to specify the python interpreter that was downloaded in previous.

Now we can create a python file and run it . Here I create a python file name welcome.



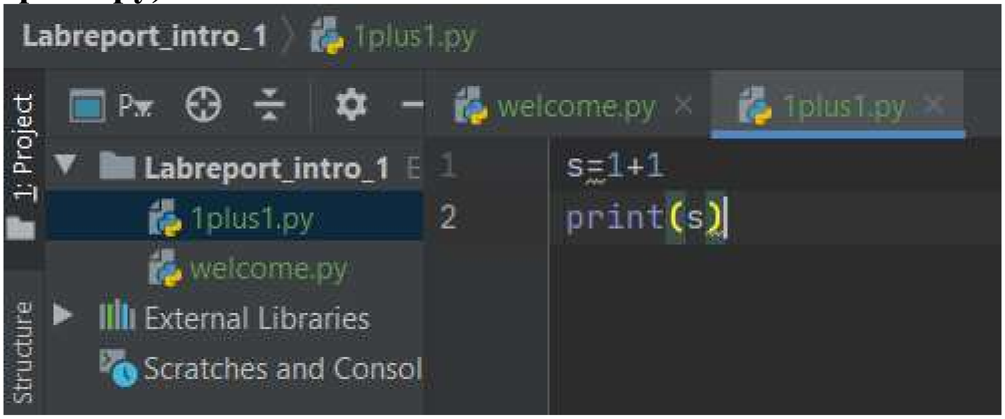
```
print("WELCOME IN PYTHON in computer network 2021")
```



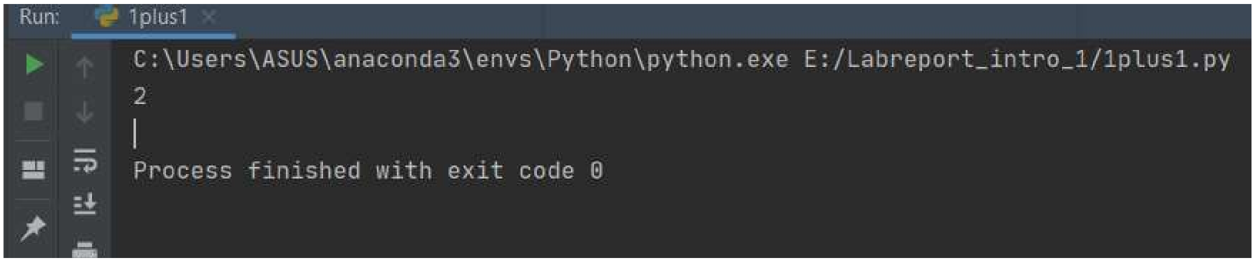
```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labreport_intro_1/welcome.py
WELCOME IN PYTHON in computer network 2021

Process finished with exit code 0
```

Basic Program in python :
Exercise 4.1.3: Compute 1+1
The first exercise concerns some very basic mathematics and programming: assign the
result of 1+1 to a variable and print the value of that variable (save as 1plus1.py).



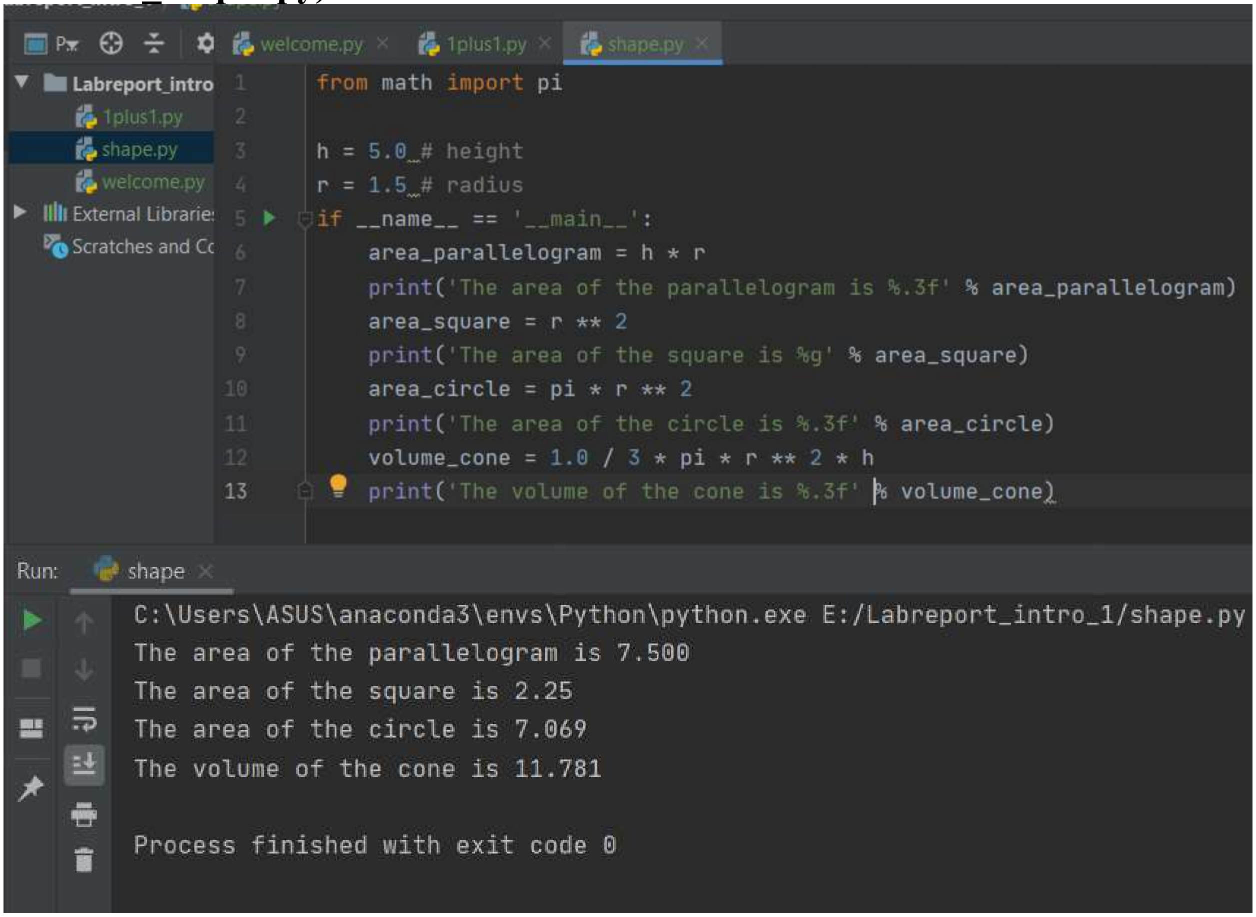
```
s=1+1
print(s)
```



```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labreport_intro_1/1plus1.py
2

Process finished with exit code 0
```

Exercise 4.1.4: Type in program text
Type the following program in your editor and execute it. If your program does not work,
check that you have copied the code correctly and debug it (save as formulas_shapes.py).



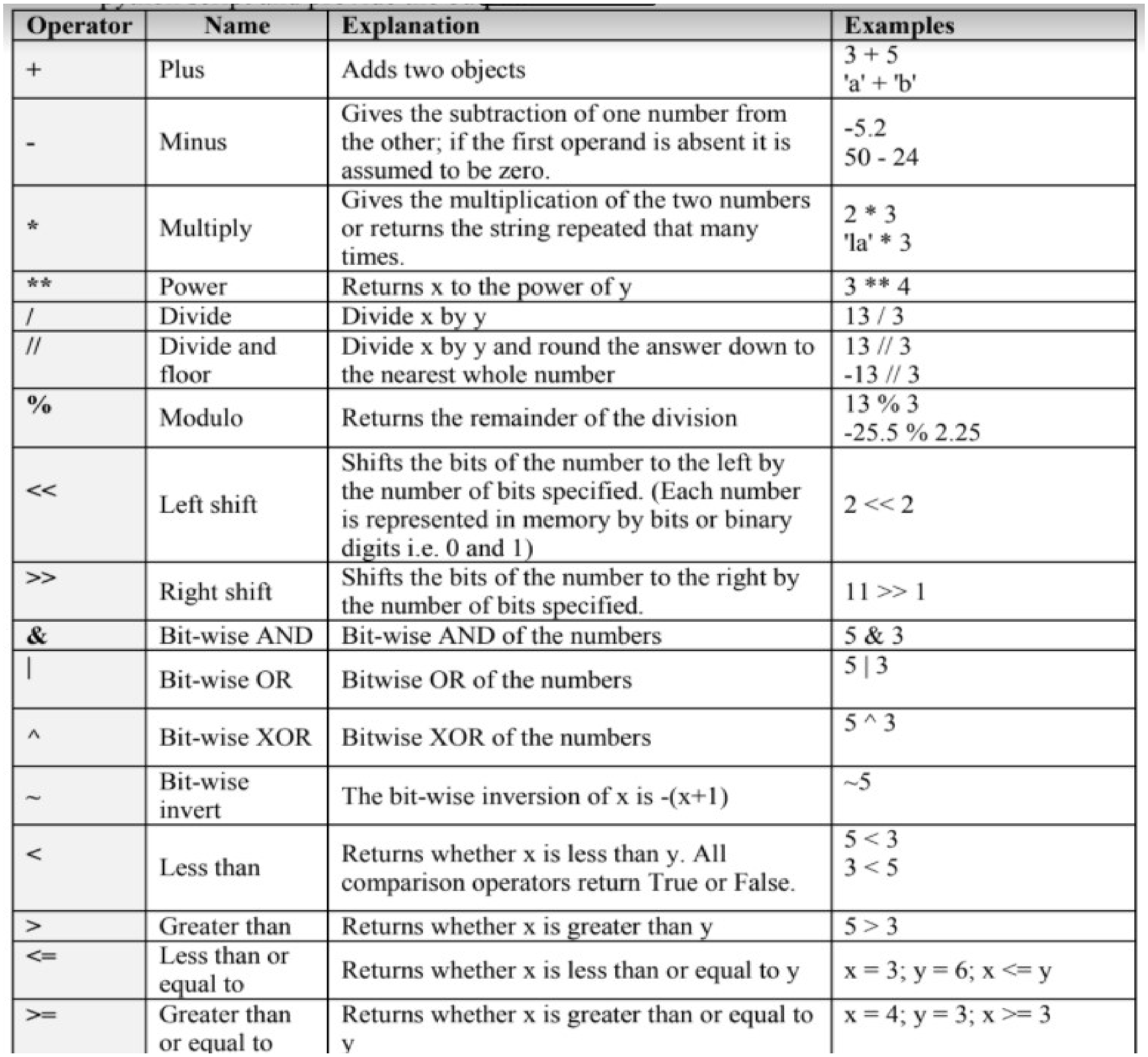
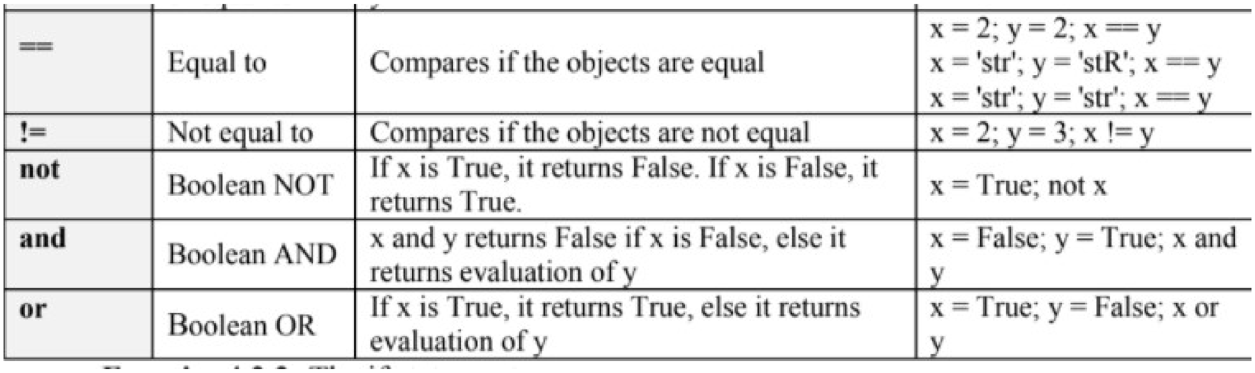
```
from math import pi

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

```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labreport_intro_1/shape.py
The area of the parallelogram is 7.500
The area of the square is 2.25
The area of the circle is 7.069
The volume of the cone is 11.781

Process finished with exit code 0
```

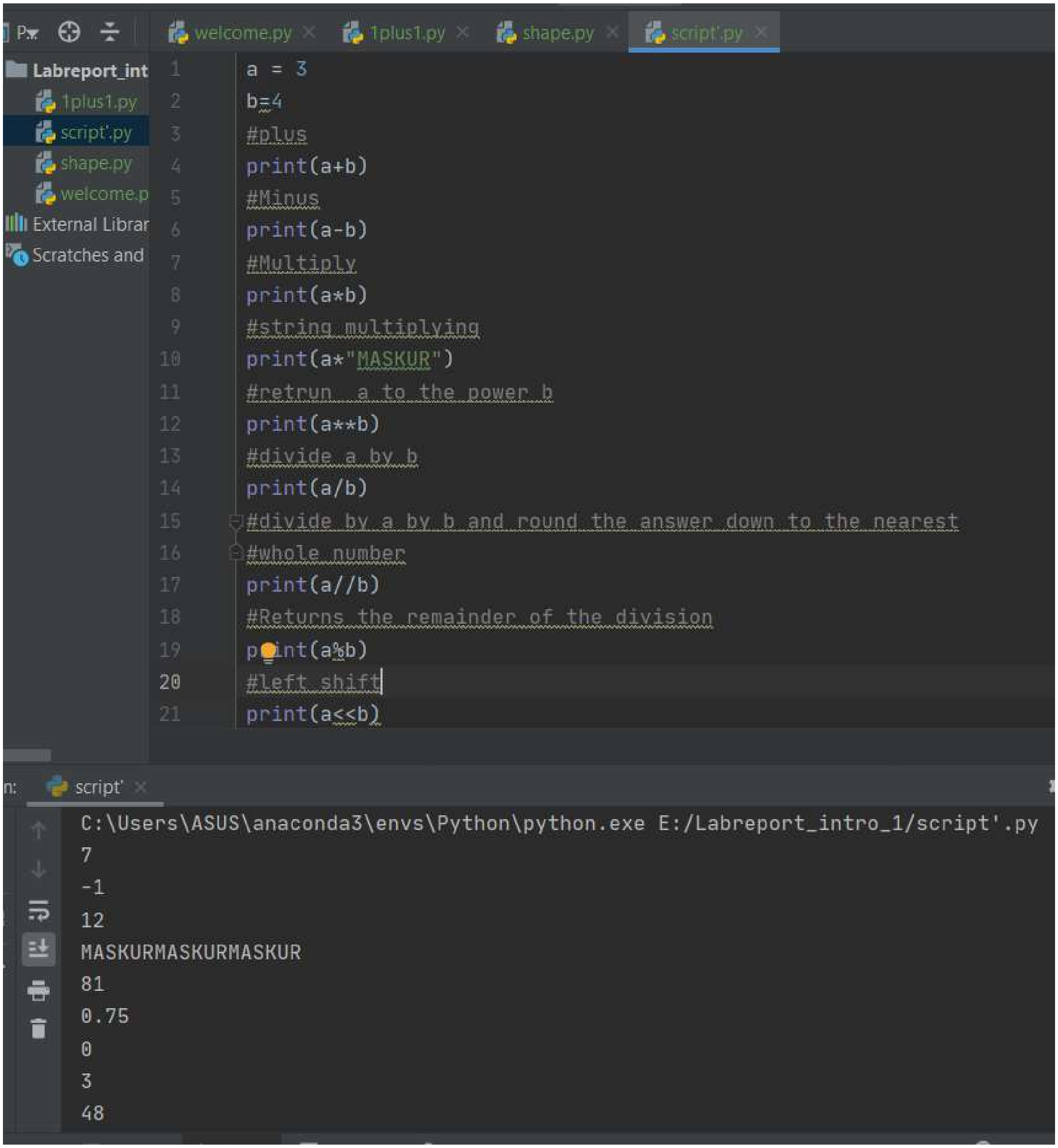
Exercise 4.2.1: Verify the use of the following operator. Execute the example code in
python script and provide the output.

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

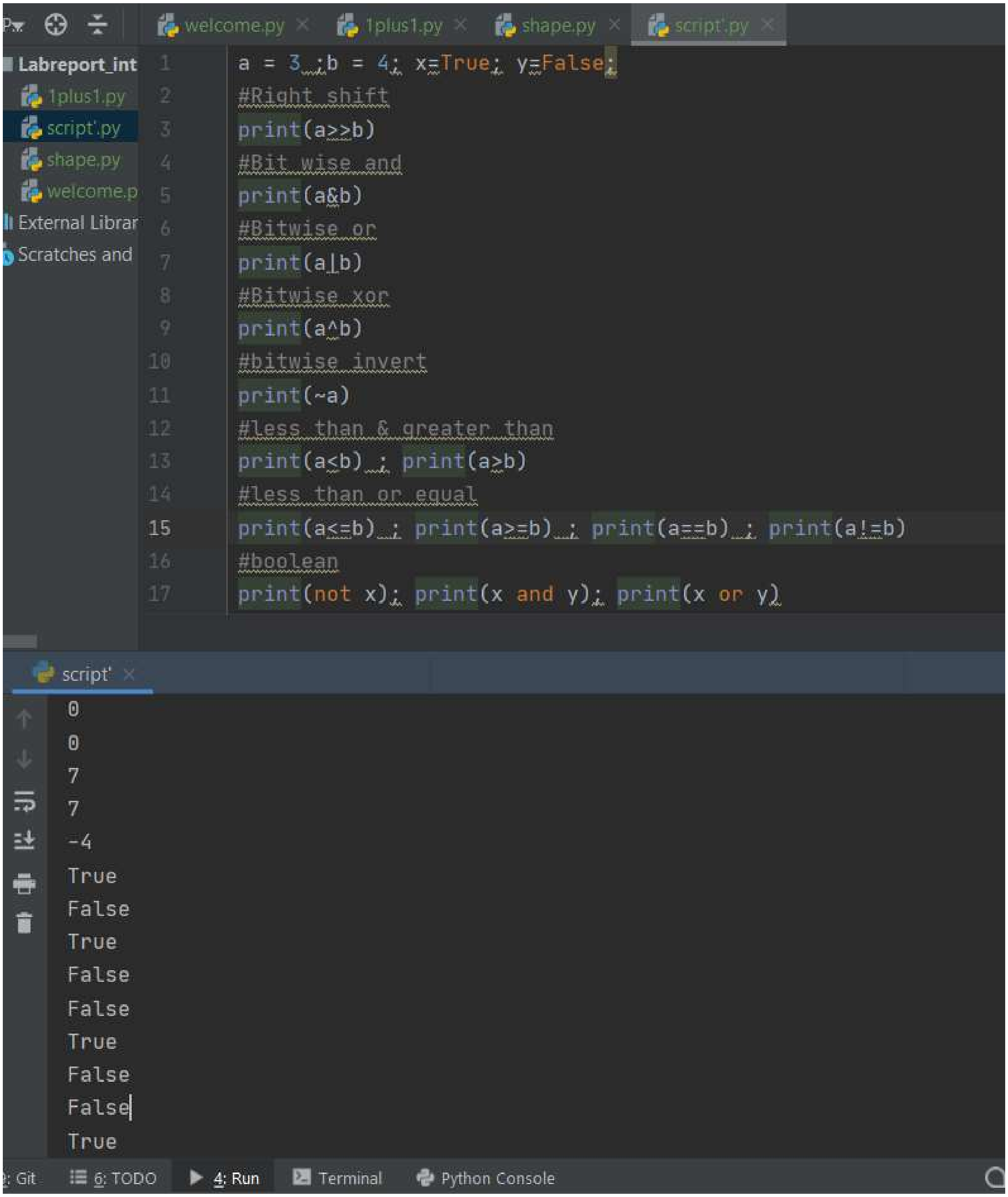
==	Equal to	Compares if the objects are equal	x = 2; y = 2; x == y
			x = 'str'; y = 'stR'; x == y
			x = 'str'; y = 'str'; x == y

!=	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



```
a = 3
b=4
#plus
print(a+b)
#Minus
print(a-b)
#Multiply
print(a*b)
#string multiplying
print(a*"MASKUR")
#retrun  a to the power b
print(a**b)
#divide a by b
print(a/b)
#divide by a by b and round the answer down to the nearest
#whole number
print(a//b)
#Returns the remainder of the division
print(a%b)
#left shift
print(a<<b)
```

```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labreport_intro_1/script'.py
7
-1
12
MASKURMASKURMASKUR
81
0.75
0
3
48
```



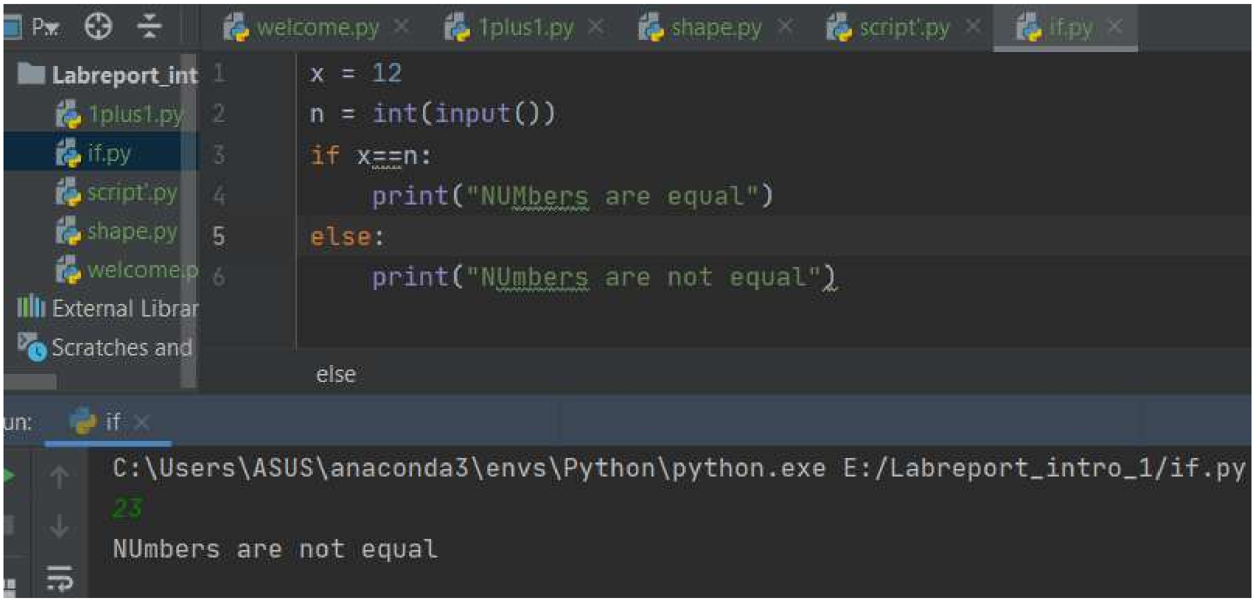
```python
a = 3 ;b = 4; x=True; y=False;
#Right shift
print(a>>b)
#Bit wise and
print(a&b)
#Bitwise or
print(a|b)
#Bitwise xor
print(a^b)
#bitwise invert
print(~a)
#less than & greater than
print(a<b) ; print(a>b)
#less than or equal
print(a<=b) ; print(a>=b) ; print(a==b) ; print(a!=b)
#boolean
print(not x); print(x and y); print(x or y)
```

```
0
0
7
7
-4
True
False
True
False
False
True
False
False
True
```

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

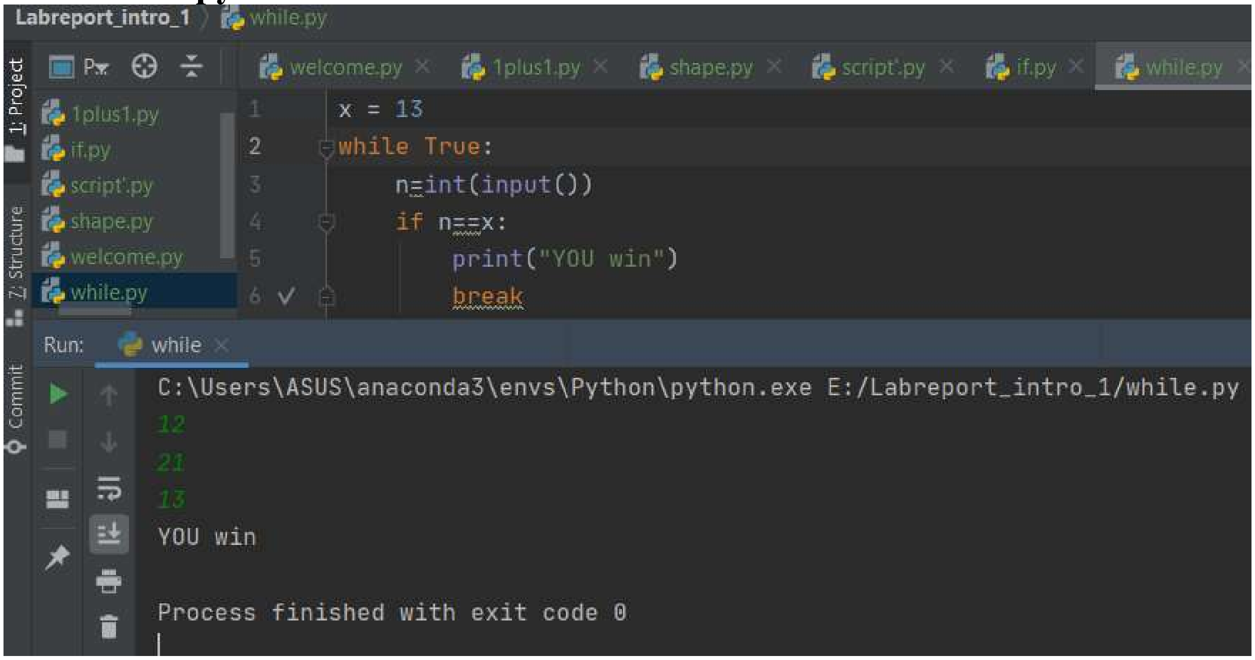
have saved in the code (TIP: use input command). Save the file as if.py



```
x = 12
n = int(input())
if x==n:
    print("NUMbers are equal")
else:
    print("NUmbers are not equal")
```

```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labreport_intro_1/if.py
23
NUmbers are not equal
```

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. Save the
file as while.py

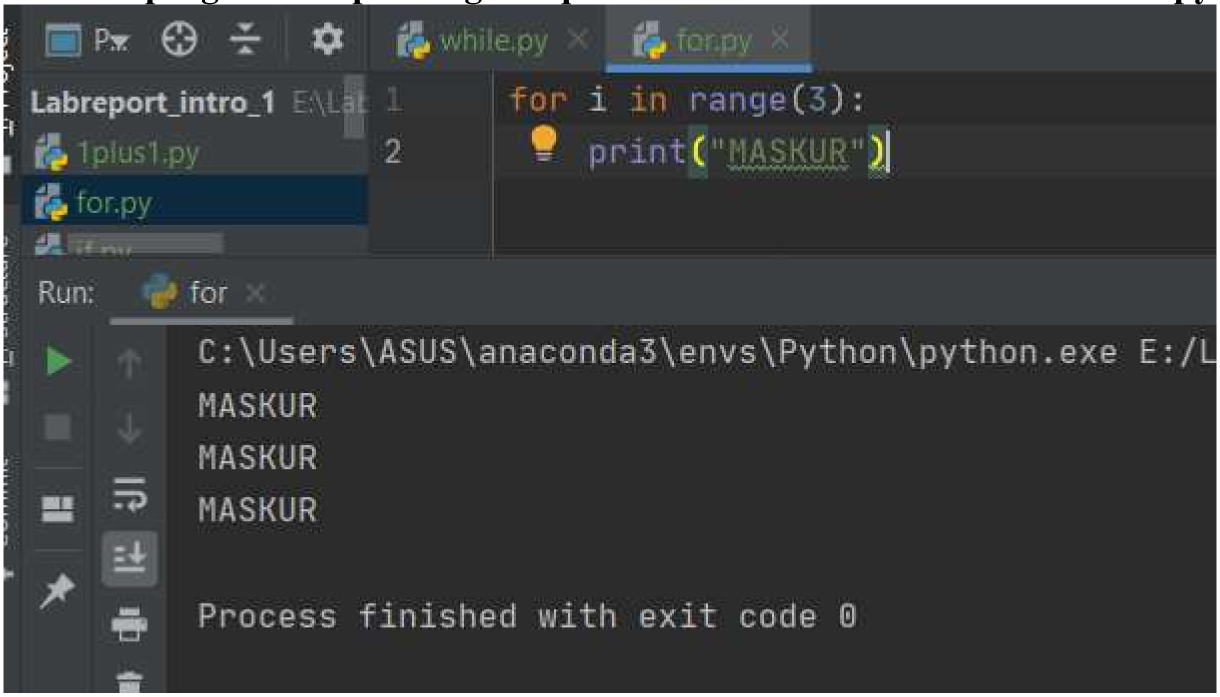


```
x = 13
while True:
    n=int(input())
    if n==x:
        print("YOU win")
        break
```

```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labreport_intro_1/while.py
12
21
13
YOU win

Process finished with exit code 0
```

Exercise 4.2.4: The for Statement
Create a program for printing a sequence of numbers. Save the file as for.py



```
for i in range(3):
    print("MASKUR")
```

```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/L
MASKUR
MASKUR
MASKUR

Process finished with exit code 0
```

5. Questions

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

Ans: In the field of computer science Eclipse is an IDE that use for developing applications using various programming language such as java,python,c,c++ ,Ruby etc.

The reason behind it uses :

1. This IDE is composed of plug-ins and is designed to be extensible using additional plugins.
2. It is easy to write the code and interpret.
3. It is free to download and use .

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

Ans :

1. Python has a simple syntax similar to English Language .
2. It runs on interpreter.
3. It has the feature like all other language such as loop, if condition, bollean besides it has many modules that make it more special .

**Question 5.3: Which is the difference between empty module and main module when
creating a python script?**

A *module* is a file containing Python code. Python modules have the `.py` extension.

Python code can be managed using:

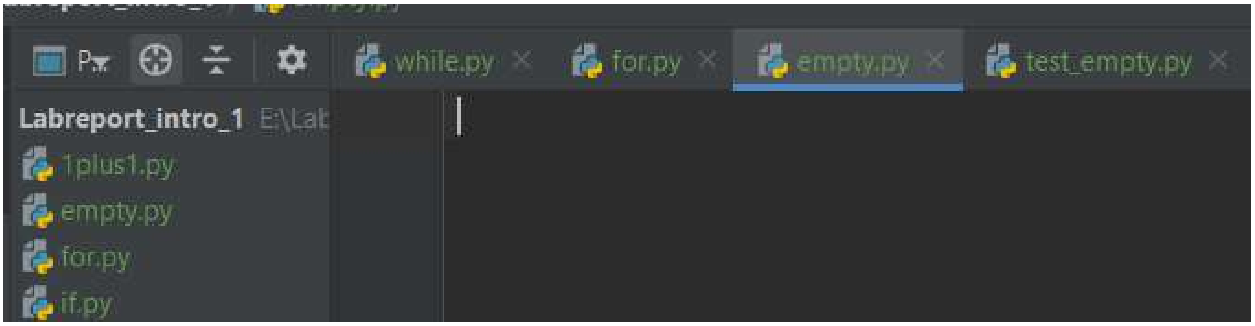
- functions
- classes
- modules
- packages

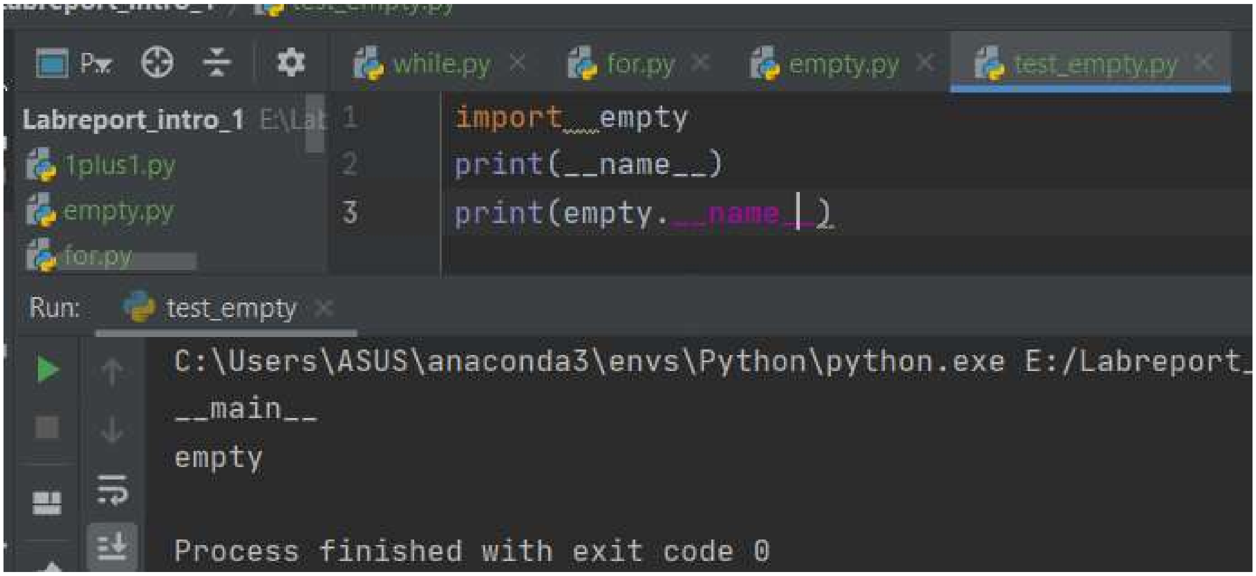
Python modules are used to organize Python code. For example, database related code is placed inside a database module, security code in a security module etc. Smaller Python scripts can have one module. But larger programs are split into several modules. Modules are grouped together to form packages.

Python module names

A module name is the file name with the `.py` extension. When we have a file called `empty.py`, empty is the module name. The `__name__` is a variable that holds the name of the module being referenced. The current module, the module being executed (called also the main module) has a special name: `'__main__'`. With this name it can be referenced from the Python code.

We have two files in the current working directory: `empty.py` and `test_empty.py`. The second module is the main module, which is executed. It imports the first module. Modules are imported using the `import` keyword.





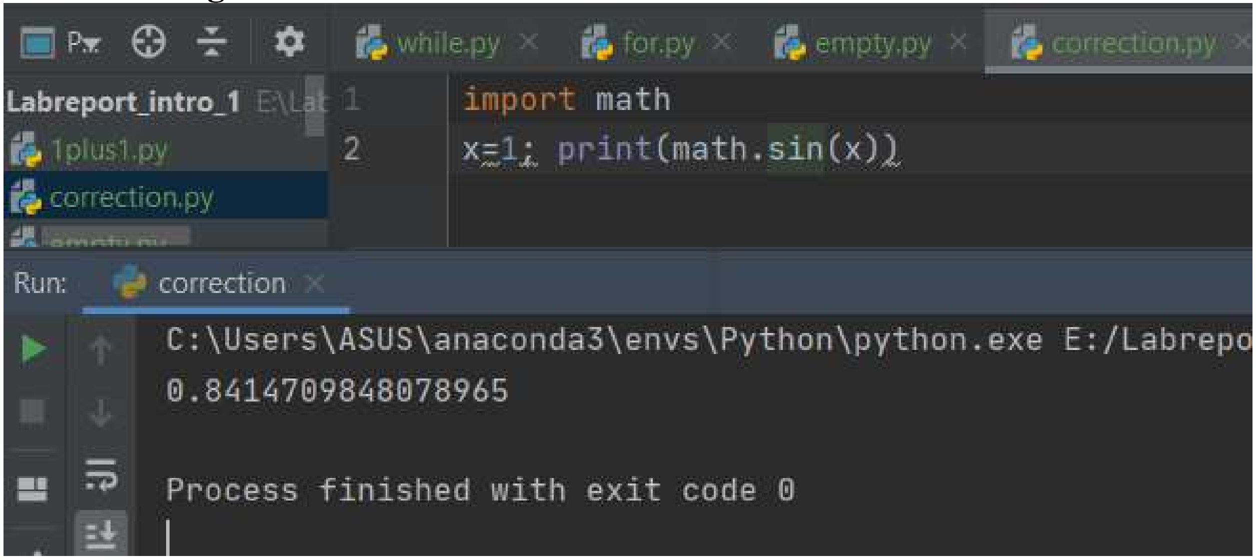
```
import empty
print(__name__)
print(empty.__name__)
```

```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labreport_
__main__
empty

Process finished with exit code 0
```

So the differences between the main module and the empty module is that the main module is default module where the empty module is user define module . There is no need to import the main module where empty module have to import.

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 : Here we need to import the math module to use sin function. The correction is given below :

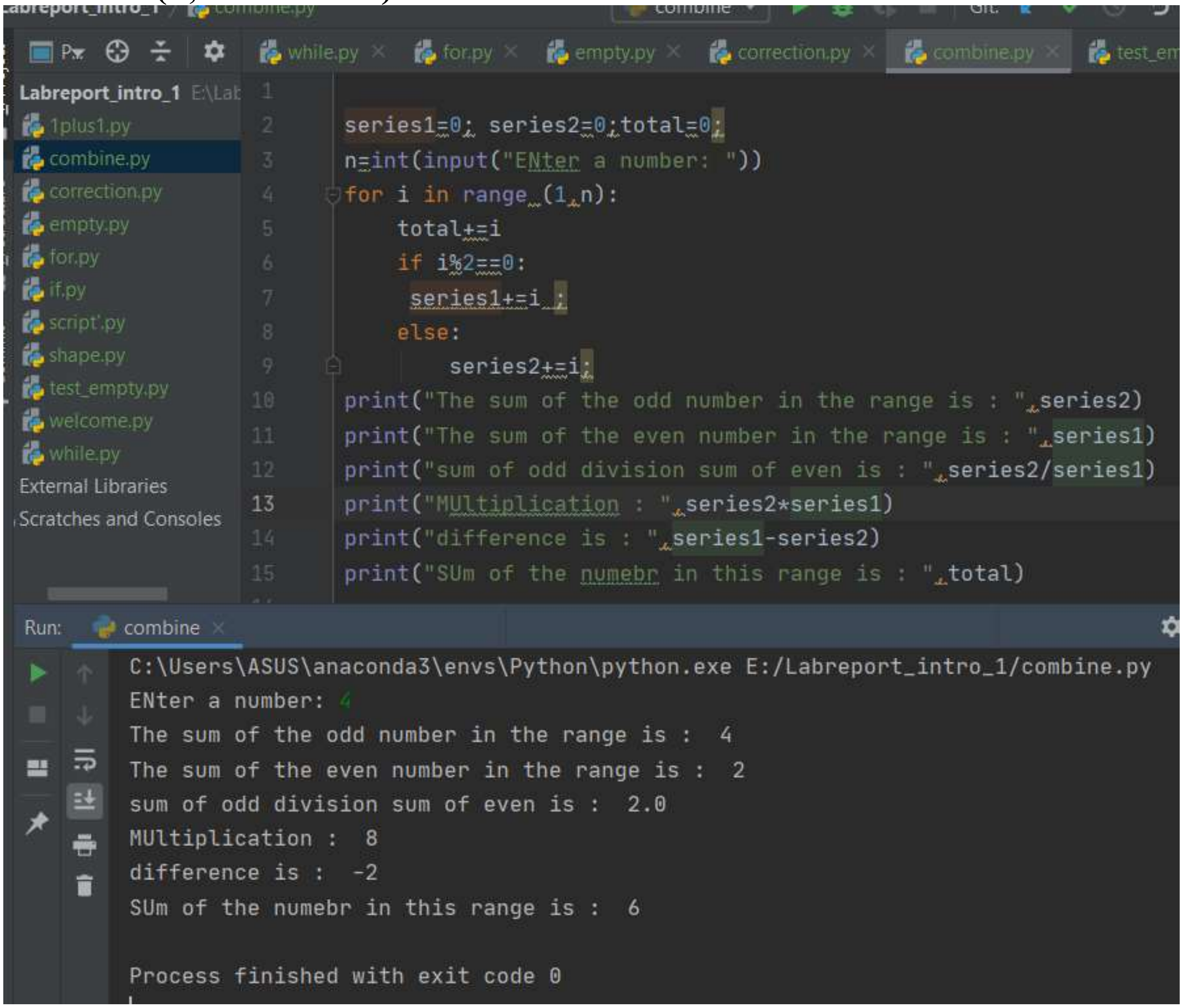


```
import math
x=1; print(math.sin(x))
```

```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labrepo
0.8414709848078965

Process finished with exit code 0
```

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



```python

series1=0; series2=0;total=0;
n=int(input("ENter a number: "))
for i in range (1,n):
    total+=i
    if i%2==0:
     series1+=i ;
    else:
        series2+=i;
print("The sum of the odd number in the range is : ",series2)
print("The sum of the even number in the range is : ",series1)
print("sum of odd division sum of even is : ",series2/series1)
print("MUltiplication : ",series2*series1)
print("difference is : ",series1-series2)
print("SUm of the numebr in this range is : ",total)
```

```
C:\Users\ASUS\anaconda3\envs\Python\python.exe E:/Labreport_intro_1/combine.py
ENter a number: 4
The sum of the odd number in the range is :  4
The sum of the even number in the range is :  2
sum of odd division sum of even is :  2.0
MUltiplication :  8
difference is :  -2
SUm of the numebr in this range is :  6

Process finished with exit code 0
```

Conclusion : I have completed all this program in my laptop taking help from the slide given by my class teacher. As I have introduced with the python few months ago , it is not so hard to do this basic introduction lab report . Here I have additionally learned the python module which was not clear to me at that time .