**Lab report no: 01**

**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:**

**Definition of python:** 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:** The main features of Python are:

* **Simple**: Python is a simple and minimalistic language. This pseudo-code nature of Python is one of its greatest strengths.
* **Easy to Learn:** Python is extremely easy to get started. Python has an extraordinarily simple syntax.
* **Free and Open Source:** Python is an example of FLOSS (Free/Libré and Open Source Software). In simple terms, i can freely distribute copies of this software, read it's source code, make changes to it, use pieces of it in new free programs, and i know that can do these things. FLOSS is based on the concept of a community which shares knowledge.
* **High-level Language:** When i write programs in Python, I never need to bother about the low-level details such as managing the memory used by your program, etc.
* **Portable:** Due to its open-source nature, Python has been ported (i.e. changed to make it work on) to many platforms. All Python programs can work on any of these platforms without requiring any changes.
* **Multi-Plarform:** Python can be used on Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and even Pocket PC.
* **Interpreted:** Python does not need compilation to binary .I just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called byte codes and then translates this into the native language of my computer and then runs it.
* **Object Oriented:** Python supports procedure-oriented programming as well as object oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object oriented languages, the program is built around objects which combine data and functionality.

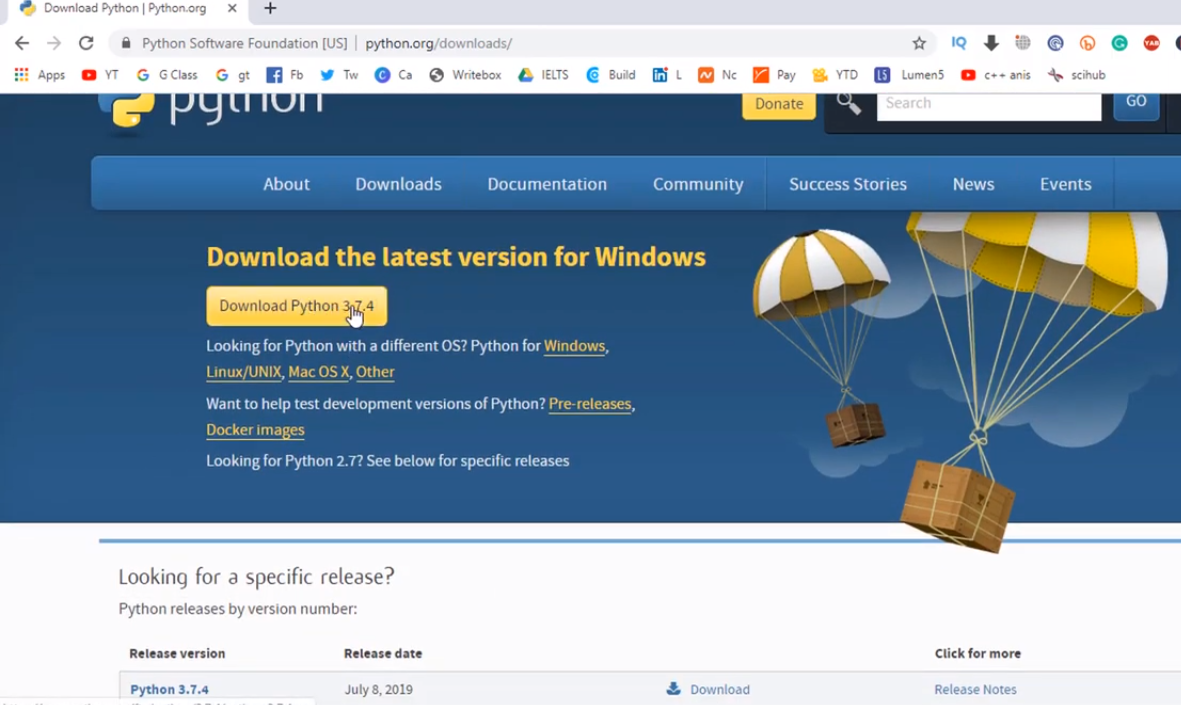
**Extensive Libraries:** The Python Standard Library is huge indeed. It can help our do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed.

**Methodology:** Setup of Python Environment

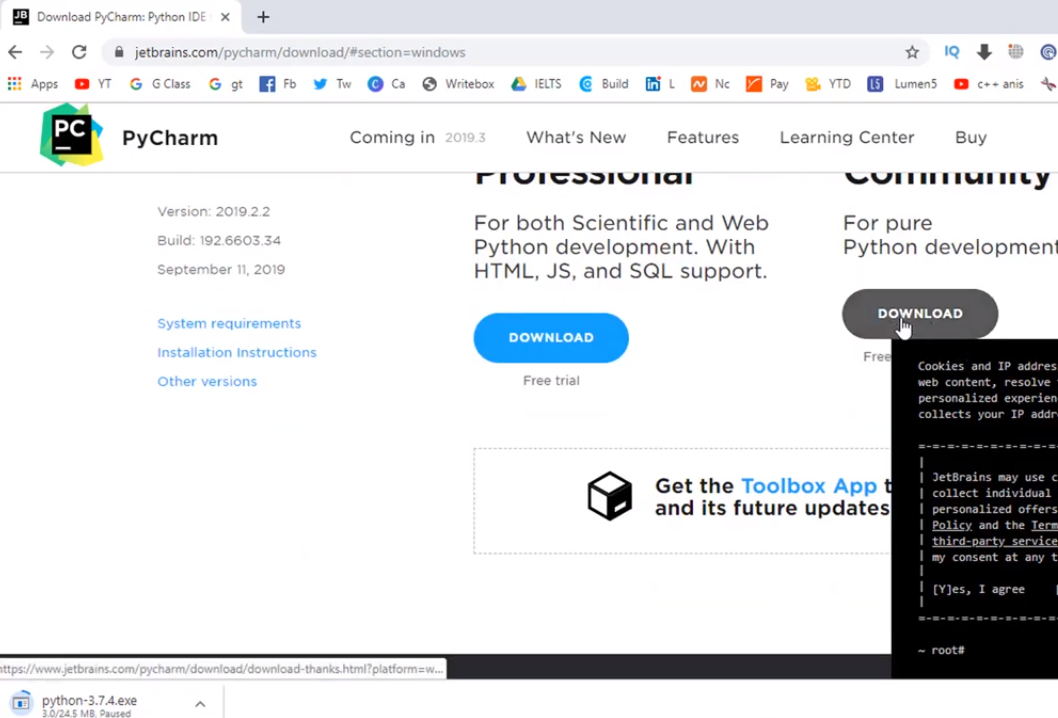
**First install**: 1.Python

2. pyCharm

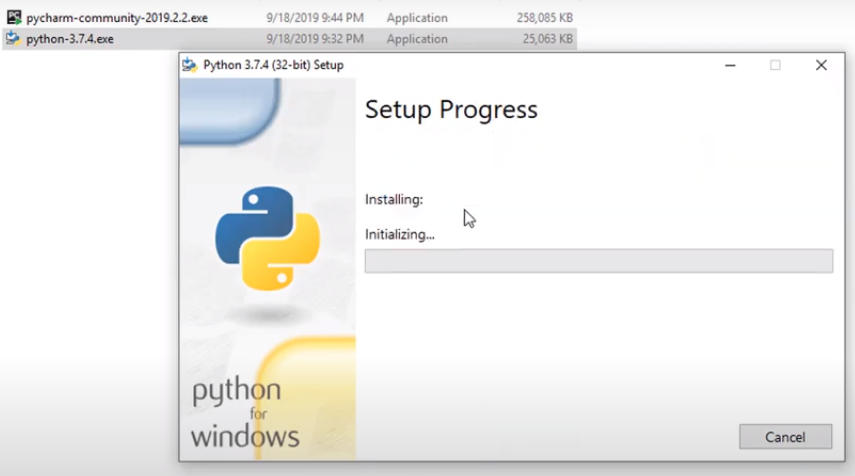
**Step**-1**:** Download the latest version python 3.7.4



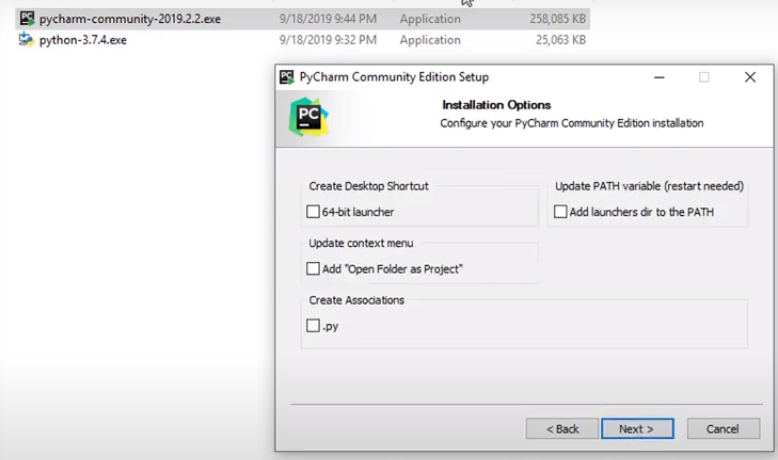
**Step-2:** Download pyCharm community version.



**Step-3:** python 3.7.4 setup process



**Step-4:** pyCharm setup and select the option (64-bit... and .py)



**4. Exercises:**

**Exercise 4.1.2:** **Write a Hello World program**

Ans:

print('hello world')

Output:



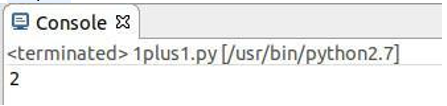
**Exercise 4.1.3: Compute 1+1**

**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

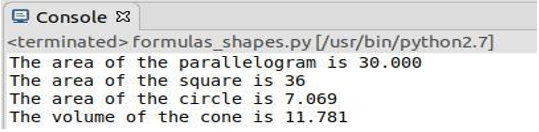
print ('The area of the square is %g' % area\_square)

area\_circle = 3.1416\*r\*\*2 print ('The area of the circle is %.3f' % area\_circle)

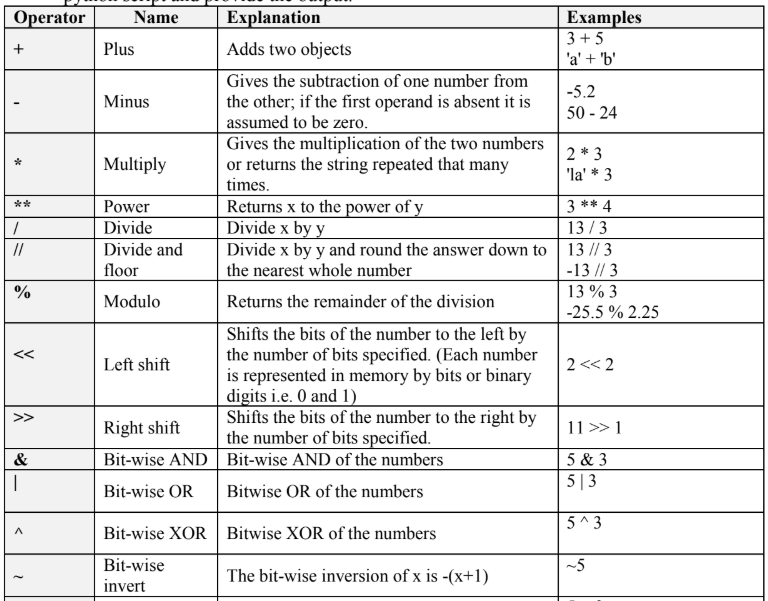
volume\_cone = 1.0/3\*3.1416\*r\*\*2\*h

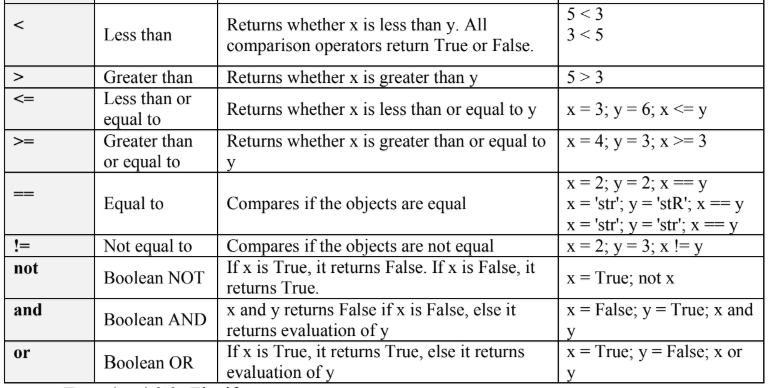
print ('The volume of the cone is %.3f' % volume\_cone)

**Output:**



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





**Ans:**

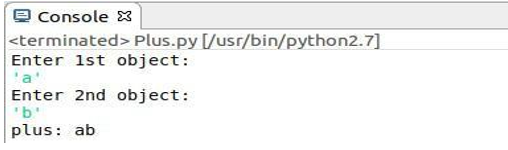
**plus (+) operator:**

a= input('Enter 1st object:\n');

b= input('Enter 2nd object:\n');

plus = a+b

print 'plus:',plus



**Minus (-) operator:**

a=input('Enter 1st object:\n');

b=input('Enter 2nd object:\n');

minus = a-b

print 'minus:’, minus



**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



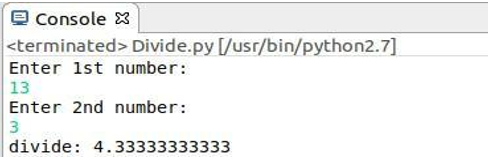
**Divide (/) operator:**

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



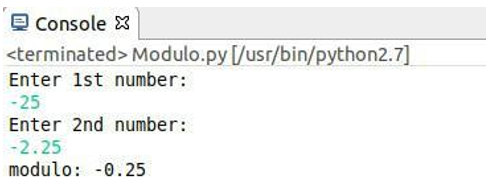
**Modulo (%) operator:**

a=input('Enter 1st number:\n')

b=input('Enter 2nd number:\n')

modulo=a%b

print 'modulo:', modul



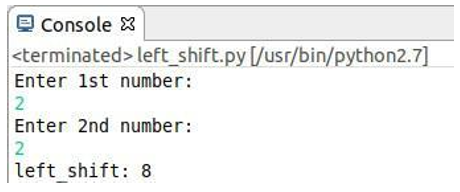
**Left shift (<<) operator:**

**a=input('Enter 1st number:\n')**

**b=input('Enter 2nd number:\n')**

**left\_shift=a<<b**

**print 'left\_shift:', left\_shift**



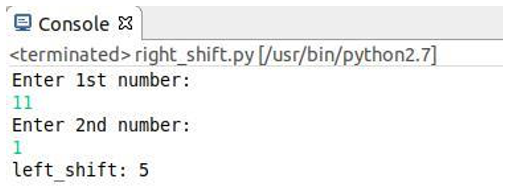
**Right shift (>>) operator:**

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



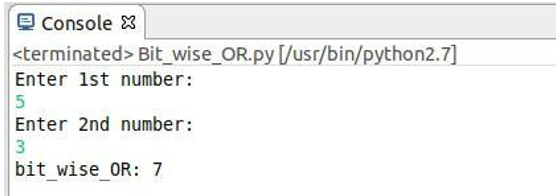
**Bit-wise OR (|) operator:**

a=input('Enter 1st number:\n')

b=input('Enter 2nd number:\n')

bit\_wise\_OR= a|b

print 'bit\_wise\_OR:',bit\_wise\_OR



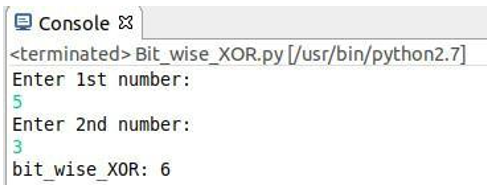
**Bit-wise XOR (^) operator:**

a=input('Enter 1st number:\n')

b=input('Enter 2nd number:\n')

bit\_wise\_XOR=a^b

print 'bit\_wise\_XOR:',bit\_wise\_XO



**Less than (<)operator:**

a=input('Enter 1st number:\n')

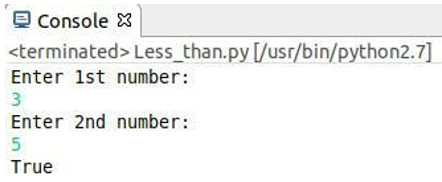
b=input('Enter 2nd number:\n')

if a<b:

print True

else:

print False



**Greater than(>) operator:**

a=input('Enter 1st number:\n')

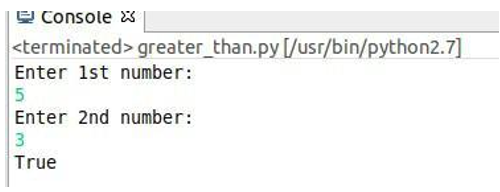
b=input('Enter 2nd number:\n')

if a>b:

print True

else:

print False



**Less than or equal to(<=) operator:**

a=input('Enter 1st number:\n')

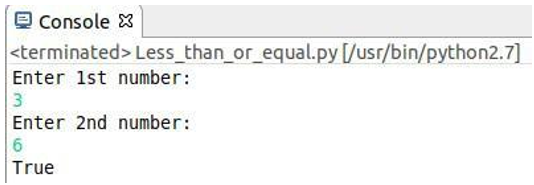
b=input('Enter 2nd number:\n')

if a<=b:

print True

else:

print False



**Equal to (==) operator:**

a=input('Enter 1st number:\n')

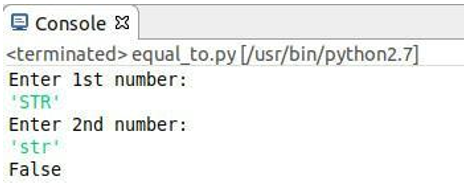
b=input('Enter 2nd number:\n')

if a==b:

print True

else:

print False



**Not equal to(!=) operator:**

a=input('Enter 1st number:\n')

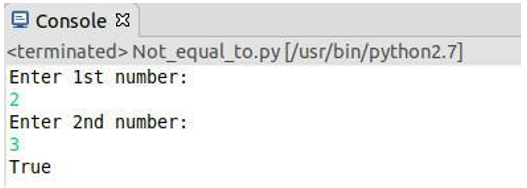
b=input('Enter 2nd number:\n')

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



**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')

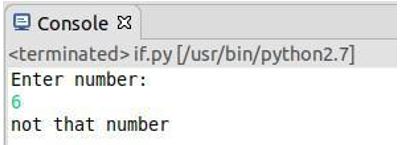
b=5

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

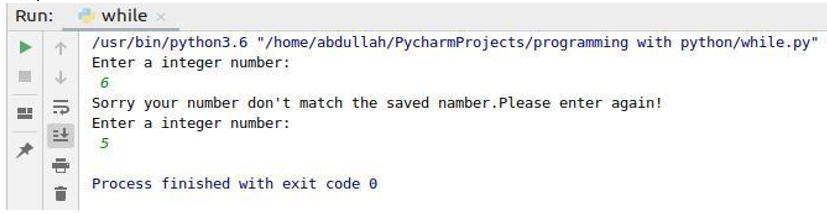
number=int(input('Enter a integer number:\n '))

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:**



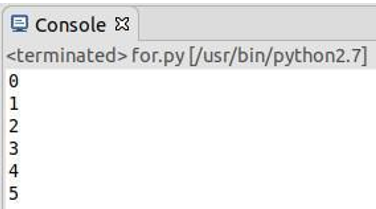
**Exercise 4.2.4: The for Statement**

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)



**Correct code:**

import math as m

x=1

print("sin (%g) = %g"%(x, m.sin(x)))



**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;

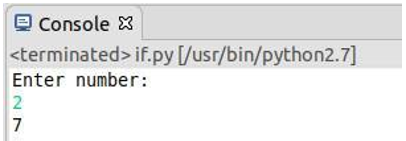
if a>b:

print a-b

else:

print a+b

**Output:**



**Discussion:**

I learned many things from this lab. This lab helps me to understand the basic of python programming. I also know how to download the python and python IDE(pyCharm) and how to setup the python environment. I learn how python programming works, structure and many things. I also learn how to run a python program. Also, learn about variables, operators, keywords in python programming. This was an interesting lab, I can be able to run successfully all the above program as screenshot given above