Mawlana Bhashani Science and Technology University

Lab-Report

Report No:01

Course code: ICT-3202

Course title: Computer Network

Date of Performance:06-01-21

Date of Submission:07-01-21

Submitted by

Name: Sabikun Nahar Piya

ID:IT-18020

3rd year 2nd semester

Session: 2016-2017

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Lab-1:Introduction to Python

1.Objectives:

The objective of the lab-1 is to:

1)Set up python environment for programming

2)Learn the basics of python

3)Create and run basic examples using python

2.Theory:

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming.In Python, we don't need to declare the type of variable because it is a dynamically typed language.
For example, x = 10.Here, x can be anything such as String, int, etc.

Features of Python: The main features of python are:

1.Simple: Python is a simple and minimalistic language. This pseudo-code nature of Python is one of its greatest strengths.

2. Easy to Learn: Python is extremely easy to get started with. Python has an extraordinarily simple syntax.

3. Free and Open Source: Python is an example of FLOSS (Free/Libré and Open Source Software). In simple terms, you 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 that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge.

4. High-level Language: When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

5. Portable: Due to its open-source nature, Python has been ported (i.e. changed to make it work on) to many platforms. All your Python programs can work on any of

these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

6. 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 PocketPC.

7. Interpreted: Python does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an Page | 5 SDN-Labs intermediate form called byte codes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

8. 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 objectoriented languages, the program is built around objects which combine data and functionality.

9. Extensible: If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your programing C or C++ and then use them from your Python program.

10. Embeddable: You can embed Python within your C/C++ programs to give 'scripting' capabilities for your program's users.

11. Extensive Libraries: The Python Standard Library is huge indeed. It can help you 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), Tk, and other system-dependent stuff. Remember, all this is always available wherever Python is installed.
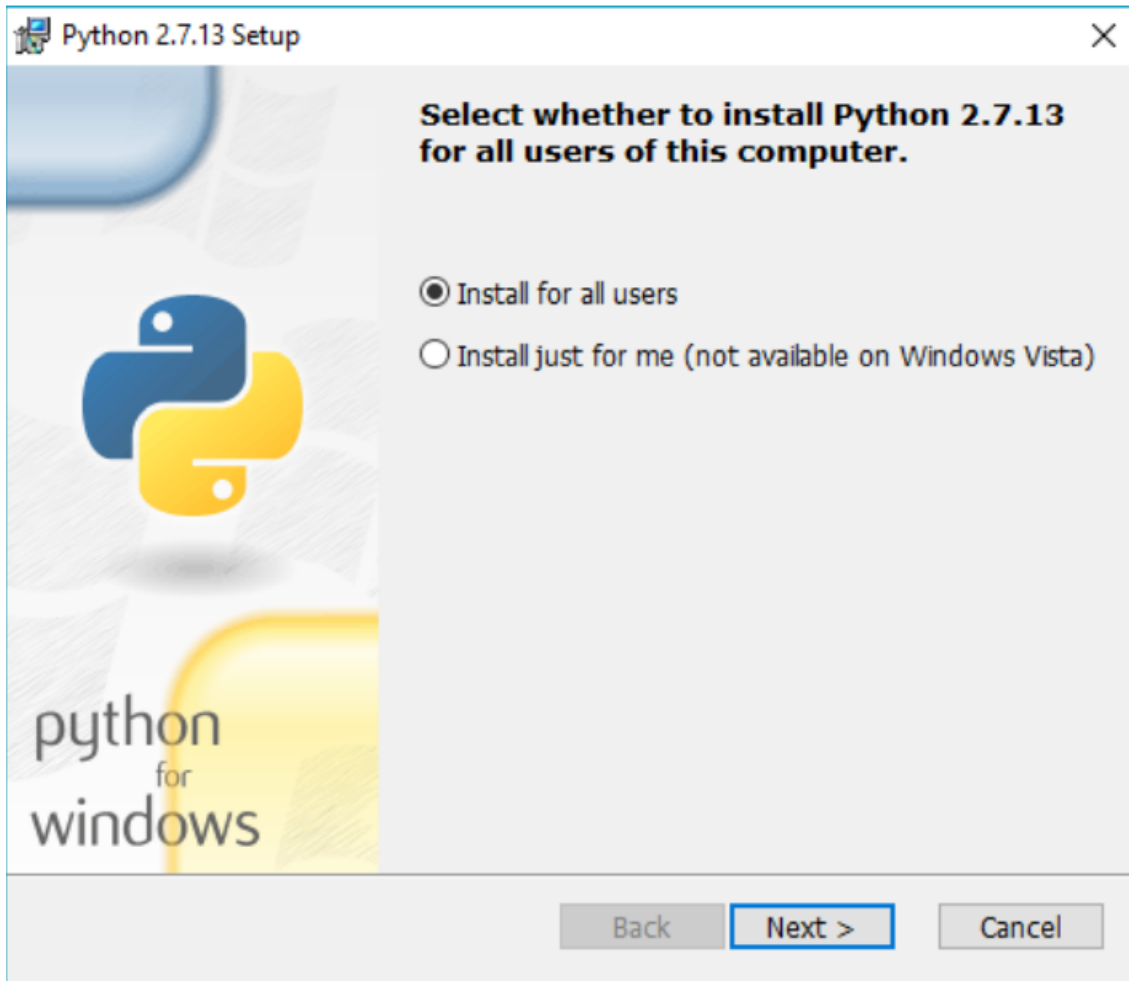
3.Methodology:

**Section-3.1:**Set up of python environment

Step-1:Installing Python

1. Go to Python's downloads page. https://www.python.org/downloads/

2. From the page, choose to download the second edition as illustrated in the figure below.

3. Run the installation wizard that you have just downloaded. At this screen, click "Next"

4. In this screen, you can specify the location where Python to be installed. Leave as default and click on Next.
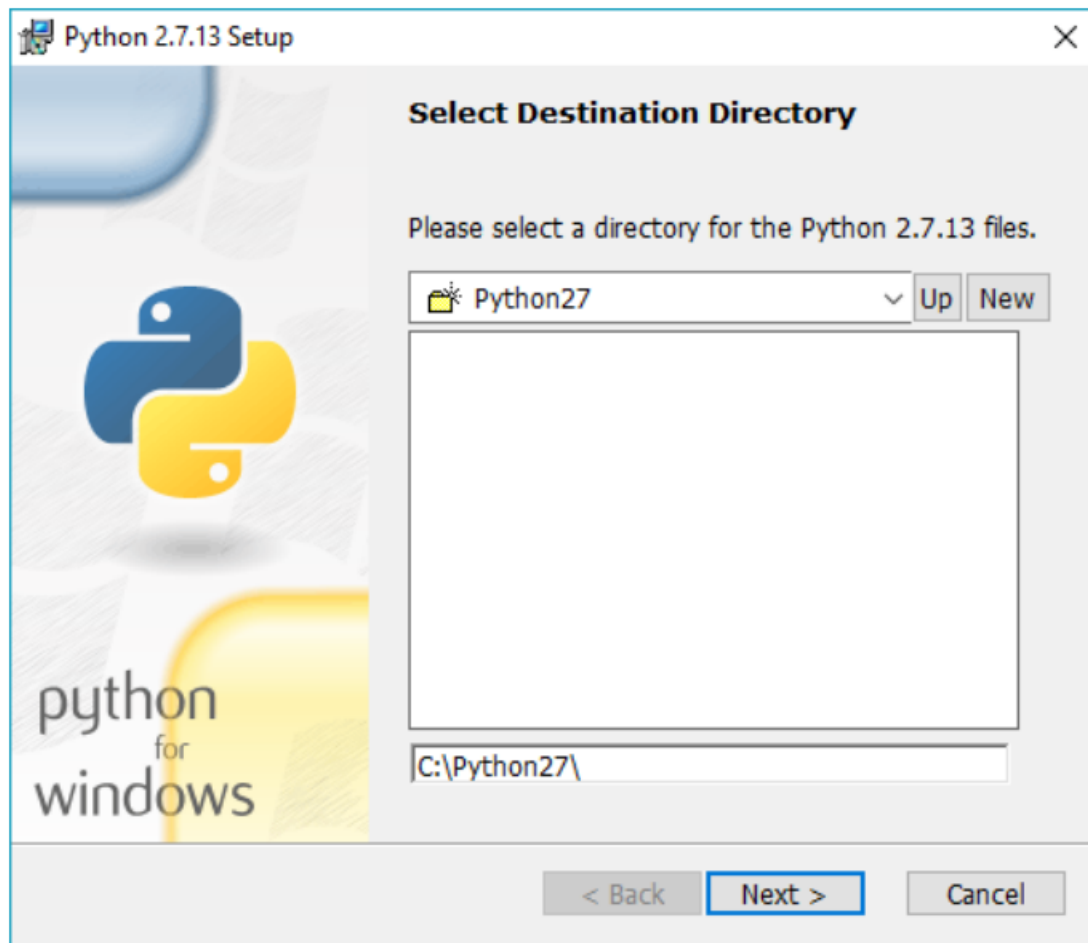
5. Click on "Add python.exe to Path" and choose "Will be installed on local hard drive", then click Next
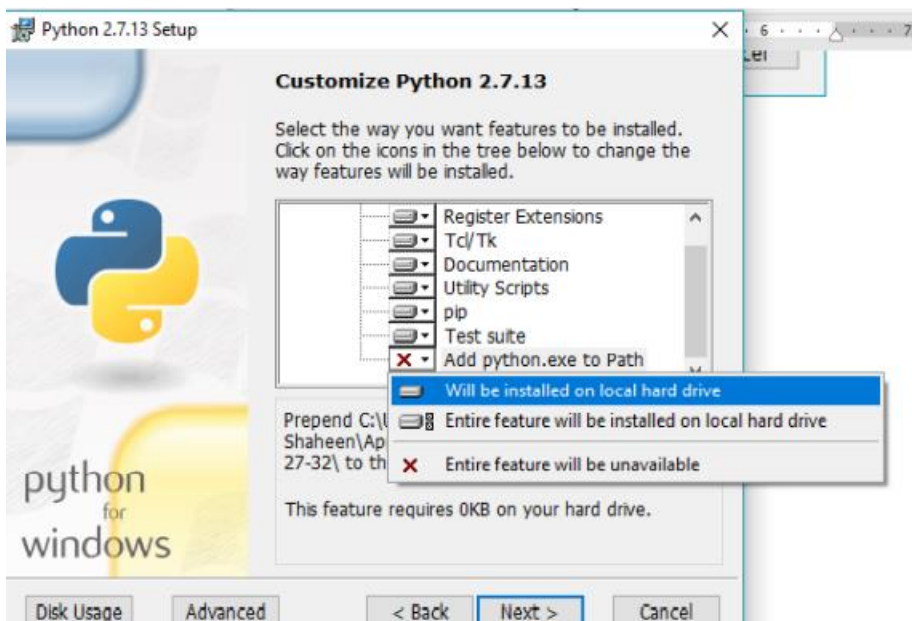
6. The installation process will start. At the end, you'll have Python installed on your machine and ready to be used.

Step-2:Check Python Version

To check Python version installed on your machine follow the following steps.

1. To run the command prompt, press Win + R at the same time, then type cmd in the wizard. Or you can search for cmd from the start menu.

2. In the command prompt print the following command to test python's version.

Python --version

3. Press Enter. The following result will be displayed on the screen. Python 2.7.13

Step-3:Installing Pycharm

As may you have noticed, that process is tedious and cumbersome. And we need a more productive model than that. So, developers around the world use those things called Integrated Development Environments (IDE) that allows us to type in our code, compile, debug and format it, everything in one place.

PyCharm Community Edition is the free version of PyCharm, a premier IDE for Python. In this lab, we are going to use it.

The following steps show you how to install it:

1. Go to the download page and download the latest version of PyCharm Community Edition. https://www.jetbrains.com/pycharm/download/

2. After download is complete, installation process is easy and straightforward. Do it yourself.

• As a feature of being a student �, you can get PyCharm Professional Edition, and all other JetBrains products, for free as long as you are a student. Go to this link
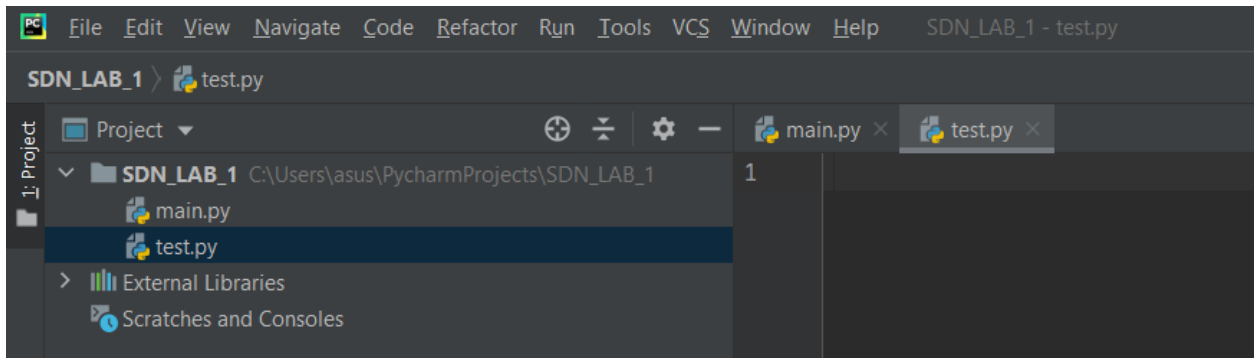
https://www.jetbrains.com/student/

4.Exercises:

**Section-4.1:**Basics of python and programming

Exercise-4.1.1: Create a python project,click in **File->New Project.**Provide a name for the project(SDN_LAB_1 for this lab) as shown below:
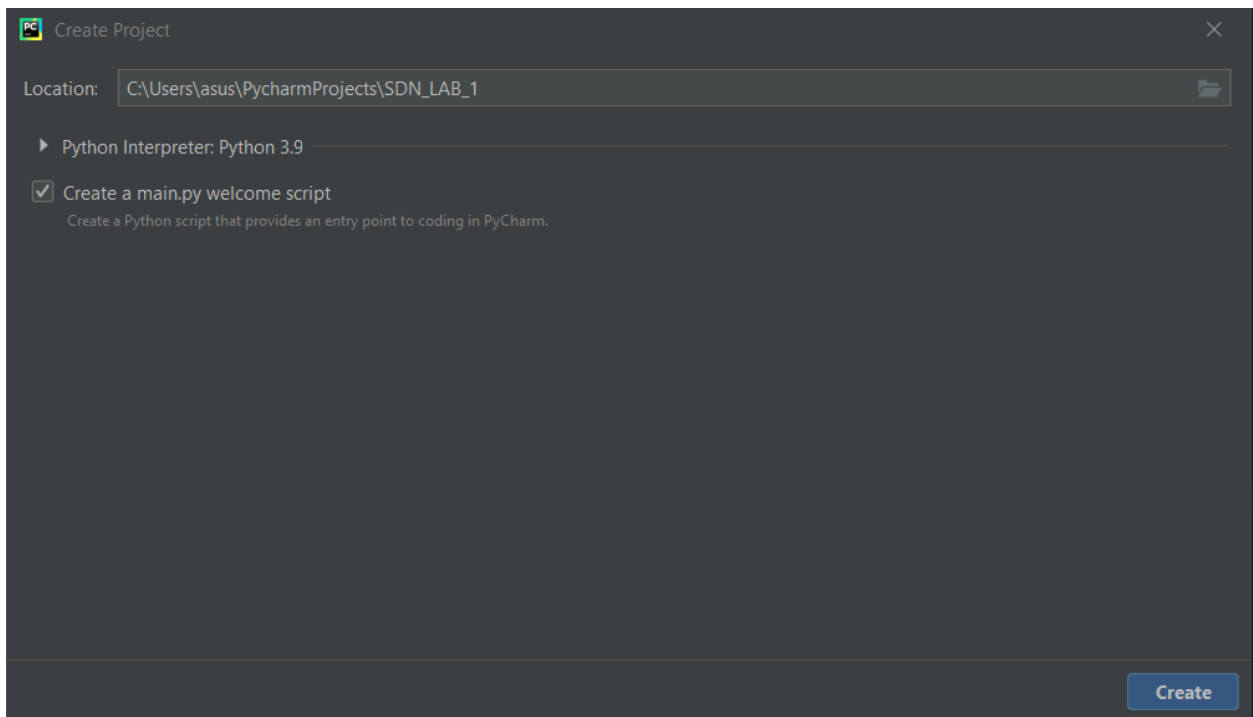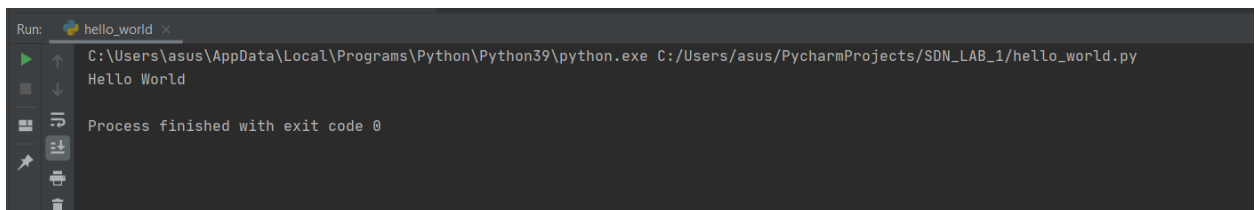
Right click on project name, choose **new -> Python File** and write the name you want for that file. You will notice a new python file is added.
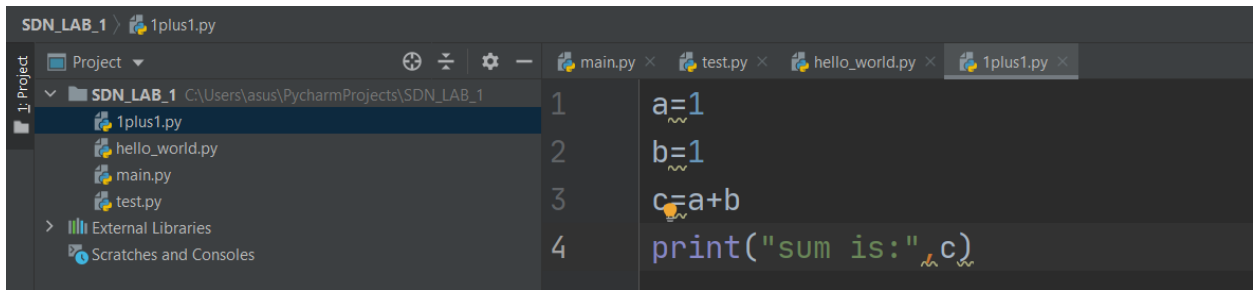
Exercise 4.1.2: Write a Hello World program Almost all books about programming languages start with a very simple program that prints the text Hello, World! to the screen. Make such a program in Python. (save as hello_world.py).
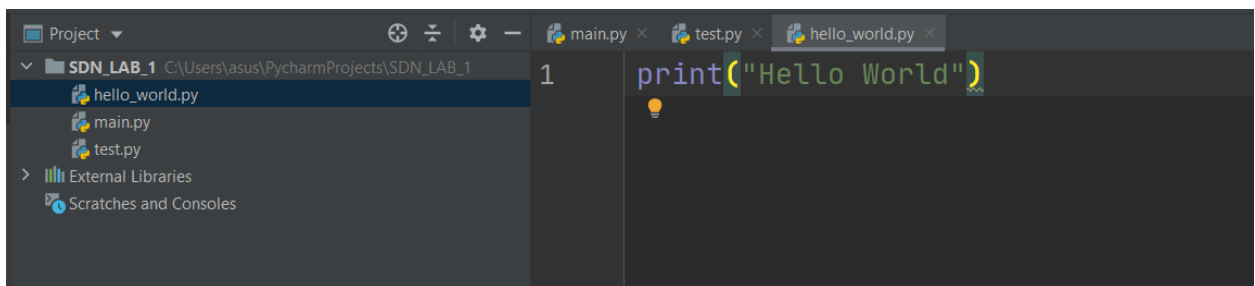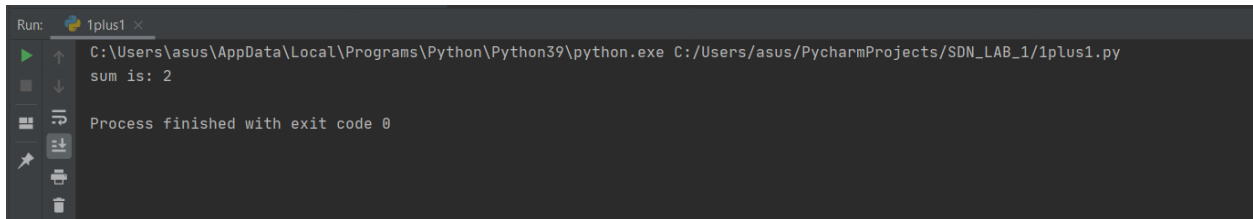

Output:

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).

```
a=1
b=1
c=a+b
print("sum is:",c)
```
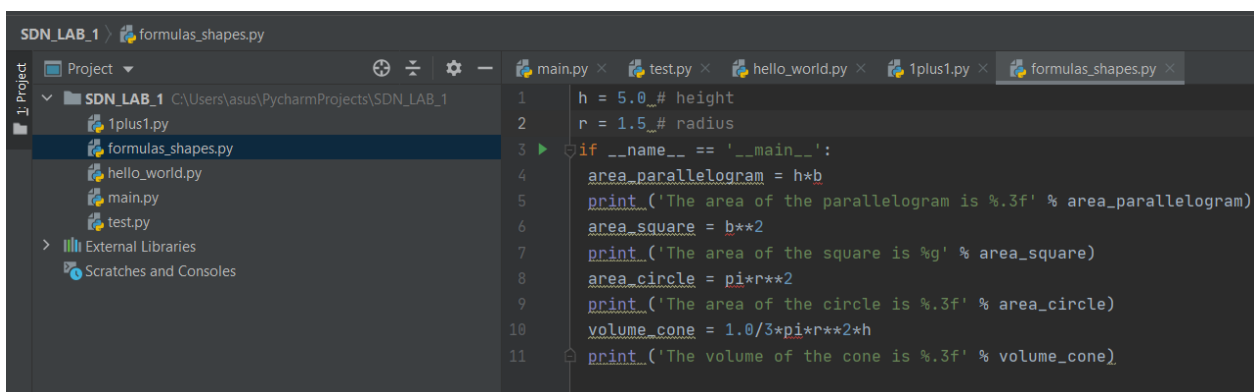
Output:

```
Run:    1plus1
   C:\Users\asus\AppData\Local\Programs\Python\Python39\python.exe C:/Users/asus/PycharmProjects/SDN_LAB_1/1plus1.py
   sum is: 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).

```python
h = 5.0 # height
r = 1.5 # radius
if __name__ == '__main__':
    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 = 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)
```

```python
h = 5.0 # height
r = 1.5 # radius
if __name__ == '__main__':
    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 = 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)
```

```
Run:    formulas_shapes ×
    C:\Users\asus\AppData\Local\Programs\Python\Python39\python.exe C:/Users/asus/PycharmProjects/SDN_LAB_1/formulas_shapes.py
    Traceback (most recent call last):
      File "C:\Users\asus\PycharmProjects\SDN_LAB_1\formulas_shapes.py", line 4, in <module>
        area_parallelogram = h*b
    NameError: name 'b' is not defined

    Process finished with exit code 1
```

Note: Sir,The program does not work because the variables: b and pi are not defined.So,I define these variables and the program works correctly as shown below:


```
SDN_LAB_1 >  formulas_shapes.py
 Project ▼                                    main.py ×    test.py ×    hello_world.py ×    1plus1.py ×    formulas_shapes.py ×
 SDN_LAB_1 C:\Users\asus\PycharmProjects\SDN_LAB_1    1    h = 5.0 # height
    1plus1.py                                         2    r = 1.5 # radius
    formulas_shapes.py                                3    b = 7.0 # base
    hello_world.py                                    4    pi = 3.1416
    main.py                                           5 ▶  if __name__ == '__main__':
    test.py                                           6        area_parallelogram = h*b
 > External Libraries                                 7        print ('The area of the parallelogram is %.3f' % area_parallelogram)
    Scratches and Consoles                            8        area_square = b**2
                                                      9        print ('The area of the square is %g' % area_square)
                                                     10        area_circle = pi*r**2
                                                     11        print ('The area of the circle is %.3f' % area_circle)
                                                     12        volume_cone = 1.0/3*pi*r**2*h
                                                     13        print ('The volume of the cone is %.3f' % volume_cone)
```

Output:


```
Run:    formulas_shapes ×
    C:\Users\asus\AppData\Local\Programs\Python\Python39\python.exe C:/Users/asus/PycharmProjects/SDN_LAB_1/formulas_shapes.py
    The area of the parallelogram is 35.000
    The area of the square is 49
    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

```
if.py ×
1    a=10
2    b=5
3    print(a+b)
4    print(a-b)
5    print(a*b)
6    print(a/b)
7    print(a%b)
8    print(a//b)
9    print(a**b)
10   print(a&b)
11   print(a|b)
12   print(a^b)
13   print(~b)
14   print(a>b)
15   print(a<b)
16   print(a<=b)
17   print(a>=b)
18   print(a==b)
19   print(a!=b)
```
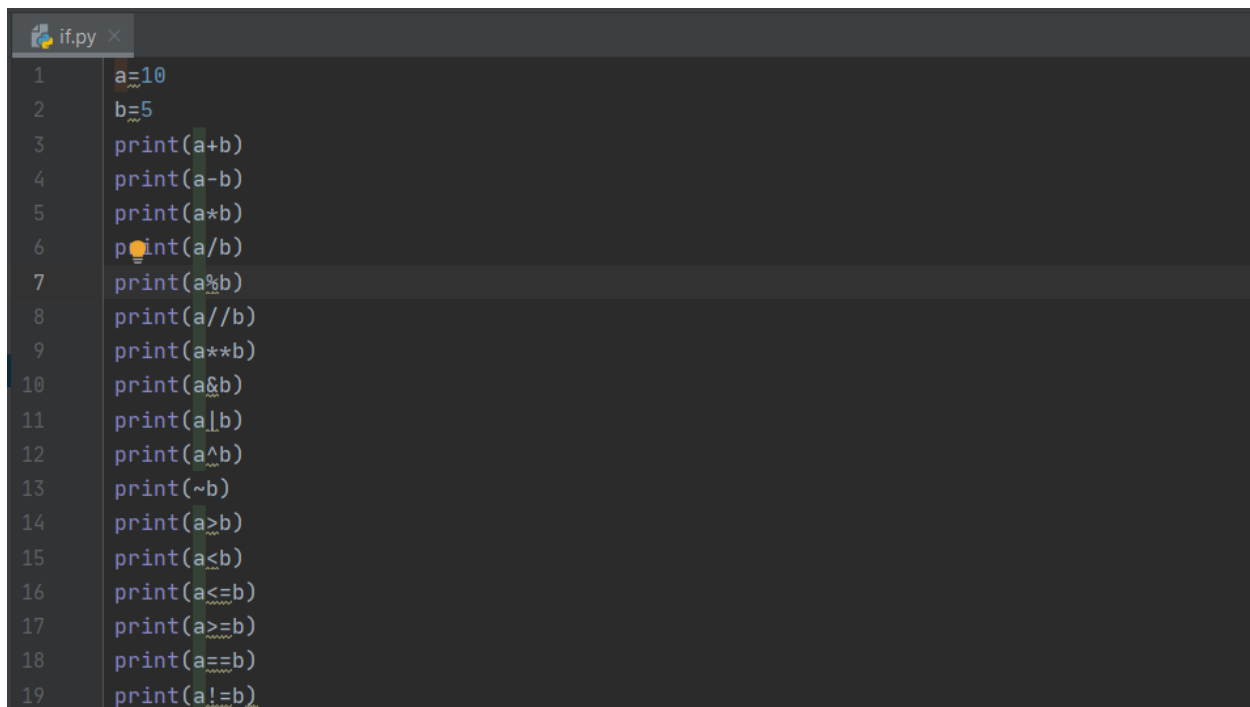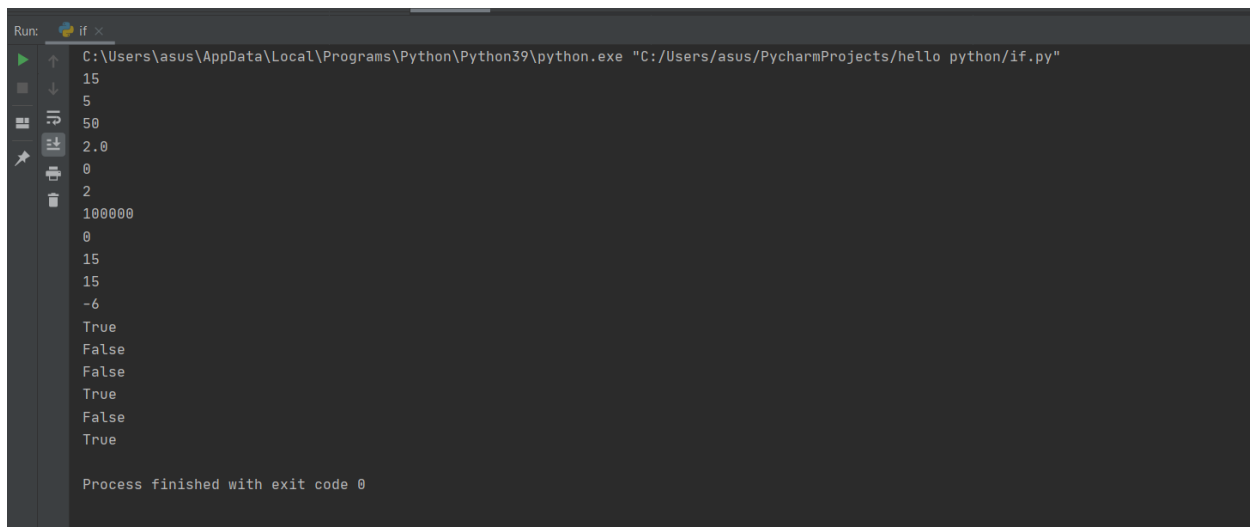
Output:

```
Run:    if ×
    C:\Users\asus\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/asus/PycharmProjects/hello python/if.py"
    15
    5
    50
    2.0
    0
    2
    100000
    0
    15
    15
    -6
    True
    False
    False
    True
    False
    True

    Process finished with exit code 0
```

5.Questions:

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

Answer:

Eclipse: In the context of computing, 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.

The Eclipse platform which provides the foundation for the Eclipse IDE is composed of plug-ins and is designed to be extensible using additional plug-ins. Developed using Java, the Eclipse platform can be used to develop rich client applications, integrated development environments and other tools. Eclipse can be used as an IDE for any programming language for which a plug-in is available.

why we use it for programing on python?

The Java Development Tools (JDT) project provides a plug-in that allows Eclipse to be used as a Java IDE, PyDev is a plugin that allows Eclipse to be used as a Python IDE, C/C++ Development Tools (CDT) is a plug-in that allows Eclipse to be used for developing application using C/C++, the Eclipse Scala plug-in allows Eclipse to be used an IDE to develop Scala applications and PHPeclipse is a plug-in to eclipse that provides complete development tool for PHP.

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

Answer:

There are three main characteristics of python :

1) Easy to Learn and Use

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

2) Expressive Language

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type **print("Hello World")**. It will take only one line to execute, while Java or C takes multiple lines.

3) Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

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

Answer:

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.

Question 5.4: Find error(s) in a program Suppose somebody has written a simple one-line program for computing

```
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?

Answer:

This program will produce an error because sin has not been imported.

To make this program functional do a from math import sin.

```
1    from math import sin
2    x=1; print ('sin(%g)=%g' % (x, sin(x)))
```

```
C:\Users\asus\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/asus/PycharmProjects/hello python/if.py"
sin(1)=0.841471

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)

Answer:

This is a python program to check leap year number.

```python
# Python Program to Check Leap Year using If Statement

year = int(input("Please Enter the Year Number you wish: "))

if (( year%400 == 0)or (( year%4 == 0 ) and ( year%100 != 0))):
    print("%d is a Leap Year" %year)
else:
    print("%d is Not the Leap Year" %year)
```

Output:

```
C:\Users\asus\AppData\Local\Programs\Python\Python39\python.exe "C:/Users/asus/PycharmProjects/hello python/if.py"
Please Enter the Year Number you wish: 2020
2020 is a Leap Year

Process finished with exit code 0
```