

# **MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**

**Santosh,Tangail – 1902**



**Course Title : Computer Networks Lab**

**Lab Report : Introduction to Python**

**Name**

**Lab Report No. : 04**

**Submitted by,**

**Name : ANIKA JAHIN**

**ID: IT-17056**

**Session: 2016-17**

**Dept. of ICT,MBSTU.**

**Submitted to,**

**NAZRUL ISLAM**

**Assistant Professor**

**Dept. of ICT,MBSTU.**

## 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.

## Setup of Python Environment

**STEP 1:** Open Eclipse and setup a correct access to Internet (This is required only in RMIT network). In order to set up Manual Proxy follow the instructions (see also figure 1): a. Go to **Windows > Preferences > General > Network Connections**.

b. Change Active Provider to Manual.

c. Input proxy details, including username/password if required.

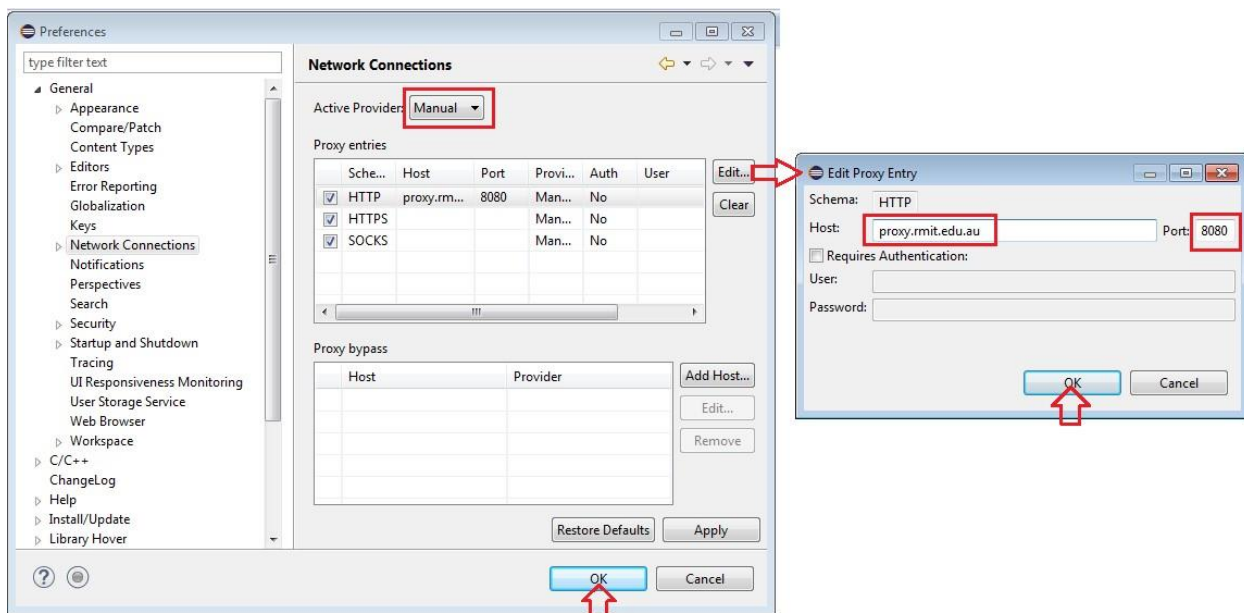
❗ **Host:** proxy.rmit.edu.au

❗ **Port:** 8080

❗ **Username/password:** No required

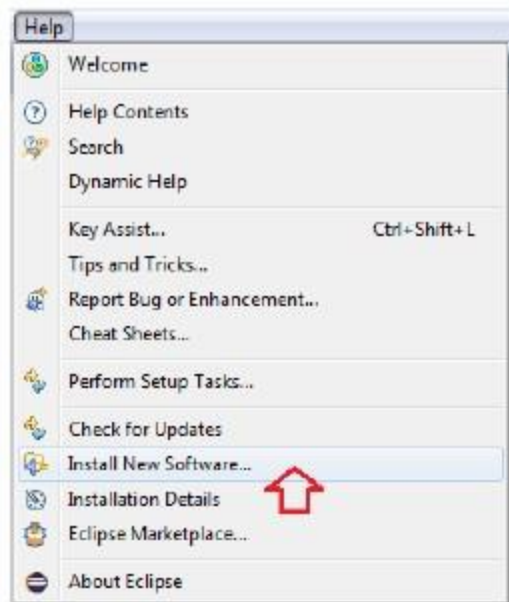
d. Clear SOCKS proxy.

e. Restart Eclipse.



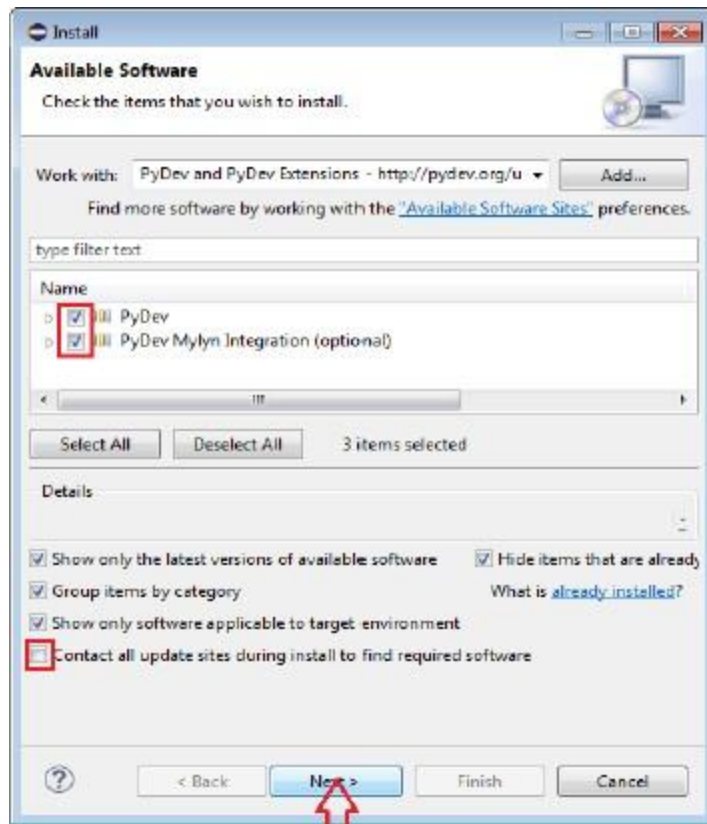
**STEP 2:** Installing python environment using Eclipse Graphical Interface1.

a. To install PyDev and PyDev Extensions using the Eclipse Update Manager, you need to use the **Help > Install New Software...** menu (note that in older versions, this would be the 'Find and Install' menu) as shown in the following figure:

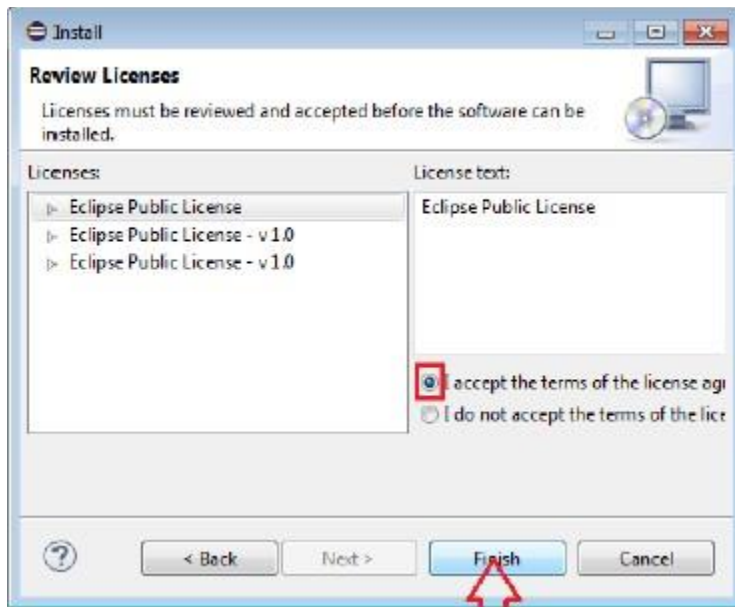


In the next screen, add the update site(s) you want to work with (see the figure below). The available update sites are :

<http://pydev.org/updates>

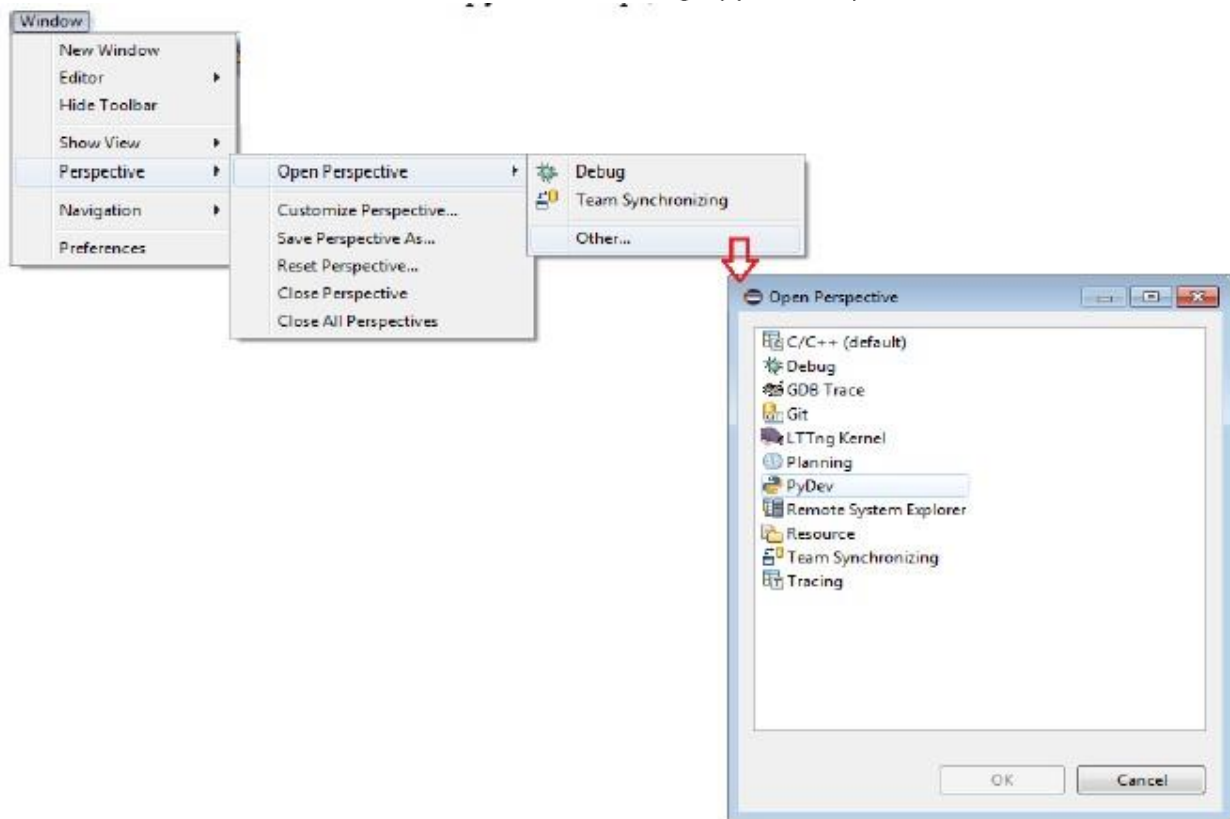


And finally, read the license agreement and if you accept, select the accept radio button and click 'Finish'



**STEP 2: Checking the installation:** You can verify if it is correctly installed going to the menu '**window> preferences**' and checking if there is a **PyDev** item under that (see Figure 7). After that eclipse will display the graphical interface for python perspective, the main components are (see Figure 8):

- ❑ Project space is the section where all your py
- ❑ Project Editor is the section where python scripts can be edited,
- ❑ Console allows the visualization of results father running a python script,



thon projects are visualized,

## Exercises

### Section 4.1: Basics of python and programing

#### Exercise 4.1.1: Create a python project.

Answer:

**PyDev Project**  
Create a new PyDev Project.

Project name:

Project contents:  
☒ Use default

Directory:

Project type  
Choose the project type  
☒ Python ☐ Jython ☐ IronPython

Grammar Version  
Same as interpreter

Interpreter  
Default -- currently: python  
[Click here to configure an interpreter not listed.](#)

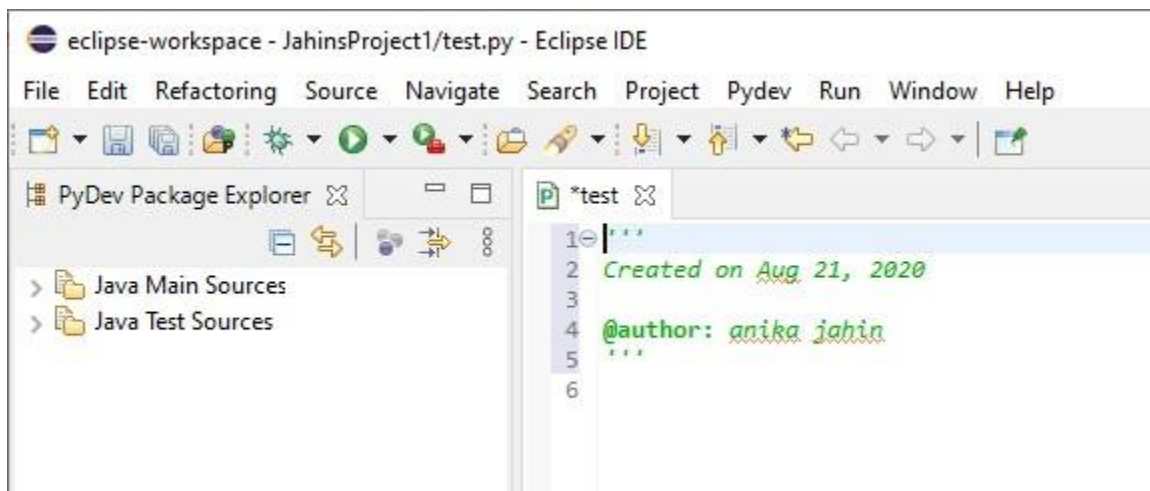
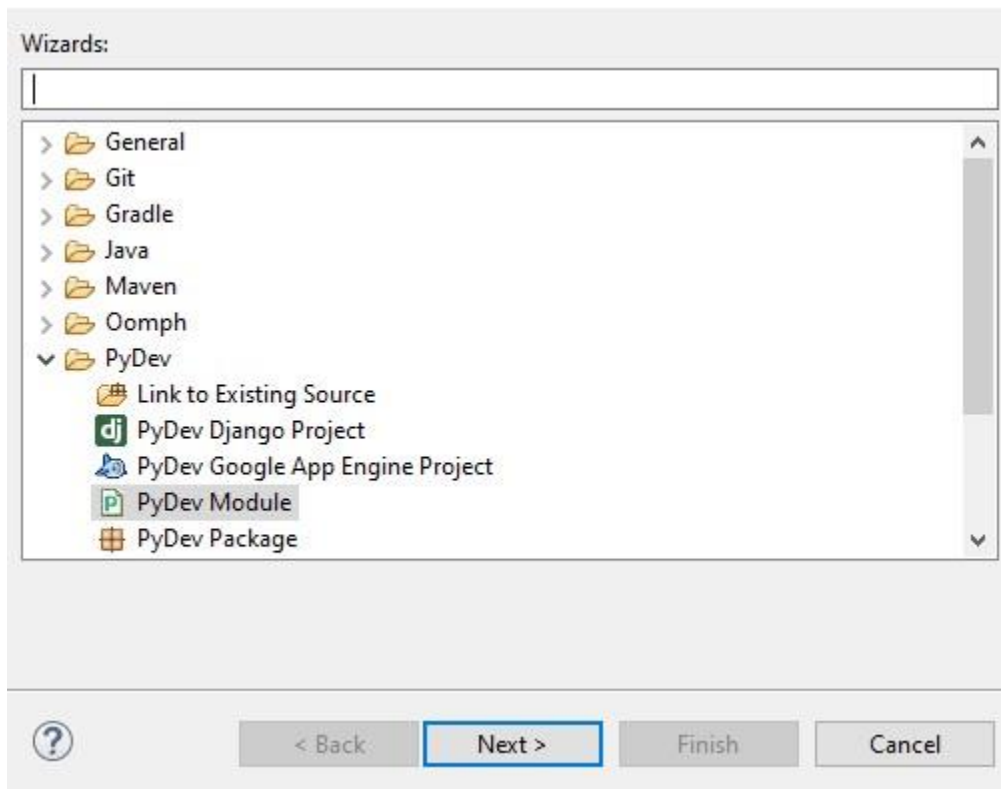
Additional syntax validation: <no additional grammars selected>

☒ Add project directory to the PYTHONPATH  
☐ Create 'src' folder and add it to the PYTHONPATH  
☐ Create links to existing sources (select them on the next page)  
☐ Don't configure PYTHONPATH (to be done manually later on)

Working sets  
☐ Add project to working sets   
Working sets:

Create a python script, click in **File > New > PyDev Module**. Select the folder source name. Then, provide a name for the project (Hello\_world), then select empty module or main module as shown below:





Project created successfully.

Exercise: 4.1.2 : Write a Hello World Program.



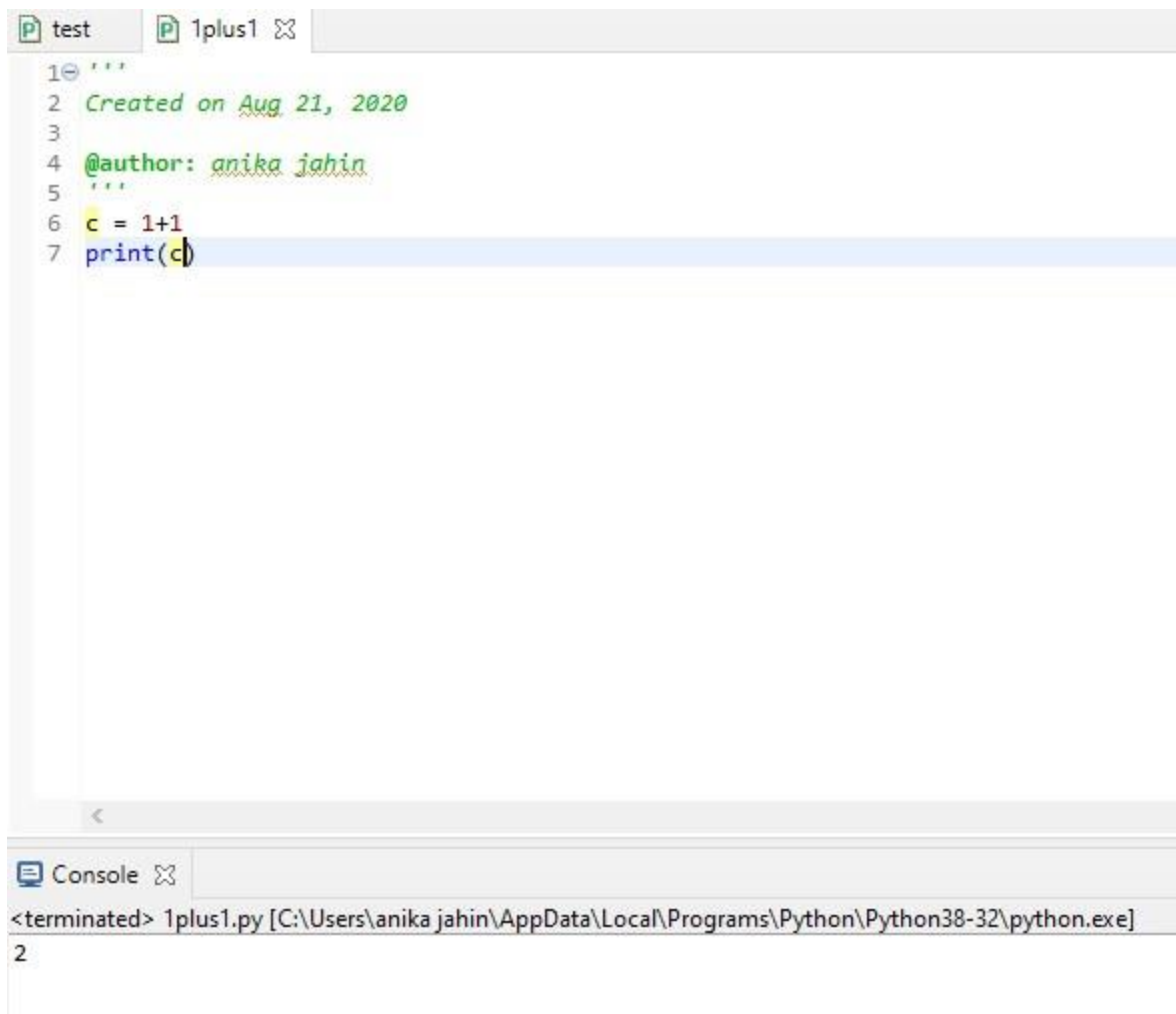
The screenshot shows a Python IDE with a file named 'test.py'. The code in the editor is as follows:

```
1 '''  
2 Created on Aug 21, 2020  
3  
4 @author: anika jahin  
5 '''  
6 print('Hello_World!')
```

Below the editor is a console window. It shows the command prompt output after running the program:

```
<terminated> test.py [C:\Users\anika jahin\AppData\Local\I  
Hello_World!  
|
```

**Exercise 4.1.3:** Compute 1+1



The image shows a Python IDE window with two tabs: 'test' and '1plus1'. The '1plus1' tab is active, displaying a Python script. The script contains a docstring with the creation date and author, followed by a calculation and a print statement. The console window below shows the output of the script.

```
1 '''  
2 Created on Aug 21, 2020  
3  
4 @author: anika jahin  
5 '''  
6 c = 1+1  
7 print(c)
```

Console output:

```
<terminated> 1plus1.py [C:\Users\anika jahin\AppData\Local\Programs\Python\Python38-32\python.exe]  
2
```

**Exercise 4.1.4:** Type in program text

```
test 1plus1 formulas_shapes
```

```
1 '''
2 Created on Aug 21, 2020
3
4 @author: anika jahin
5 '''
6 h = 5.0 # height
7 r = 1.5 # radius
8 pi = 3.1416
9 if __name__ == '__main__':
10     area_parallelogram = h*r
11     print ('The area of the parallelogram is %.3f' % area_parallelogram)
12     area_square = h**2
13     print ('The area of the square is %g' % area_square)
14     area_circle = pi*r**2
15     print ('The area of the circle is %.3f' % area_circle)
16     volume_cone = 1.0/3*pi*r**2*h
17     print ('The volume of the cone is %.3f' % volume_cone)
```

```
Console
```

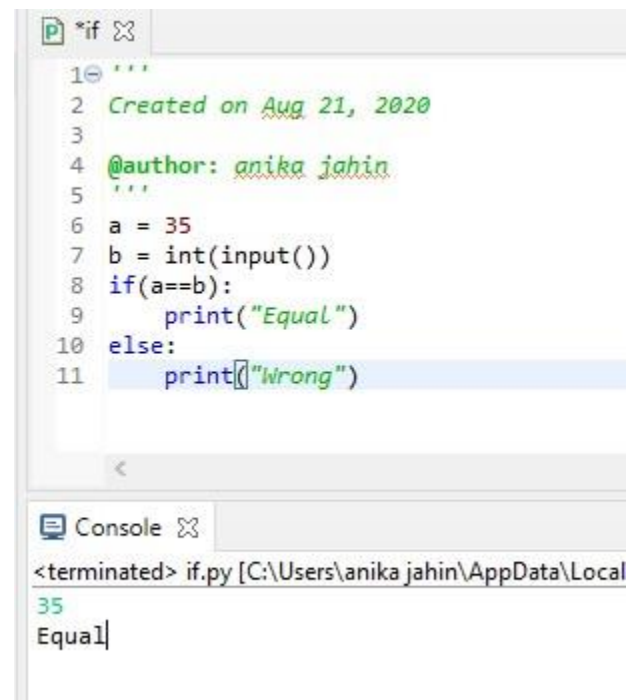
```
<terminated> formulas_shapes.py [C:\Users\anika jahin\AppData\Local\Programs\Python\Python38-32\pythc
The area of the parallelogram is 7.500
The area of the square is 25
The area of the circle is 7.069
The volume of the cone is 11.781
```

**Section 4.1:** Create and run basic example.

```
test 1plus1 formulas_shapes *sample
1 '''
2 Created on Aug 21, 2020
3
4 @author: anika jahin
5 '''
6 a = int(input())
7 b = int(input())
8
9 print(a+b) #addition
10 print(a-b) #minus
11 print(a*b) #multiply
12 print(a**b) #power
13 print(a/b) #divide
14 print(a//b) #floor
15 print(a%b) #modulo
16 print(a<<b) #left shift
17 print(a>>b) #right shift
18 print(a&b) #bitwise and
19 print(a|b) #bitwise or
20 print(a^b) #bitwise xor
21 print(a<b) #less than
22 print(a>b) #greater than
23 print(a<=b)
24 print(a>=b)
25 print(a==b)
26 print(a!=b)
```

```
Console
<terminated> sample.py [C:\Users\
5
6
11
-1
30
15625
0.8333333333333334
0
5
320
0
4
7
3
True
False
True
False
False
True
```

#### Exercise 4.2.2: The if statement:



```
1 '''  
2 Created on Aug 21, 2020  
3  
4 @author: anika jahin  
5 '''  
6 a = 35  
7 b = int(input())  
8 if(a==b):  
9     print("Equal")  
10 else:  
11     print("Wrong")
```

<terminated> if.py [C:\Users\anika jahin\AppData\Local  
35  
Equal|

#### Exercise 4.2.3: The while Statement

```
if while
```

```
1 '''
2 Created on Aug 21, 2020
3
4 @author: anika jahin
5 '''
6 a = 10
7 while(a>1):
8     print(a)
9     a = a-1
```

```
Console
```

```
<terminated> while.py [C:\Users\anika jahin]
10
9
8
7
6
5
4
3
2
```

#### Exercise 4.2.4: The for Statement

```
if while for
```

```
1 '''
2 Created on Aug 21, 2020
3
4 @author: anika jahin
5 '''
6 for x in range(7):
7     print(x)
```

```
Console
```

```
<terminated> for.py [C:\Users\anika jahin]
0
1
2
3
4
5
6
```

**Conclusion:** Python is a language that is remarkably easy to learn, and it can be used as a stepping stone into other programming languages and frameworks. If you're an absolute beginner and this is your first time working with any type of coding language, that's something you definitely want.

Python is widely used, including by a number of big companies like Google, Pinterest, Instagram, Disney, Yahoo!, Nokia, IBM, and many others. The Raspberry Pi – which is a mini computer and DIY lover's dream – relies on Python as its main programming language too. You're probably wondering why either of these things matter, and that's because once you learn Python, you'll never have a shortage of ways to utilize the skill. Not to mention, since a lot of big companies rely on the language, you can make good money as a Python developer.

- 1)** Python can be used to develop prototypes, and quickly because it is so easy to work with and read.
- 2)** Most automation, data mining, and big data platforms rely on Python. This is because it is the ideal language to work with for general purpose tasks.
- 3)** Python allows for a more productive coding environment than massive languages like C# and Java. Experienced coders tend to stay more organized and productive when working with Python, as well.
- 4)** Python is easy to read, even if you're not a skilled programmer. Anyone can begin working with the language, all it takes is a bit of patience and a lot of practice. Plus, this makes it an ideal candidate for use among multi-programmer and large development teams.
- 5)** Python powers Django, a complete and open source web application framework. Frameworks – like Ruby on Rails – can be used to simplify the development process.
- 6)** It has a massive support base thanks to the fact that it is open source and community developed. Millions of like-minded developers work with the language on a daily basis and continue to improve core functionality. The latest version of Python continues to receive enhancements and updates as time progresses. This is a great way to network with other developers.