

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Santosh, Tangail -1902



Lab Report No : 05
Lab Report Name : Introduction to Python
Course Name : Computer Networks Lab
Course Code : ICT - 3208

Submitted by,

Name : Md. Imtyaz Ahmed

ID : IT-17017

Session : 2016-17

Dept. of ICT, MBSTU.

Submitted to,

Nazrul Islam

Assistant Professor

Dept. of ICT, MBSTU

Objectives:

- Setup python environment for programming
- Learn the basics of Python
- Create and run basic examples using Python

Q. What is Python ? What is the main features of Python?

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:

- a) Simple: Python is a simple and minimalistic language. This pseudocode nature of Python is one of its greatest strengths.
- b) Easy to Learn: Python is extremely easy to get started with. Python has an extraordinarily simple syntax.
- c) Free and Open Source: Python is an example of FLOSS (Free/Libre 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.
- d) 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.
- e) 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 systemdependent features.
- f) 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.

Q. How to setup python environment in your computer?

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

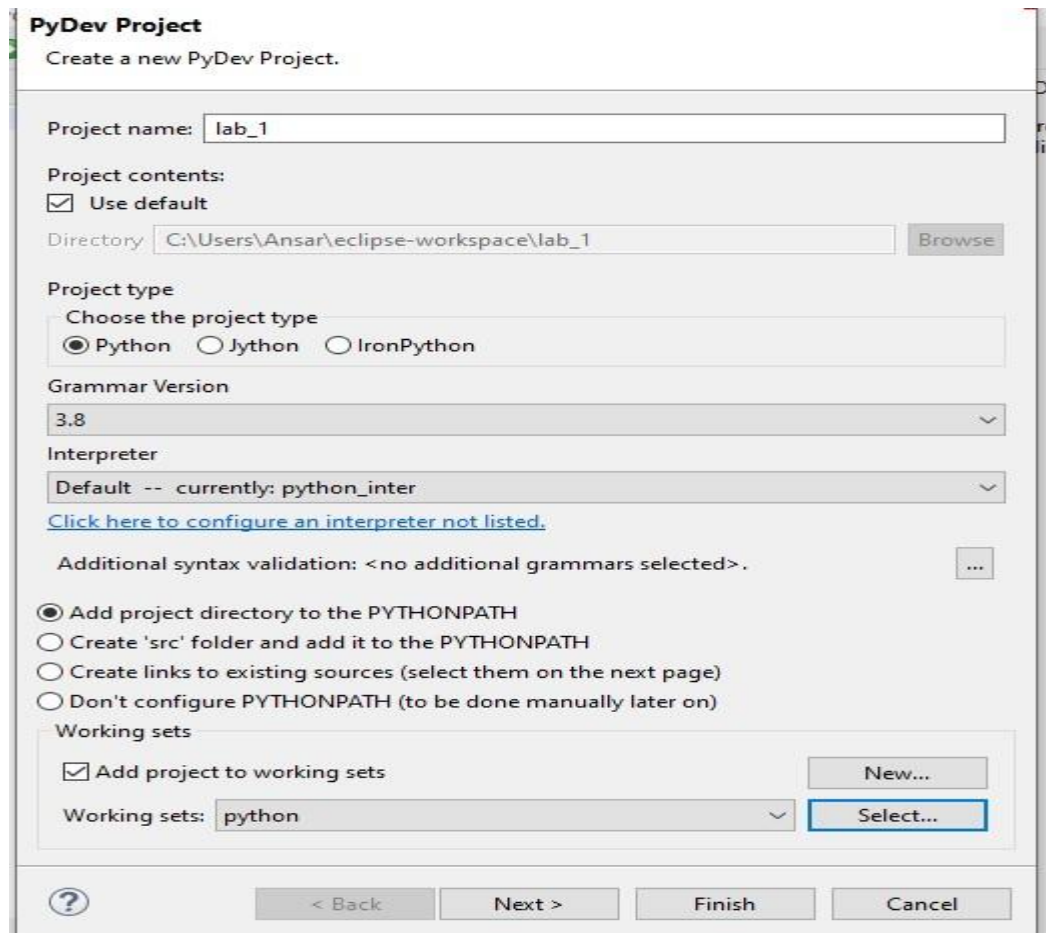
Username/password: No required d. Clear SOCKS proxy.

- e. Restart Eclipse.

STEP 2: Installing python environment using Eclipse Graphical Interface

- 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)
- b. 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> (Last version for your personal PC)  <http://pydev.org/updates4.5> (version for RMIT Labs)
- c. After entering the update sites, select the update site you entered or select "All available sites" and add a filter for PyDev, so that it shows the contents of all the update sites that have PyDev, then select what you want to install and click 'Next'
- d. Then, UNCHECK the 'Contact all update sites during install to find required software' and press 'Next' again to confirm your selection .
- e. And finally, read the license agreement and if you accept, select the accept radio button and click 'Finish'
- f. At that point, Eclipse should automatically download the plugin contents and present you to a dialog asking if you want to restart (to which you should say yes). 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 python projects are visualized, Project Editor is the section where python scripts can be edited, Console allows the visualization of results father running a python script,

Exercise 4.1.1 : Create a python project, click in File > New > PyDev Project. Provide a name for the project (LAB_1 for lab), then select the version of python to be used and select to add the project to working set as shown below:



The image shows the 'PyDev Project' dialog box in Eclipse. The 'Project name' field is set to 'lab_1'. Under 'Project contents', the 'Use default' checkbox is checked, and the 'Directory' is 'C:\Users\Ansar\eclipse-workspace\lab_1'. In the 'Project type' section, 'Python' is selected. The 'Grammar Version' is set to '3.8' and the 'Interpreter' is 'Default -- currently: python_inter'. There is a link to 'Click here to configure an interpreter not listed.' Below this, 'Additional syntax validation' is set to '<no additional grammars selected>'. In the 'Working sets' section, the checkbox 'Add project to working sets' is checked, and the 'Working sets' dropdown is set to 'python'. The 'Select...' button is highlighted with a blue border. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

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
 ▾

Interpreter
 ▾
[Click here to configure an interpreter not listed.](#)

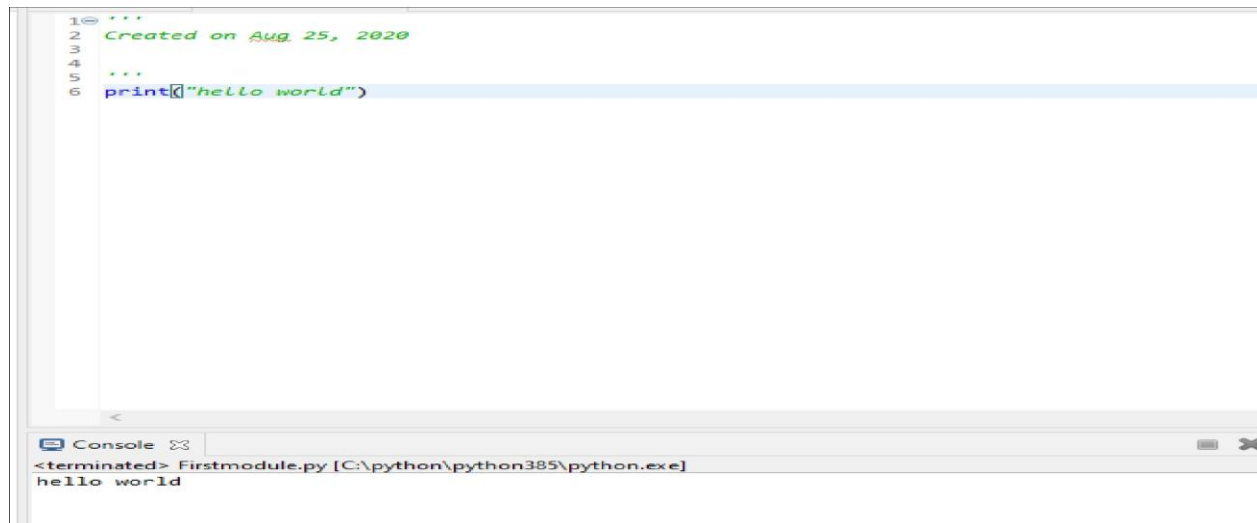
Additional syntax validation: <no additional grammars selected>.

☒ 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: ▾

Exercise 4.1.2 : Write a Hello World program.

Ans:



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

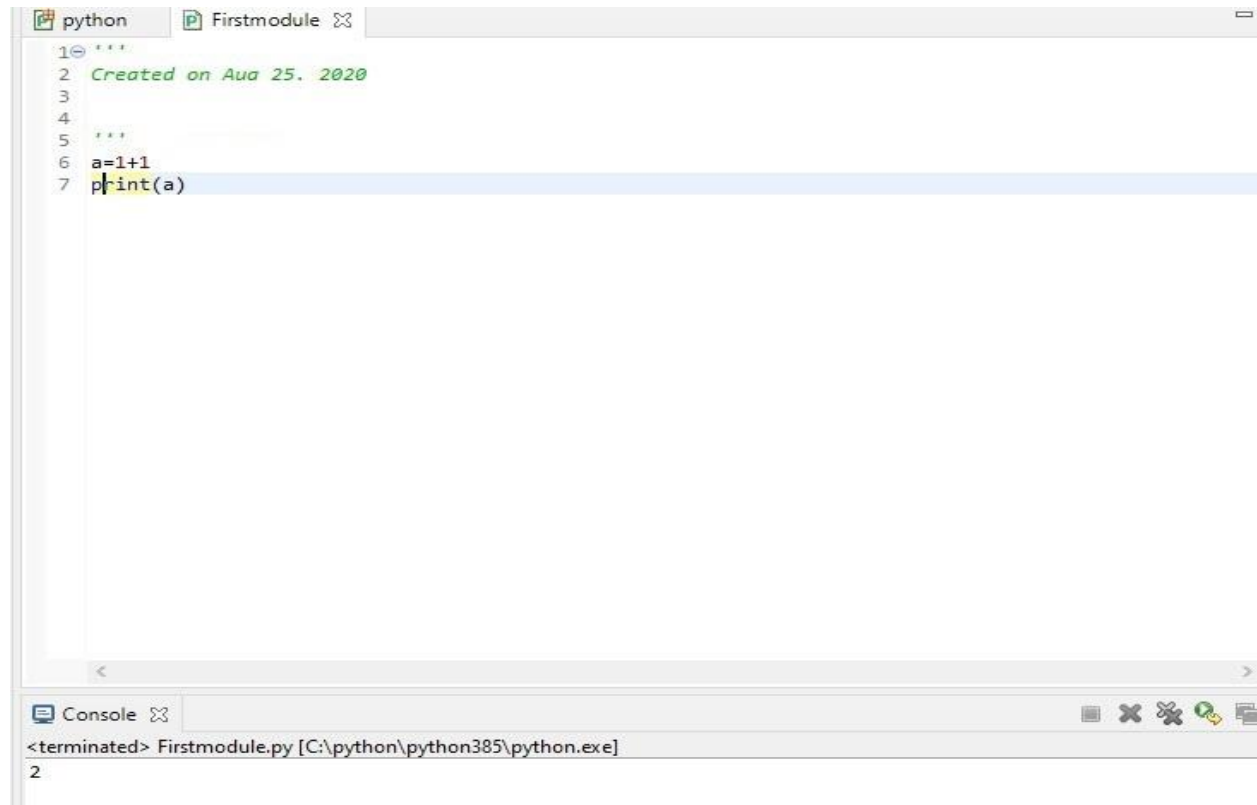
```
1 '''  
2 Created on Aug 25, 2020  
3 '''  
4  
5 '''  
6 print("hello world")
```

The console window at the bottom shows the output of the program:

```
<terminated> Firstmodule.py [C:\python\python385\python.exe]  
hello world
```

Exercise 4.1.3: Compute 1+1

Ans:



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

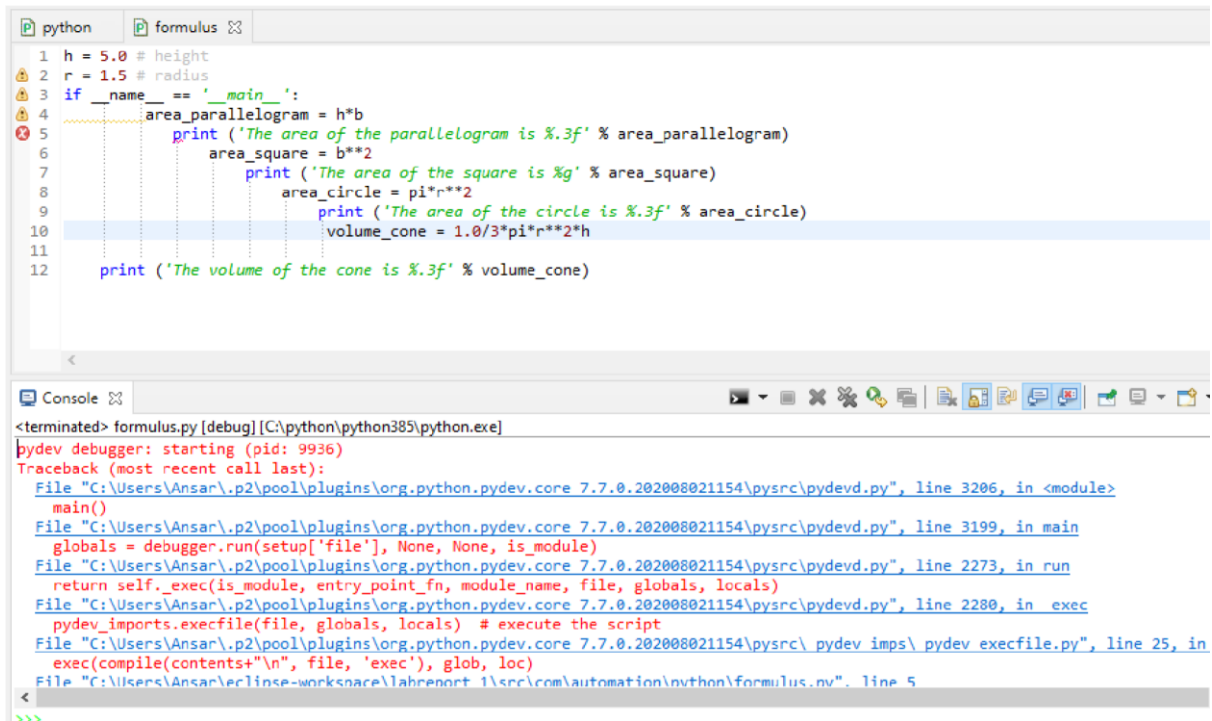
```
1 '''  
2 Created on Aug 25, 2020  
3 '''  
4  
5 '''  
6 a=1+1  
7 print(a)
```

The console window at the bottom shows the output of the program:

```
<terminated> Firstmodule.py [C:\python\python385\python.exe]  
2
```

Exercise 4.1.4: Type in program text

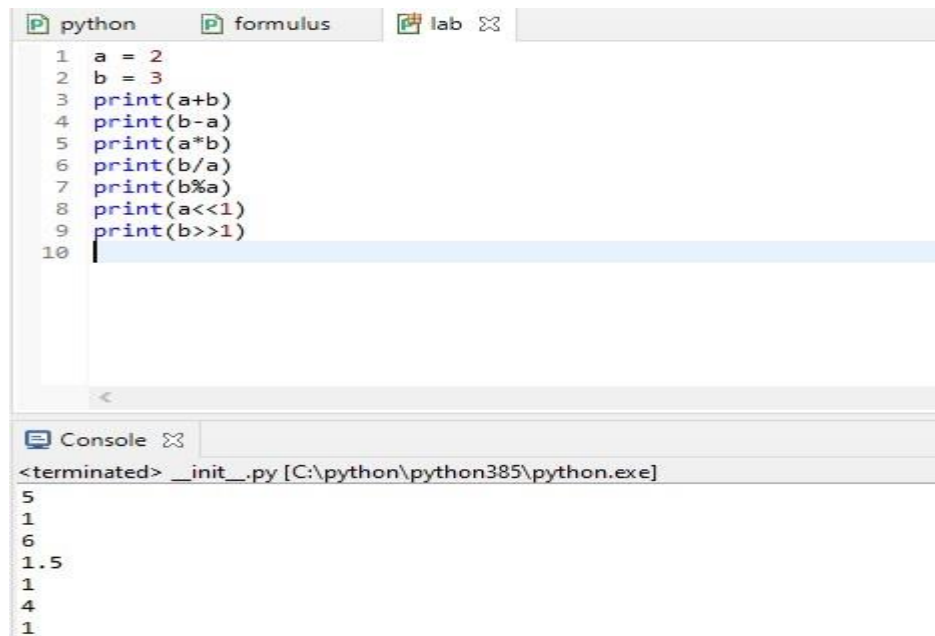
Ans:



```
python  formulus
1 h = 5.0 # height
2 r = 1.5 # radius
3 if __name__ == '__main__':
4     area_parallelogram = h*b
5     print ('The area of the parallelogram is %.3f' % area_parallelogram)
6     area_square = b**2
7     print ('The area of the square is %g' % area_square)
8     area_circle = pi*r**2
9     print ('The area of the circle is %.3f' % area_circle)
10    volume_cone = 1.0/3*pi*r**2*h
11
12    print ('The volume of the cone is %.3f' % volume_cone)
```

```
<terminated> formulus.py [debug] [C:\python\python385\python.exe]
pydev debugger: starting (pid: 9936)
Traceback (most recent call last):
  File "C:\Users\Ansar\.p2\pool\plugins\org.python.pydev.core 7.7.0.202008021154\pysrc\pydevd.py", line 3206, in <module>
    main()
  File "C:\Users\Ansar\.p2\pool\plugins\org.python.pydev.core 7.7.0.202008021154\pysrc\pydevd.py", line 3199, in main
    globals = debugger.run(setup['file'], None, None, is_module)
  File "C:\Users\Ansar\.p2\pool\plugins\org.python.pydev.core 7.7.0.202008021154\pysrc\pydevd.py", line 2273, in run
    return self._exec(is_module, entry_point_fn, module_name, file, globals, locals)
  File "C:\Users\Ansar\.p2\pool\plugins\org.python.pydev.core 7.7.0.202008021154\pysrc\pydevd.py", line 2280, in _exec
    pydev_imports.execfile(file, globals, locals) # execute the script
  File "C:\Users\Ansar\.p2\pool\plugins\org.python.pydev.core 7.7.0.202008021154\pysrc\pydevimps\pydev_execfile.py", line 25, in
    exec(compile(contents+"\n", file, 'exec'), glob, loc)
  File "C:\Users\Ansar\elcine-workspace\labreport 1\src\com\automation\python\formulus.py", line 5
>>>
```

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

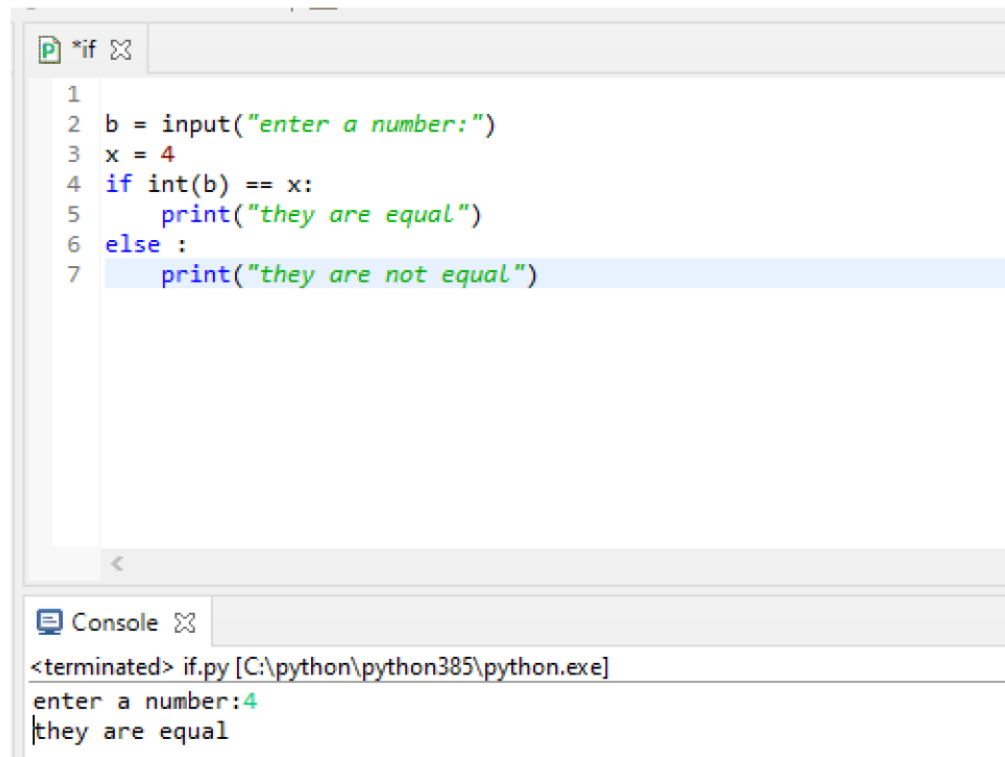


```
python  formulus  lab
1 a = 2
2 b = 3
3 print(a+b)
4 print(b-a)
5 print(a*b)
6 print(b/a)
7 print(b%a)
8 print(a<<1)
9 print(b>>1)
10
```

```
<terminated> __init__.py [C:\python\python385\python.exe]
5
1
6
1.5
1
4
1
```

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



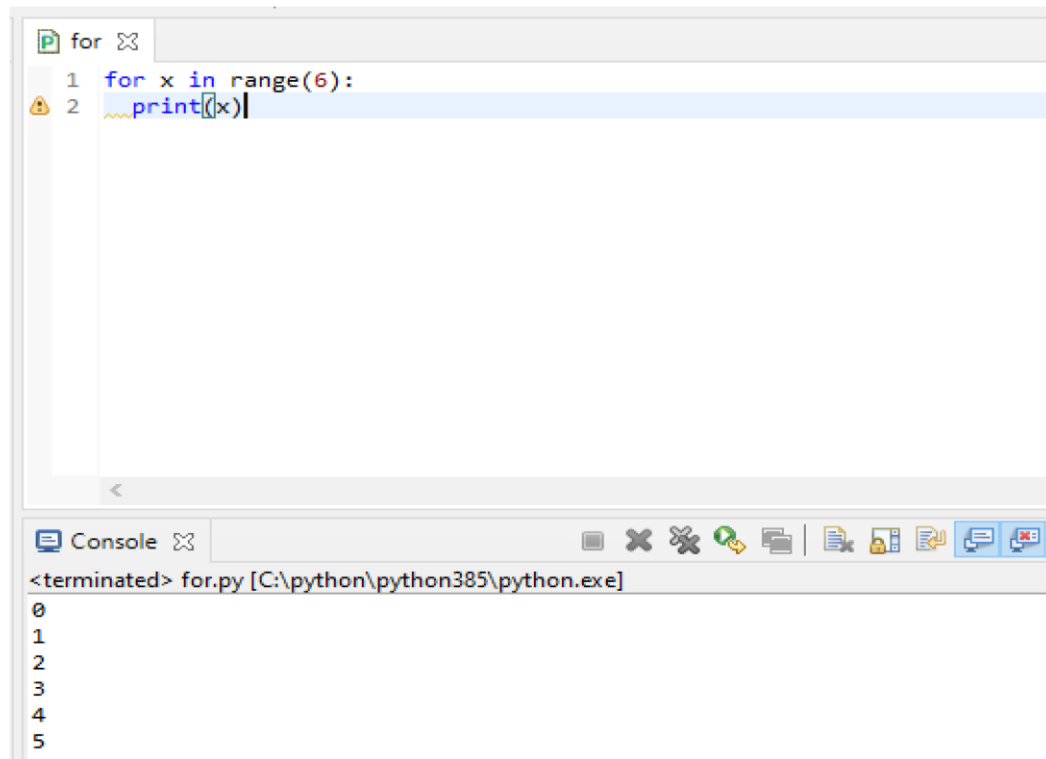
```
*if
1
2 b = input("enter a number:")
3 x = 4
4 if int(b) == x:
5     print("they are equal")
6 else :
7     print("they are not equal")

Console
<terminated> if.py [C:\python\python385\python.exe]
enter a number:4
they are 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

Ans:

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



The screenshot shows the Eclipse IDE with a file named `for.py` open. The code in the editor is:

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

Below the editor is the Console window, which shows the output of the program:

```
<terminated> for.py [C:\python\python385\python.exe]  
0  
1  
2  
3  
4  
5
```

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

Ans:

Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. ... It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

In fact, once we start using Eclipse for real programming, whenever it starts it ... in Eclipse is the PyDev perspective, which we use to develop Python modules ..

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

There are many features in Python, some of which are discussed below –

1. Easy to code:

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.

2. Free and Open Source:

Python language is freely available at the official website and you can download it from the given download link below click on the **Download Python** keyword.

[Download Python](#)

Since it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

3. Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports objectoriented language and concepts of classes, objects encapsulation, etc.

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

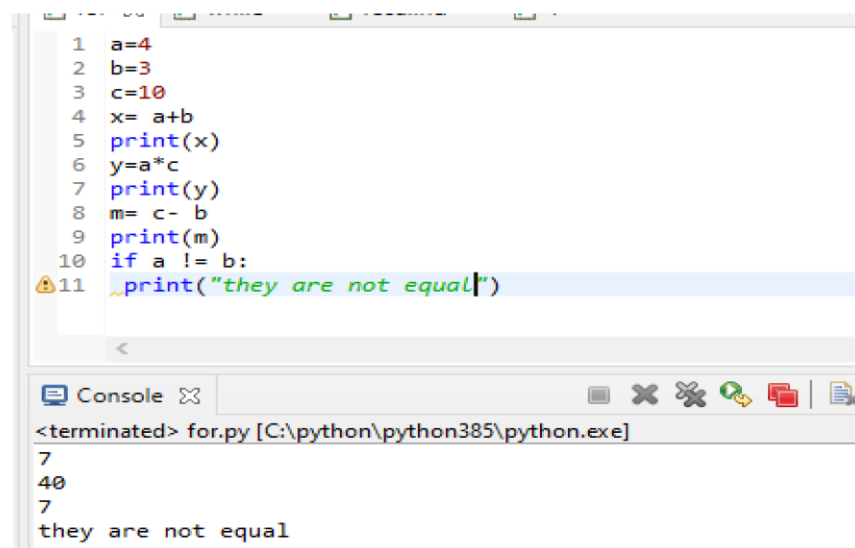
Ans: A package is a collection of Python modules: while a module is a single Python file, a package is a directory of Python modules containing an additional `__init__.py` file, to distinguish a package from a directory that just happens to contain a bunch of Python scripts.

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))
```

Ans: In this code `print()` is incomplete.so there is an error.

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



```
1 a=4
2 b=3
3 c=10
4 x= a+b
5 print(x)
6 y=a*c
7 print(y)
8 m= c- b
9 print(m)
10 if a != b:
11 ~print("they are not equal")
```

Console

```
<terminated> for.py [C:\python\python385\python.exe]
7
40
7
they are not equal
```