Mawlana Bhashani Science and Technology University

Lab-Report

Lab Report No: 01

Course Title: Computer Network Lab

Lab Report Name : Introduction to Python

Submitted by

Name: Moniruzzaman & Mst.Zakia Sultana

ID: IT-18007 & IT-18027

3rd year 2nd semester

Session: 2017-2018

Dept. of ICT

MBSTU.

Submitted To

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

Theory:

Definition of Python: The official definition of Python is

Python is an easy to learn, powerful programming language. It has efficient highlevel 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 with. 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, 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.

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.

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.

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.

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 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!

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.

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.

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

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.

Other Libraries: Besides, the standard library, there are various other high-quality libraries such as:

wxPython [http://www.wxpython.org],

Twisted [http://www.twistedmatrix.com/products/twisted],

Python Imaging Library [http://www.pythonware.com/products/pil/index.htm]

Methodology:

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.

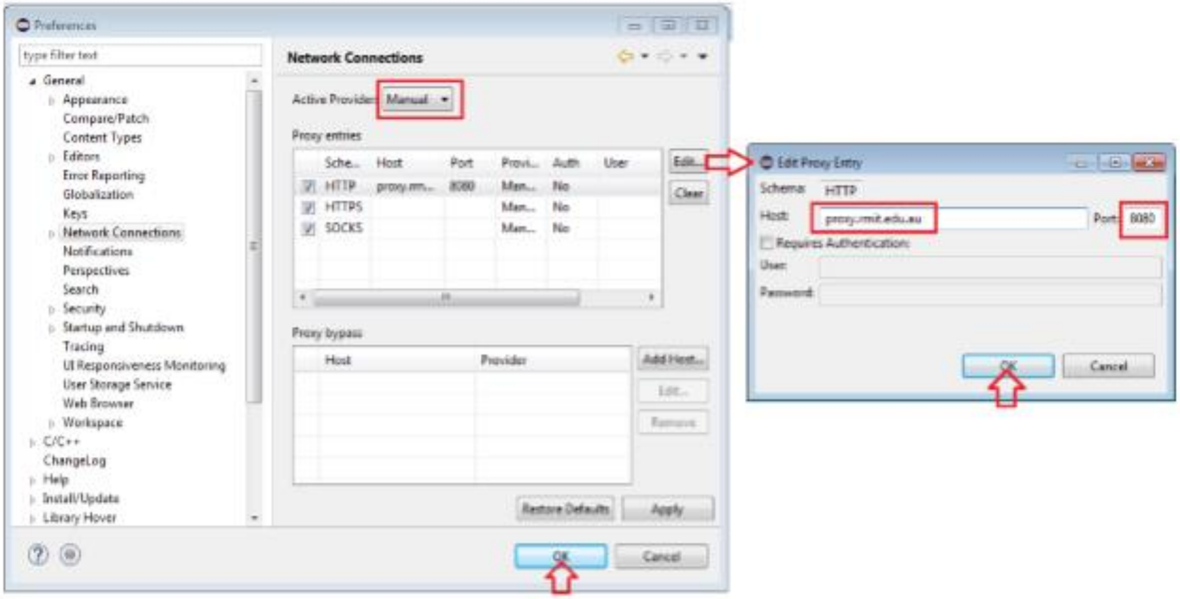
Host: proxy.rmit.edu.au

Port: 8080

Username/password: No required

d. Clear SOCKS proxy.

e. Restart Eclipse.



Network Connections

Active Provider: Manual

Proxy entries

	Sche...	Host	Port	Provi...	Auth	User
☑	HTTP	proxy.rm...	8080	Man...	No	
☑	HTTPS			Man...	No	
☑	SOCKS			Man...	No	

Proxy bypass

Host	Provider

Edit Proxy Entry

Schema: HTTP

Host: proxy.rmit.edu.au Port: 8080

□ Requires Authentication:

User:

Password:

Figure 3-1. Eclipse setup for Internet.

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:

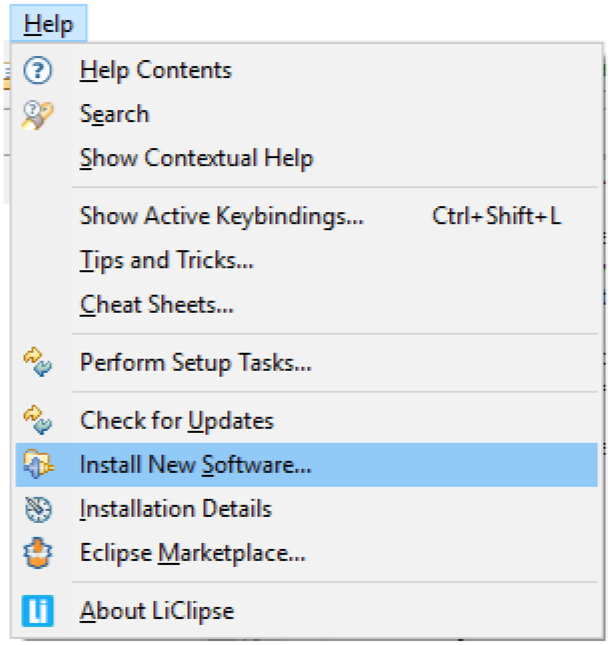
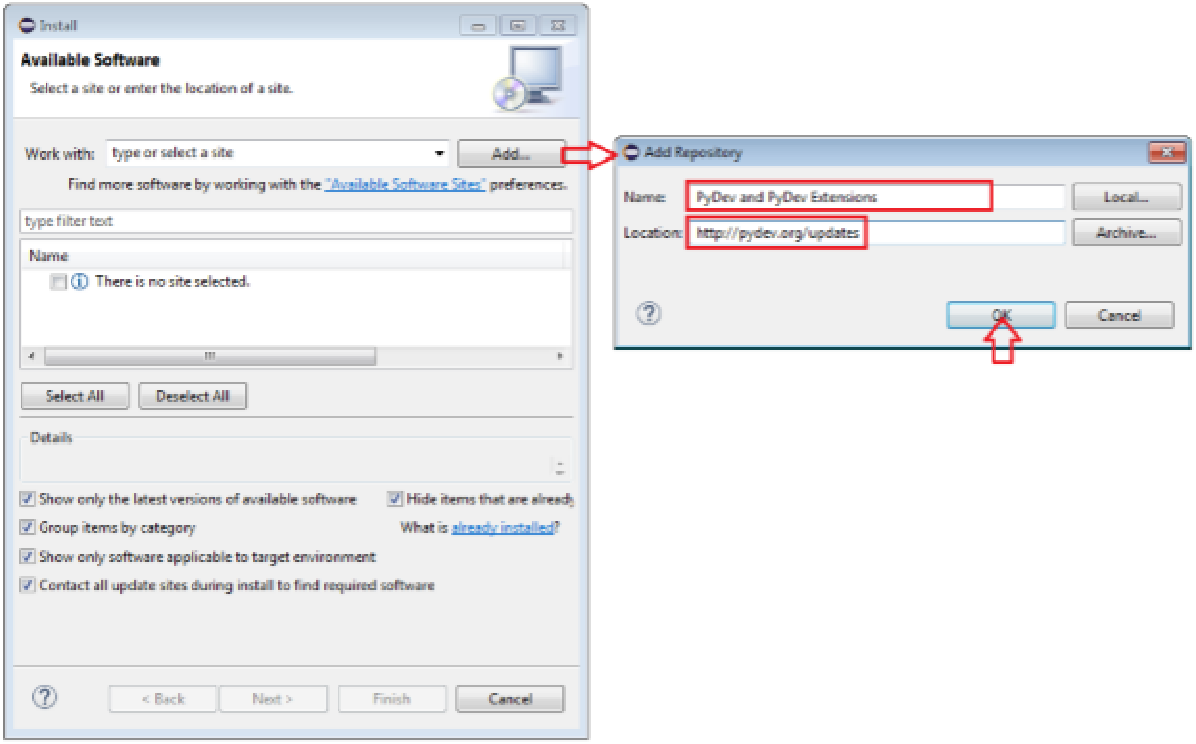


Figure 3-2. Step 2.

b. In the next screen, add the update site(s) you want to work with (see the figure below). The available update sites are (see Figure 3):
http://pydev.org/updates (Last version for your personal PC)
http://pydev.org/updates4.5(version for RMIT Labs**)**



Available Software

Select a site or enter the location of a site.

Work with: type or select a site

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name

□ There is no site selected.

Details

☑ Show only the latest versions of available software

☑ Hide items that are already

What is already installed?

☑ Group items by category

☑ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

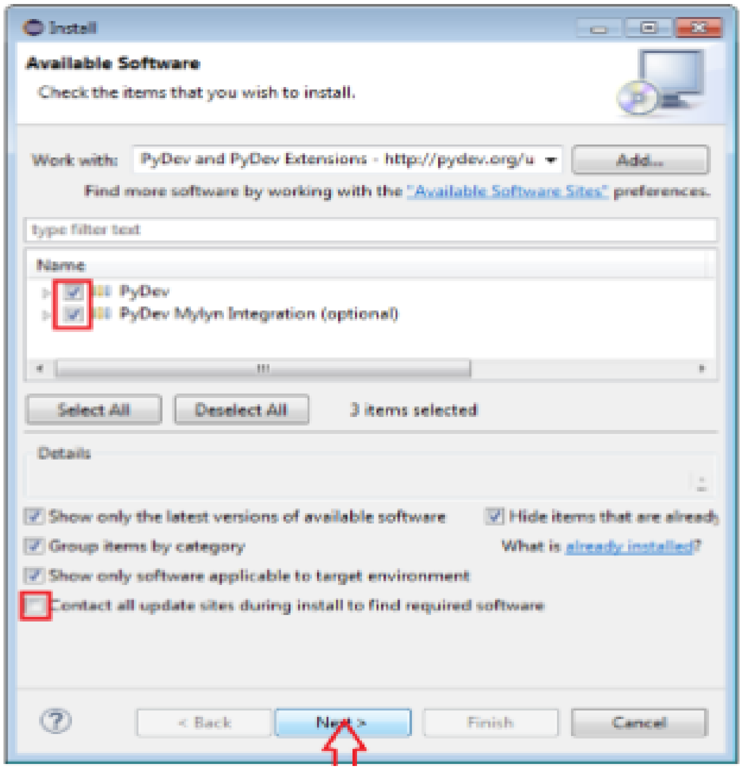
Add Repository

Name: PyDev and PyDev Extensions

Location: http://pydev.org/updates

Figure 3-3. Set up Python on Eclipse.

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' (see Figure 4).



Available Software

Check the items that you wish to install.

Work with: PyDev and PyDev Extensions - http://pydev.org/u

Find more software by working with the "Available Software Sites" preferences.

☑ PyDev

☑ PyDev Mylyn Integration (optional)

3 items selected

☑ Show only the latest versions of available software

☑ Hide items that are already

☑ Group items by category

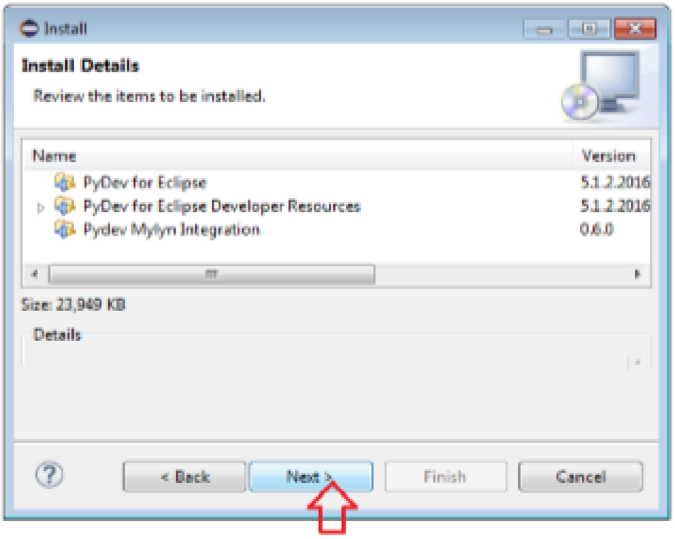
What is already installed?

☑ Show only software applicable to target environment

□ Contact all update sites during install to find required software

Figure 3-4. Set up Python on Eclipse.

d. Then, UNCHECK the 'Contact all update sites during install to find required software' and press 'Next' again to confirm your selection (see Figure 5).



Install Details

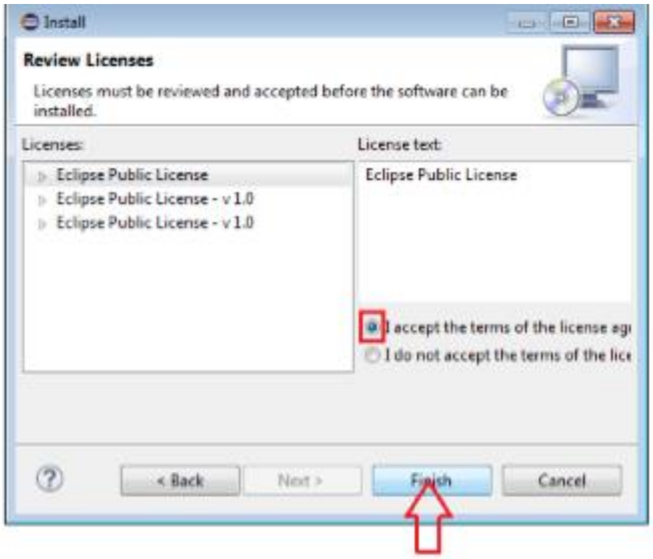
Review the items to be installed.

Name	Version
PyDev for Eclipse	5.1.2.2016
PyDev for Eclipse Developer Resources	5.1.2.2016
Pydev Mylyn Integration	0.6.0

Size: 23,949 KB

Figure 3-5. Set up Python on Eclipse.

e. And finally, read the license agreement and if you accept, select the accept radio button and click 'Finish' (see Figure 6).



Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

Licenses:

- Eclipse Public License
- Eclipse Public License - v 1.0
- Eclipse Public License - v 1.0

License text:

Eclipse Public License

● I accept the terms of the license agı

○ I do not accept the terms of the lic

Figure 3-6. Set up Python on Eclipse

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,
Run bottom allows to run a python script,

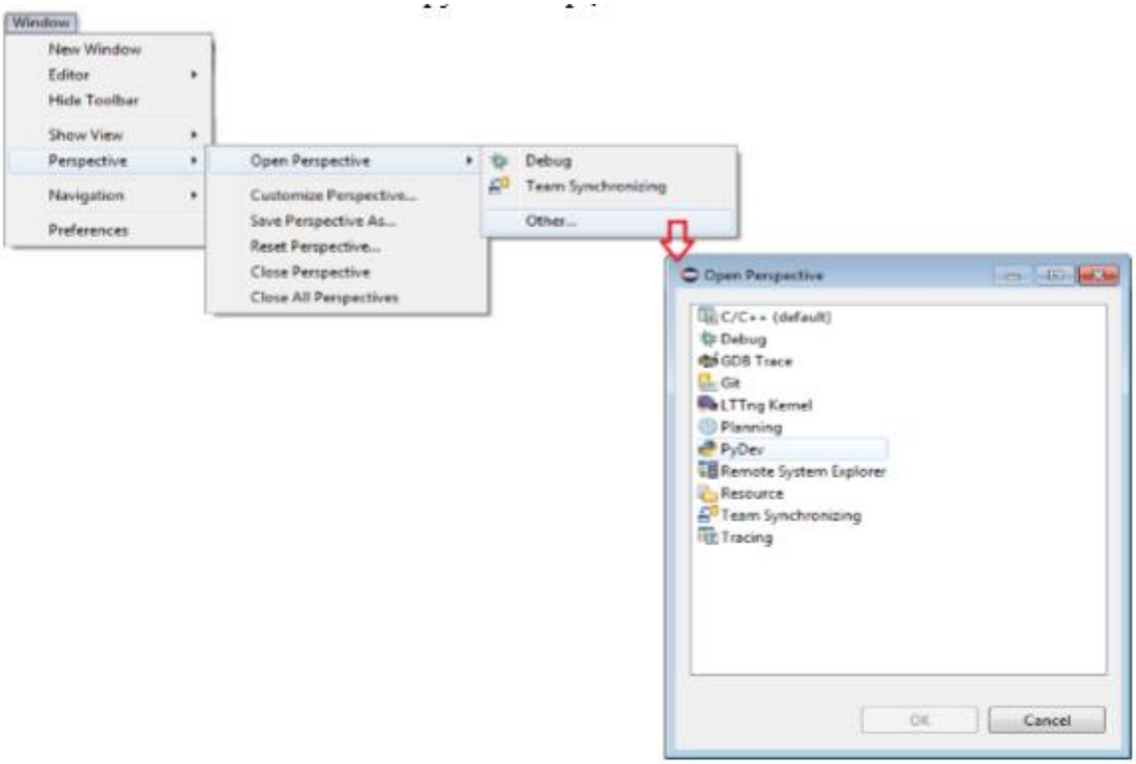
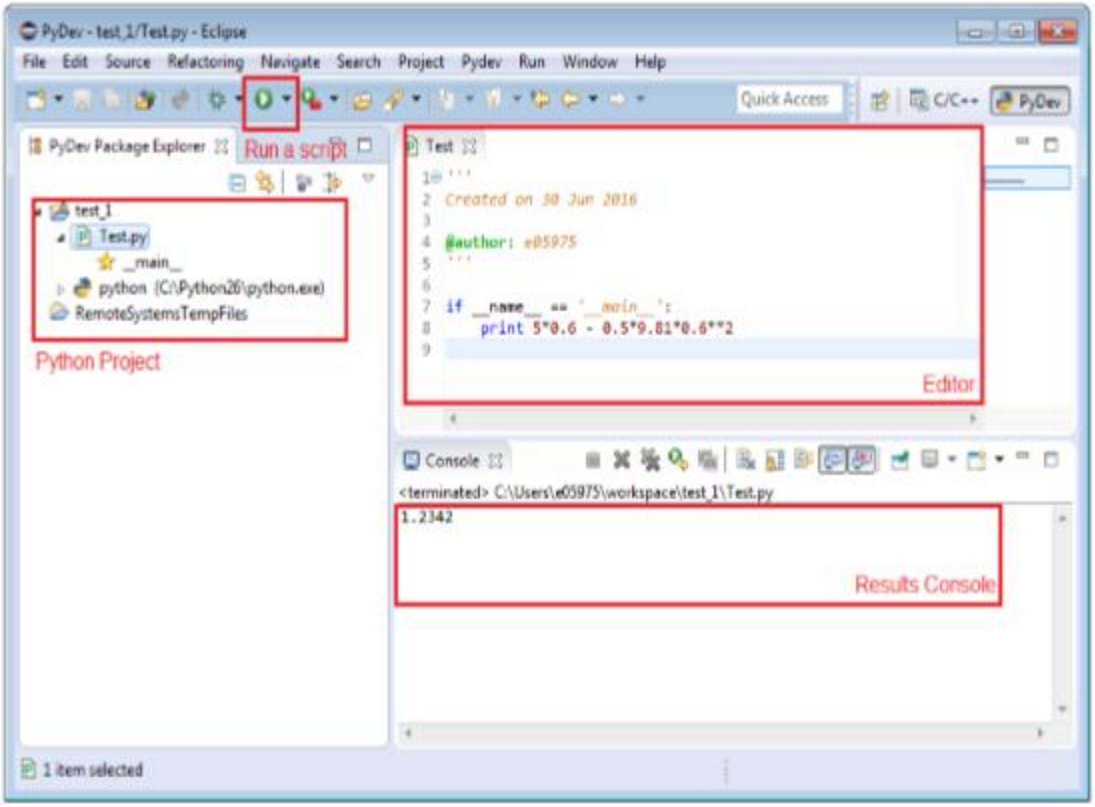


Figure 3-7. Python perspective in Eclipse.



```
'''
Created on 30 Jun 2016

@author: e05975
'''


if __name__ == '__main__':
    print 5*0.6 - 0.5*9.81*0.6**2
```

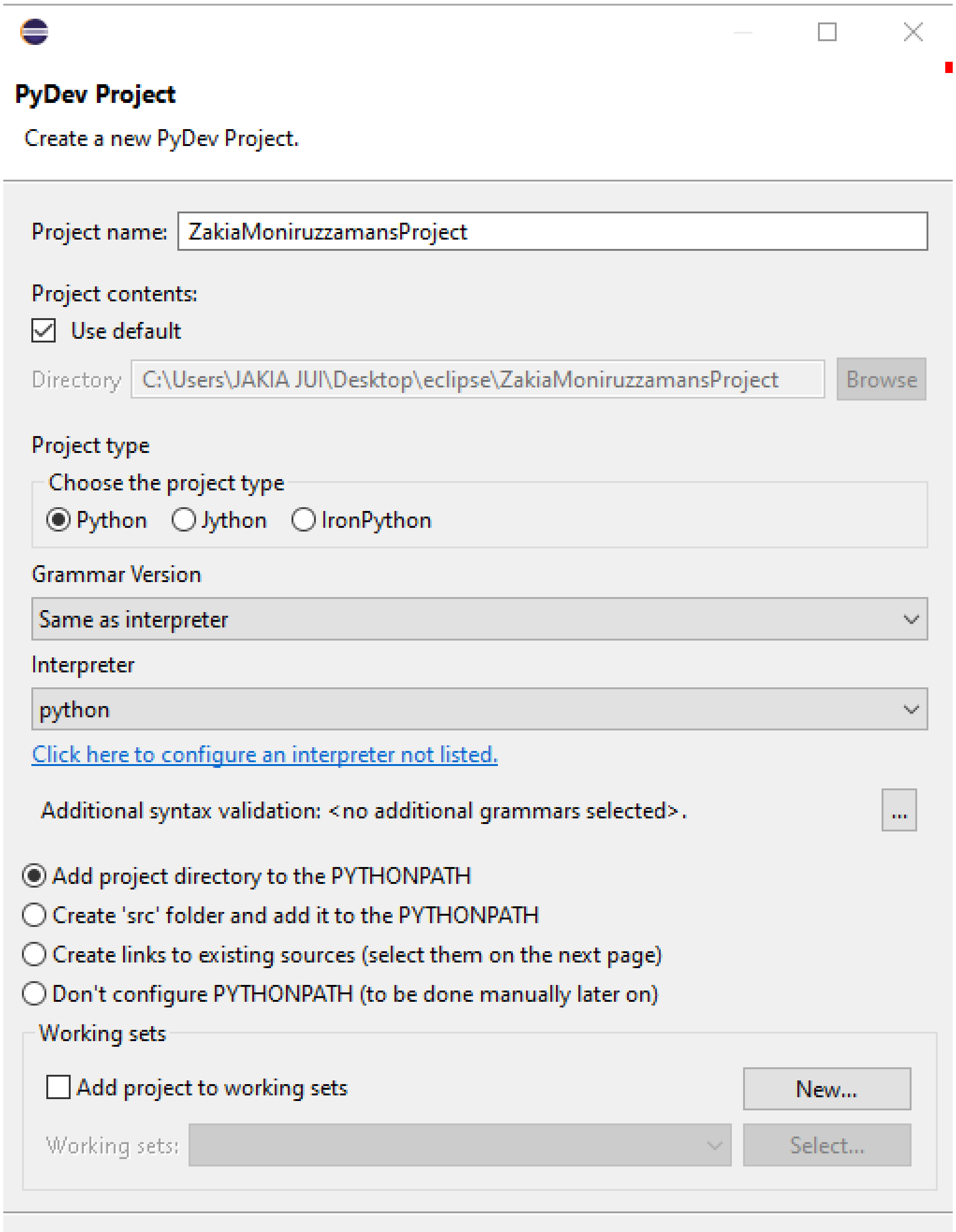
```
<terminated> C:\Users\e05975\workspace\test_1\Test.py
1.2342
```

Figure 3-8. Python perspective environment.

Exercises

Section 4.1: Basics of python and programin

Exercise 4.1.1: Create a python project.



PyDev Project

Create a new PyDev Project.

Project name: ZakiaMoniruzzamansProject

Project contents:

☑ Use default

Directory C:\Users\JAKIA JUI\Desktop\eclipse\ZakiaMoniruzzamansProject

Project type

Choose the project type

● Python ○ Jython ○ IronPython

Grammar Version

Same as interpreter

Interpreter

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)

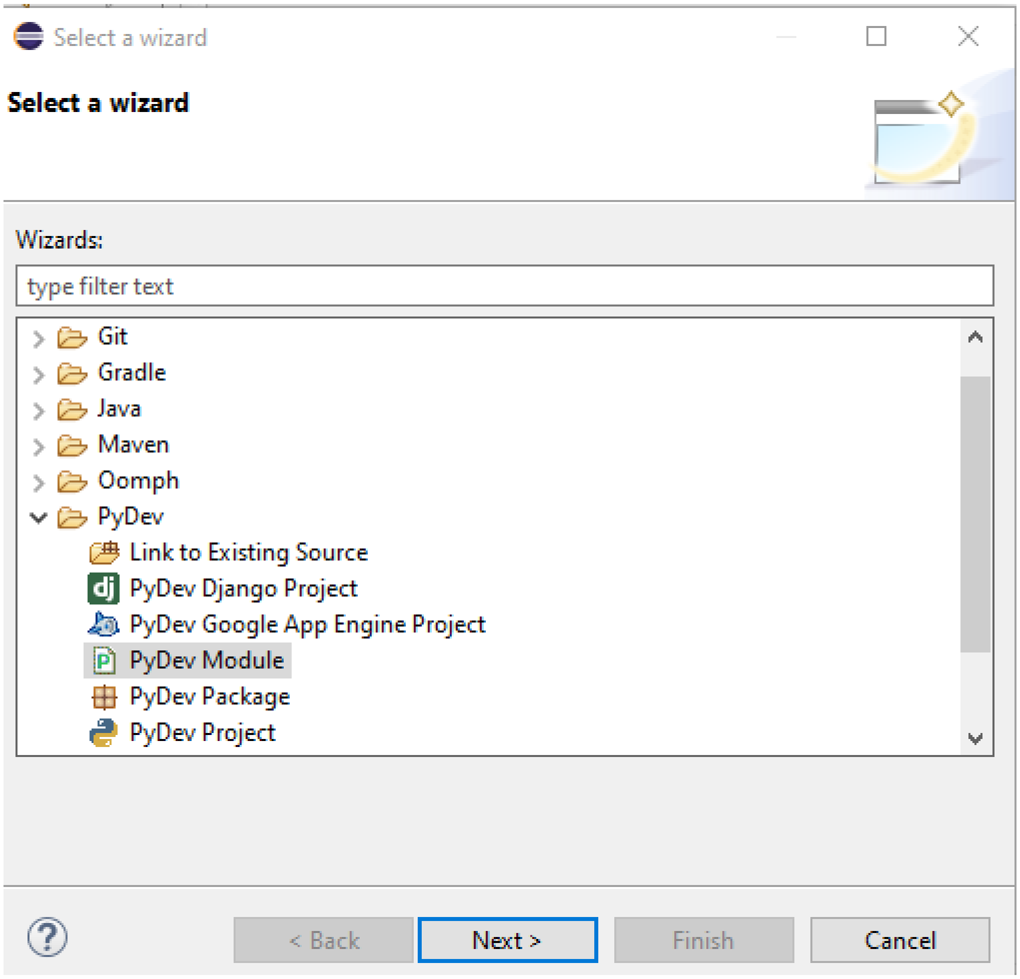
Working sets

☐ Add project to working sets

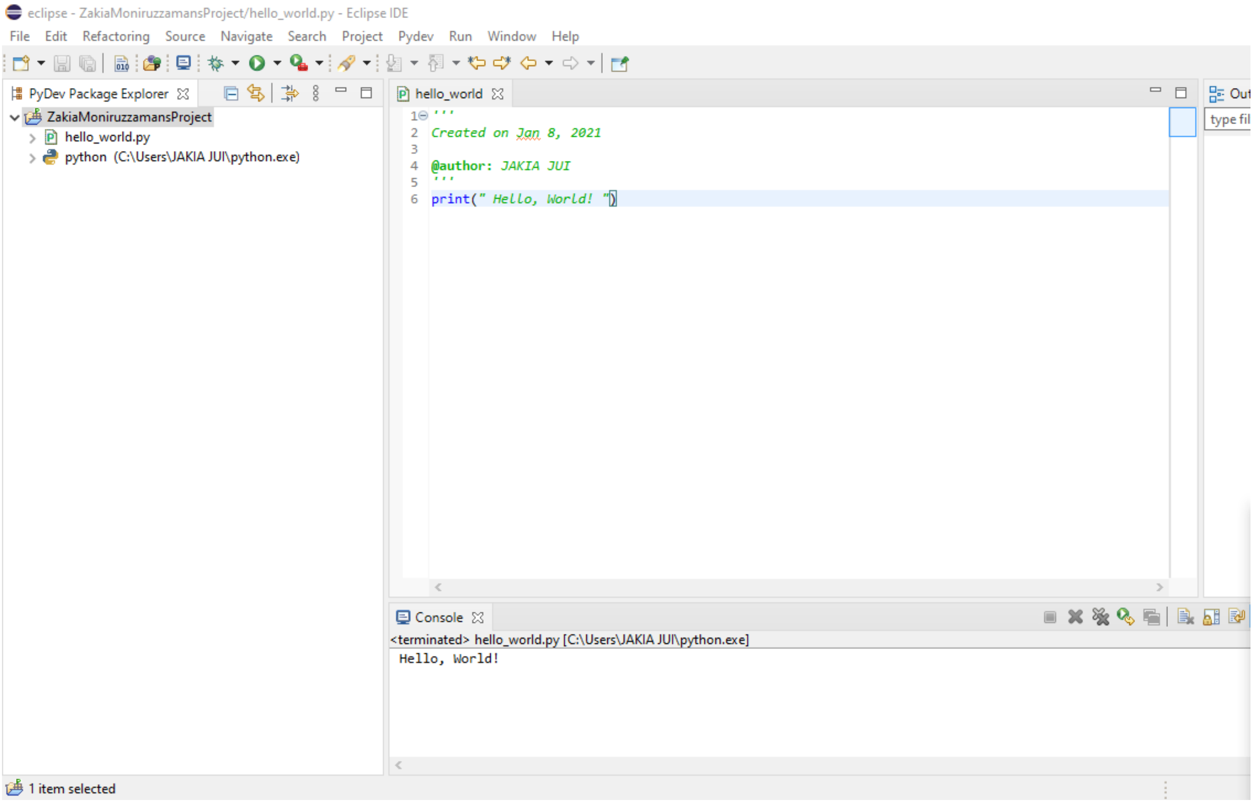
Working sets:

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:



Exercise: 4.1.2 : Write a Hello World Program.

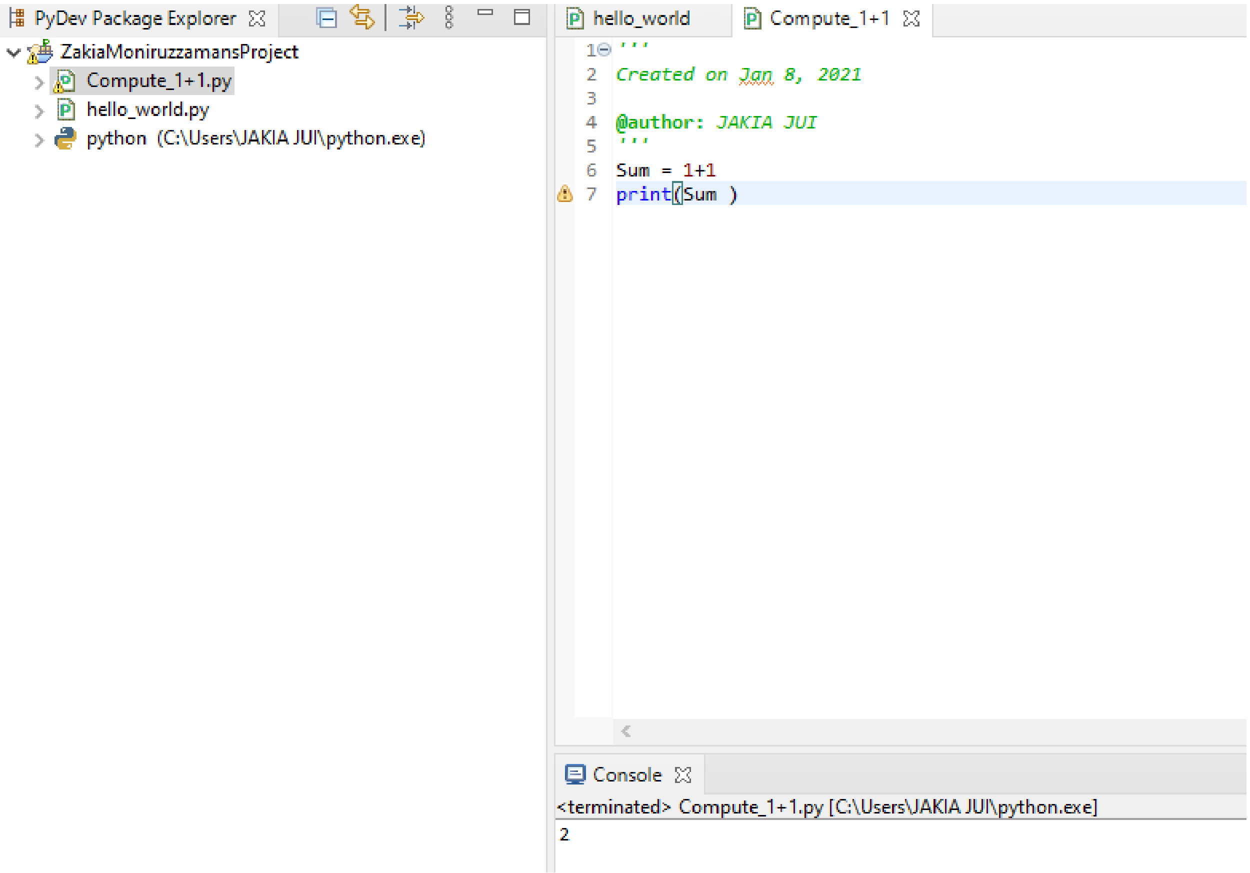

eclipse - ZakiaMoniruzzamansProject/hello_world.py - Eclipse IDE
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help
PyDev Package Explorer
ZakiaMoniruzzamansProject
hello_world.py
python (C:\Users\JAKIA JUI\python.exe)
hello_world
Out
type fil
Console
1 item selected

```
'''
Created on Jan 8, 2021

@author: JAKIA JUI
'''
print(" Hello, World! ")
```

```
<terminated> hello_world.py [C:\Users\JAKIA JUI\python.exe]
Hello, World!
```

Exercise 4.1.3: Compute 1+1

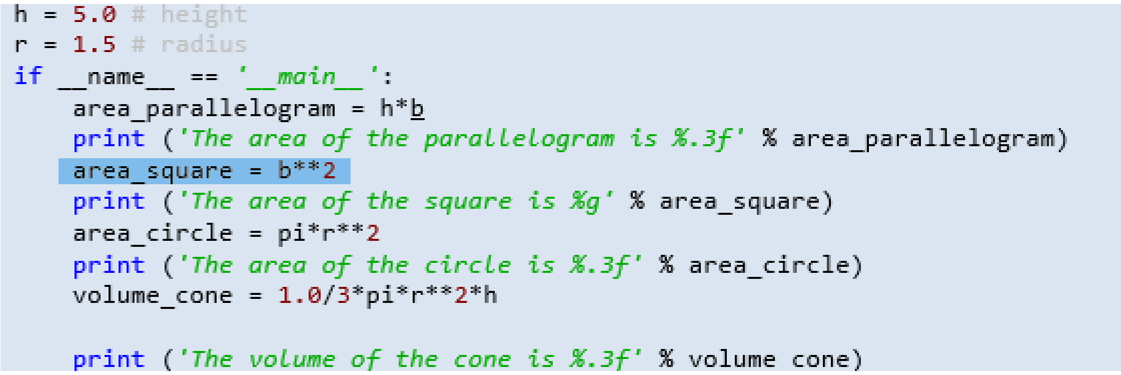


```python
'''
Created on Jan 8, 2021

@author: JAKIA JUI
'''
Sum = 1+1
print(Sum )
```

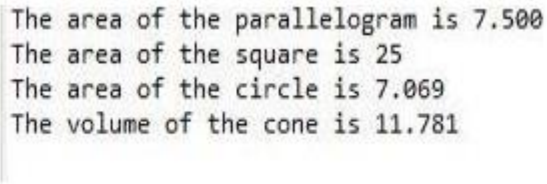
```
<terminated> Compute_1+1.py [C:\Users\JAKIA JUI\python.exe]
2
```

Exercise 4.1.4: Type in program text



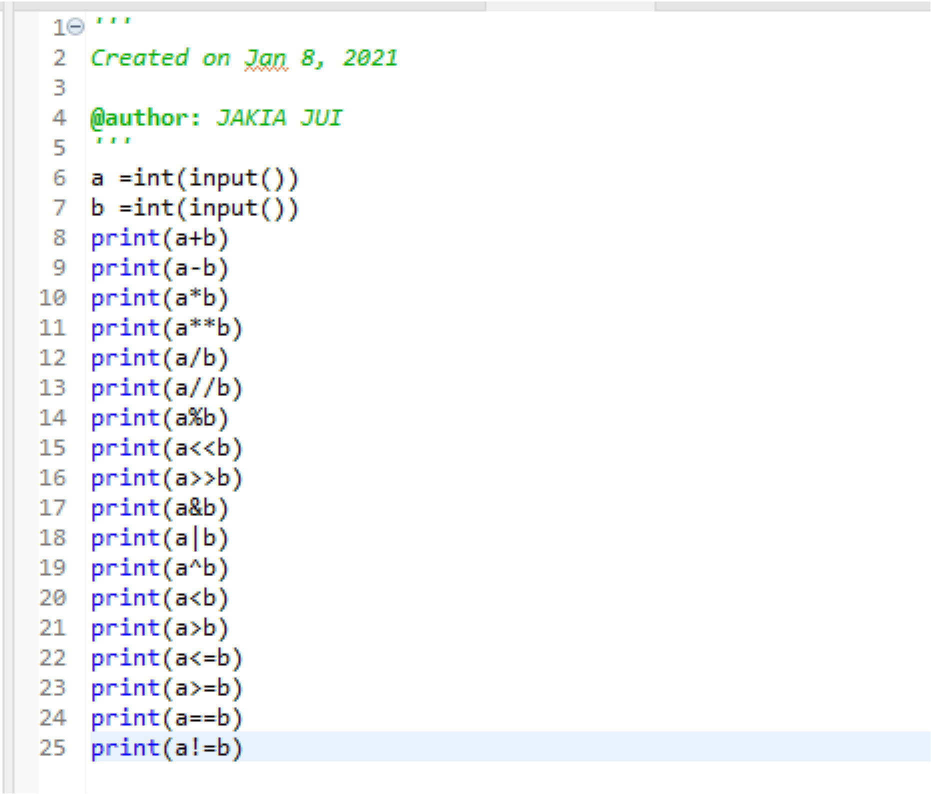
```python
h = 5.0 # height
r = 1.5 # radius
if __name__ == '__main__':
    area_parallelogram = h*b
    print ('The area of the parallelogram is %.3f' % area_parallelogram)
    area_square = b**2
    print ('The area of the square is %g' % area_square)
    area_circle = pi*r**2
    print ('The area of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



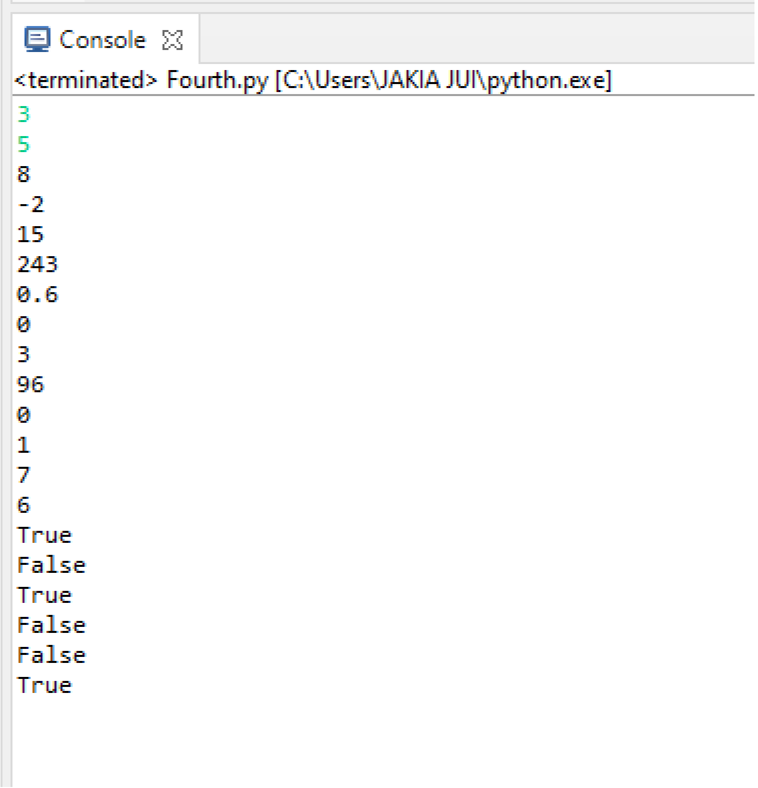
```
The area of the parallelogram is 7.500
The area of the square is 25
The area of the circle is 7.069
The volume of the cone is 11.781
```

Section 4.1: Create and run basic example.



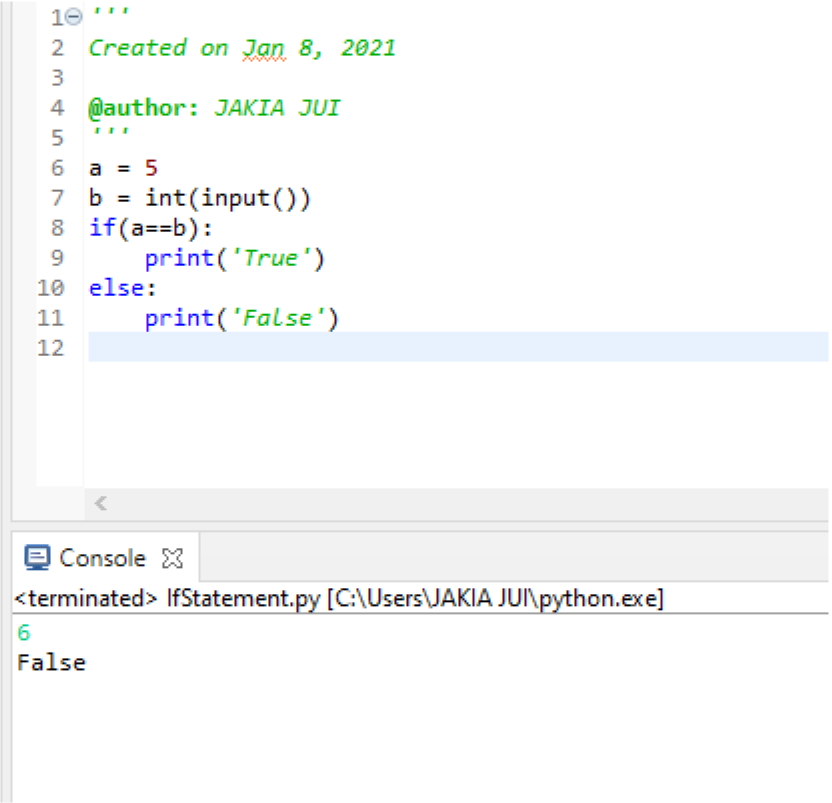
```
'''
Created on Jan 8, 2021

@author: JAKIA JUI
'''
a =int(input())
b =int(input())
print(a+b)
print(a-b)
print(a*b)
print(a**b)
print(a/b)
print(a//b)
print(a%b)
print(a<<b)
print(a>>b)
print(a&b)
print(a|b)
print(a^b)
print(a<b)
print(a>b)
print(a<=b)
print(a>=b)
print(a==b)
print(a!=b)
```



```
<terminated> Fourth.py [C:\Users\JAKIA JUI\python.exe]
3
5
8
-2
15
243
0.6
0
3
96
0
1
7
6
True
False
True
False
False
True
```

Exercise 4.2.2: The if statement:



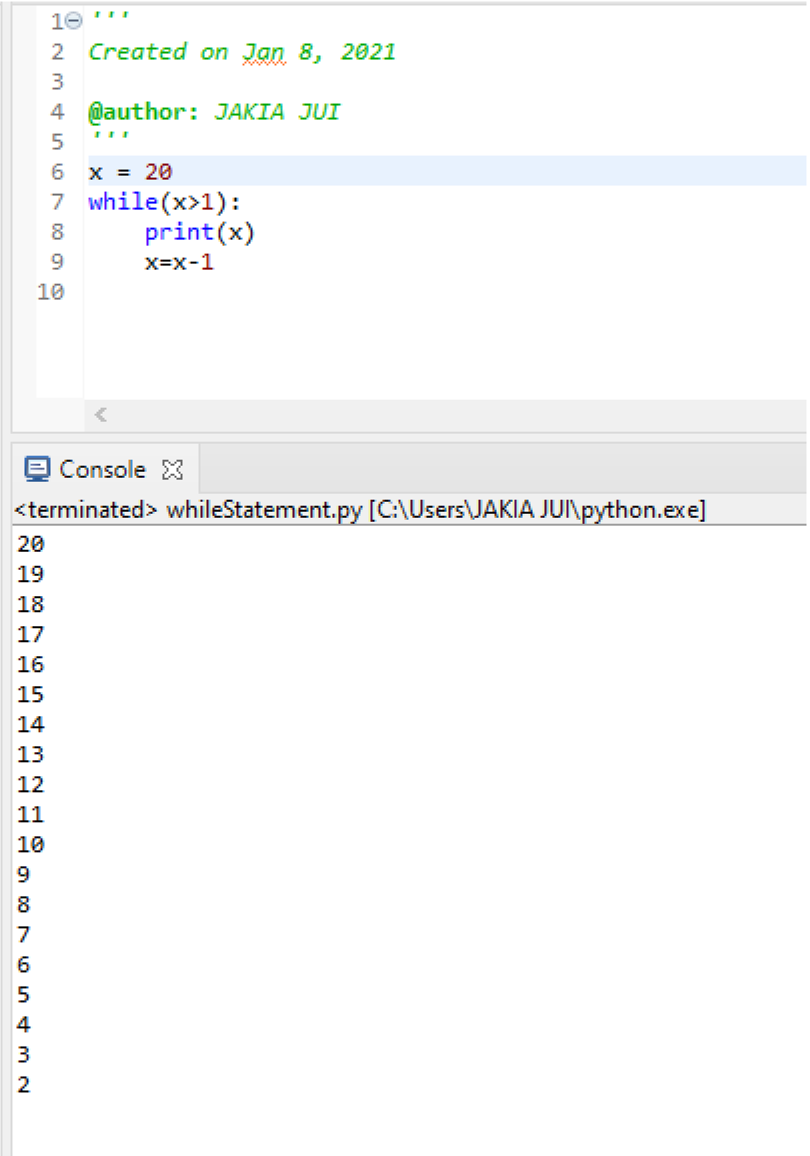
```
'''
Created on Jan 8, 2021

@author: JAKIA JUI
'''
a = 5
b = int(input())
if(a==b):
    print('True')
else:
    print('False')

```

```
<terminated> IfStatement.py [C:\Users\JAKIA JUI\python.exe]
6
False
```

Exercise 4.2.3: The while Statement



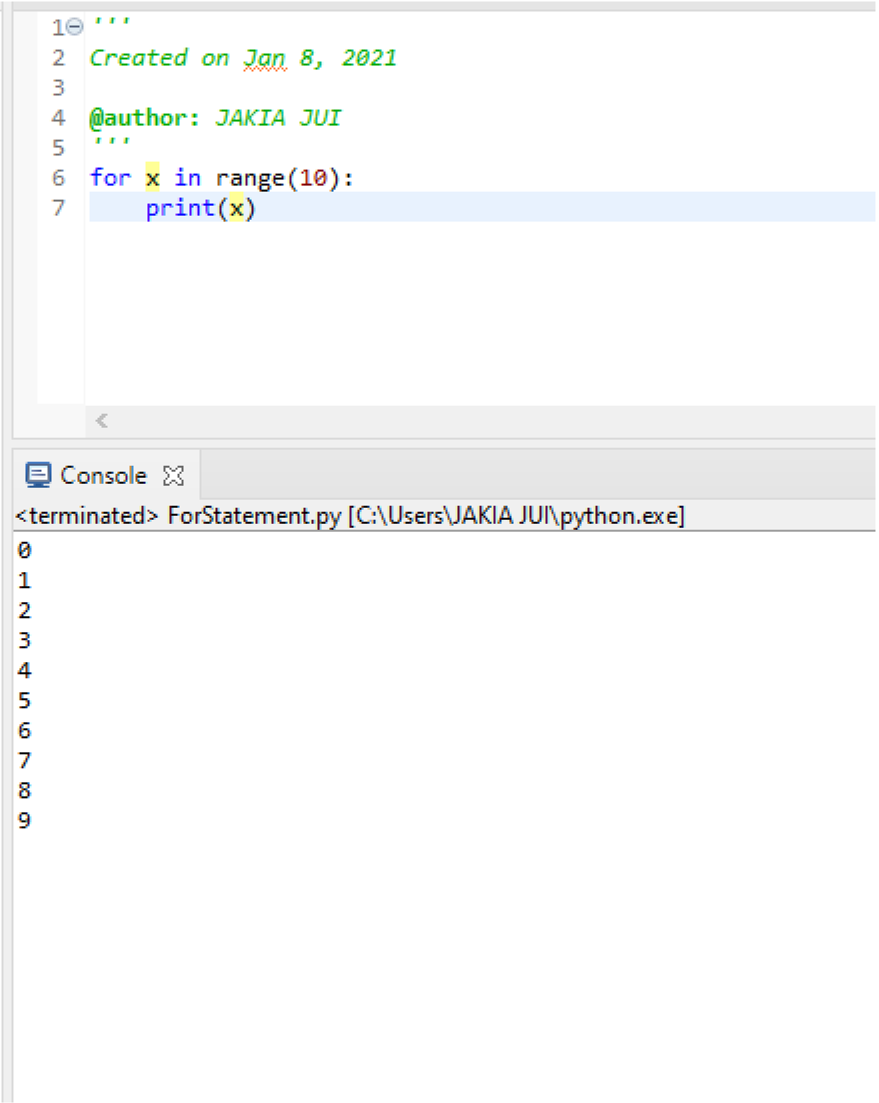
```
'''
Created on Jan 8, 2021

@author: JAKIA JUI
'''
x = 20
while(x>1):
    print(x)
    x=x-1
```

<terminated> whileStatement.py [C:\Users\JAKIA JUI\python.exe]

```
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
```

Exercise 4.2.4: The for Statement



```
'''
Created on Jan 8, 2021

@author: JAKIA JUI
'''
for x in range(10):
    print(x)
```

```
<terminated> ForStatement.py [C:\Users\JAKIA JUI\python.exe]
0
1
2
3
4
5
6
7
8
9
```

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. Ifyou'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 it's 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.

5. Questions

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

Answer:

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.

In the sections below you will learn how to

- Start and Stop Eclipse
- Recognize the parts of the Eclipse Workbench
- Test code in the Python Interpreter
- Create a New a Project in the Eclipse Workbench
- Enter and Edit a Script (a module that runs as a program)
- Run a Script
- Find and Correct Errors in a Script
- Finish a Project
- Use an Old/Existing/Instrutor-distributed Project in the Eclipse Workbench

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

Answer:

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 object-oriented language and concepts of classes, objects encapsulation, etc.