

## Lab Report: 01

**Report Name:** Introduction to Python

**Course code:** ICT-3208

**Course title:** Computer Networks Lab

**Date of**

**Performance:**    /    /2021

**Date of**

**Submission:** 08/01/2021

**SUBMITTED BY**

**Name:** Shakhera khanom

**ID:** IT-18033

**3<sup>rd</sup> year 2<sup>nd</sup> semester**

**Session:** 2017-18

**Department of ICT,**

**MBSTU.**

**SUBMITTED TO**

**Nazrul Islam**

**Assistant Professor**

**Department of ICT,**

**MBSTU.**

## **Experiment No: 01**

### **Experiment Name: Introduction to Python**

#### **Objectives:**

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

**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.

#### **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.

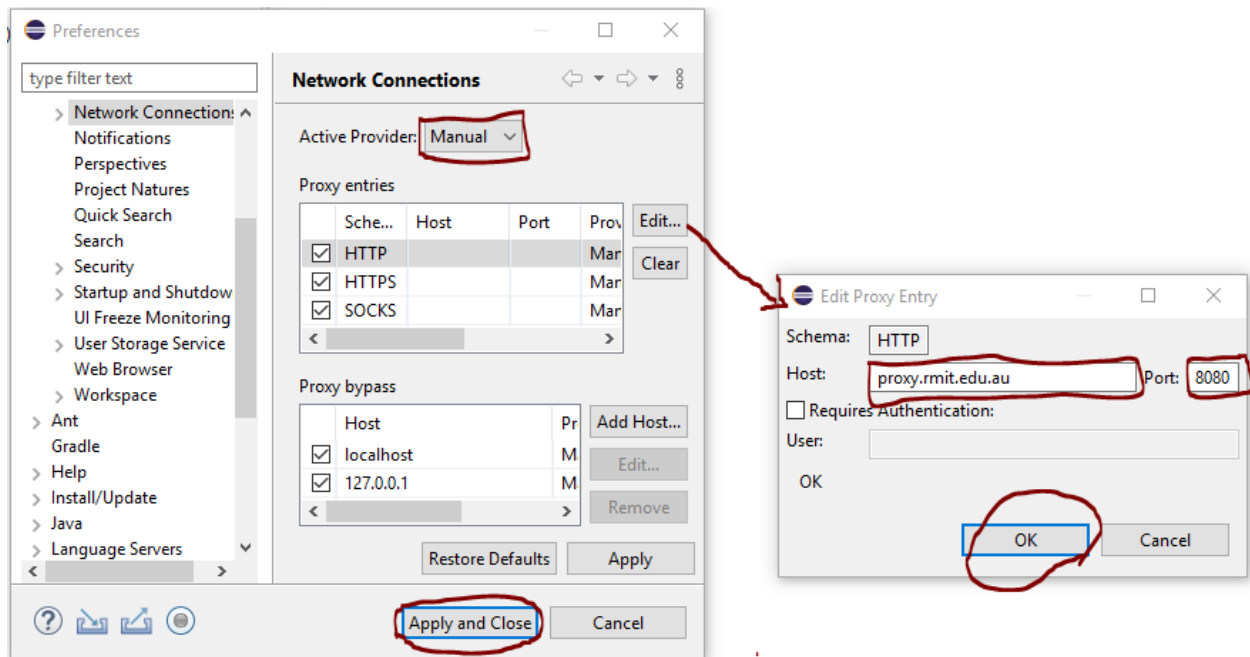


Figure 1. Eclipse setup for Internet

## 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:

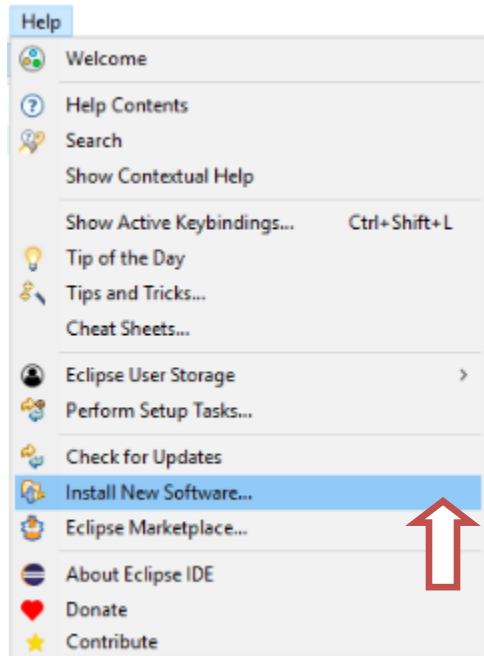


Figure 2. Step 2.

**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):

<https://pydev.org/updates>

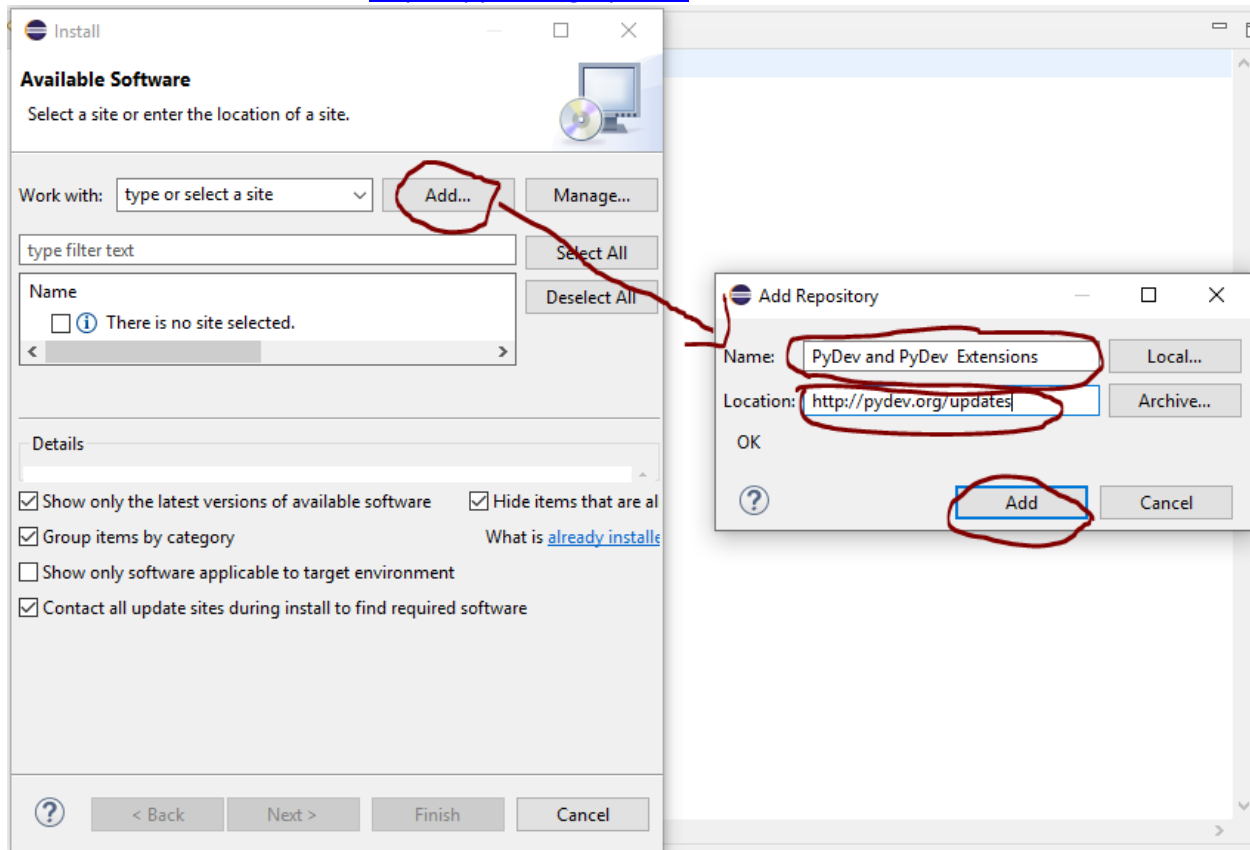


Figure 3. Set up Python on Eclipse

**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).

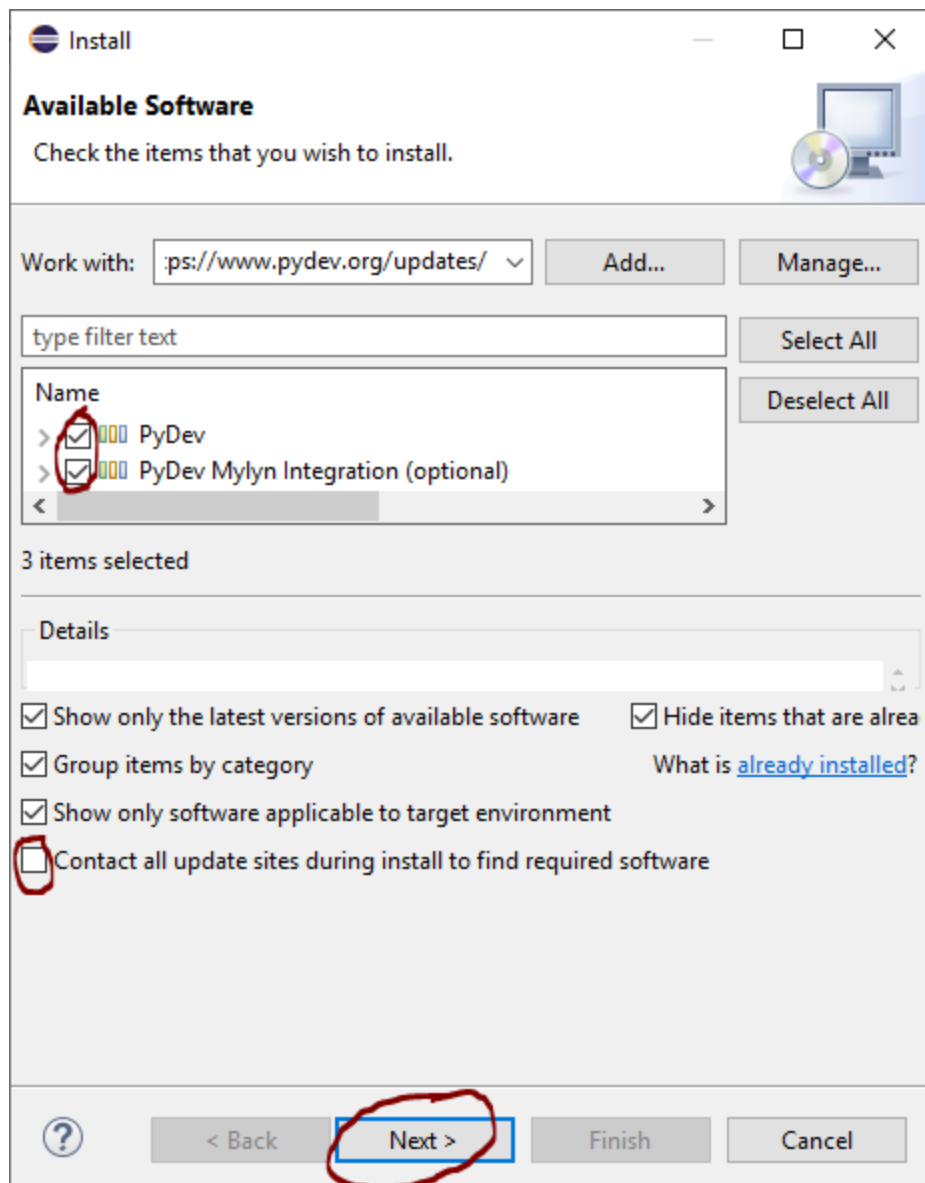


Figure 4. Set up Python on Eclipse.

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).

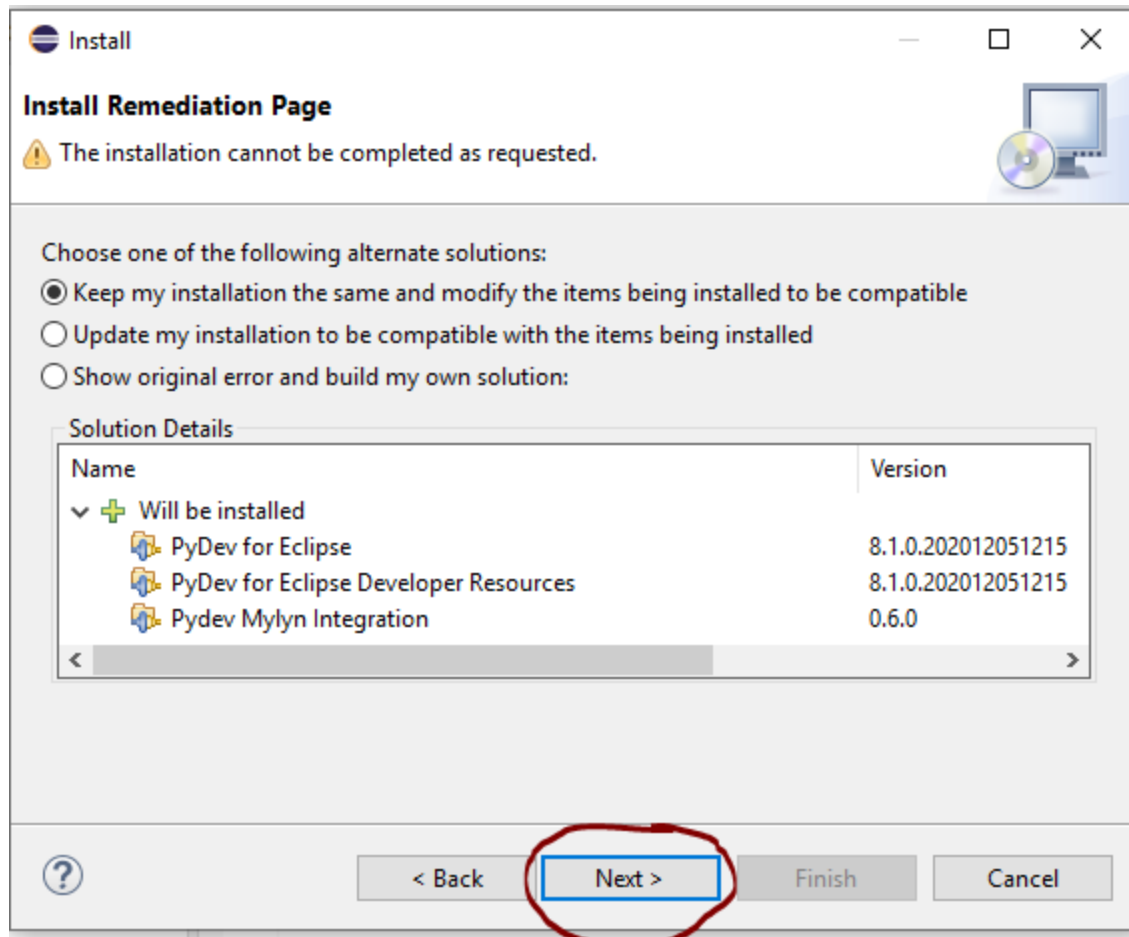


Figure 5. Set up Python on Eclipse.

e.

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

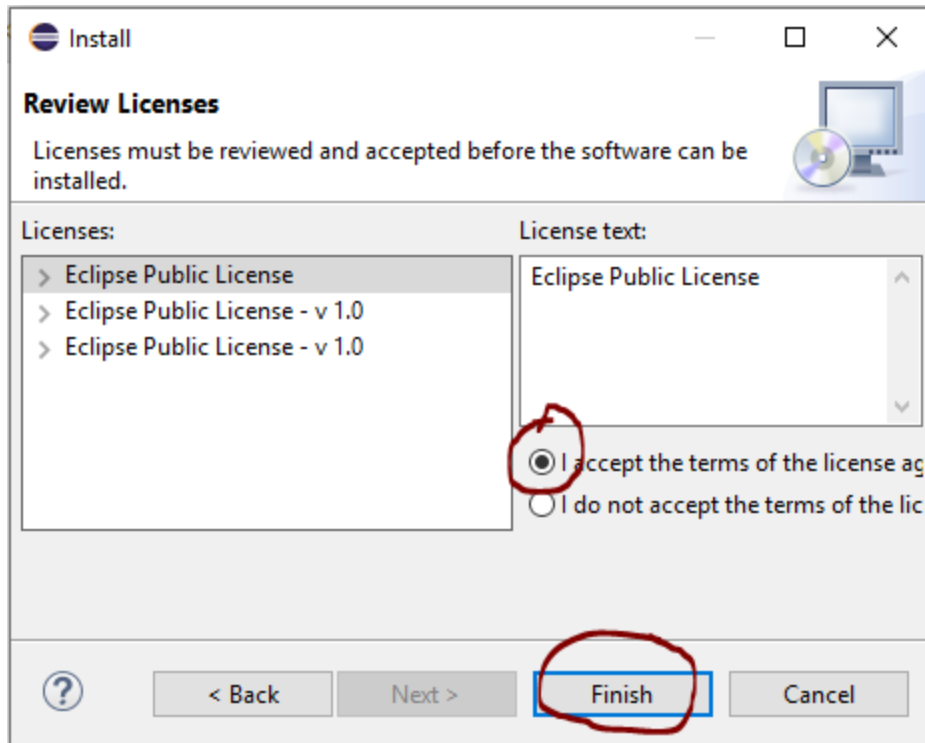


Figure 6. Set up Python on Eclipse

**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 after running a python script,
- Run bottom allows to run a python script,

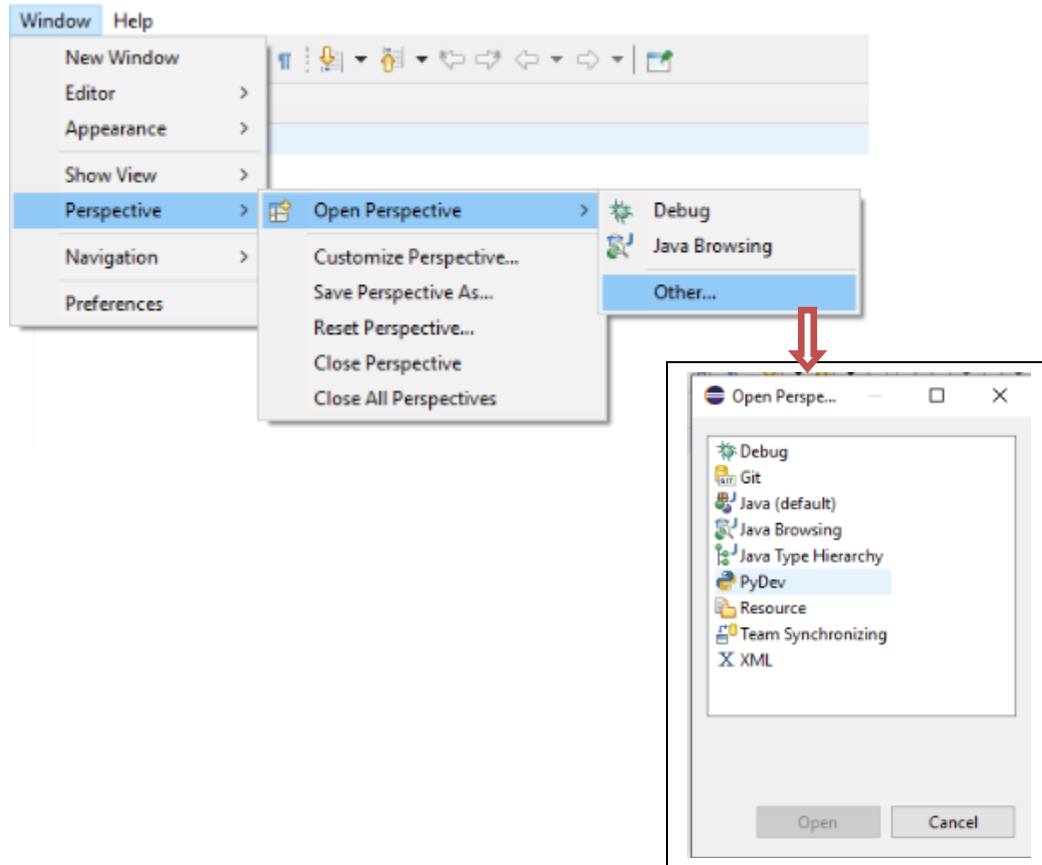
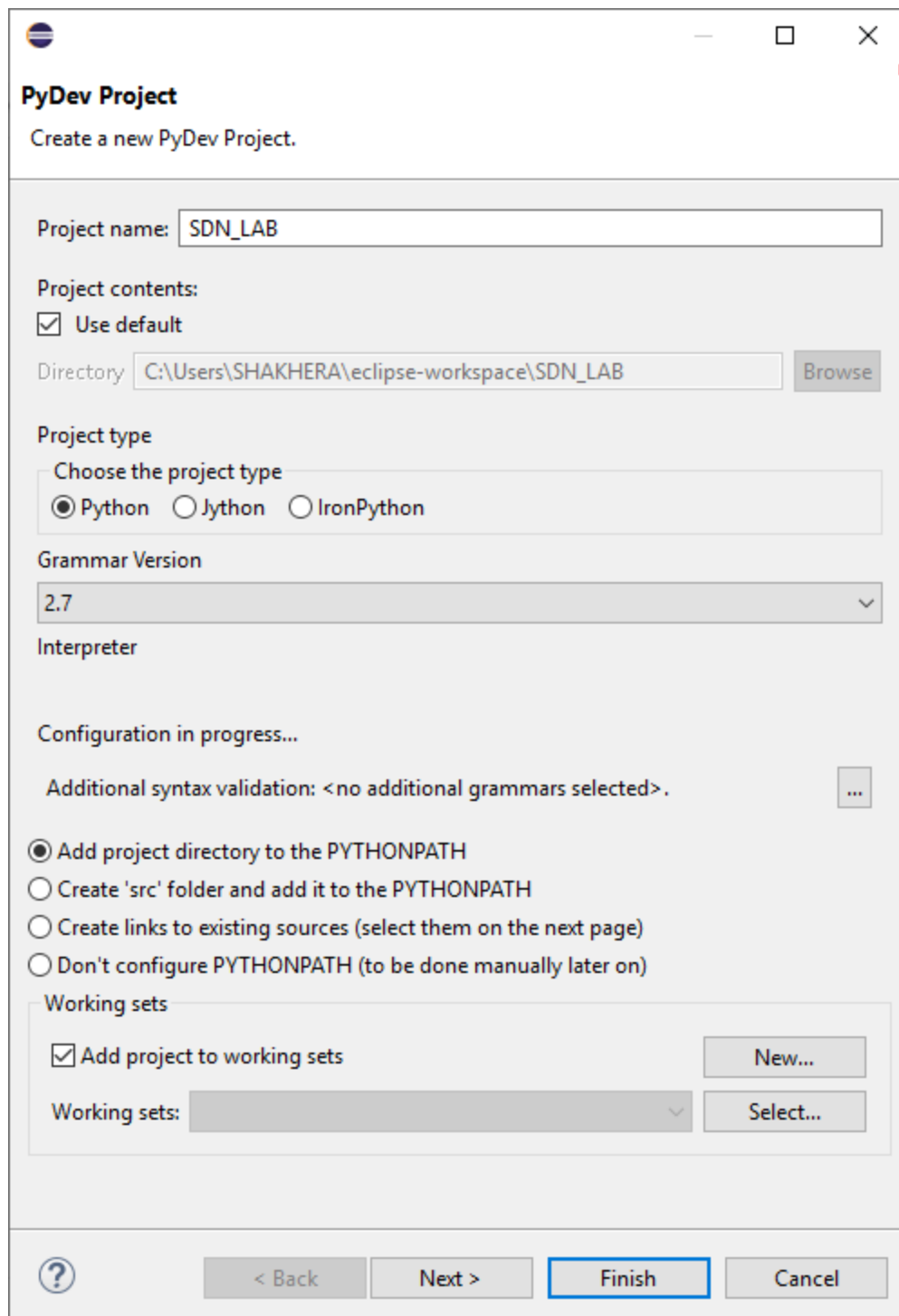


Figure 7. Python perspective in Eclipse.

### Exercises:

**Exercise 4.1.1:** Create a python project, click in **File > New > PyDev Project**. Provide a name for the project (SDN\_LAB\_1 for the fits lab), then select the version of python to be used and select to add the project to working set as shown below:



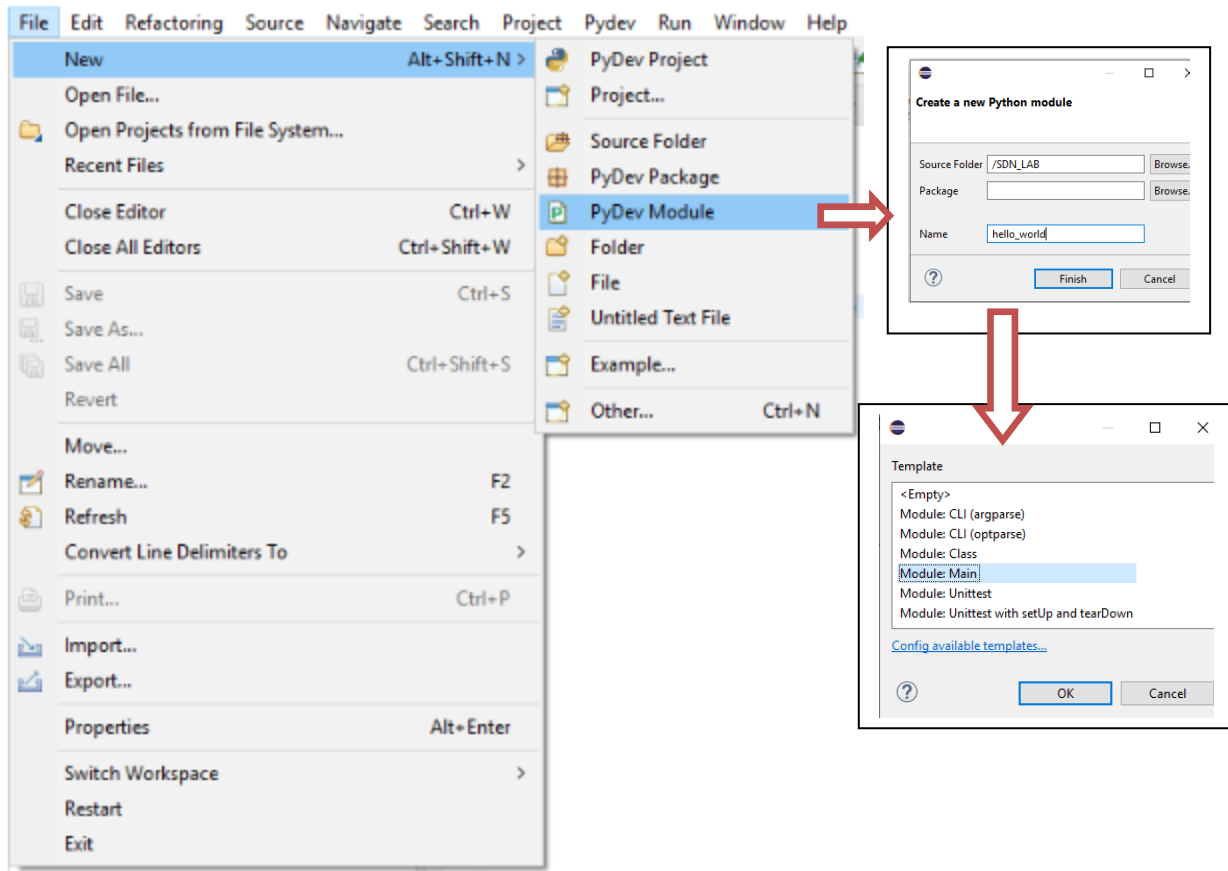


The image shows a 'PyDev Project' dialog box with the following fields and options:

- Project name:** A text box containing 'SDN\_LAB'.
- Project contents:**
  - ☒ Use default
  - Directory:** A text box containing 'C:\Users\SHAKHERA\eclipse-workspace\SDN\_LAB' and a 'Browse' button.
- Project type:** A section titled 'Choose the project type' with three radio buttons: ☒ Python, ☐ Jython, and ☐ IronPython.
- Grammar Version:** A dropdown menu showing '2.7'.
- Interpreter:** A section titled 'Configuration in progress...' with the text 'Additional syntax validation: <no additional grammars selected>' and a small '...' button.
- Working sets:** A section with four radio buttons: ☒ 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), and ☐ Don't configure PYTHONPATH (to be done manually later on).
- Working sets:** A section with ☒ Add project to working sets, a 'New...' button, a 'Working sets:' dropdown menu, and a 'Select...' button.

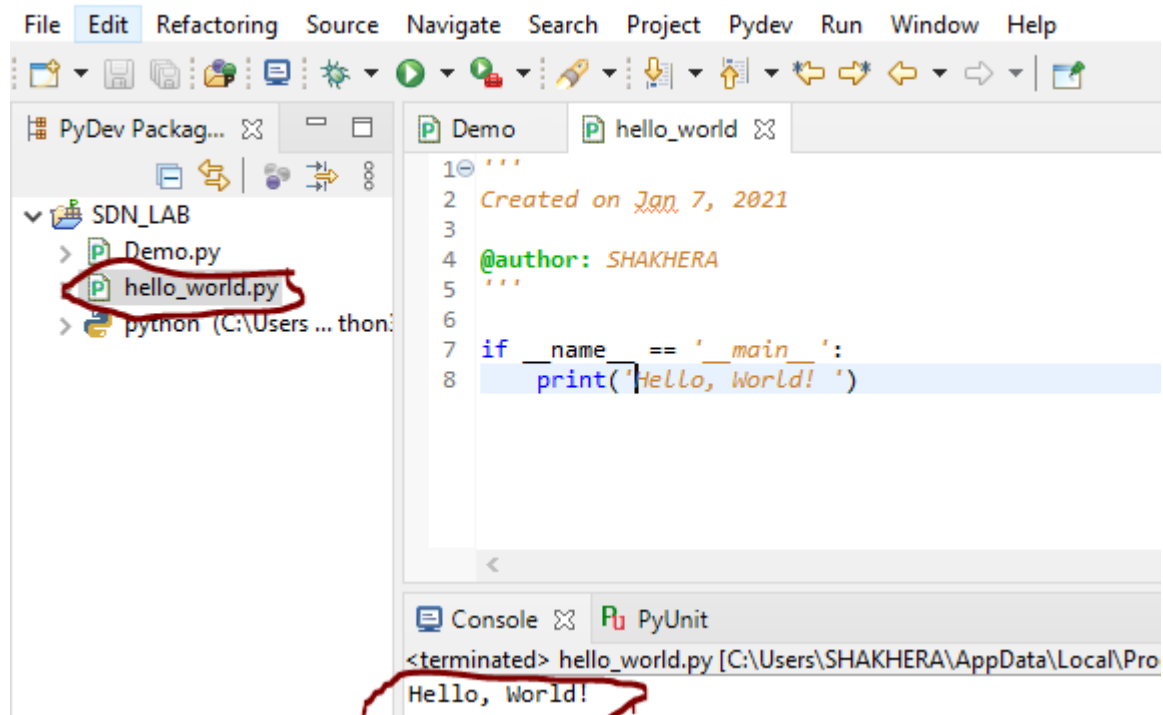
At the bottom, there are navigation buttons: a help icon (?), '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

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:

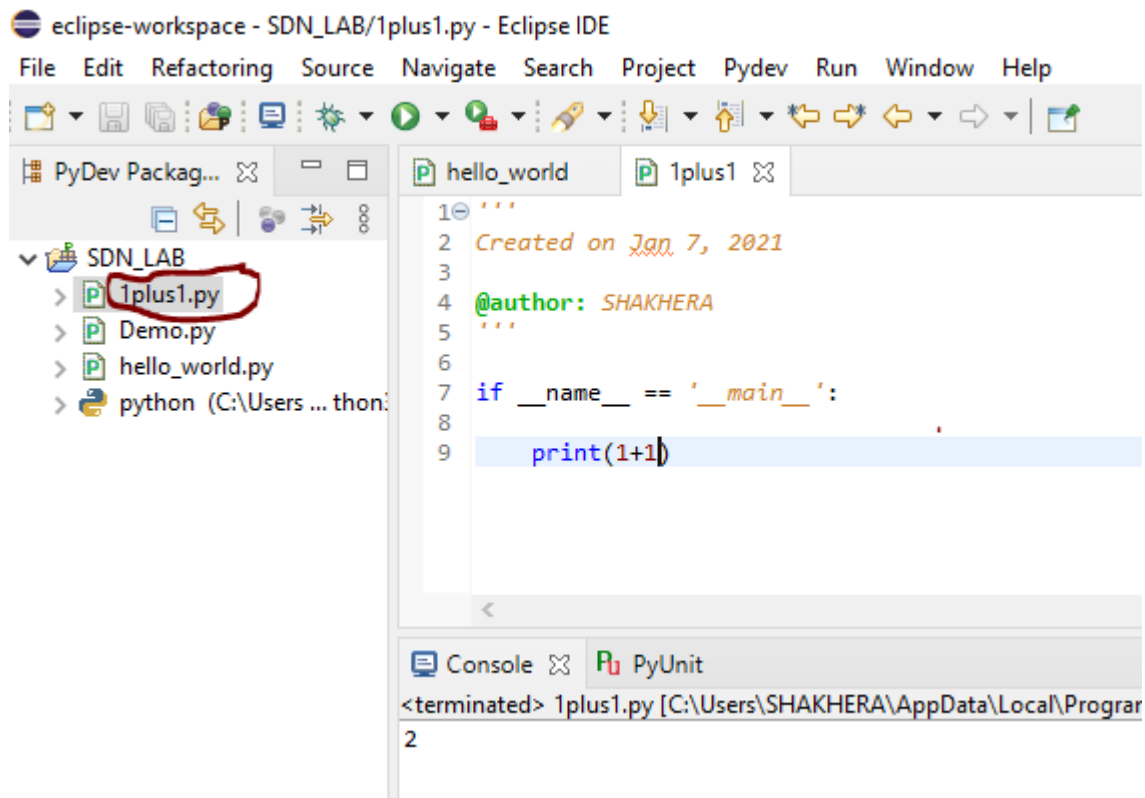


## Exercise 4.1.2: Write a Hello World

eclipse-workspace - SDN\_LAB/hello\_world.py - Eclipse IDE



### Exercise 4.1.3: Compute 1+1



### Exercise 4.1.4: Type in program text

```
hello_world 1plus1 shapes
1 '''
2 Created on Jan 7, 2021
3
4 @author: SHAKHERA
5 '''
6
7 h = 5.0 # height
8 r = 1.5 # radius
9 pi = 3.1416
10 if __name__ == '__main__':
11
12     area_parallelogram = h*r
13     print('the area of the parallelogram is %.3f' % area_parallelogram)
14
15     area_square = h**2
16     print('the area of the square is %g' % area_square)
17
18     area_circle = pi*r**2
19     print('the area of the circle is %.3f' % area_circle)
20
21     volume_cone = 1.0/3*pi*r**2*h
22     print('the volume of the cone is %.3f' % volume_cone)
```

Console PyUnit

<terminated> shapes.py [C:\Users\SHAKHERA\AppData\Local\Programs\Python\Python39\python.exe]

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 4.2.1: Create and run basic example.

hello\_world

1plus1

shapes

\*operator

```
1 '''
2 Created on Jan 7, 2021
3
4 @author: SHAKHERA
5 '''
6 if __name__ == '__main__':
7     x = int(input('Enter first num : '))
8     y = int(input('Enter second num : '))
9     plus = x+y
10    print('Plus of {0} and {1} is = {2}'.format(x,y,plus))
11    minus = x-y
12    print('Minus of {0} and {1} is = {2}'.format(x,y,minu))
13    multiply = x*y
14    print('Multiply of {0} and {1} is = {2}'.format(x,y,multiply))
15    power = x**y
16    print('Power of {0} and {1} is = {2}'.format(x,y,power))
17    divide = x/y
18    print('Divide of {0} and {1} is = {2}'.format(x,y,divide))
19    floor = x//y
20    print('Divide and Floor of {0} and {1} is = {2}'.format(x,y,floor))
21    modulo = x%y
22    print('Modulo of {0} and {1} is = {2}'.format(x,y,modulo))
23    left_shift = x<<y
24    print('Left Shift of {0} and {1} is = {2}'.format(x,y,left_shift))
25    right_shift = x>>y
26    print('Right Shift of {0} and {1} is = {2}'.format(x,y,right_shift))
27    bit_wise_and = x&y
28    print('Bit-wise AND of {0} and {1} is = {2}'.format(x,y,bit_wise_and))
29    bit_wise_or = x|y
30    print('Bit-wise OR of {0} and {1} is = {2}'.format(x,y,bit_wise_or))
31    bit_wise_xor = x^y
32    print('Bit-wise XOR of {0} and {1} is = {2}'.format(x,y,bit_wise_xor))
33
```

hello\_world1plus1shapesoperator

34less\_than = x<y

35print('Less than of {0} and {1} is = {2}'.format(x,y,less\_than))

36greater\_than = x>y

37print('Greater than of {0} and {1} is = {2}'.format(x,y,greater\_than))

38less\_or\_equal = x<=y

39print('Less than or equal to of {0} and {1} is = {2}'.format(x,y,less\_or\_equal))

40great\_or\_equal = x>=y

41print('Greater than or equal to of {0} and {1} is = {2}'.format(x,y,great\_or\_equal))

42equal = x==y

43print('Equal to of {0} and {1} is = {2}'.format(x,y,equal))

44not\_equal = x!=y

45print('Not equal to of {0} and {1} is = {2}'.format(x,y,not\_equal))

...

ConsolePyUnit

<terminated> operator.py [C:\Users\SHAKHERA\AppData\Local\Programs\Python\Python39\python.exe]

Enter first num : 5

Enter second num : 8

Plus of 5 and 8 is = 13

Minus of 5 and 8 is = -3

Multiply of 5 and 8 is = 40

Power of 5 and 8 is = 390625

Divide of 5 and 8 is = 0.625

Divide and Floor of 5 and 8 is = 0

Modulo of 5 and 8 is = 5

Left Shift of 5 and 8 is = 1280

Right Shift of 5 and 8 is = 0

Bit-wise AND of 5 and 8 is = 0

Bit-wise OR of 5 and 8 is = 13

Bit-wise XOR of 5 and 8 is = 13

Less than of 5 and 8 is = True

Greater than of 5 and 8 is = False

Less than or equal to of 5 and 8 is = True

Greater than or equal to of 5 and 8 is = False

Equal to of 5 and 8 is = False

Not equal to of 5 and 8 is = True

### Exercise 4.2.2: The if statement:

```
hello_world 1plus1 shapes operator if ✕
1 '''
2 Created on Jan 8, 2021
3
4 @author: SHAKHERA
5 '''
6
7 if __name__ == '__main__':
8     num1 = 33
9     num2 = int(input('Enter the num : '))
10
11     if(num1==num2):
12         print('Equal')
13     else:
14         print('Wrong')
15
```

Console ✕ PyUnit

```
<terminated> if.py [C:\Users\SHAKHERA\AppData\Local\Programs\Python\Python39\python.exe]
Enter the num : 34
Wrong
```

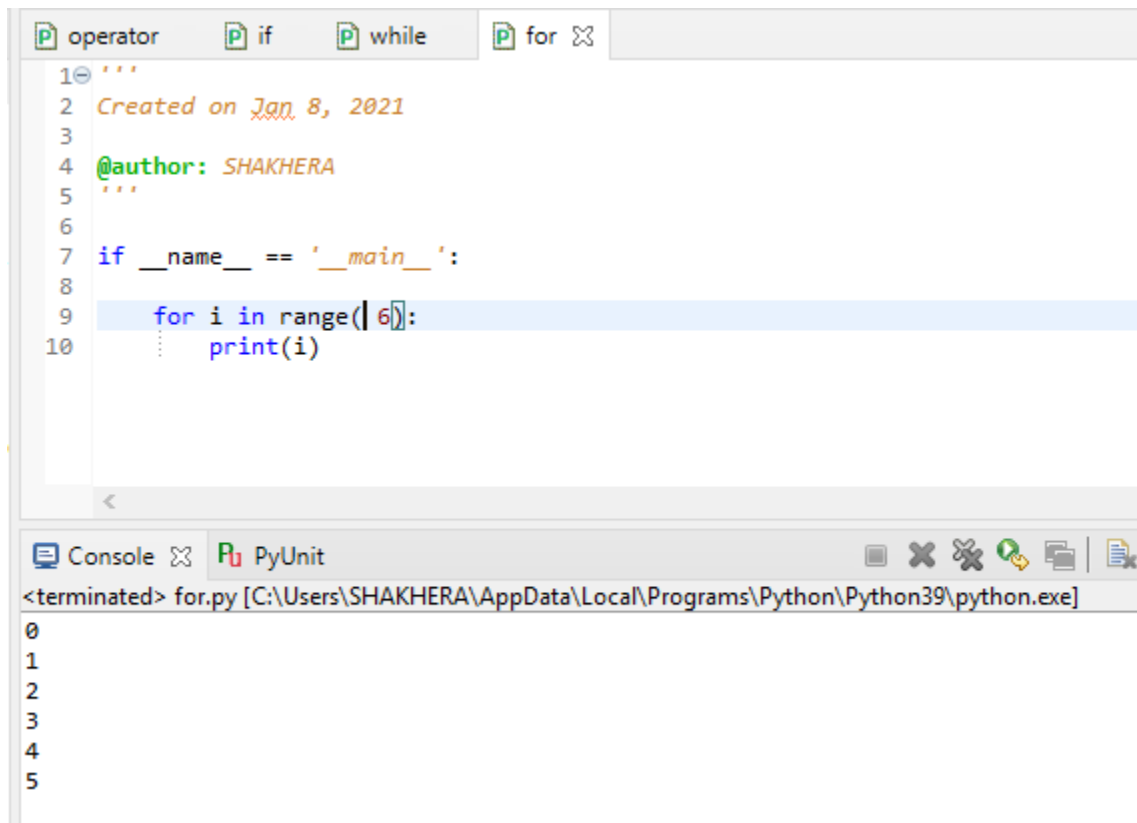
### Exercise 4.2.3: The while Statement

```
operator if while ✕
1 '''
2 Created on Jan 8, 2021
3
4 @author: SHAKHERA
5 '''
6
7 if __name__ == '__main__':
8     num1 = 33
9     num2 = int(input('Enter the num : '))
10
11     while(num1<=num2):
12         print(num1)
13         num1=num1+1
```

Console ✕ PyUnit

```
<terminated> while.py [C:\Users\SHAKHERA\AppData\Local\Programs\Python\Python39\python.exe]
Enter the num : 40
33
34
35
36
37
38
39
40
```

### Exercise 4.2.4: The for Statement



The screenshot shows the Eclipse IDE interface. The top toolbar includes buttons for 'operator', 'if', 'while', and 'for'. The main editor window displays a Python script with the following code:

```
1 """  
2 Created on Jan 8, 2021  
3  
4 @author: SHAKHERA  
5 """  
6  
7 if __name__ == '__main__':  
8  
9     for i in range(6):  
10        print(i)
```

The line `for i in range(6):` is highlighted. Below the editor, the 'Console' tab is active, showing the output of the script:

```
<terminated> for.py [C:\Users\SHAKHERA\AppData\Local\Programs\Python\Python39\python.exe]  
0  
1  
2  
3  
4  
5
```

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

**Answer:**

**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.

For python development under Eclipse you can use the PuDev Plugin which is an open source project. So, we use it for programming on python.

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

**Answer:**

#### **Features in Python**

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



### **1. Easy to code:**

Python is a very developer-friendly language which means that anyone and everyone can learn to code it in a couple of hours or days. As compared to other object-oriented programming languages like Java, C, C++, and C#, Python is one of the easiest to learn.

### **2. Open and Free Source:**

Python is an open-source programming language which means that anyone can create and contribute to its development. Python has an online forum where thousands of coders gather daily to improve this language further. Along with this python is free to download and use in any operating system, be it Windows, Mac or Linux.

### **3. Python is Portable language:**

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

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

**Answer:**

A module is a file containing Python code. Python modules can be managed using functions, classes etc.

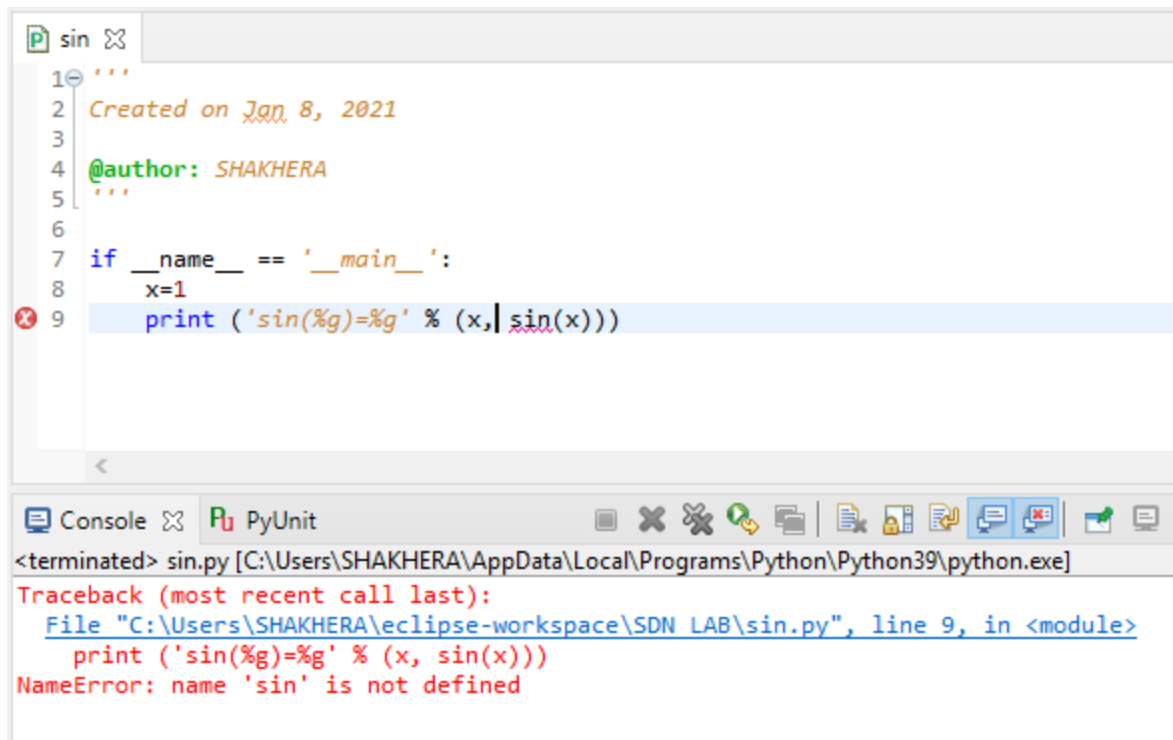
A module name is the file name with the .py extension. When we have a file called empty.py empty is the module name. The `__name__` is a variable that holds the name of the modules being executed called also the main module, has a special name: `'__main__'`. With this name it can be referenced from the Python code.

**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?**

**Answer:**



```
1 '''  
2 Created on Jan 8, 2021  
3  
4 @author: SHAKHERA  
5 '''  
6  
7 if __name__ == '__main__':  
8     x=1  
9     print ('sin(%g)=%g' % (x, sin(x)))
```

Console PyUnit  
<terminated> sin.py [C:\Users\SHAKHERA\AppData\Local\Programs\Python\Python39\python.exe]  
Traceback (most recent call last):  
File "C:\Users\SHAKHERA\workspace\SDN LAB\sin.py", line 9, in <module>  
 print ('sin(%g)=%g' % (x, sin(x)))  
NameError: name 'sin' is not defined

In this program we can see sin is not defined. To run mathematical functions such as sin, cos, tan, exp, log, etc, we must importing a module, say math, is to write,

import math

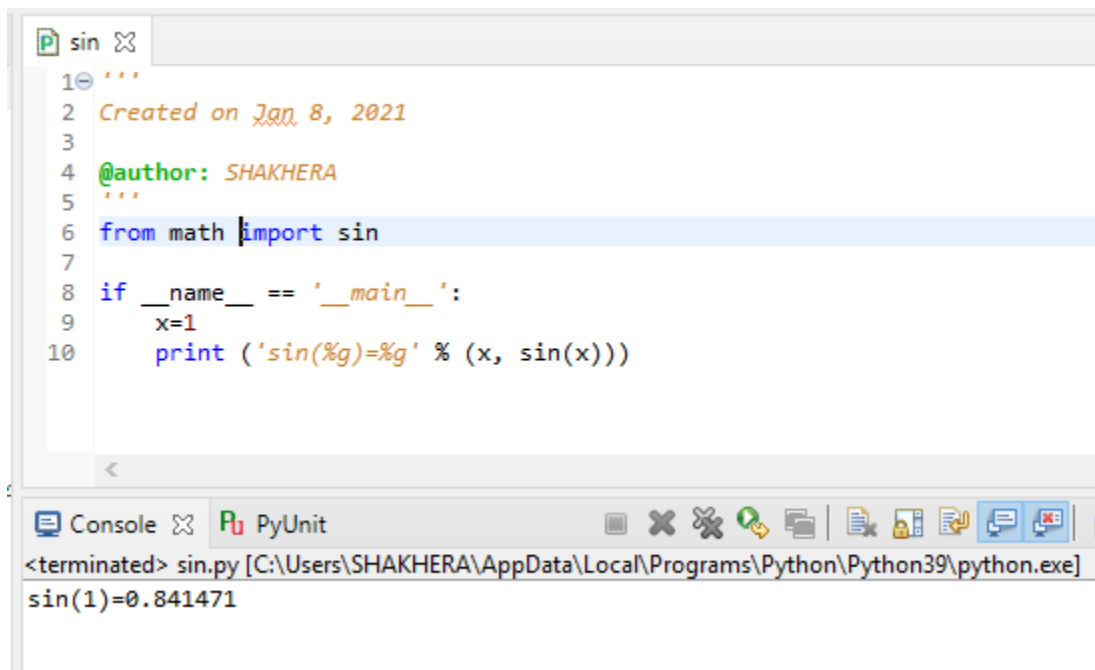
More than one function can be imported:

from math import sqrt, exp, log, sin

Sometimes one just writes,

from math import \*

So, correct code is:



```
1 '''
2 Created on Jan 8, 2021
3
4 @author: SHAKHERA
5 '''
6 from math import sin
7
8 if __name__ == '__main__':
9     x=1
10    print ('sin(%g)=%g' % (x, sin(x)))
```

Console | PyUnit

<terminated> sin.py [C:\Users\SHAKHERA\AppData\Local\Programs\Python\Python39\python.exe]  
sin(1)=0.841471

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

**Answer:**

```
simple_example ❏
2 Created on Jan 8, 2021
3
4 @author: SHAKHERA
5 """
6
7 if __name__ == '__main__':
8     num = int(input('Enter num of element : '))
9     b = []
10    for i in range(0, num):
11        a = int(input('Enter the elements : '))
12        b.append(a) # Adding the elements
13    sum1 = 0
14    sum2 = 0
15    sum3 = 0
16    for j in b:
17        if(j>0):
18            if(j%2==0):
19                sum1=sum1+j
20            else:
21                sum2 =sum2+j
22        else:
23            sum3 = (sum3+j)*(-1)
24    print('sum of all positive even numbers: ',sum1)
25    print('sum of all positive odd numbers: ',sum2)
26    print('sum of all negative numbers without sign : ',sum3)
27
<
Console ❏ PyUnit
<terminated> simple_example.py [C:\Users\SHAKHERA\AppData\Local\Programs\Python\Python39\python.exe]
Enter num of element : 4
Enter the elements : -12
Enter the elements : 33
Enter the elements : 68
Enter the elements : 13
sum of all positive even numbers: 68
sum of all positive odd numbers: 46
sum of all negative numbers without sign : 12
```

**Discussion:** Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming language. Python runs on an interpreter system, meaning that code can be executed as soon as it is written.

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.

In this lab, we can learn setup the python in eclipse, and can execute a python code successfully.

