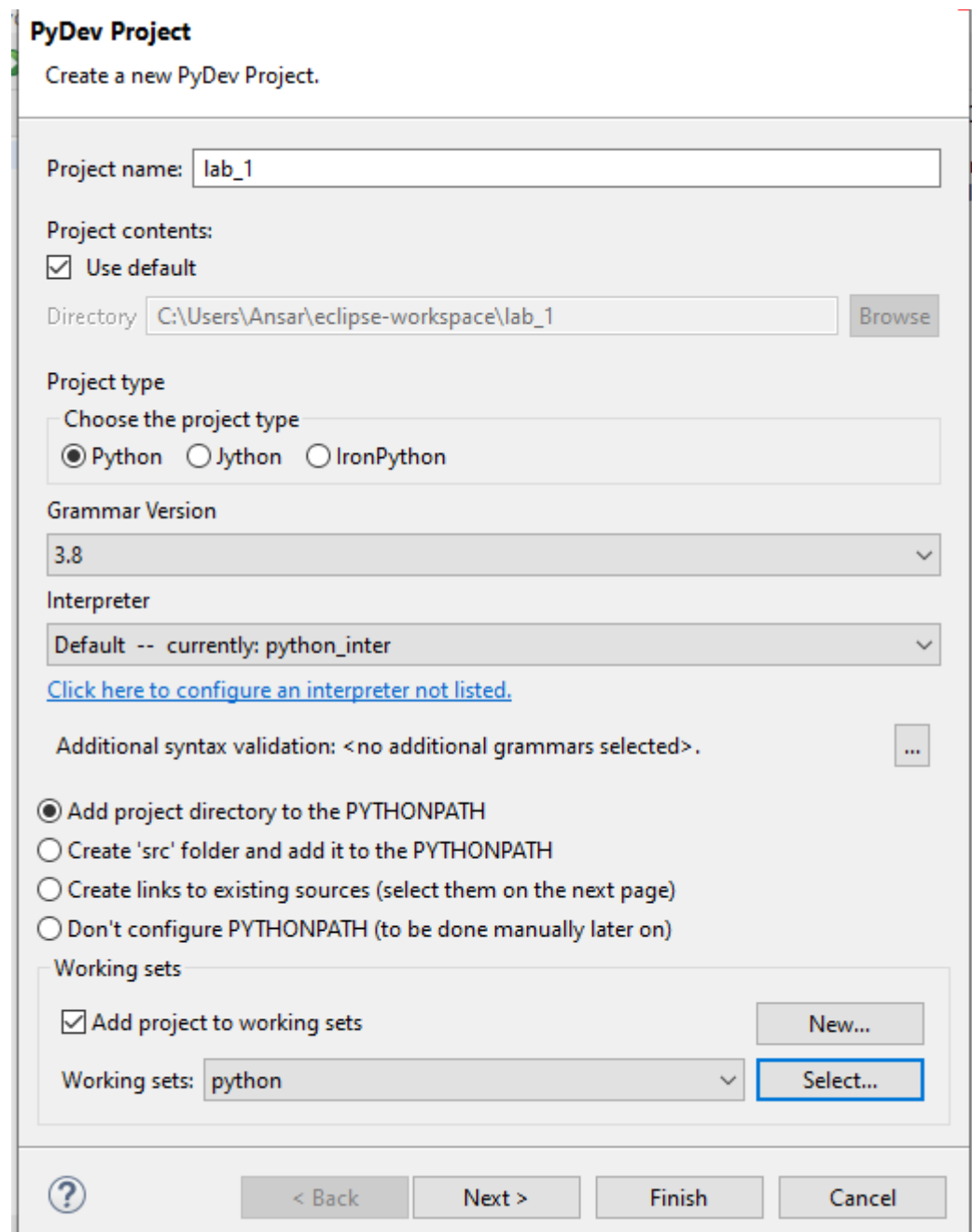


Exercise 4.1.1 : Create a python project, click in File > New > PyDev Project. Provide a name for the project (LAB_1 for lab), then select the version of python to be used and select to add the project to working set as shown below:



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. The 'Directory' field shows 'C:\Users\Ansar\eclipse-workspace\lab_1' with a 'Browse' button. Under 'Project type', 'Python' is selected. The 'Grammar Version' is set to '3.8'. The 'Interpreter' is set to 'Default -- currently: python_inter'. There is a link to 'Click here to configure an interpreter not listed.' Under 'Additional syntax validation', it says '<no additional grammars selected>'. For 'PYTHONPATH', 'Add project directory to the PYTHONPATH' is selected. Under 'Working sets', the 'Add project to working sets' checkbox is checked, and the 'Working sets' dropdown is set to 'python'. There are 'New...' and 'Select...' buttons. 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:

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

<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 @author: Nadira  
5 '''  
6 a=1+1  
7 print(a)
```

The console output shows the command prompt running the script:

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

Exercise 4.1.4: Type in program text

Ans:

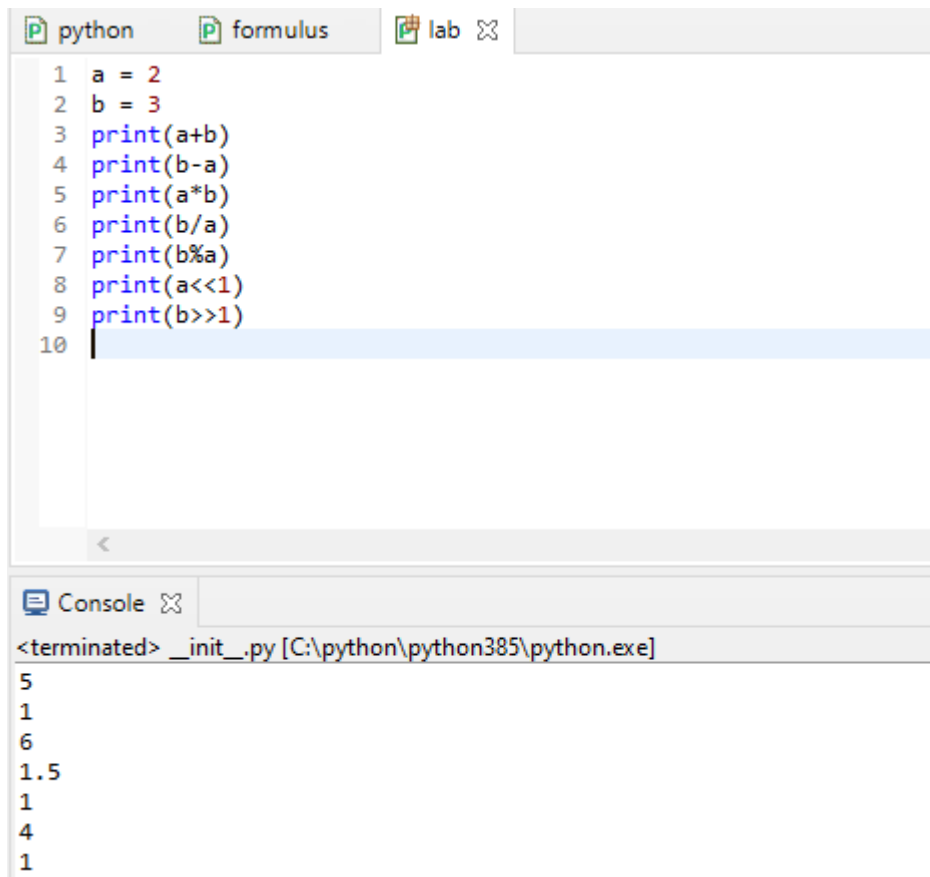
The screenshot shows a Python IDE with a file named 'formulus.py'. The code in the editor is as follows:

```
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)
```

The console output shows the command prompt running the script, followed by a traceback:

```
<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\pydev\imps\pydev_execfile.py", line 25, in  
    exec(compile(contents+"\n", file, 'exec'), glob, loc)  
  File "C:\Users\Ansar\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.



The screenshot shows a Python IDE with two tabs: 'python' and 'formulus'. The 'python' tab is active and contains the following code:

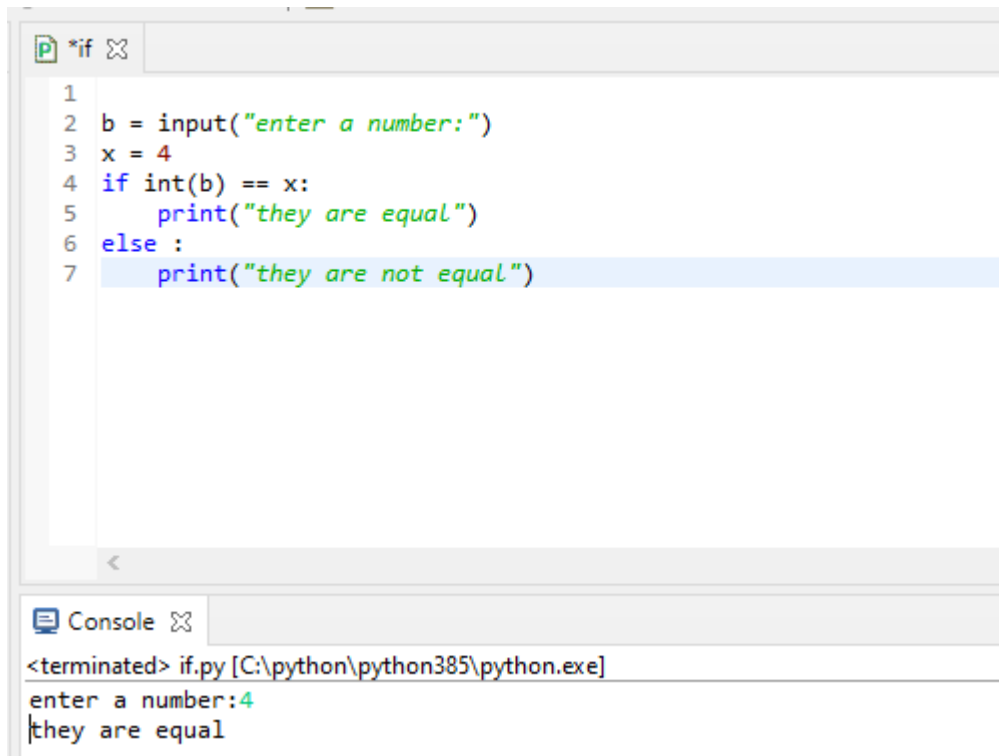
```
1 a = 2
2 b = 3
3 print(a+b)
4 print(b-a)
5 print(a*b)
6 print(b/a)
7 print(b%a)
8 print(a<<1)
9 print(b>>1)
10
```

Below the code editor is a 'Console' tab showing the output of the script:

```
<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



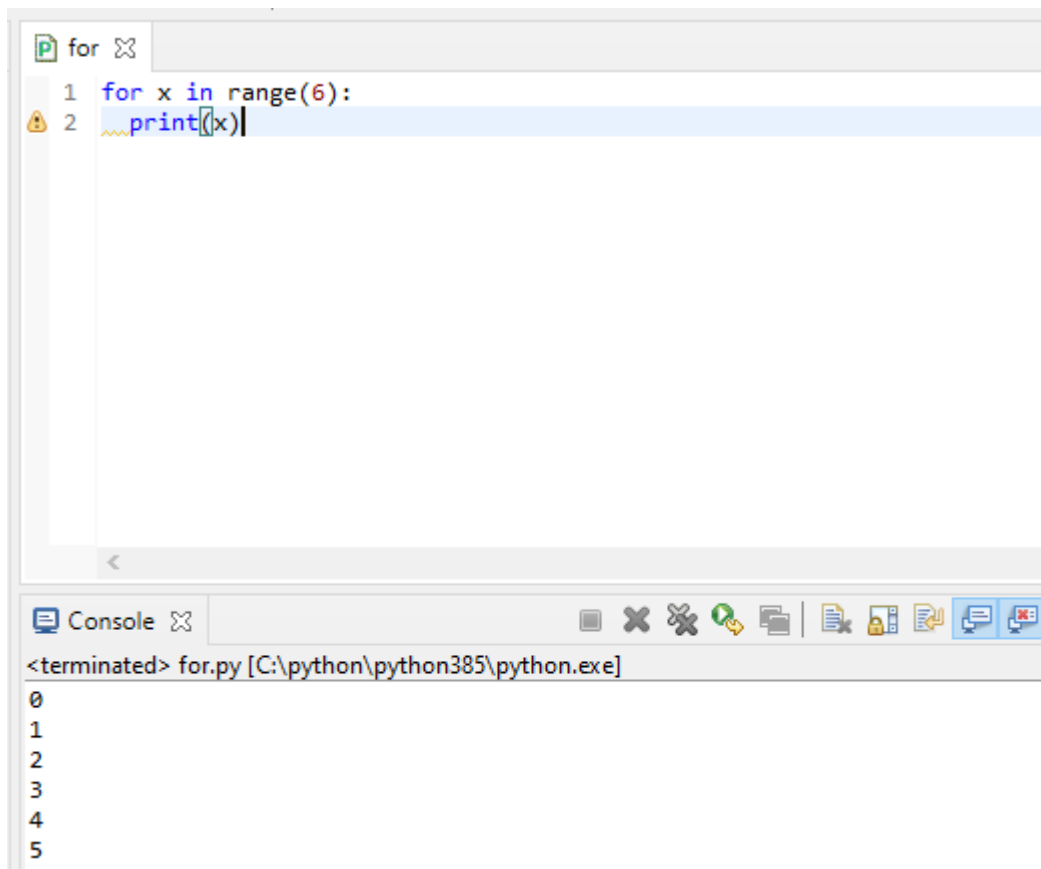
```
1
2 b = input("enter a number:")
3 x = 4
4 if int(b) == x:
5     print("they are equal")
6 else :
7     print("they are not equal")
```

<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 interface. The top editor window, titled 'for.py', contains the following Python code:

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

The bottom console window 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 object-oriented 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?

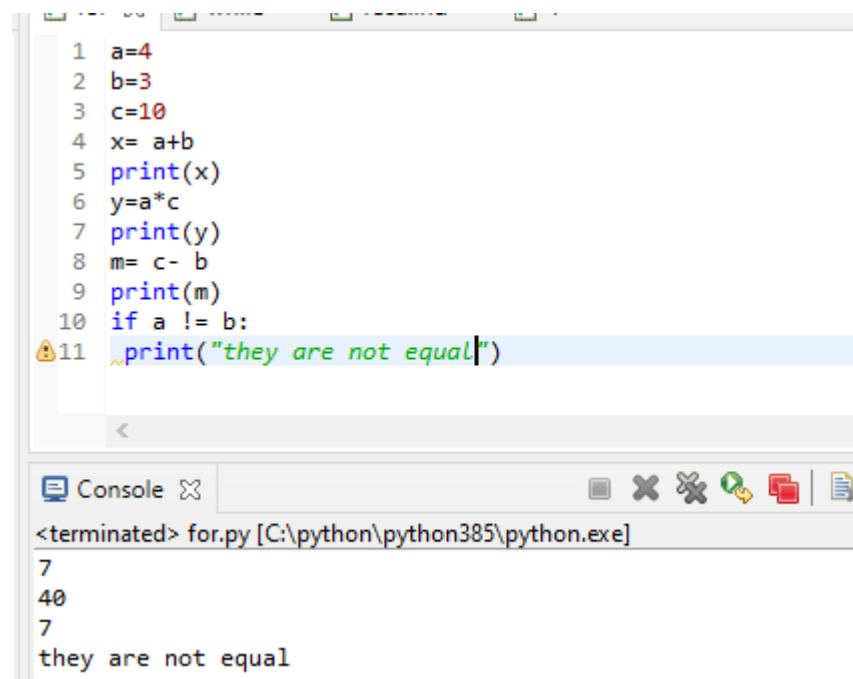
Ans: A package is a collection of Python modules: while a module is a single Python file, a package is a directory of Python modules containing an additional `__init__.py` file, to distinguish a package from a directory that just happens to contain a bunch of Python scripts.

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 Output:

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

