Zhihua Shang

05/25/2020

Fundations of Programming(python)

Assignment06

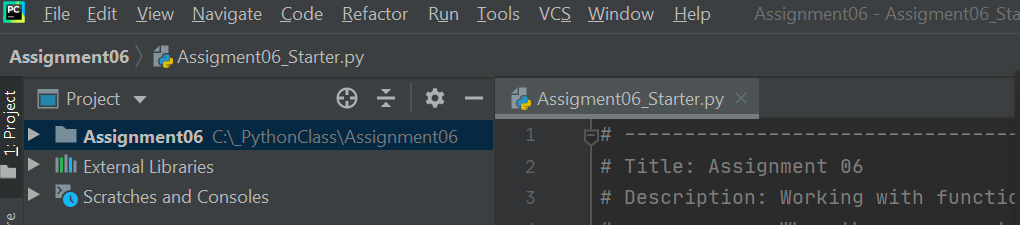
Work with Functions

# Introduction

Python is a very powerful tool to deal with data. However, it is very unorganized to write a long script. Function is a very useful tool to group and one and more statement. By using it, python scripts will look much professional and easier to read.

# Create a folder and Open the downloaded file

Open file via PyCharm. Click create new project, name it as Assignment06, and save it under \_PythonClass in C: drive, then open and find the location where you have downloaded the Assginment06\_Starter.py (Figure 1).



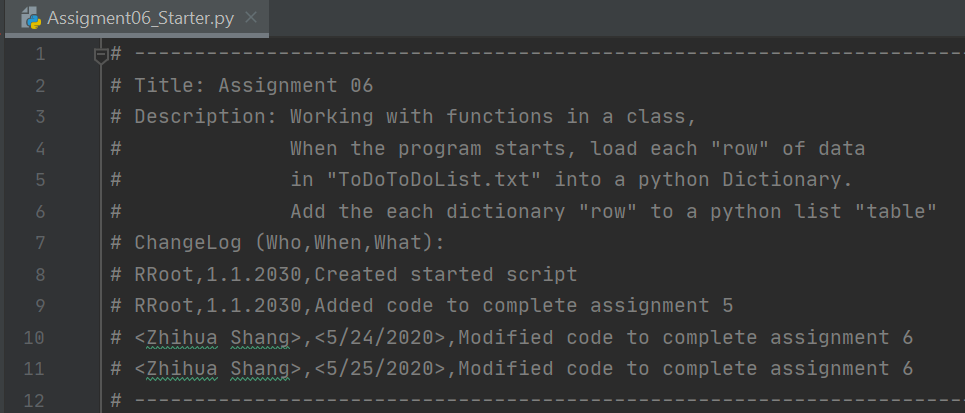
*Figure 1: Showing the location of Assginment06\_Starter.py*

# Coding

This is the main and the most important part of the whole process because it would not work if the codes are wrong. The first step is to read the existing code and check where the codes should be updated, then fix it. This part shows the steps and some pictures to demonstrate the results.

## Update the changelog in the header

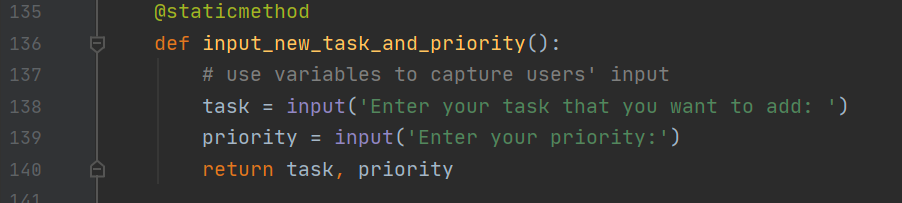
A header is very important for both the creators and other users. It provides some basic information about this script. For example, the title, the creator, and the created date. Some codes were update each time in this script, so the changes notification could be added to the header (Figure 2).



*Figure 2: Showing the updated changelog of file.*

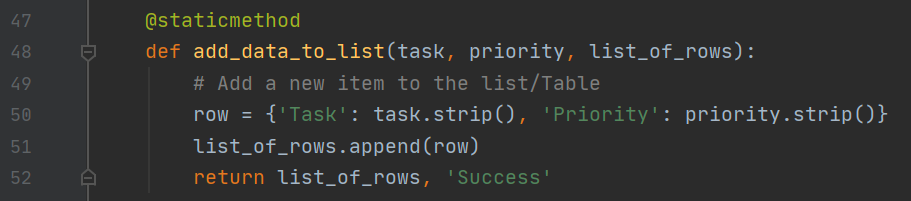
## Add a new Task

No data can be updated in steps 1 to 3. In step 4, there are several choices need to be update. The first one is to add a new Task. The first step is to capture users’ input. In this scrip, the variables are defined in a function which named as input\_new\_task\_and\_priority in the class of IO (Figure 3).



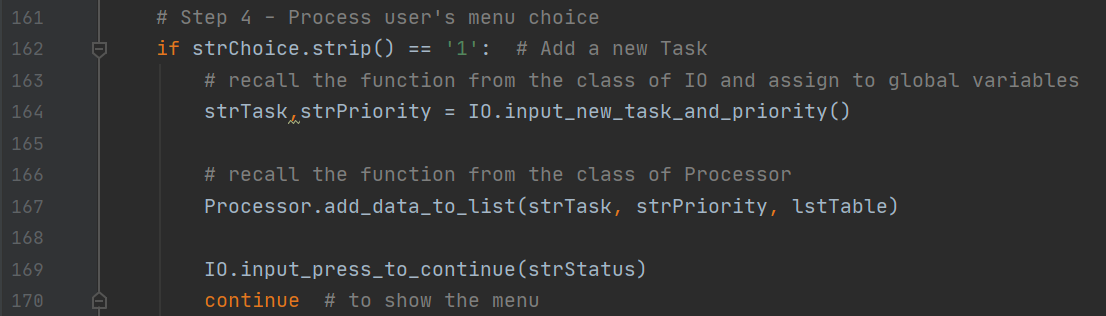
*Figure 3:* *Showing the input function of file.*

The second step is to update the function of add\_data\_to\_list in the class of Processor (Figure 4).



*Figure 4: Showing the function of add\_data\_to\_list().*

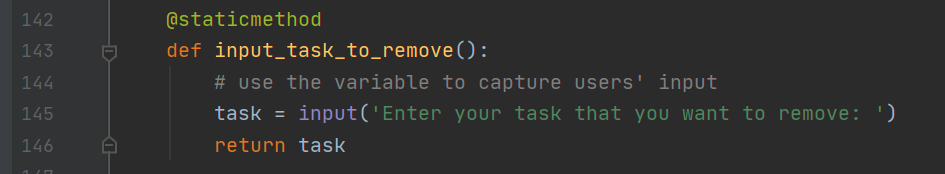
The third step is to recall the function in the main body. By using the global variables to get the value from input function and pass the value to the add function (Figure 5).



*Figure 5: Showing the code of Add a new Task part.*

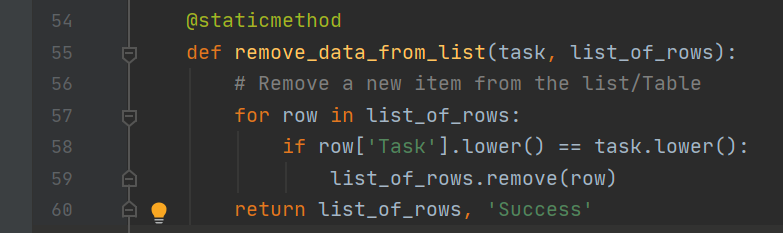
## Remove an existing task

Same as adding a new task, the first step is to capture users’ input. In this scrip, the variables are defined in a function which named as input\_task\_to\_remove in the class of IO (Figure 6).



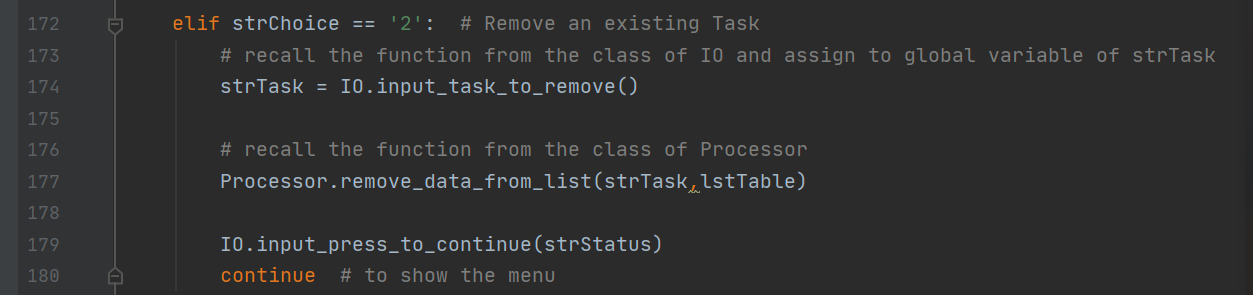
*Figure 6: Showing the function of input\_task\_to\_remove().*

The second step is to update the function of remove\_data\_from\_list in the class of Processor (Figure 7).



*Figure 7: Showing the function of remove\_data\_from\_list().*

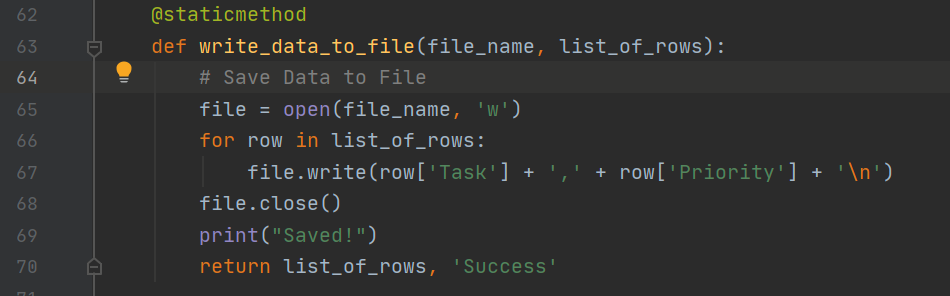
The third step is to recall the function in the main body. By using the global variables to get the value from input function and pass the value to the add function (Figure 8).



*Figure 8: Showing the code of removing an existing task part.*

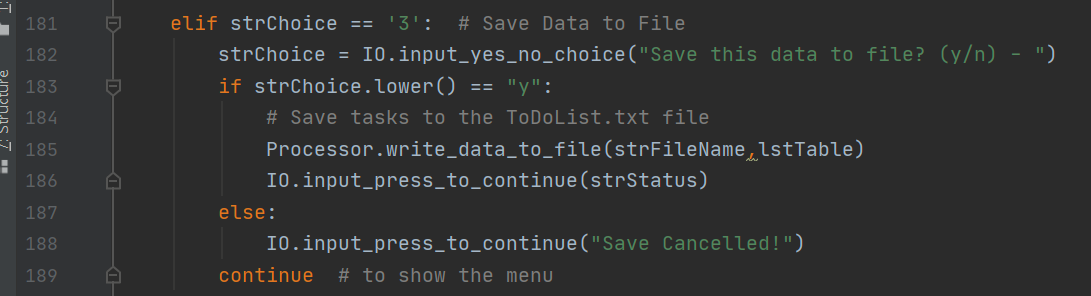
## Save data to file

This step is to save the new data into the file. It means that open the file and write the data to the file (Figure 9).



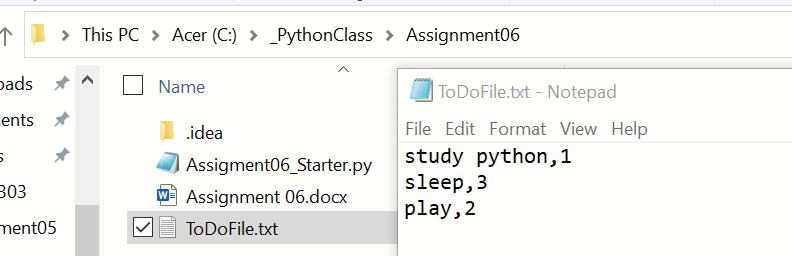
*Figure 9: Showing the function of write\_date\_to\_file().*

After finished the function, call it at the main body (Figure 10).



*Figure 10: Showing the code of saving data to file part.*

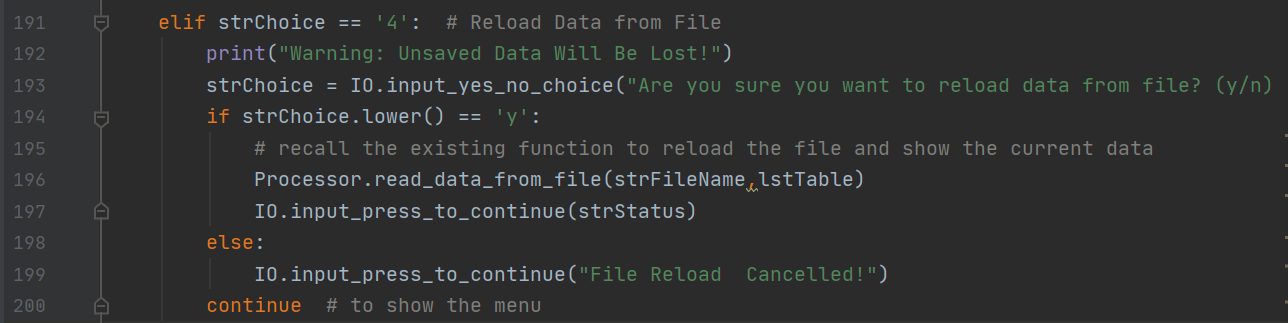
If run the scrip after adding or deleting the data, the update information will be shown in the text file (Figure 11).



*Figure 11: Showing the code and location of text file.*

## Reload data from file

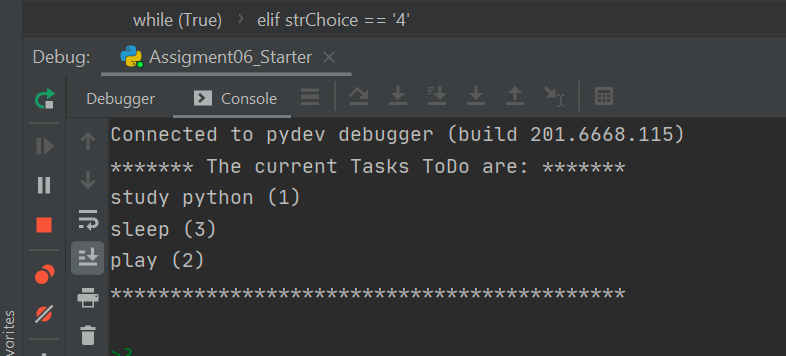
This step is to read the file. If the new information is saved, the new information is included in the file. If new information is not saved, it will not show up. To finish this step, we just simple recall the function of read\_data\_from\_file from the class of Processor (Figure 12).



*Figure 12: Showing the code of reloading data from file part.*

## Debugger in python

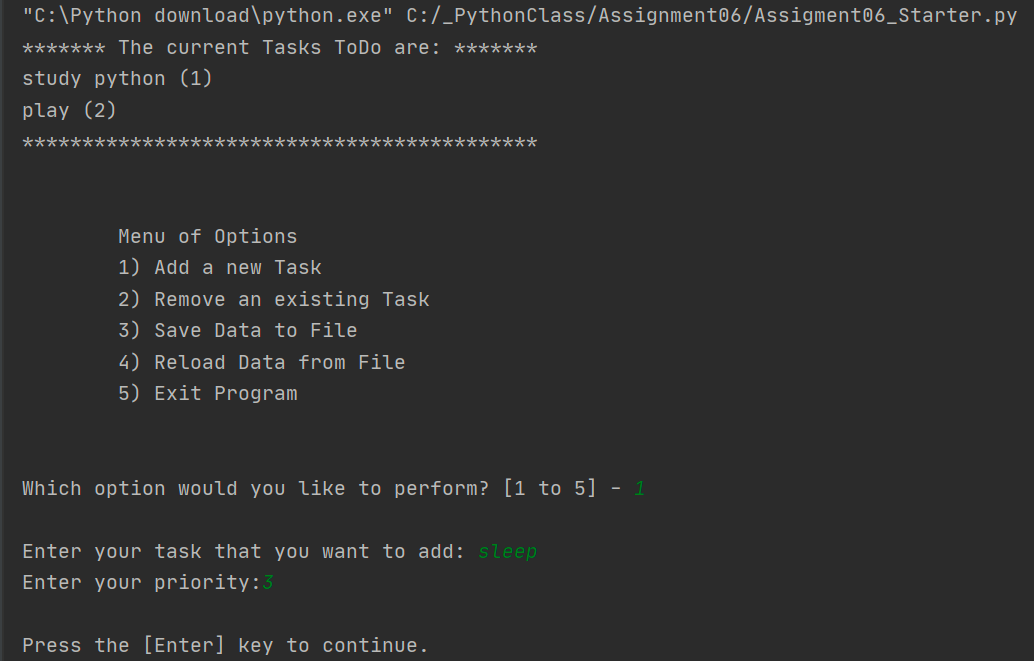
Debugger can be used to find and fix bug much easier during editing the codes by separating codes into small sets. Just right click the mouse and click debug (Figure 13).

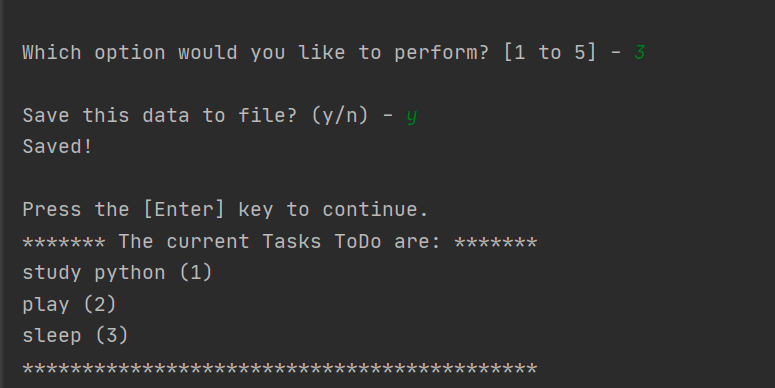


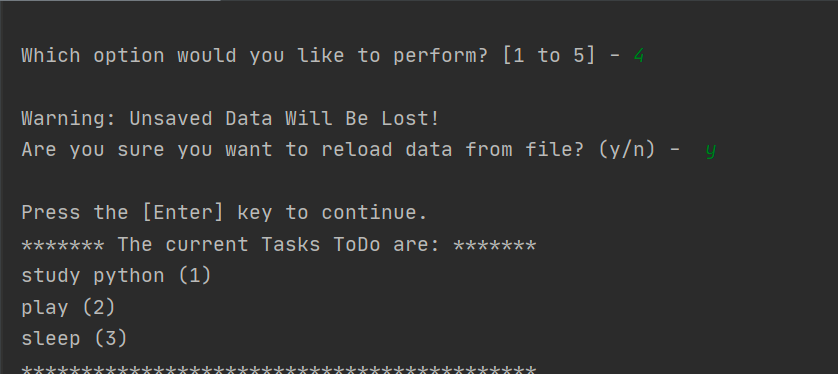
*Figure 13: Showing the debugger in PyCharm.*

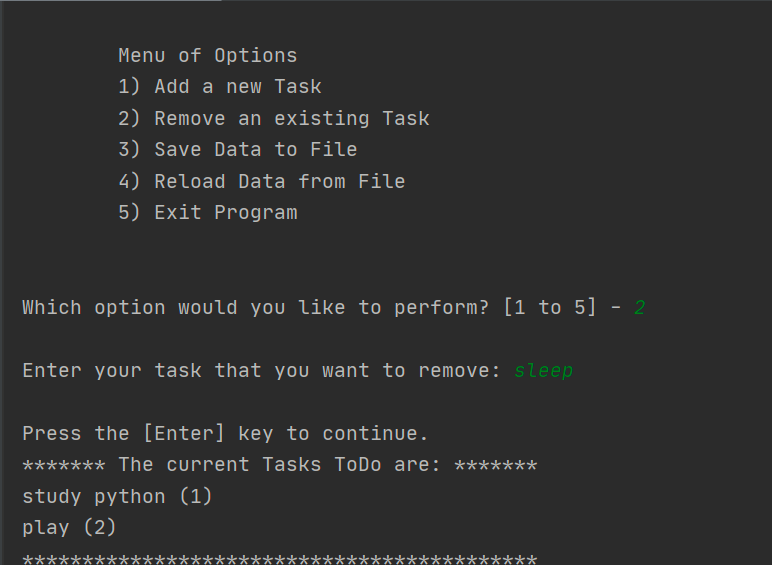
## Demonstrate in PyCharm

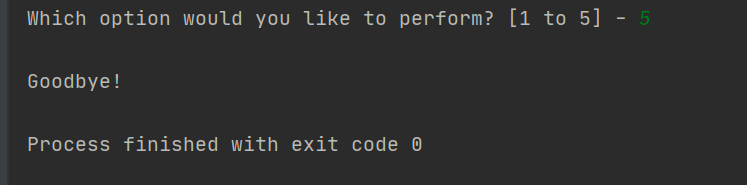
This is the result that showing in PyCharm (Figure 14).







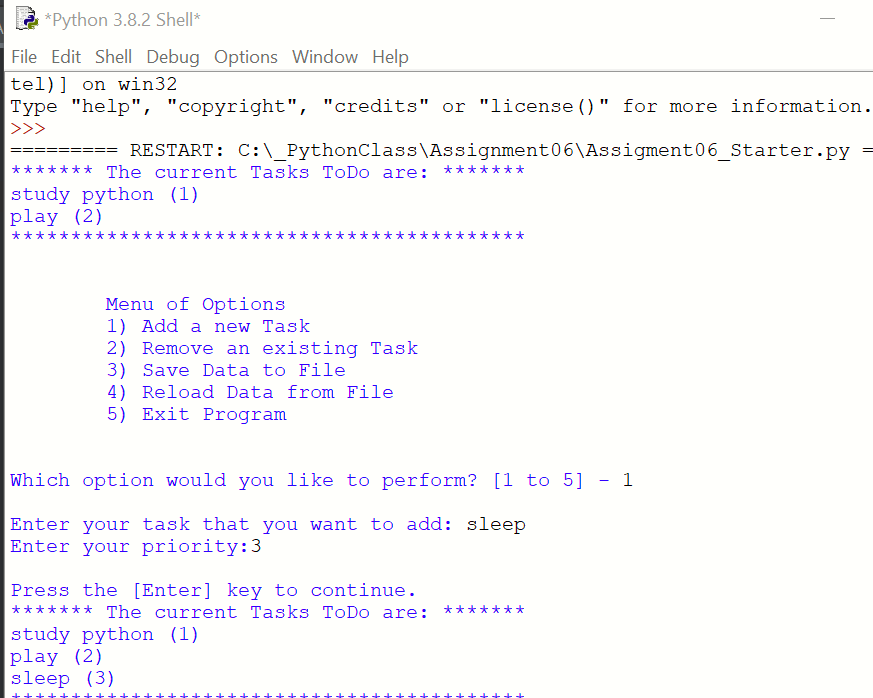


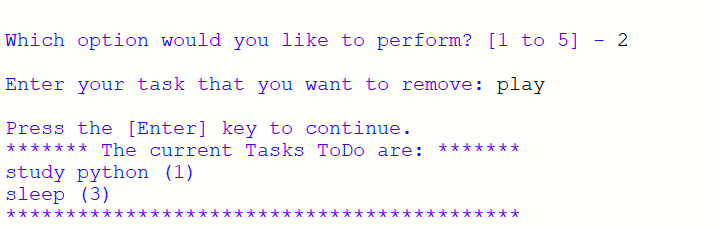


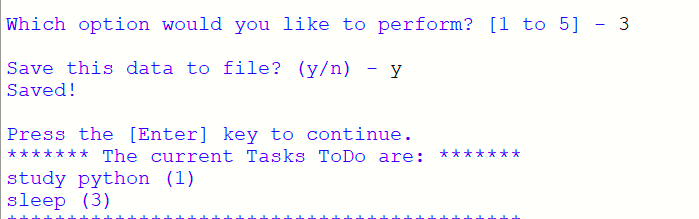
*Figure14: Showing the result in PyCharm.*

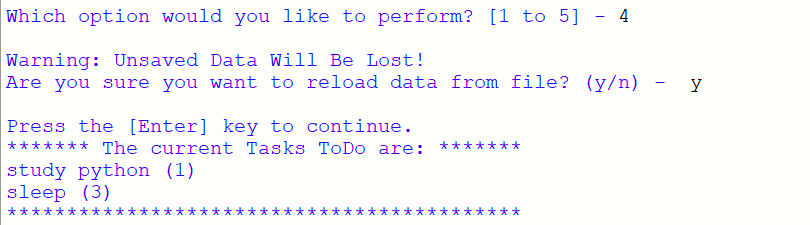
## Demonstrate in Shell window

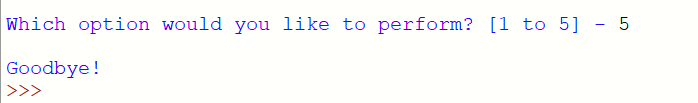
Beside working in PyCharm, the same script could also be demonstrated in IDEL shell. Go to start, search and click IDEL. The shell window shows up. Find file and open from the Assignment05 which is under\_PythonClass in C drive. Run it, and the same result will be shown (Figure 15).











*Figure 15: script running from Shell window*

# Summary

The functions are very helpful to organize the data in python. It is not only to make the information easier to read but also it is a way to split jobs in a team. For example, when a project is assigned, managers can divide the whole project into many functions and assigned different functions to different employees. After all of the functions are completed, they will be pooled and connected together. In this way, the work efficiency will be greatly improved at the workplace.