

Kate Beverly

August 25, 2022

Foundations of Programming: Python

Assignment 07

<https://github.com/kbev12/IntroToProg-Python-Mod07>

To Do Script

Introduction

For assignment 07 for the class Foundations of Programming: Python I worked on a python script that pickles a file, loads and prints the unpickled file, and ends the program.

Creating the Script

Pickling

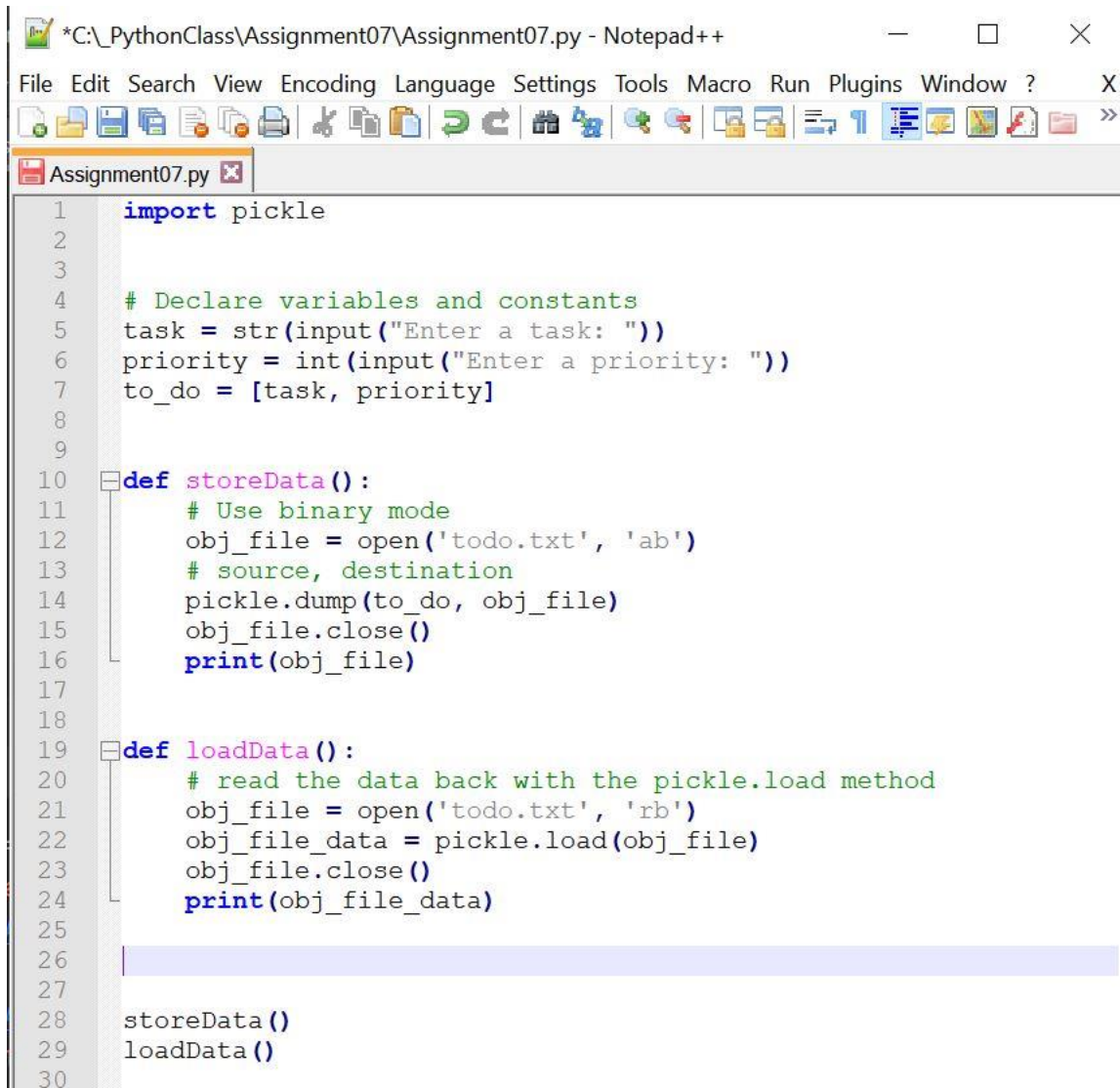
The script begins with importing the pickle module and declaring the variables (**Figure 1**).

```
import pickle

# Declare variables and constants
task = str(input("Enter a task: "))
priority = int(input("Enter a priority: "))
to_do = [task, priority]
```

Figure 1. Importing pickle module and declaring the program variables

I started creating the script (Figure 2) by taking a string input for a task, int input for the priority, and assigning them to a dictionary. I created a function to append to 'to_do.txt in binary. Using pickle.dump the to_do object is serialized to obj_file. The function loadData unpickles todo.txt by opening the file in read binary and loading the file to the variable obj_file_data.

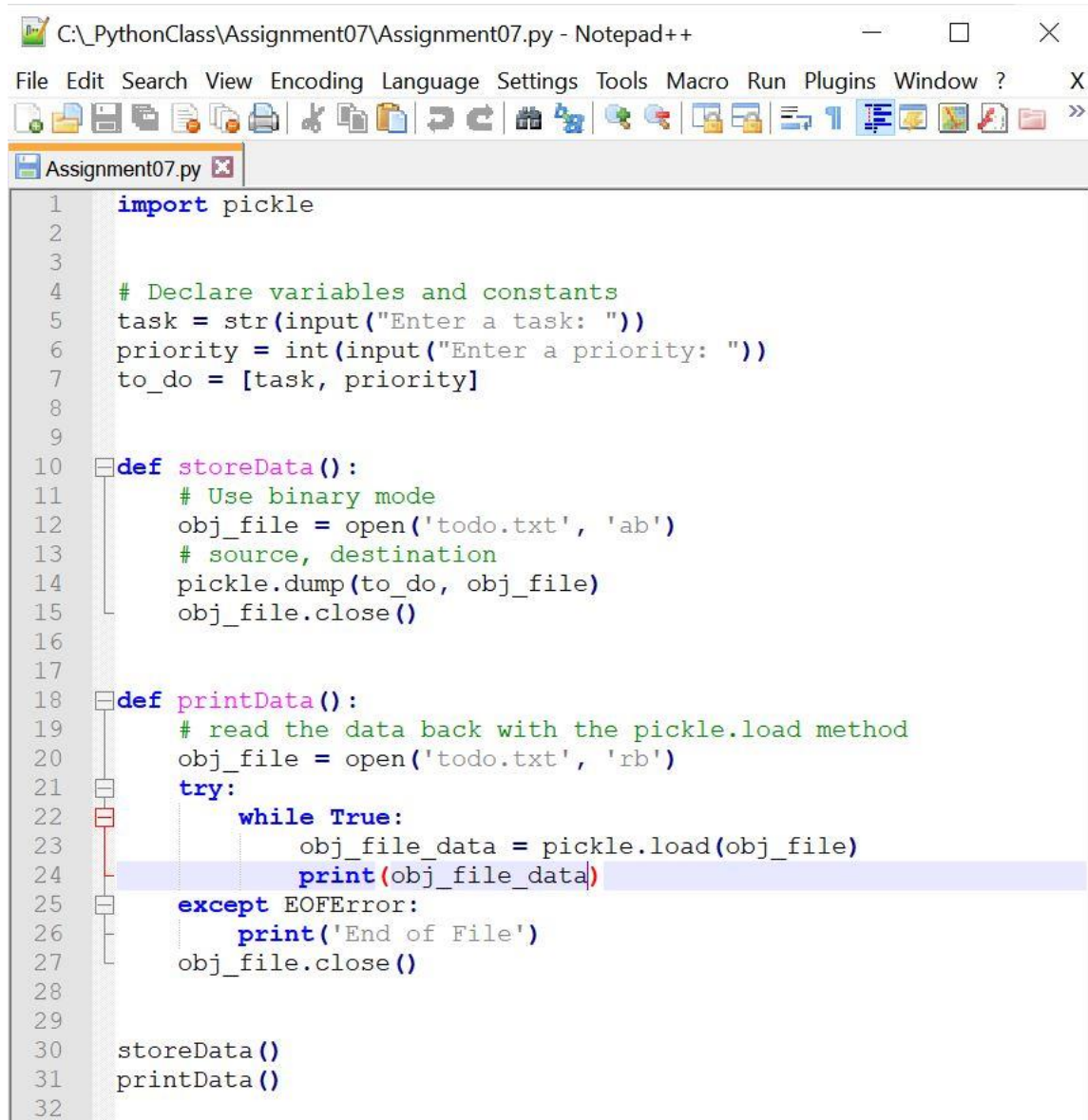
The image shows a Notepad++ window with the title bar "*C:_PythonClass\Assignment07\Assignment07.py - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar contains various icons for file operations and editing. The editor shows a Python script with line numbers 1 through 30. The script imports the pickle module, prompts the user for a task and priority, stores the data in a file named 'todo.txt', and provides functions to load the data back. The script is as follows:

```
1  import pickle
2
3
4  # Declare variables and constants
5  task = str(input("Enter a task: "))
6  priority = int(input("Enter a priority: "))
7  to_do = [task, priority]
8
9
10 def storeData():
11     # Use binary mode
12     obj_file = open('todo.txt', 'ab')
13     # source, destination
14     pickle.dump(to_do, obj_file)
15     obj_file.close()
16     print(obj_file)
17
18
19 def loadData():
20     # read the data back with the pickle.load method
21     obj_file = open('todo.txt', 'rb')
22     obj_file_data = pickle.load(obj_file)
23     obj_file.close()
24     print(obj_file_data)
25
26
27
28 storeData()
29 loadData()
30
```

Figure 2. Pickling and unpickling todo.txt

Exceptions

To enhance the script I updated the loadData function to printData function to add in a try/except (Figure 3). The function still opens todo.txt in read binary but now it loads each line and prints it to the terminal. Except once it receives an End of File error it prints 'End of File' and then closes the file.



The image shows a Notepad++ window titled "C:_PythonClass\Assignment07\Assignment07.py - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar contains various icons for file operations and editing. The script "Assignment07.py" is open, showing the following Python code:

```
1  import pickle
2
3
4  # Declare variables and constants
5  task = str(input("Enter a task: "))
6  priority = int(input("Enter a priority: "))
7  to_do = [task, priority]
8
9
10 def storeData():
11     # Use binary mode
12     obj_file = open('todo.txt', 'ab')
13     # source, destination
14     pickle.dump(to_do, obj_file)
15     obj_file.close()
16
17
18 def printData():
19     # read the data back with the pickle.load method
20     obj_file = open('todo.txt', 'rb')
21     try:
22         while True:
23             obj_file_data = pickle.load(obj_file)
24             print(obj_file_data)
25     except EOFError:
26         print('End of File')
27     obj_file.close()
28
29
30 storeData()
31 printData()
32
```

End of Script

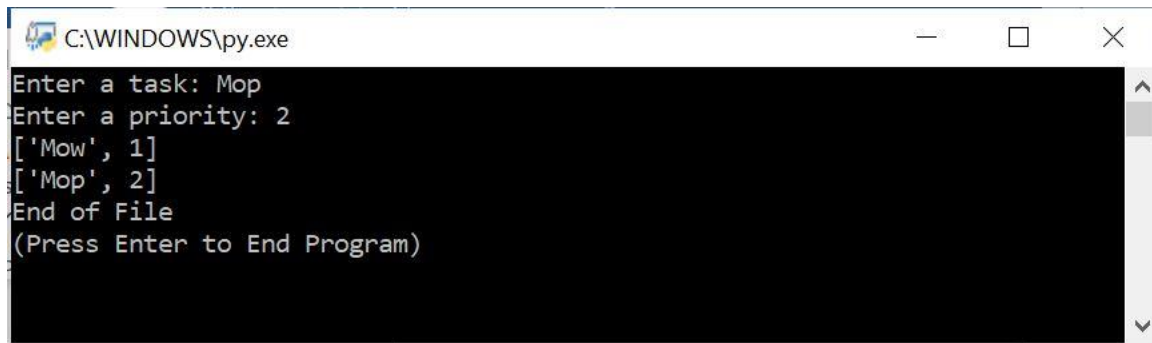
The script ends by asking the user to press enter to end the program. (Figure 4) Each of the functions are called in order.

```
12 # Declare variables and constants
13 task = str(input("Enter a task: "))
14 priority = int(input("Enter a priority: "))
15 to_do = [task, priority]
16
17
18 def storeData():
19     # Use binary mode
20     obj_file = open('todo.txt', 'ab')
21     # source, destination
22     pickle.dump(to_do, obj_file)
23     obj_file.close()
24
25
26 def printData():
27     # read the data back with the pickle.load method
28     obj_file = open('todo.txt', 'rb')
29     try:
30         while True:
31             obj_file_data = pickle.load(obj_file)
32             print(obj_file_data)
33     except EOFError:
34         print('End of File')
35     obj_file.close()
36
37
38 def endProgram():
39     #End program
40     input('(Press Enter to End Program)')
41
42
43 storeData()
44 printData()
45 endProgram()
46
```

Figure 4 Program ends with user's input

Testing the Script

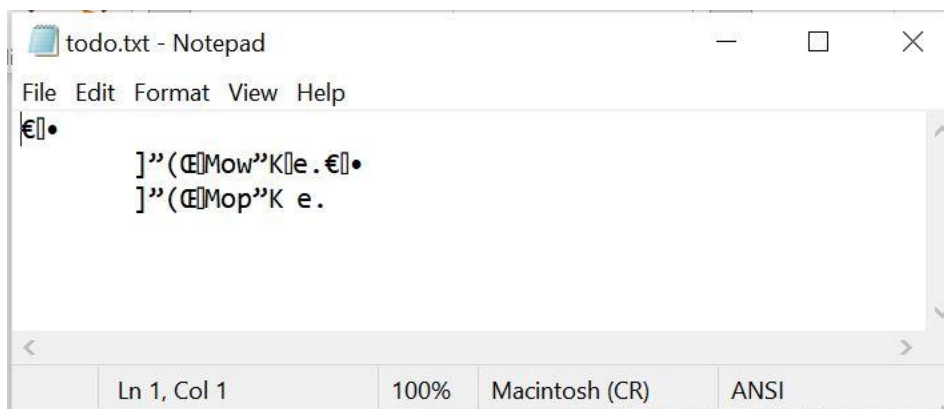
I tested the program in both PyCharm and in the command prompt and it successfully completed using both (**Figure 5**).



```
C:\WINDOWS\py.exe
Enter a task: Mop
Enter a priority: 2
['Mow', 1]
['Mop', 2]
End of File
(Press Enter to End Program)
```

Figure 5. Testing the program

The script successfully ran and updated the text file todo.txt (**Figure 6**).



```
todo.txt - Notepad
File Edit Format View Help
€•
]”(Mow”K e. €•
]”(Mop”K e.
```

Figure 6. Updated todo.txt

Summary

For Foundations of Programming: Python seventh assignment I worked on a python script that pickles a file, loads and prints the unpickled file, and ends the program.