Samuel Nunn

May 29, 2020

Foundations of Programming: Python

Assignment 07

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

Working with Exception Handling

and Pickling in Python

# Introduction and Assumptions

In this assignment, we modify the code from Assignment 06 into Assignment 07, so that it stores the ToDoFile as a .dat binary file (pickling). We also add some advanced Exception Handling to handle scenarios where the .dat file is missing or not setup properly.

Part of this assignment was researching websites which would help us with Exception Handling and Picking. Below were the useful websites I found, and why I found them useful.

Useful website for learning about Exception Handling:

<https://stackoverflow.com/questions/438894/how-do-i-stop-a-program-when-an-exception-is-raised-in-python>

The website above was useful in learning how to exit the program after the exception was raised. First, we import “sys”, then we use sys.exit() in the appropriate place.

Useful websites for learning about Pickling:

<https://www.geeksforgeeks.org/understanding-python-pickling-example/>

The website above was useful in learning more about the advantages of using the Pickle Module, specifically: Recursive Objects, Object Sharing, and User-defined classes.

# Instructions

1. Start a new PyCharm project called “Assignment07” within the C:\\_PythonClass\ directory. Create a Header section with your information, and create a section for importing Modules, as seen in Figure 1.

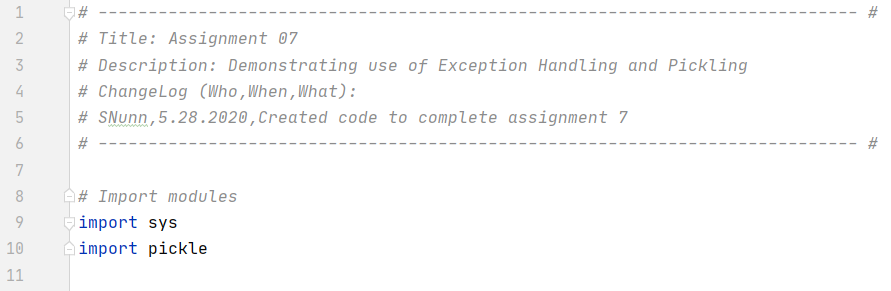


Figure . Script Title, Header Section, and Import Modules

1. In the “Data” section, declare your variables and constants, as seen in Figure 2.

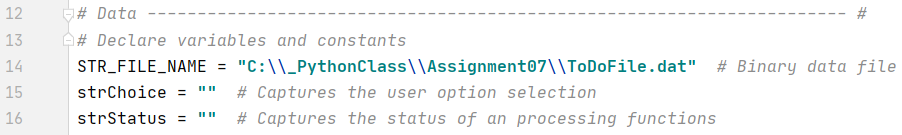


Figure . – Data. Declare variables and constants.

1. In the “Processing” section, create a “class” called “Processor”. Then, indent and create all the “Processor” class functions., as seen in Figure 3. The “class” allows us to better organize our functions. Adapt the code from Assignment 6 to work with Exception Handling and Pickling, as highlighted in Figure 3. Exception Handing allows us to handle for particular (or general) error messages. As can be seen, I use sys.exit() to stop the program after informing the user of what they should do. The highlighted pickle.load(file) command loads the pickled .dat file into a list called fileData that can be read by Python and the user, as seen in Figure 3.

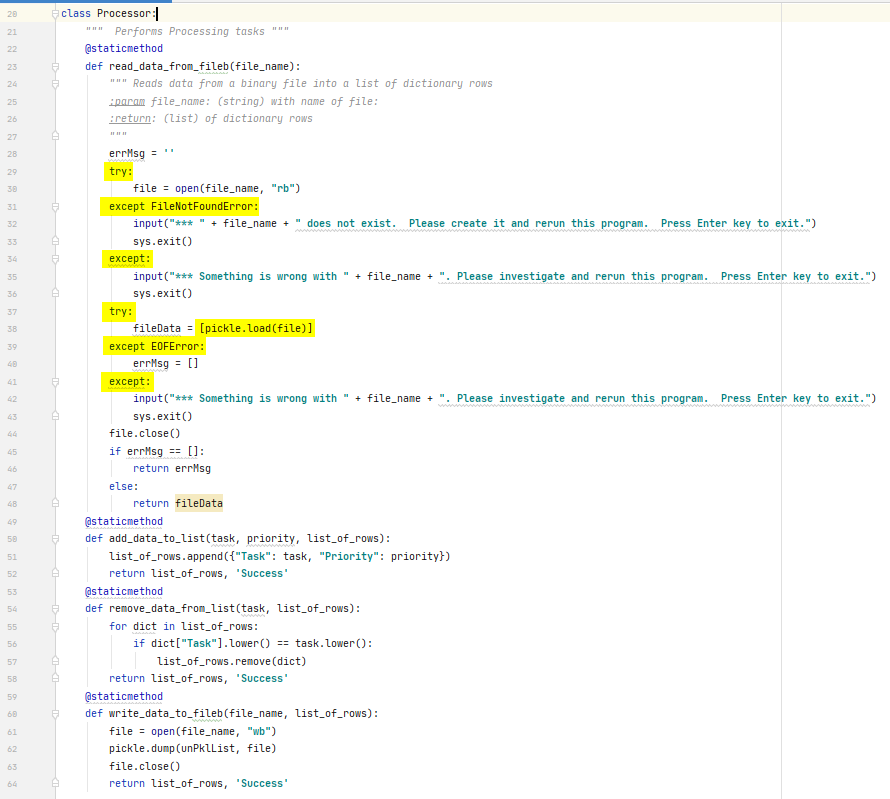


Figure . Process the Data

1. In the Presentation (Input/Output) section, create a new class called IO. Adapt the code from Assignment 6 to work within the new functions, as seen in Figure 4 and Figure 5.



Figure - IO Class

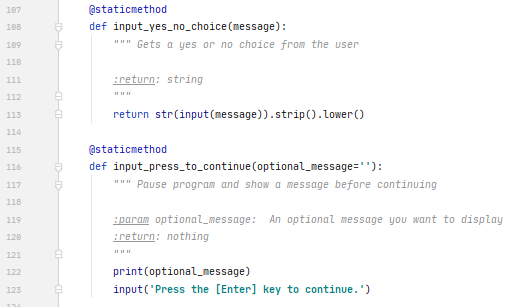


Figure . IO Class

1. In the Main Body of the Script, first use the Processor.read\_data\_from\_fileb() function to call the function which will load all the data from the file, and assign it to variable unPklList (Un-Pickled List). Note the use of the “class” prefix “Processor”, then the “.”, then the function name. We continue to use conditional logic and calling our IO functions to determine the user choice. Based on user choice, we prompt for inputs and pass the parameters/arguments back to the functions via function calls. This makes our Main Body much more readable, and allows us to reuse our functions, as seen in Figure 6 and Figure 7.



Figure . Main Body

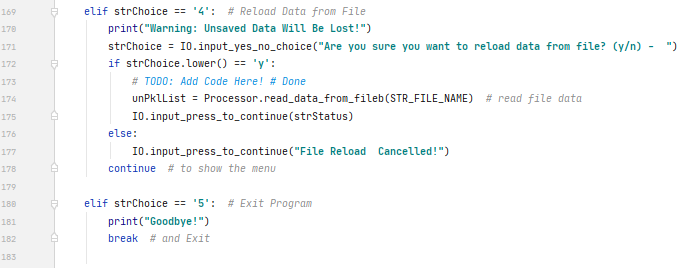


Figure - Main Body (cont')

1. Run the program in the command prompt, as seen in Figures 8 – 11, below.

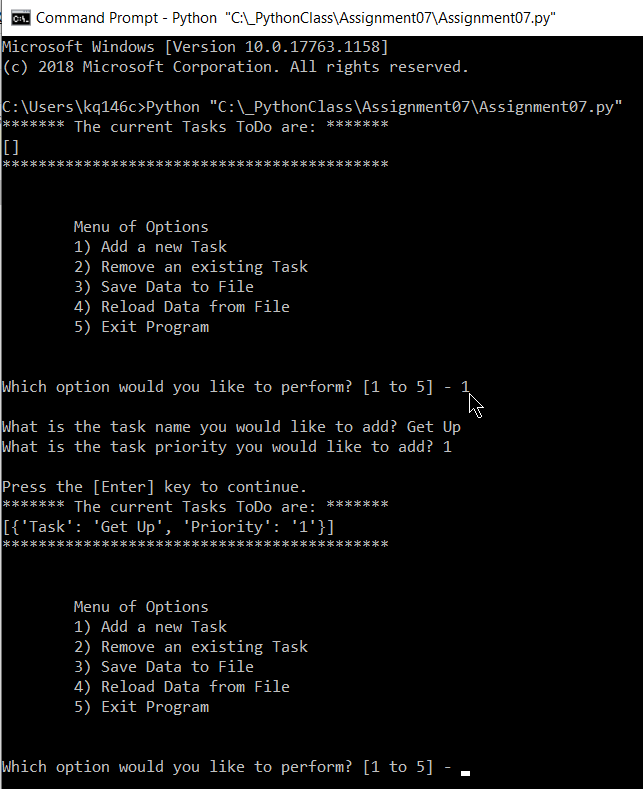


Figure – List is Loaded and Displayed on Program Start. Option 1 Selected.

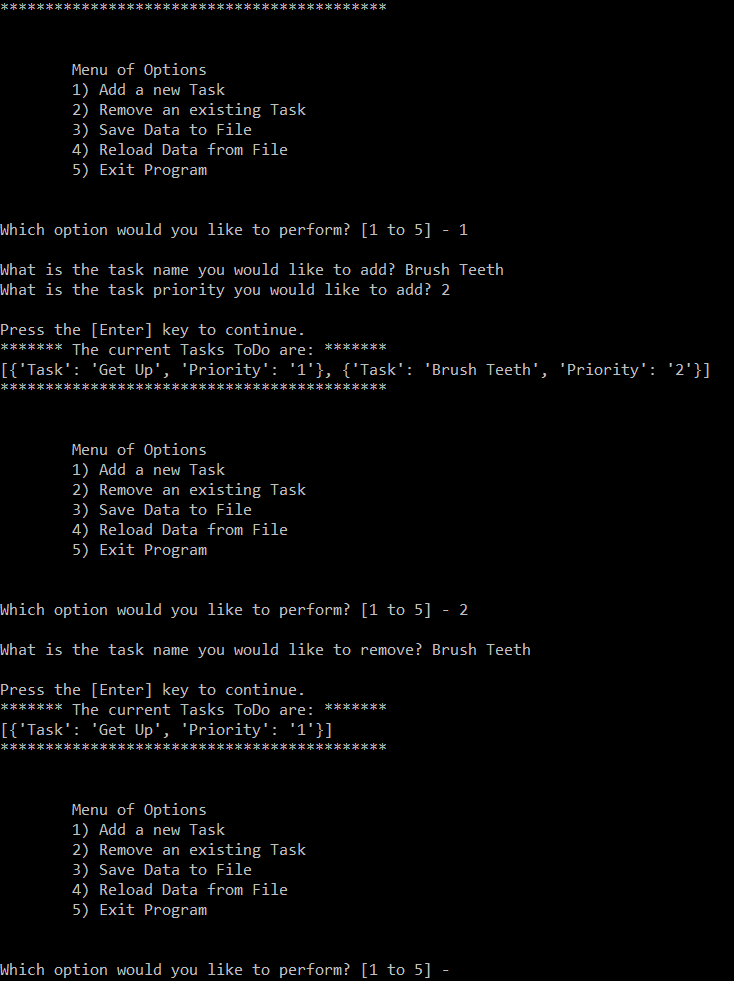


Figure - Option 2

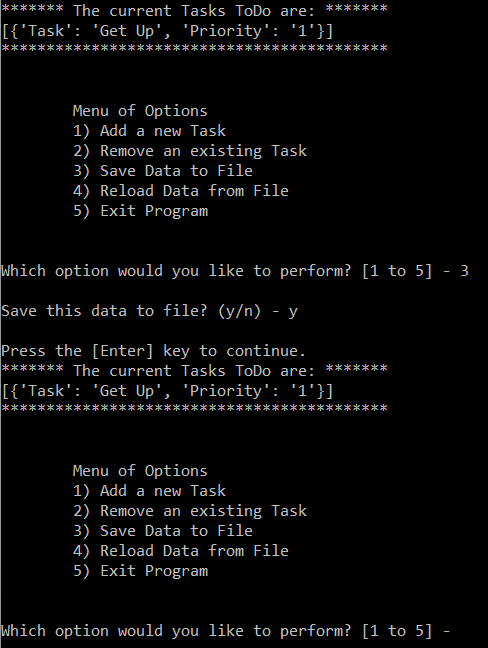


Figure - Option 3

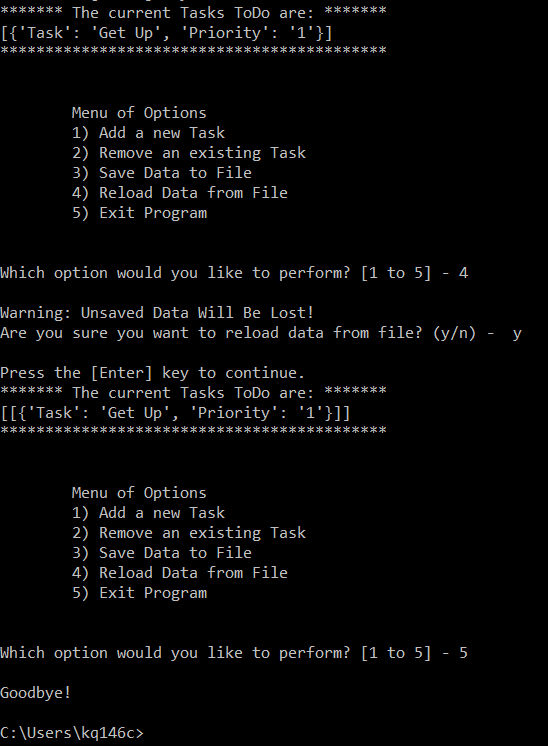


Figure – Option 4 and Option 5

1. Open the .dat file to see the Tasks and Priorities which have been saved in a Pickled/Binary/Obscured format, as seen in Figure 12.

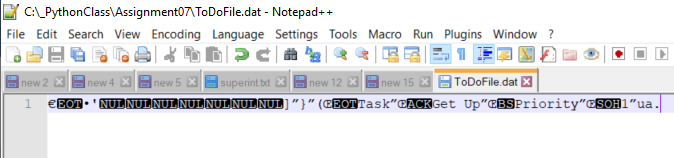


Figure . Open ToDoFile.txt to see the saved output

# Summary

In this assignment, we learned how to modify our Assignment 6 code into Assignment 7 code that utilizes both Exception Handling and Pickling to create a more advanced program. Exception Handling allows us to present the user with messages that are more useful and less confusing than a Python-generated error. Pickling allow us to store data more easily in a .dat file. Binary files (.dat) store data more efficiently than .txt files.