Mark Stoetzl  
11/22/2023  
IT FDN 110 AAu 23:Foundations Of Programming Python  
Assignment 06

GitHub Link: https: [//github.com/Stetz33/IntroToProg-Python-Mod06](Assignment06_Stoetzl.docx)

Creating Python Program Assignment 06

Introduction  
Module #6 consisted of creating a program that uses functions. By creating a python program that is comprised of functions you avoid rewriting the same logic or code again and again. In a single program you can call functions anywhere and use the functions multiple times. The functions are grouped into classes that bundle different types of data and functionality together.

Step 1  
The first section involves creating the code that defines the constants and the variables as is done in all programs. After the constants and variables have been defined you create the FileProcessor class framework for the program. The FIleProcessor class contains two functions. The first function reads data from a file. The file read function contains error handling messages also. This function was similar to the function used in Assignment #5. The second function that is nested in the File Processor class writes to the file. The file is written to by opening it, using the json.dump for the list of student\_data. The information is passed into the student\_data list through the IO.output\_dtudent\_and\_course\_names(student\_data=student\_data) logic that is contained in the while loop. A snap shot of the code is shown in illustration #1 below.

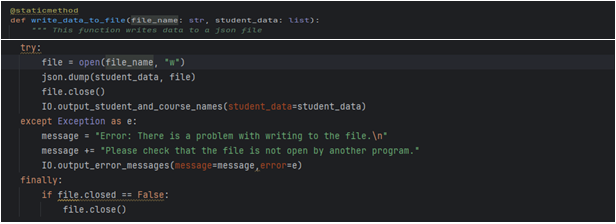


Illustration #1 Write to file function

Step 2

The second section involves creating all of the input & output functions for the program. All of the user input and output functions are nested inside of the I.O. class. Specifically it contains the functions that perform the following tasks: (1) output error message, (2) output menu that was previously defined in the constants, (3) input menu choice to choose what process you want the program to perform. (4) output student and course name information that is in the file or has been previously input. (5) input the student registration data for the first, last and course name. If any of the information needs to be accessed by other parts of the program it is passed through using the return function followed by the variable name containing the data that is passed through. A example of this is shown in illustration #2 below. In this case the menu number that is input is passed through to other functions.

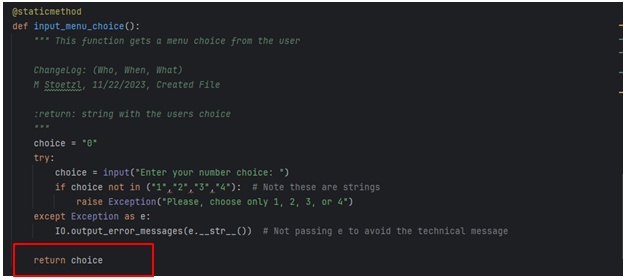


Illustration #2 – showing the data return code that is passed through

Step 3

The third section involves using a while loop to access the code for each individual menu choice in the program. The data is passed through to both the I.O. functions and the File Processor functions by defining the respective processor or I.O. code as needed. Illustration #3 shows how the File Processor is defined.

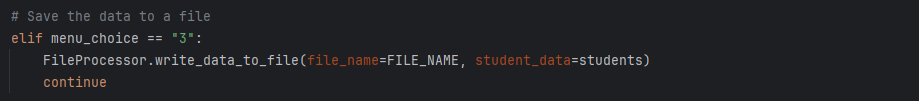


Illustration #3 – File processor

Summary  
By separating out each function and grouping them into classes the program is easily expanded or changed without have to re-write the entire file. It becomes much more modular. For this assignment the overall concept was described very well in the notes and labs. I encountered some data transfer errors which resulted in me spending a large amount of time trying to trouble shoot them. Even after going to the TA section (Thanks Ning) I had problems getting my program to run. Ultimately used the starter code from lab #7 to determine that my “Enrollments” file was corrupted and this was causing the error even after I had the Assignment06 code correct.