Lab Exercise 7

Focus

1. Batch input and output with files
2. Exception handling

This lab maps to learning objectives 1 through 5 in Competency Module Seven – Write a Working Program that Uses Both Input and Output Files as well as Exception Handling.

Part A: Building on an Existing Solution

For this portion of the lab, you will start by downloading the program Lab7a_euro_conversion.py from Blackboard. This program has the user input a number of euros and converts it to US dollars. It then has the user input a number of dollars and converts it to euros. This program includes input validation loops to ensure that the number of euros entered and the number of dollars entered are zero or larger.

Run the program a few times to be sure you understand how it works. Now try running it and try inputting a value of *10a* for the number of euros. You should see that this causes the program to abort. Look on the last line of the traceback output to see what type of exception is raised.

Redesign the program to make two major changes. First, you will add a menu-driven loop so that the user can perform as many conversions as they want. On each iteration of the loop they can choose if they want to convert euros to dollars, convert dollars to euros, or end the program. The second change will be to add an exception handler so that the program does not abort if the user enters invalid characters while trying to input a number. Here are some more detailed instructions:

1. Add a menu-driven loop similar to Program 5-28 in chapter 5 of your textbook. Be sure to include an error message if the user enters an invalid menu selection.
2. Instead of inputting the number of euros in the main function, define a separate value-returning function that reads and returns the number of euros.
3. Add a try/except statement in your function to ensure that your function will not abort if the user enters invalid characters. The error handler should set the number of euros to -1. (Note: Be sure to leave the validation loop in your main function.)
4. Instead of inputting the number of dollars in the main function, do the input in a value-returning function that handles bad input (follow the instructions in steps 2 and 3).

Save the program as firstname_lastname_Lab7a.py where you will replace firstname and lastname with your actual first and last name.

Part B: Write Something New

Write a complete and syntactically correct Python program to solve the following problem:

Write a program for Professor Polanco at Austin Community College that allows him to keep a record of each student's name and average grade for his class. The program will read the input from the keyboard and write it to a text file. Then to verify that the file was created successfully, it will read the file and display it on the screen. Here are more detailed instructions:

1. The input must be interactive from the keyboard. You will take input for 6 students.
2. For each student, you will input the student's name and grade.
3. Write a separate value-returning function to input and return the student's name. You should call this function from your main function.
4. Write a separate value-returning function to input and return one grade. Include a try/except statement in your function to ensure that your function will not abort if the user enters invalid characters. The error handler should set the grade to -1.
5. A grade cannot be below 0 or above 100. Include a validation loop to ensure that the grade falls in this range. You can put the validation loop in your main function. **(LO 5)**
6. Be sure that all input is done in a function other than main.
7. Write a record for each student to an output file named *grades.txt*. You should write the name on one line and the grade on the next line. The output file should only contain names and grades. **(LO 1, 3, 4)**
8. When you have written records for all students to the output file, close it. **(LO 3, 4)**
9. Open the file *grades.txt* for input. **(LO 2)**
10. Your program will raise and handle an exception if the file is not found. The error handler should display an informative error message. **(LO 5)**
11. Read the file line by line and display each line to the screen. **(LO 2)**

Please save your file as firstname_lastname_Lab7b.py where you will replace firstname and lastname with your actual first name and last name. Remember to use the extension .py.

Run and test your program for all conditions. Once you are sure it works you will turn in the items listed in the next section.

Turn In

All labs will be graded in Blackboard. Once you are done with the lab turn it in to the Lab 7 link. Please read the How To Submit instructions if you have any questions or contact the instructor / academic coach. For this lab you will turn into Blackboard:

1. The Python *code file(s)* you saved in part A.
2. The Python code file you saved in part B and the *grades.txt* file created in part B