Simon Fraser University
School of Computing Science

CMPT 120 – Assignment 4

Students' Result Application

Objective

Our objective, in this Assignment 4, is …

- To solve a problem by developing, testing and debugging, in an incremental fashion, a Python program, using any of the Python building blocks we have learnt so far.
- More specifically, to practice opening, reading from, writing to and closing text files, and to manipulate data.

On our own

We are to do Assignment 4 on our own, not in a group.

A gentle reminder about Academic Honesty: we are expected to do this entire assignment on our own, i.e., not using code we find on the Internet and/or not having someone else do this assignment (entirely or partially) for us. Doing any of the above will not allow us to learn. The only exception to all this is that we can reuse any code we wrote for this course's labs, and/or assignments and we can reuse any code we find on our CMPT 120 course web site.

Problem Statement

In this assignment, we create a **Students' Result Application** that reads data (student marks) from a text file and writes the result (total mark and grade) in another text file.

Your **Students' Result Application** should **read** data from one input file and **write** data into **three** output files. The output files are listed in the *Output Files* section below.

See the **File Structure and Sample Runs** section for further file details, layouts, and data.).

Input file (**score1.txt**)

1. **score1.txt** Contains student marks. One line for each student. Sample data is shown below.

```
sid,lname,fname,lab,assignments,midterm,final
00000,bbb,aaa,10,20,20,50
30115,Tim,Michael,10,20,20,50
30116,He,yuxuan,9,17,10,40
```

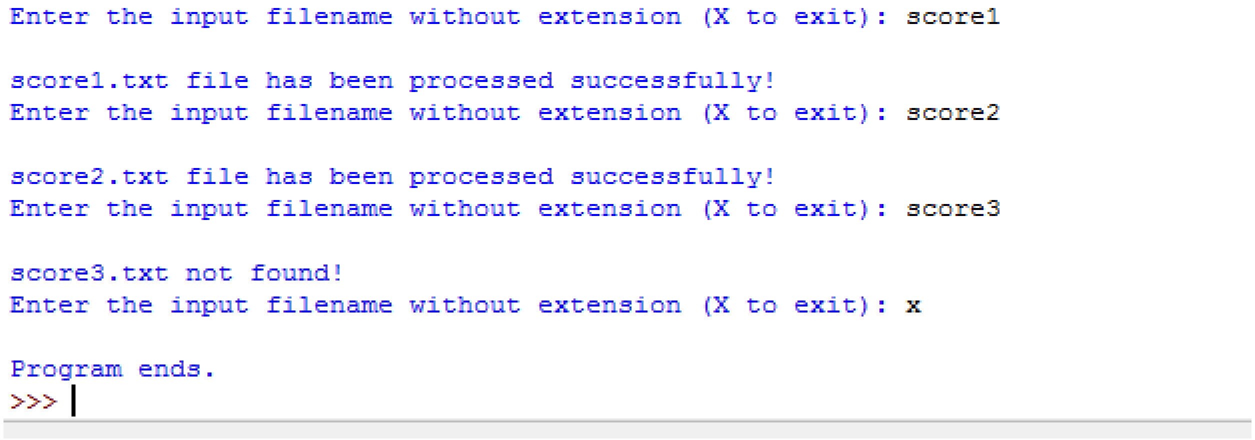
- First two lines of the file contain **headers** and data information.
    - Line 1: contains data headers. (Call it a **header line**.) In score1.txt file, the headers are:
        - sid,
        - lname,
        - fname,
        - lab,
        - assignments,
        - midterm,
        - final
    - Line 2: contains total marks. (Call it a **total marks line**.) 00000 indicates that student id must be **5-digit** long. bbb and aaa indicate that last name and first name are alphabetic.
- Line 3 until the last line: contains students' marks data. (Call it **data lines**.)
    - Values on each data line are separated by commas.
        - Values may have extra spaces around them. We need to get rid of those spaces. The first or last names may be in lower or upper cases, we need to ensure they are all capitalized.
        - Remember that there is a newline character ('\n') following each line in the data file. We need to get rid of those unwanted newline characters as well.
            - How do we get rid of unwanted newline characters? **Answer**: see the program **File_IO_Demo_Read_File.py** on our course web site and our lecture notes on Files I/O.
            - (**Note:** Getting rid of those extra spaces, newline characters is called data cleansing. We always try to clean our data before using it in our calculations.)

Output files: (See **File Structure and Sample Runs** section for further details):

2. **score1_log.txt** Logs (writes) each line of input data as it is being processed.
3. **score1_error.txt** Contains incorrect data. Data is incorrect if:
    - student id is not of the same length (**each sid should be of same length**),
    - a value is missing (i.e. the number of values on the line are not same as number of values on the first, header, line).
4. **score1_grade.txt** Contains total marks and grade for each student.

Suggested Algorithm

- Prompt user to enter an input filename.
- While the user has not terminated the application and file exists:
  - Read the first line (header line) from the data file.
    - Move the line into a list. Strip extra space or newline characters.
    - Write data into the log file as you read and strip the data. See the sample run log file above for exact messages.
  - Read the second line (total mark line) from the data file.
    - Move the line into a list. Strip extra space or newline characters.
    - Write data into the log file as you read and strip the data. See the sample run log file above for exact messages.
  - Read line 3 until the last line one by one
    - Move each line into a list. Strip extra space or newline characters.
    - Write data into the log file as you read and strip the data. See the sample run log file above for exact messages.
    - Write line to the error file, if it is in error. (See **#3 on page 1 for error detail**.)
    - Compute total marks.
    - Compute grade.
    - Write data in the grade file.
  - If the filename does not exist, print error message. (Hint: Use **try** and **except**.) or **X** if the user wishes to exit.
  - Define and use functions for generating list, computing total mark, and grade.
  - You may also define functions to write data into text files.
  - Here is a sample prompt and possible input values and messages:



```
Enter the input filename without extension (X to exit): score1

score1.txt file has been processed successfully!
Enter the input filename without extension (X to exit): score2

score2.txt file has been processed successfully!
Enter the input filename without extension (X to exit): score3

score3.txt not found!
Enter the input filename without extension (X to exit): x

Program ends.
>>> 
```

Incremental Development and Some Hints!

Not quite sure where to start? Here is a suggestion:

1. First, do we understand what we are asked to do in this Assignment 4?

   If the answer is **No**, then let's have a look at the **sample run** and the input/output files. If this does not clarify Assignment 4 for us, let' seek help by seeing either the instructor or the TA's.

2. Once Assignment 4 is clear, let's start. Feel free to reuse (and tweak) any code we have written so far in this course or any code we find on our course web site.

3. Then implement **incrementally** each of the steps of the Suggested Algorithm above. Move on to the next step only when the current step has been implemented and tested successfully.

   Hurray, we are done!

Grading Criteria

```
A+ start from 95, A  start from 90, A- start from 85
B+ start from 80, B  start from 75, B- start from 70
C+ start from 65, C  start from 60, C- start from 55
D  start from 50, F  below 50
```

Submission

- Let's submit our file: **studentsResultApplication.py** file on CourSys by Friday, July 27, no later than 16:30. **Code must be written in Python 3.**
- We can submit our work to CourSys as many times as we wish, but it is the last submission that counts. So, let's make sure that our last submission is done before the deadline.
- No late submission will be accepted: any submission made after the due date and time stated above will not be marked for grades. However, they will be marked to provide feedback to students.
- The only exception is for medical reasons. If we are not able to submit our assignment on time due to a medical reason, we must notify the instructor by email right away and provide her with a doctor's note as soon as possible.

Marking

When marking Assignment 4, we will look at some of the following criteria:

- Are the requirements related to the data files we are to submit satisfied?
- Is our Students Result Application producing the same output as the one illustrated in the provided Sample Runs when the user enters the provided data file score1.txt or score2.txt?
- Have the GPS guidelines described on our course web site been used? Let's make sure we have read these guidelines posted on the GPS web page and that we are familiar with them.
- Have we created functions and these functions are called at least once within our program. At this point in the semester, we should no longer be creating programs that do not include functions.
- Have we included a header comment block? Does it contain the name of the file, a description of the program, the name of the author and the creation date of the program?
- Is our program easy to read? Is it commented?
- Have we selected the most appropriate/efficient conditional and iterative statements for each situation? Are we opening and closing our data files properly? Are we using lists appropriately?
- Are we the only author of the code we have submitted to CourSys?
- Is our Python program executing? Are there any syntax and/or runtime errors?
- Is our Python program solving the problem? Are there any semantic errors?

File Structure and Sample Runs

score1.txt: In order to be able to read the student marks stored in a data file and create other data files, we need to understand the structure of this data file, i.e., what the data found on each line of the file signify.

(**Note**: a score file may contain different data. Your program should handle that as well. See another data set **score2.txt**, for instance.)

<u>Sample Run 1</u>

score1.txt data:

```
sid,lname,fname,lab,assignments,midterm,final
00000,bbb,aaa,10,20,20,50
30115,Tim,Michael,10,20,20,50
30116,He,yuxuan,9,17,10,40
30117,Chua, Marc ,10,10,8,25
30118,Gil,esther,5,20,17,38
30119,Chan,Lim,10,20,15,43
30120,Chan ,Jacob,10,20,5,20
30121,long,Mehtab,10,20,,30
30122,Hayre,Zaid,10, 20,10,30
30123,Lari,Tracy,7,16,13,29
30124,Lion,Amy,3,20,19,50
30125,patel,suri,3,17,19,10
3012,Keith,Andrew,3,20,10,50
30127,Miller,James,3,20,50
```

---

score1_log.txt: A sample run is shown below. Complete sample run is posted on the website. Your program must generate the following output and write it in the score1_log.txt file. (Log Time may vary each time we run the program.)

```
********************
Processing header lines.
********************
```

```
----------------------
Processing line 1.
----------------------
Log Time: 2018-07-12 07:49:56.462630

Input line is...
sid,lname,fname,lab,assignments,midterm,final

Generated list is...
['sid', 'lname', 'fname', 'lab', 'assignments', 'midterm', 'final']

The list contains 7 values.

----------------------
Processing line 2.
----------------------
Log Time: 2018-07-12 07:49:56.462630

Input line is...
00000,bbb,aaa,10,20,20,50

Generated list is...
['00000', 'bbb', 'aaa', '10', '20', '20', '50']

The list contains 7 values.

**********************
Processing data lines.
**********************

----------------------
Processing line 3.
----------------------
Log Time: 2018-07-12 07:49:56.462630

Input line is...
30115,Tim,Michael,10,20,20,50

Generated list is...
['30115', 'Tim', 'Michael', '10', '20', '20', '50']

The list contains 7 values.
```

score1_error.txt: A sample run is shown below. Also posted on the course website. Your program must generate the following output and write it in the score1_error.txt file.

```
Line 14 in error: 3012,Keith,Andrew,3,20,10,50
Line 15 in error: 30127,Miller,James,3,20,50
```

---

score1_grade.txt: A sample run is shown below. Also posted on the course website. Your program must generate the following output and write it in the score1_grade.txt file.

```
                                **************
                                Students' Result
                                **************

Student Id    Name                              Total Mark    Grade
-------       ----                              ---------     -----
30115         Michael Tim                       100.0         A+
30116         Yuxuan He                         76.0          B
30117         Marc Chua                         53.0          D
30118         Esther Gil                        80.0          B+
30119         Lim Chan                          88.0          A-
30120         Jacob Chan                        55.0          C-
30121         Mehtab Long                       60.0          C
30122         Zaid Hayre                        70.0          B-
30123         Tracy Lari                        65.0          C+
30124         Amy Lion                          92.0          A
30125         Suri Patel                        49.0          F
```

Sample Run 2

score2.txt data:

```
sid,lname,fname,assignments,midterm,final

000000,bbb,aaa,30,20,50

300145,Reza,Ahmed,30,20,50

300146,Howe,Ayan,27,10,40

300147,Nitasha, Li ,21,8,25

300148,Hashir,ali,27,17,38

30019,Chau,Kim,30,15,43

300151,long,Mehtab,10,,30

300152,Hayre,Zaid,10,30

300153,kim,Baker,,,29

300154,John,Ally,50
```

---

score2_log.txt: A sample run is shown below. Complete sample run is posted on the website. Your program must generate the following output and write it in the score2_log.txt file. (Log Time may vary each time we run the program.)

```
********************
Processing header lines.
********************

------------------------
Processing line 1.
------------------------
Log Time: 2018-07-12 07:41:19.693816

Input line is...
sid,lname,fname,assignments,midterm,final

Generated list is...
['sid', 'lname', 'fname', 'assignments', 'midterm', 'final']

The list contains 6 values.
```

```
------------------------
Processing line 2.
------------------------
Log Time: 2018-07-12 07:41:19.693816

Input line is...
00000,bbb,aaa,30,20,50

Generated list is...
['000000', 'bbb', 'aaa', '30', '20', '50']

The list contains 6 values.

******************
Processing data lines.
******************

----------------------
Processing line 3.
----------------------
Log Time: 2018-07-12 07:41:19.693816

Input line is...
30145,Reza,Ahmed,30,20,50

Generated list is...
['300145', 'Reza', 'Ahmed', '30', '20', '50']

The list contains 6 values.
```

---

score2_error.txt: A sample run is shown below. Also posted on the course website. Your program must generate the following output and write it in the score2_error.txt file.

```
Line 7 in error: 30019,Chau,Kim,30,15,43
Line 9 in error: 300152,Hayre,Zaid,10,30
Line 11 in error: 300154,John,Ally,50
```

---

score2_grade.txt: A sample run is shown below. Also posted on the course website. Your program must generate the following output and write it in the score2_grade.txt file.

```
                              **************
                              Students' Result
                              **************

Student Id    Name                             Total Mark    Grade
--------      ----                             ---------     -----
300145        Ahmed Reza                       100.0         A+
300146        Ayan Howe                        77.0          B
300147        Li Nitasha                       54.0          D
300148        Ali Hashir                       82.0          B+
300151        Mehtab Long                      40.0          F
300153        Baker Kim                        29.0          F
```

Have fun!