CIS 41B - Lab assignment 1: iterables, callables

Write a program that works with a list of student grade records. The program reads the student records from a file, and lets the user view and search the student records.

**Input file**

The input text file is lab1in.txt. Each line of the file is one student record, with comma separated fields.

The first field of each record is a student name, followed by at least one grouping of 3 fields: class name, units, letter grade. There can be one or more groupings of the 3 fields.

Example of one line of the input file: Grace Hopper,CIS 22A,4.5,A+,PHIL 4,4,B+

There is also an input text file, lab1in\_error.txt, where some student records are not in the correct format. After the name field, there is one grouping of 3 that doesn't contain all 3 fields: Lisa Su,CIS 22A,4.5,EWRT 2,5,B

**Program requirements**

The application is made of 2 source files: lab1.py and record.py

It is recommended that you write code for the green portion first for both files. It is the base code for the lab.

Then write the black portion later, this is the extra features that should be added only if the base code works.

record.py

Contains the Record class, which holds information for 1 student record, and has:

* Instance variables that hold the student name and the grouping of 3 for each course (class name, units, grade). You decide which data structure to store the student record.
* *For 1pt of the lab*: at the top of the record.py file, in a comment block, explain clearly what data structure you use. Example: My data structure is a set of dictionaries, one dictionary for each class that the student took.

The key is the student name, the value is a list of class name, units, grade

(This example data structure is not the best way to organize the data, so don't follow it)

* The constructor:
  + Accepts one line of the input file, parses (separates) the line into data values and stores them in the data structure. You should do the parsing *without using loops* (recall that comprehension doesn't count as a loop because the comprehension loop doesn't run at the Python level).
  + If the line cannot be parsed properly (lab1in\_error.txt), raise an exception so that main can print an error message and end the program.
* A print method that prints the student record in multiple lines:
  + first line: student name
  + subsequent lines: class name : units, letter grade (in column format). See sample output.  
    Challenge: use only one for loop in this method.
  + The class names should be in alphabetical order.
* A getName method that returns the student name
* A hasGrade method that:
  + accepts a letter grade
  + returns True if the student record has that grade, False otherwise.

You should not have to use a loop (same criterion as the constructor not using loops).

[My record.py has 15 lines of code, not counting comments and blank lines. This is meant to give you an estimate of the code you would write, you don't have to match this]

lab1.py

This file imports the Record class in order to work with it, but lab1.py is not an OO program.

The file has the following functions.

1. A getData function that will:
   * Accept a filename or use the default filename lab1in.txt
   * Read in from the file and create a list of Record objects
   * If the file open is not successful or if the Record object raises an exception, print a descriptive error message and end the program
2. A printAll function:

* Call the print method of each Record object in the list.
* The student names should be printed in the same order as in the input file.

1. A getChoice function that will get a choice from the user. The function will:

* Print a menu of 3 choices: s. search for one record

g. grade count

q. quit

* Keep prompting the user until there is a valid choice and then return the choice.
* Challenge: use one loop condition and don't use a break statement

1. A userProcess function that will

* Call the getChoice function above to get a valid choice
* As long as the choice is not 'q', call a search function or a countGrade function to process the user choice.
* Keep looping until the user choice is 'q'
* Don't use any if statement to select the search function or countGrade function. Take advantage of the fact that functions are first class objects in Python.

1. A search function that will:

* Create a generator to return one student record at a time. The student records are returned in alphabetical order by student *last* name.
* Use the generator to get the first student record and print it.
* Loop to let the user press the Enter key to get the next student record, or enter any other key to stop the search.
* Each time the user hits Enter, use the generator to get the next student record and print it.
* End the search function when the user enters any other key, or print "End of record" if the user has gone through all of the student records.

1. A countGrade function that will:

* Keep prompting the user for a grade until there is a valid grade (A+, A, A-, B+, B, B-, C+, C, C-, D+, D, D-, F).
* Print an error message to remind the user of the valid grades if the grade is not valid.
* Call a findGrade function to print the number of students that have the user input grade.

1. A findGrade function that will:

* Call the hasGrade method of each Record object, and use the return value to count the number of students with the user input grade.
* Print the number of students with that grade.

1. A decorator called 'check' that will:

* Keep track of whether an input *string* argument has already been passed to the function that it decorates.
* If the input string argument has not been passed previously, then it runs the function.
* If the input string argument has been used, it doesn't run the function and prints an "already run" message.
* Recall that a decorator should be somewhat general purpose and can decorate different functions. This means don't check for a specific string value and don't look for a specific positional argument or keyword argument.
* Use the check decorator with the findGrade function, so that during testing, if the user entered a letter grade that is already checked previously, then the "already run" message is printed.

1. A main function that will:

* call the getData function
* call the printAll function
* call the userProcess function

**Additional requirements**

* At the top of each file, put your name and one line description of the code.
* Add a one line docstring to describe each function and public method.

[My lab1.py is about 70 lines of code, not counting comments and blank lines]

**Test**

Test your code with both lab1in.txt and lab1in\_error.txt, and test with invalid input.

**Sample output** (user input is shown in green):

Guido van Rossum # output of printAll function, same student order as input file

ARTS 1A : 4 B-

CIS 41A : 4.5 A-

POLI 1 : 5 C

Lisa Su

CIS 22A : 4.5 B+

EWRT 2 : 5 B

Linus Torvalds

BIOL 10 : 5 C

CIS 18B : 4.5 A

Grace Hopper

CIS 22A : 4.5 A+

PHIL 4 : 4 B+

Katherine Johnson

CIS 22C : 4.5 A-

MATH 1B : 5 A

PHYS 4A : 6 B

Carlos Guestrin

BUS 18 : 5 B+

BUS 55 : 5 A

CIS 41A : 4.5 A

HUMI 16 : 4 A-

Tammarrian Rogers

CIS 35A : 4.5 A-

COMM 2 : 5 B

Ima Student

CIS 40 : 4.5 C

KNES 1 : 0.5 F

Steve Chen

BUS 56 : 5 C+

CIS 41A : 4.5 A

COMM 1 : 5 A

Satya Nadella

BUS 65 : 5 A+

CIS 22C : 4.5 B+

MATH 1DH: 5 A-

=========================

s: search for one record

g: filter by grade

q: quit

Enter your choice: a

Enter your choice: 12

Enter your choice: s

Printing one student record one at a time

Steve Chen # output of generator, alphabetical order by student last name

BUS 56 : 5 C+

CIS 41A : 4.5 A

COMM 1 : 5 A

Press Enter for next name, anything else to quit: <Enter key>

Carlos Guestrin

BUS 18 : 5 B+

BUS 55 : 5 A

CIS 41A : 4.5 A

HUMI 16 : 4 A-

Press Enter for next name, anything else to quit: done

=========================

s: search for one record

g: filter by grade

q: quit

Enter your choice: g

Enter letter grade: A

{'A'} # optional print out from decorator, showing grades that have been searched

4 student(s) with grade A

=========================

s: search for one record

g: filter by grade

q: quit

Enter your choice: g

Enter letter grade: A+

{'A', 'A+'} # optional print out from decorator, showing grades that have been searched

2 student(s) with grade A+

=========================

s: search for one record

g: filter by grade

q: quit

Enter your choice: g

Enter letter grade: a

Grade should be A-F with optional + and -

Enter letter grade: A

already searched this grade # output from decorator, instead of from function findGrade

=========================

s: search for one record

g: filter by grade

q: quit

Enter your choice: g

Enter letter grade: F

{'F', 'A', 'A+'} # optional print out from decorator, showing grades that have been searched

1 student(s) with grade F

=========================

s: search for one record

g: filter by grade

q: quit

Enter your choice: g

Enter letter grade: D-

{'F', 'A', 'D-', 'A+'} # optional print out from decorator, showing grades that have been searched

0 student(s) with grade D-

=========================

s: search for one record

g: filter by grade

q: quit

Enter your choice: s # begin of search through all student records

Printing one student record one at a time

Steve Chen

BUS 56 : 5 C+

CIS 41A : 4.5 A

COMM 1 : 5 A

Press Enter for next name, anything else to quit: <Enter key>

Carlos Guestrin

BUS 18 : 5 B+

BUS 55 : 5 A

CIS 41A : 4.5 A

HUMI 16 : 4 A-

Press Enter for next name, anything else to quit: <Enter key>

Grace Hopper

CIS 22A : 4.5 A+

PHIL 4 : 4 B+

Press Enter for next name, anything else to quit: <Enter key>

Katherine Johnson

CIS 22C : 4.5 A-

MATH 1B : 5 A

PHYS 4A : 6 B

Press Enter for next name, anything else to quit: <Enter key>

Satya Nadella

BUS 65 : 5 A+

CIS 22C : 4.5 B+

MATH 1DH: 5 A-

Press Enter for next name, anything else to quit: <Enter key>

Tammarrian Rogers

CIS 35A : 4.5 A-

COMM 2 : 5 B

Press Enter for next name, anything else to quit: <Enter key>

Guido van Rossum

ARTS 1A : 4 B-

CIS 41A : 4.5 A-

POLI 1 : 5 C

Press Enter for next name, anything else to quit: <Enter key>

Ima Student

CIS 40 : 4.5 C

KNES 1 : 0.5 F

Press Enter for next name, anything else to quit: <Enter key>

Lisa Su

CIS 22A : 4.5 B+

EWRT 2 : 5 B

Press Enter for next name, anything else to quit: <Enter key>

Linus Torvalds

BIOL 10 : 5 C

CIS 18B : 4.5 A

Press Enter for next name, anything else to quit: <Enter key>

End of student records

=========================

s: search for one record

g: filter by grade

q: quit

Enter your choice: q

Don't forget to test with lab1in\_error.txt:

line: Lisa Su,CIS 22A,4.5,EWRT 2,5,B

is not valid

Process terminated with an exit code of 1

And with an non-existing input file:

wrongfile.txt not found

Process terminated with an exit code of 1