CIS 41A - Assignment 8: Regular Expression and review of concepts covered in the quarter

This assignment takes time to design, code, and debug so it is recommended that you start early. And if you can do most of this lab by yourself, then you can confidently say that you are an intermediate level Python programmer.

Write an application that helps a user look up CIS classes. The app uses data from this [CIS web page](http://deanza.edu/cis/schedule.html), which has been saved into a text file called lab8input.txt. (In CIS 41B your program will go directly to the website for the data.)

The app lets the user look up CIS classes by entering a class number, or a CIS topic, or a CIS topic and a quarter.

Lab requirements:

1. Create a class called CISclasses in the cisclasses.py file.  
   The CISclasses object will:
   1. read in all data from the lab8input.txt file *one time* and store only the data that are described in part b.
   2. use regular expression to extract the class number (such as "41A"), the class name (such as "Python Programming"), and the quarters that the class is offered (such as Fall, Winter, Spring).  
      You will need to study the data format in the lab8input.txt file in order to write the regular expression that can match these data. Your code will have multiple regex searches since the data are on separate lines.
   3. provide a method called *searchByNumber* that will accept a class number (such as "41A") as input argument.   
      If the class number matches one of the CIS classes, then the method prints the class number, class name, and the quarters that the class is offered. (See sample output).  
      If the class number doesn't match any CIS classes, then the method prints an error message.
   4. provide a method called *searchByTopic* that will accept an input string (such as "java") as an input argument. If the input string is part of any class name, then the method prints the class number and class name. If there are multiple class names that match the topic, then the names are sorted in alphabetical order.  
      If there is no class name that matches the topic, then the method prints an error message.
   5. provide a method called *searchByTopicQuarter* that will accept a topic input string (such as "java") and a quarter input string (such as "fall") as input arguments.  
      If the topic string is part of a class name and the class is offered in the input quarter, then the method prints the class number and the class name.  
      If no class matches the topic and the quarter, then the method prints an error message.
2. Create a class called UI (for user interface) in the lab8.py file.  
   The UI class is the front end to your application. It will have methods to do the following tasks:
   1. create a CISclasses object
   2. have a loop to give the user a menu with 4 choices and process each choice until choice 4 is entered:  
      1. Search by number  
      2. Search by topic  
      3. Search by topic and quarter  
      4. End search
   3. process choice 1 by asking the user for a class number and calling the searchByNumber method of the CISclasses object.
   4. process choice 2 by asking the user for a CIS topic and calling the searchByTopic method of the CISclasses object.
   5. process choice 3 by asking the user for a topic and a quarter, and calling the searchByTopicQuarter method of the CISclasses object.
3. It is up to you to decide what data structure(s) to store the data for easy data look up. Remember that the user wants to look up by class number, or by topic, or by topic and quarter.  
   At the top of lab8.py, write a short description (2-3 sentences) to explain what data structures you use and what you store it them.

It is also up to you to decide which methods in the 2 classes will do the tasks outlined above.

1. It goes without saying that you should:

- have a docstring for every method  
- check for file open success or end the program

- do user input validation in 2 places in the UI class:  
 - when the user enters a number for a menu choice (should be 1 - 4 only)  
 - when the user types in a quarter name (should be fall, winter, spring, summer only)

- the user should be able to enter text strings in uppercase, lowercase, or a combination of them.

Use the str class title() method to help you. At the shell prompt, try the following example code to see what title()   
 does: "abc".title() "tHIS is A stRIng".title() "LETTERS aNd numbers 123!!".title()

1. At the end of lab8.py, write these 2 lines of code to run the program. This should be the only 2 lines.  
    app = UI()  
    app.run()

Sample output

The sample output only shows cases with no user input error and cases where classes are found.

1. Search by class number

2. Search by topic

3. Search by topic and quarter

4. End search

Enter a number: 1

Enter a class number: 4

CIS 4 Computer Literacy(formerly CIS 93): Fall,Winter,Spring,Summer

1. Search by class number

2. Search by topic

3. Search by topic and quarter

4. End search

Enter a number: 1

Enter a class number: 170f

CIS 170F Windows Administration: Spring

1. Search by class number

2. Search by topic

3. Search by topic and quarter

4. End search

Enter a number: 1

Enter a class number: 89C

CIS 89C Client-Side Programming with Javascript: Fall

1. Search by class number

2. Search by topic

3. Search by topic and quarter

4. End search

Enter a number: 2

Enter a CIS topic: java

CIS 35B : Advanced Java Programming

CIS 36B : Intermediate Problem Solving in Java

CIS 36A : Introduction to Computer Programming in Java

CIS 35A : Introduction to Java Programming

CIS 53 : Java for Mobile Development

1. Search by class number

2. Search by topic

3. Search by topic and quarter

4. End search

Enter a number: 2

Enter a CIS topic: javascript

CIS 89C : Client-Side Programming with Javascript

1. Search by class number

2. Search by topic

3. Search by topic and quarter

4. End search

Enter a number: 2

Enter a CIS topic: c++

CIS 15BG : (Has been replaced with CIS 22B - Intermediate Methodologies in C++ beginning Fall 2013)

CIS 29 : Advanced C++ Programming

CIS 22A : Beginning Programming Methodologies in C++

CIS 22B : Intermediate Programming Methodologies in C++

CIS 27 : Programming in C++ for C Programmers

1. Search by class number

2. Search by topic

3. Search by topic and quarter

4. End search

Enter a number: 3

Enter a CIS topic: C++

Enter a quarter: SPRING

CIS 29 : Advanced C++ Programming

CIS 22A : Beginning Programming Methodologies in C++

CIS 22B : Intermediate Programming Methodologies in C++