CIS 41A - Lab 4: OOP, exception handling, review of containers

Change your Lab 3 to an OO program and add exception handling code to make the program more robust.

Data files

The input file is still *languages.txt*, and there’s a new output file called *lab4out.csv*.

Lab requirement

Convert lab3.py file into 3 modules: lab4.py, languages.py, and country.py

* The file country.py is for a Country class that represents one country. It stores the country name and the country’s languages.
* The file languages.py is for a Languages class, which has a container of Country objects, and it provides the same searches for languages as in Lab 3.
* The file lab4.py is for a UI class, which interacts with the user: provide the menu, run the searches and print the results.

**country.py**:

* Contains the Country class with 2 instance attributes:
  + Name: a string which is the name of the country
  + Languages: a container of languages of the country
* Methods as needed to access the instance attributes.
* Make sure the Country object does not interact with the user (no print() or input() functions). All user interactions belong in the UI class.

**languages.py**:

* Contains the Languages class with:
  + 1 class attributes for the input filename. Make sure your code uses the class attribute and not the “languages.txt” text string.
  + 1 instance attribute: a dictionary of {country name : Country object} for each country.
  + Any other instance attributes as needed.
  + At least 3 methods to support the 2 searches and to find the most common languages.
    - Think carefully about whether the searches should come from the Languages object or the Country object. Keep the following in mind:  
      -- The Country object is responsible for any data or search results within one country  
      -- The Languages object is responsible for search results that are across all countries
    - Make sure the Languages object does not access private data of the Country object directly. Use methods of the Country class instead.
    - Make sure the Languages object does not interact with the user: no input() or print() functions. All user interactions belong to the UI class.

Recommendation as you code the Languages class:

For each method, write test code to call the method at least one time to make sure that all methods work as intended, before continuing with the UI class. This is called unit testing.

Below are the suggested testing steps, in the order of how you would build up the Languages class. You should test one step at a time since each step is for a method of the Languages class. Fix all errors of one step before going to the next step.

1. Create the Languages object.
2. Call the method to get the most common languages, then print the result.
3. Call the method to get data for 1 country: name and languages, then print the result.
4. Call the method to compare 2 countries’ languages: common languages and all languages, then print the result.

When you’re done with all the steps in the unit testing, comment out the block of testing code and move on the UI class.

**lab4.py**  
Contains the UI class with:

* 1 class attribute which has the output filename *lab4out*.csv. Make sure your code uses the class attribute and not the “lab4out.csv” text string.
* An instance attribute which is the Languages object, fully initialized with data from the input files.

After creating the instance attribute, call the method to print the most common languages.

* A run() method, which loops to print the same menu as Lab 3 and processes each user choice until the user chooses to quit.

New for lab 4: when the user enters the choice from the menu, process each choice as a number, not a string.

* One method for each user menu choice. Each method will:
  + Get any input needed from the user
  + Call the appropriate method of the Languages class
  + Print the result with formatting. Note that the UI object does the formatting of all printed results.
  + Note also that any checking of valid country names or searching of countries belongs to the Languages class.
* Any supporting methods if needed.
* New for Lab 4:
  + When the user searches for a country’s data, use a container to keep track of the searched country and its languages. The container should keep only one copy of each searched country. For example, if the user searches for “Mexico” 3 times, then only one instance of “Mexico” will be in the container.
  + After the user has chosen to quit the application, use CSV writer to write to the output file all the country names and their languages, sorted by country names. The output should be in the format such that the file can be read in as a csv file. See sample output file.
  + Then print a message to tell the user that their searches are saved in the output filename.

At the end of the lab4.py file, add these 2 lines of code to run the application:

obj = UI()

obj.run()

**Exception handling**

You can assume that the data from the input file will be good data: each line has the correct format, no missing data, correct data type, etc.

Handle exceptions for the following cases:

* The input file can’t be opened.   
  Print the error message with the filename, then terminate the program. See sample output.

Note that the input file belongs to the Languages class. The UI class should not have to keep track of this file.

* The user enters something non-numeric when asked for a numeric choice.  
  Let the user know the range of correct input, and prompt again. See sample output.

For both input file error and user input error, your code should not have to use an if statement to check for the error. Use exception handling instead.

Documentation

- Create a beginning comment block with your name, lab number, and lab module name (such as languages.py)

- Write a docstring for public methods. Docstrings for private methods are encouraged, but not required.

Sample output

The output from lab4.py should be similar to the output of lab3.py. User input in blue below.

Enter a filename or press Enter for default file: wrongfile

[Errno 2] No such file or directory: 'wrongfile'

Enter a filename or press Enter for default file: languages.txt # Enter key would be the same

The most common languages

Language Num of countries

English 85

French 46

Arabic 29

Spanish 23

Russian 17

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: canada

English, French, other

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: us

Us doesn't exist in database

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: united states

English, Spanish

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 2

Compare languages

Name of first country: canada

Name of second country: us

Us doesn't exist in database

Name of second country: united states

Common language: English

All languages: English, French, Spanish, other

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: canada

English, French, other

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: mexico

Spanish, various Mayan, Nahuatl, and other regional indigenous languages

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 1

Name of country: canada

English, French, other

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: quit

1-3 only

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: 3

Output of country searches are saved in lab4out.txt

lab4out.txt file:

A screenshot of a computer

Description automatically generated

Second test case with different input filenames:

Enter a filename or press Enter for default file:

[Errno 2] No such file or directory: 'languages.txt' # default file not found

Enter a filename or press Enter for default file: languages2.txt

The most common languages

Language Num of countries

English 85

French 46

Arabic 29

Spanish 23

Russian 17

1. Print languages of a country

2. Compare languages of 2 countries

3. Quit

Enter your choice: