CIS 41B - Lab 1: Review of CIS 41A topics; New topics: iterables, callables

Write an application that lets the user look up the US state population.

Input file
The app uses data from the USDA, which gets the population data from the US Census Bureau. The data are in the file *statesPop.csv*. Each line of the file is for one state. Here are the first 3 lines of the file:

Alabama	4040389	4447207	4779736	5024279	5039877	0.003105
Alaska	550043	626933	710231	733391	732673	-0.00098
Arizona	3665339	5130247	6392017	7151502	7276316	0.017453

The columns are: state name, 1990 census population, 2000 census population, 2010 census population, 2020 census population, 2021 population, change in population between 2000 and 2021.

Overview
Lab 1 consists of 2 files that you will turn in: states.py and ui.py

- states.py has the States class, which stores data from the input file and has methods to look up the data. The States class does not interact directly with the user, so there should be no print() or input() function calls.
- ui.py has a UI class (for user interface) that interacts with the user. Based on the user choice, it calls appropriate methods of the States class to get data and display the results in a user-friendly format.

The user has 3 ways to look up state population:
1. Show states and population for a specific year, sorted by population
2. Show states with positive or negative growth rate, sorted by state name
3. Show whether there is a drop in state population between 2 years

It is recommended that you write code in the order shown below.

states.py
1. Create a States class with the following:
a. A class attribute to hold the default input filename *statesPop.csv*

b. An __init__ method that will:
   - Have a filename input parameter, with a default value which is the class attribute.
   - Loop to read the input file and store data in an appropriate data structure, which is an instance variable.
     - The data to be stored are: state name, 1990 population, 2000 population, 2010 population, 2020 population, and growth rate. This means all columns except the next to last column (2021 population).
     - As you read in each line of data, use the map() function to convert the population numbers into integers.
   - The input file should be opened and read only one time in the program.
   - Initialize any other instance variable as needed.

c. A method to list the states with their population for a certain year, sorted in *descending* order by population.
   - Accept a year as input argument.
   - Use the year to select the population column, then return a generator that can produce one tuple of (state name, population) at a time, from a sequence sorted by population in descending order.
   - You should not create temporary containers to do the sorting. Instead use a lambda function with the sorted() function on the instance variable data structure.

d. A generator method that yields a state and its growth rate, based on the user choice of positive or negative growth.
   - Accept a Boolean input argument to indicate the user choice of positive or negative growth.
   - Yield one state and growth rate at a time.
   - Since this is a generator method, don't create temporary containers to do the work. The generator should work only with the instance variable data structure.

e. A method to check whether at least one state has dropped in population between 2 given years.
   - Accept a start year and an end year as input arguments.
   - Return a Boolean to indicate whether at least one state has dropped in population.
   - You should not have to use a loop to find the Boolean result.

f. A property method that returns the maximum number of states.
   - We count District of Columbia as a 'state' for this lab.

g. Create a decorator that prints the name of the function that it decorates to a log file.
   - The purpose of the decorator is to record when the function being decorated is called. The log file can be used to see how often a function is called, or to trace the order that functions are called.
   - The decorator is defined in the States class so it has scope of the States class, but it's not a method of the class.
   - For the function being decorated, the decorator adds the function name to a text file each time that the function is called.
   - Since a function is considered a first class object, we can use the attribute `__name__` of the function to get its name. For example, if a function is defined as: `def myfunction(…)`, then `myfunction.__name__` will return the string "myfunction".
   - Use the decorator with the methods in steps c, d, e above.

When done coding the States class, it's highly recommended that you add test code at the end of the states.py file to test the class, before going to the ui.py module. The code to test should create a States object, then call each of the methods c-f to check that you see the expected output.

ui.py
Contains:
1. A class called UI that has the following:

a. An __init__ method that creates a States object.
   - The States object should be created with its default input file, and the input file (a States data) should not be hard coded in the UI class.
   - If there is file exception, handle the exception by prompting the user for a filename until there is a valid file to create the States object.
   - The UI object should not open any file to check for file exception, it's the job of the States object open the file.

b. A method that prints the states and their population for a particular year.
   - Ask the user for a year and call an appropriate method of the States class to get a generator as return value.
   - Ask the user for the number of states they want to see and check that it's below or equal to the max number of states. Query the States object for the max number of states, don't hard code the value.
   - If the user number is larger than the max number of states, change it to the max number of states.

- Use the generator to print the states and their population in 2 columns, with comma to separate each thousands place, such as 1,234,567. To print a number with comma, use the `,d` for the format: `f'{num:,d}'`
- See sample output for the column format, printing commas, and user input error handling.

c. A method that prints the states and their positive or negative growth.
- Prompt the user with a menu: p. positive
  n. negative

  and keep prompting until the user enters a valid input, in uppercase or lowercase.
- Call the generator method of the States class to print all states with their positive or negative growth in 2 columns, and the growth is a percentage, with 1 digit after the decimal point and a % symbol.
- See sample output for the column format, the growth percentage, and user input error handling.

d. A method that prints whether there is at least 1 state that dropped in population between 2 given years.
- Prompt the user for a start year and an end year.
- Call a States method to get a Boolean result and print the result as a text string.
- See sample output for example text string output and user input error handling.

e. A method that prints a menu and keeps prompting the user until there's a valid choice.
Menu: 1. View most populous states
2. View growth in 2021
3. Check population drop
4. Quit

f. A run() method that will:
- Loop to print the menu and process the user choice until the user chooses to quit.
- When there is a valid choice that's not 4, call the appropriate method to process the user choice *without having to use an if elif statement (or multiple if statements).*

g. At the end of ui.py, start the code with: UI().run()

Documentation
- At the top of each file: put your name and a short description of the file (description can be as short as: States class)
- For each public method: add a docstring

You don't need to add docstrings for private method or 'get' method where the method only returns a data attribute, but you're welcome to do so.

Test
Sample output (user input is in blue)

```
        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: 1
Enter year: two thousand
Must enter a number
Enter year: 2000
Enter number of states: 5
California          33,871,653
Texas               20,851,028
```

```
New York             18,977,026
Florida              15,982,571
Illinois             12,419,927

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: 1
Enter year: 3000
3000 is not a valid census year

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: 2
p. positive
n. negative
Your choice: a
p. positive
n. negative
Your choice: n
Alaska               -0.10%
California           -0.76%
Connecticut          -0.01%
District of Columbia -2.83%
Hawaii               -0.94%
Illinois             -1.10%
Kansas               -0.11%
Louisiana            -0.72%
Maryland             -0.20%
Massachusetts        -0.64%
Michigan             -0.26%
Mississippi          -0.38%
New Jersey           -0.24%
New Mexico           -0.08%
New York             -1.81%
North Dakota         -0.53%
Ohio                 -0.16%
Pennsylvania         -0.30%
Rhode Island         -0.16%
West Virginia        -0.60%

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: 2
p. positive
n. negative
Your choice: P
Alabama              0.31%
Arizona              1.75%
Arkansas             0.48%
Colorado             0.66%
Delaware             1.36%
Florida              1.13%
Georgia              0.82%
```

```
Idaho                3.36%
Indiana              0.30%
Iowa                 0.08%
Kentucky             0.08%
Maine                0.73%
Minnesota            0.02%
Missouri             0.22%
Montana              1.85%
Nebraska             0.11%
Nevada               1.27%
New Hampshire        0.83%
North Carolina       1.07%
Oklahoma             0.69%
Oregon               0.21%
South Carolina       1.41%
South Dakota         0.98%
Tennessee            0.93%
Texas                1.31%
Utah                 2.03%
Vermont              0.39%
Virginia             0.13%
Washington           0.43%
Wisconsin            0.04%
Wyoming              0.34%

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: 3
Enter start year: 1990
Enter end year: 2000
Population drop in at least one state between 1990 and 2000

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: 3
Enter start year: 1990
Enter end year: a
Must enter a number
Enter end year: 1995
Invalid year 1995

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: 3
Enter start year: 1990
Enter end year: 2020
No population drop across all states between 1990 and 2020

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit
```

```
Enter your choice: 6
Invalid choice

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: quit
Choose 1-4

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice: 4
```

And the log file from the decorator contains the following:

```
byPop
byPop
growth
growth
drop
drop
drop
```

Your States method names don't have to match the above, of course. But the log file shows that the methods of step c was called twice, the method of step d was called twice, and the method of step e was called 3 times.

Sample run 2 to show file open error handling (use input in blue):

```
[Errno 2] No such file or directory: 'statesPop.csv'
Enter an input filename: statesPop1.csv

        1: view top populous states in a year
        2: view growth in 2021
        3: check population drop
        4: quit

Enter your choice:
```