Assignment #5: States3 **Due:** Wed. Oct. 5th **(midnight)**

Program files:

Code:	state_driver.cpp
	state.h
	state.cpp
Input:	state.txt
Output:	a5.txt

Objective:

For this assignment you will reuse the **State** class you created for assignments 3 and 4. This time you will separate your State class into multiple files, separating the class definition from the implementation. You will also be loading data from an external file into an array of objects, then searching the array.

Program Overview:

This program will utilize the C++ **State** class from your previous two assignments. The initial data set for this program will again come from an external file. This time you will load your data into an array of objects. Once your data is loaded, you will then prompt the user and ask them which state they would like information on. They may enter either the State's name or the State's abbreviation. Once you accept the user's input, you will search the array for the state. If the state exists in the array, you will display the result of the search to the console and the output file. If the state is not found in the array, you will output a message to the user that the state was not found. (Please refer the sample output at end of this program for guidance on how to format your output)

You will also be breaking down this assignment into multiple code files. We will go over this in class. The code files you will have for this assignment include:

- state_drive.cpp
  - This will contain your general function prototypes and main. This also holds the function definitions after main.
- state.h
  - This will contain your state definition only
- state.cpp
  - This will contain your state member function definitions only
- state.txt
  - This contains your input data that will populate your array
- a5.txt
  - This contains your results of your program

Program Requirements:
First, your program may NOT contain any global variables. Main() will be responsible for calling functions. Therefore, main() will be a series of function calls along with any necessary variable declarations you will need for you program.

main()
Your main() function should perform the following tasks:

- Declare any variables you will need for your program, including the array
- Declare and open two filestreams (one for reading, one for writing)
- Loop x number of times
    - Read a state *
    - Load the array, using the next available array location *
- End loop
- Loop
    - Prompt the user for the state name or abbreviation *
    - Search the array *
    - Display result (console and output file) *
- End loop
- Close both filestreams
- End program

- **void readData(State &s, fstream &inFile)**
    - This function will accept an empty **State** object by reference and load it with the next state's information from file. The filestream is received by reference.
    - You will need to read one line at a time from file, and then call the appropriate member function to set the data member
    - DO NOT close the file

- **void loadArray(State s, State stateArray[], int SIZE, int &current)**
    - This function will accept the most recent State object just read from file, the array of State objects, an integer that represents the size of the array, and the current array index.
    - The current array index is what should be used when inserting into the array.
    - After inserting the object into the array, you will need to increment current.

- **void promptUser(string &s)**
    - This function accepts a string by reference that will hold either the state name or the state abbreviation that has been entered by the user.
    - You will prompt the user to enter either a state name or a state abbreviation and you will search the array to see if it exists in the list.
    - Once the user has provided input, you will need to validate the data. Meaning, you will have to make sure that if they entered an abbreviation it is all uppercase. If they entered a state name that it is upper and lower case. If not, you will not get a match when searching.
    - Do not assume they will enter this properly.

- **bool searchName(State stateArray[], int SIZE, string name, int &location)**
  - This function accepts the State array, the name to search, and an integer by reference which represents the location (index) in the array of where the State is located, if found.
  - The function will perform a sequential search of the array beginning with the first item and compare each object to the name.
  - If found, you will set the location to the current index and return "true"
  - If not found, you will set the location to '-1' and return "false"

- **bool searchAbbr(State stateArray[], int SIZE, string abbr, int &location)**
  - This function accepts the State array, the abbreviation to search, and an integer by reference which represents the location (index) in the array of where the State is located, if found.
  - The function will perform a sequential search of the array beginning with the first item and compare each object to the abbreviation.
  - If found, you will set the location to the current index and return "true"
  - If not found, you will set the location to '-1' and return "false"

- **void outputResult(State stateArray[], string s, int location, bool result, fstream &outFile)**
  - This function will report on the result from one search at a time. <u>This info should go to both the console and file so the user can see the results and we record the results in the file.</u>
  - The function receives the state array, the user's input (abbr or name), the location, the result (bool) and the output filestream.
  - You will output what the user entered, then report whether it was found or not depending on the Boolean.
  - If found, output the state's info by using location and calling the "get" member functions.
  - If not found, tell the info was not found.
  - See sample output below.

Running Your Program

You should run your program at **least** three times using the following data:

First: Hawaii
Second: WI
Third: MO

Sample Output:

```
User entered: California
Result:       Found

Name:      California
Abbr.:     CA
Capitol:   Sacramento
Statehood: Sept. 9, 1850
Population:37253956
Nickname:  Golden State


User Entered:  Wisconsin
Result:        Not Found
```

Please make sure you include the appropriate program header information at the top of the first page of your program (see the department guidelines). When you are finished, compile, and then run the program. If it does not compile, fix any errors and try again. When your program is working, submit your **state_drive.cpp, state.h, state.cpp** files along with your output file, **a5.txt** by midnight on **WEDNESDAY, OCT. 5TH.**