Week 3 Deliverables

Overview: In this week, you have studied additional Python language syntax including Lists, Sequences, Dictionaries and Sets. The Lab for this week demonstrates your knowledge of this additional Python functionality. Be sure to use these powerful data structures you studied this week when creating your code.

Submission requirements for this project include multiple files. (Zipping them into one file is acceptable and encouraged):

- Python State Capital and Flower List Application Code
- Word or PDF file containing your test and pylint results
- Flower image set – These images should be uploaded supporting the testing of your lab.

Python Application for this Lab: (total 100 points):

The first exercise produces a command line menu-driven python application providing users with the ability to search and display U.S. State Capital, population and Flowers. The second part documents your testing and pylint analysis results.

1. **(80 points)** Python command line menu-driven application that allows a user to display, sort and update, as needed a List of U.S states containing the state capital, overall state population, and state flower. The Internet provides multiple references with these lists. For example:

https://www.crestcapital.com/tax/us_states_and_capitals

https://statesymbolsusa.org/categories/flower

https://worldpopulationreview.com/states/state-capitals/

You will need to embed the State data into your Python code in a data structure of your choice, from the readings this week. The user interface will allow the user to perform the following functions:

```
  1. Display all U.S. States in Alphabetical order along with the
Capital, State Population, and Flower

  2. Search for a specific state and display the appropriate Capital
name, State Population, and an image of the associated State Flower.

  3. Provide a Bar graph of the top 5 populated States showing their
overall population.

  4. Update the overall state population for a specific state.

  5. Exit the program
```

As before, generate an appropriate Welcome, prompt, and exit messages to help the user navigate the program.

The program should continue to allow selections until the program is exited.

If a state is not found an appropriate message should be displayed.

Hints:

1. Use the List data structure and associated sort() and searching capabilities
2. Create and use functions as often as possible.
3. Validate input data to ensure each entry from the user is correct before proceeding.
4. Prompt the user to reenter information as needed.
5. The following Python sites are excellent resources for learning more about the Python libraries mentioned in the readings that you should use as part of this exercise.
    a. https://matplotlib.org/tutorials/introductory/pyplot.html
    b. https://matplotlib.org/tutorials/introductory/images.html#sphx-glr-tutorials-introductory-images-py
6. Use comments to document your code
7. Test with many combinations.
8. Use pylint to verify the code style – the goal is a 10!
9. Before you import a third part library (e.g. matplotlib))you must install it. To install a Third Party library, you use this command at the command prompt:

```
python -m pip install -U matplotlib
```

2. **(20 points)** Document your testing results using your programming environment. You should also include and discuss your pylint results for the application. The test document should include a test table that includes the input values, the expected results and the actual results. A screen capture should be included that shows the actual test results of running each test case found in the test table. Be sure to include multiple test cases to provide full coverage for all code and for each function you develop and test.

Any submissions that do not represent work originating from the student will be submitted to the Dean's office and evaluated for possible academic integrity violations and sanctions.