Lab 3

Terrence Jackson

SDEV 300: Building Secure Python Applications

Professor Howard

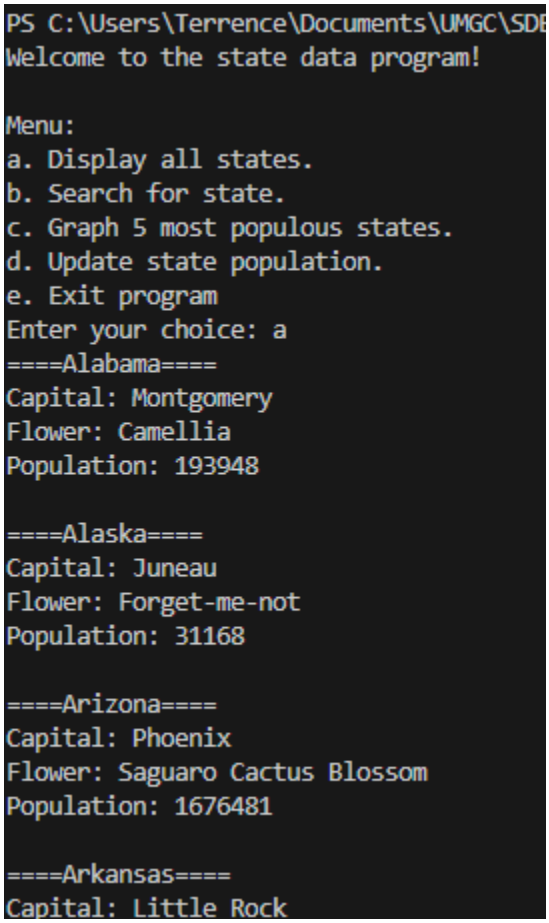
Apr 2, 2024

Test Cases

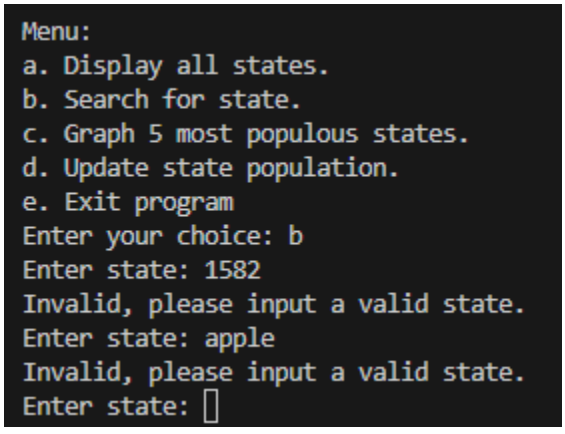
	Input	Expected	Actual	Pass?
Test Case 1: Display All States	Selection: a	Display all U.S. States in Alphabetical order with Capital, State Population, and Flower	Display all U.S. States in Alphabetical order with Capital, State Population, and Flower	Yes
Test Case 2: Search for an Invalid State	Selection: b State: 1582 State: apple	Invalid, prompt again Invalid, prompt again	Invalid, prompt again Invalid, prompt again	Yes
Test Case 3: Search for a Valid State	Selection: b State: Montana	====Montana==== Capital: Helena Flower: Bitterroot Population: 35540 Display image of the state flower.	====Montana==== Capital: Helena Flower: Bitterroot Population: 35540 Display image of the state flower.	Yes
Test Case 4: Display Population Graph	Selection: c	Display a Bar graph of the top 5 populated States, showing their overall population.	Bar graph of Arizona, Texas, Ohio, Indiana, Colorado	Yes
Test Case 5: Update Valid Population	Selection: d State: ct Pop: 123456789	====Connecticut==== Capital: Hartford Flower: Mountain Laurel Population: 123456789	====Connecticut==== Capital: Hartford Flower: Mountain Laurel Population: 123456789	Yes

Test Case 6: Update Invalid Population	Selection: d State: rhodeisland Pop: -15 Pop: tangerine Pop: 18.632	Invalid, prompt again Invalid, prompt again Invalid, prompt again	Invalid, prompt again Invalid, prompt again Invalid, prompt again	Yes
Test Case 7: Update Bar Graph	Selection: d State: AriZonA Pop: 55 Selection: c	Display a Bar graph of the top 5 populated States, updated to remove Arizona	Bar graph of Texas, Ohio, Indiana, Colorado, and Oklahoma	Yes
Test Case 8: Invalid Selection	Selection: pear	Invalid, prompt again	Invalid, prompt again	

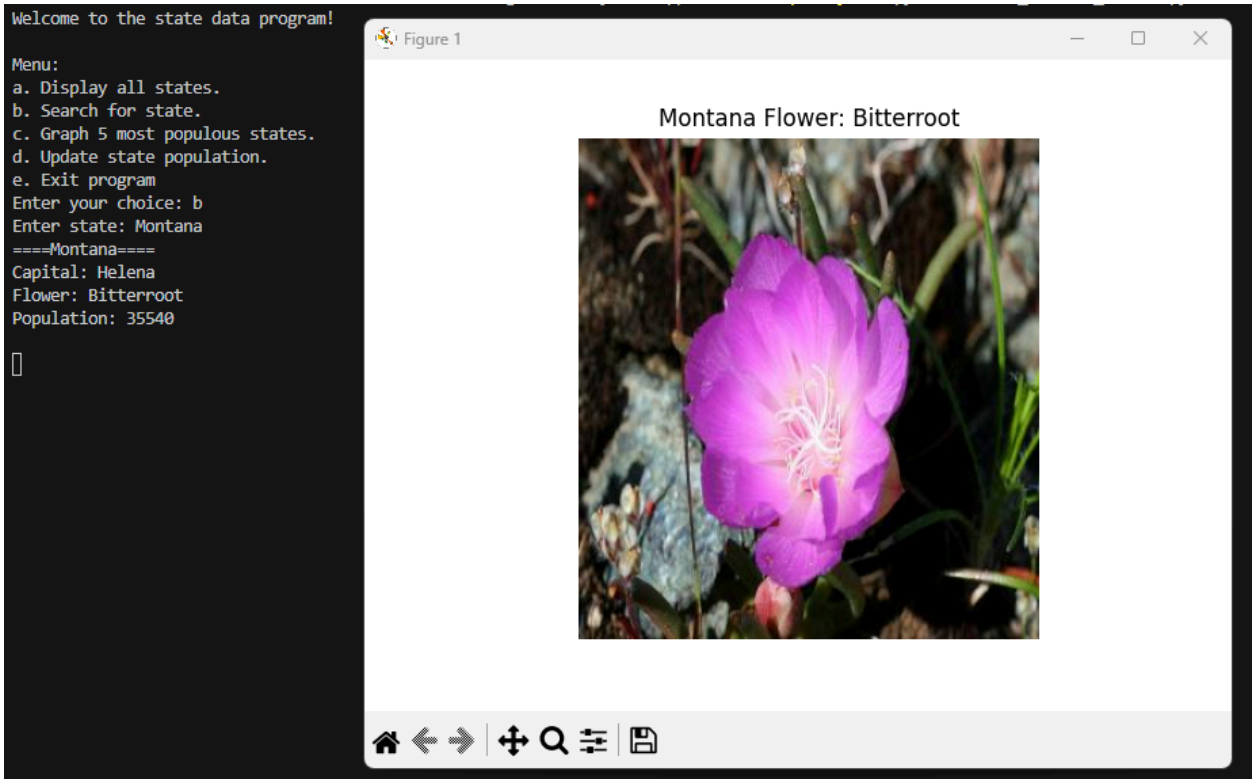
Test Case 1:



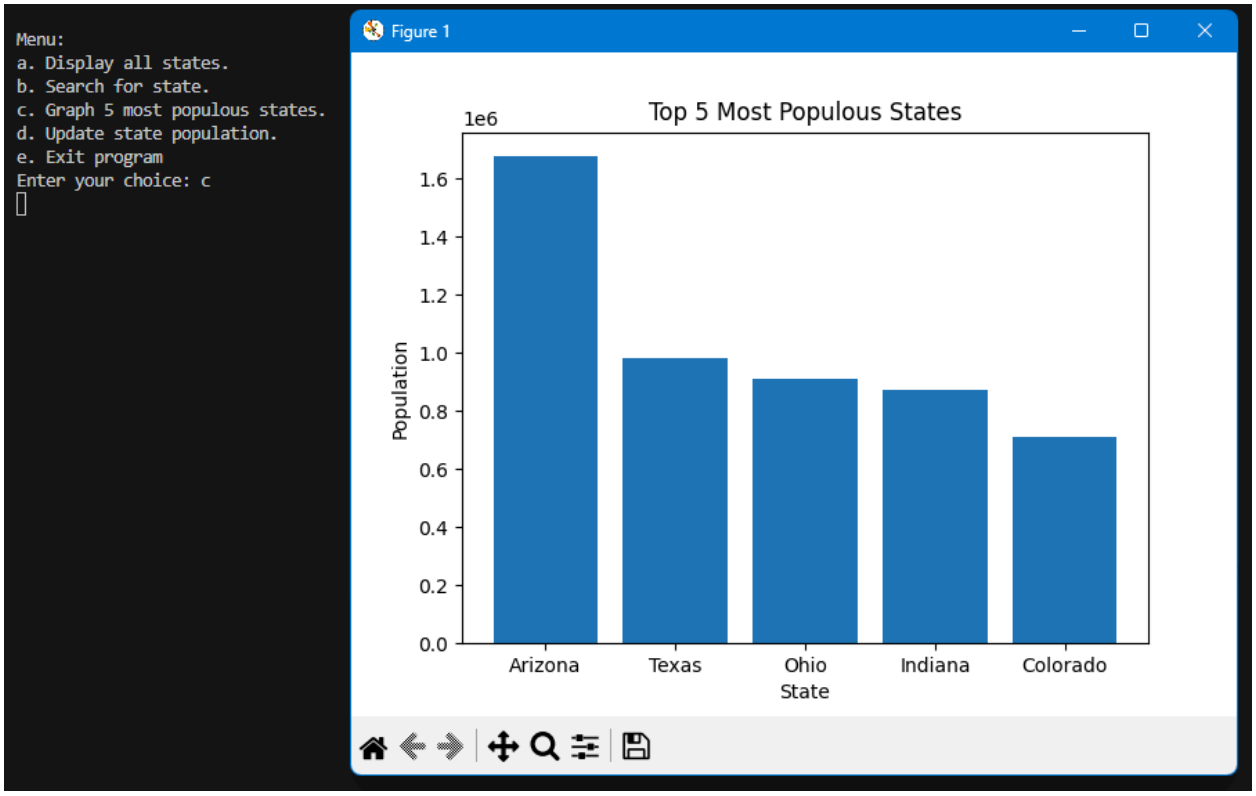
Test Case 2:



Test Case 3:


Welcome to the state data program!
Menu:
a. Display all states.
b. Search for state.
c. Graph 5 most populous states.
d. Update state population.
e. Exit program
Enter your choice: b
Enter state: Montana
====Montana====
Capital: Helena
Flower: Bitterroot
Population: 35540
Figure 1
Montana Flower: Bitterroot

Test Case 4:


Menu:
a. Display all states.
b. Search for state.
c. Graph 5 most populous states.
d. Update state population.
e. Exit program
Enter your choice: c
Figure 1
Top 5 Most Populous States
1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Population
Arizona
Texas
Ohio
Indiana
Colorado
State

Test Case 5:

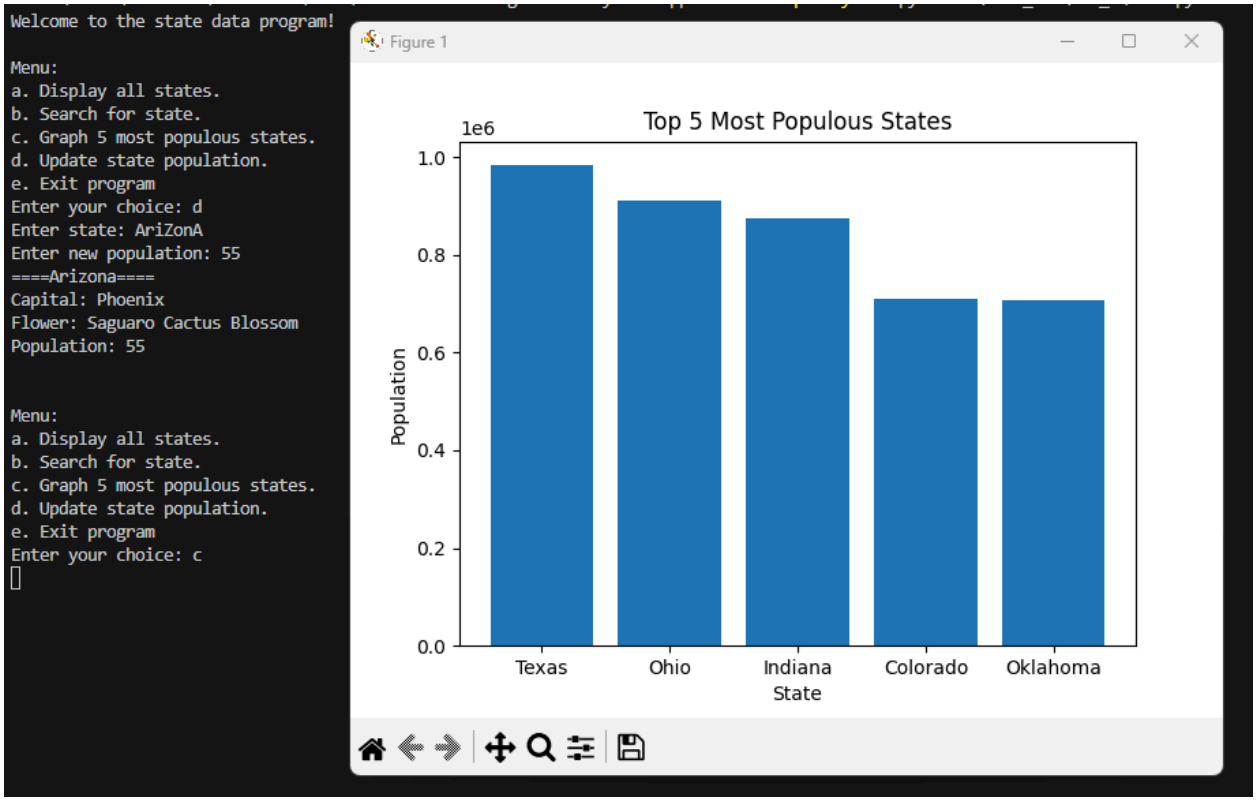
```
Menu:
a. Display all states.
b. Search for state.
c. Graph 5 most populous states.
d. Update state population.
e. Exit program
Enter your choice: d
Enter state: ct
Enter new population: 123456789
====Connecticut====
Capital: Hartford
Flower: Mountain Laurel
Population: 123456789


Menu:
a. Display all states.
b. Search for state.
c. Graph 5 most populous states.
d. Update state population.
e. Exit program
Enter your choice:
```

Test Case 6:

```
Menu:
a. Display all states.
b. Search for state.
c. Graph 5 most populous states.
d. Update state population.
e. Exit program
Enter your choice: d
Enter state: rhodeisland
Enter new population: -15
Invalid, please input an integer number greater than 0.
Enter new population: tangerine
Invalid, please input an integer number greater than 0.
Enter new population: 18.632
Invalid, please input an integer number greater than 0.
Enter new population:
```

Test Case 7:

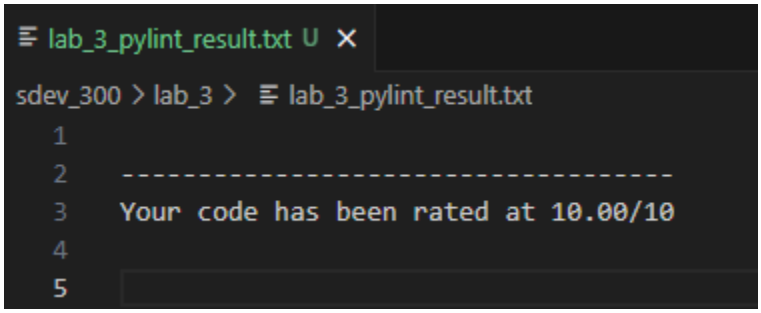


Test Case 8:

```
Menu:
a. Display all states.
b. Search for state.
c. Graph 5 most populous states.
d. Update state population.
e. Exit program
Enter your choice: pear
Invalid choice. Please enter a valid option.

Menu:
a. Display all states.
b. Search for state.
c. Graph 5 most populous states.
d. Update state population.
e. Exit program
Enter your choice:
```

Pylint



I reused a lot of code from Lab 2 in this lab. I've also learned a lot from Pylint throughout the last few assignments. Pylint did not have any issues to solve once I had finished fleshing out my ideas. However, I did make one change from Lab 2. When I had finished my initial coding in Lab 2, Pylint was throwing errors about how many local variables my main function had, so I abstracted parameter collection into a new function and passed the parameters through a single list variable. In this assignment, I initially kept the parameter collecting function. As I kept working on it, I realized that there were significantly less parameters to collect. I chose to move my parameter collection back into the main function, and luckily found that Pylint did not have any issues with the number of local variables.

I am also using an extension called SonarLint, which functions similarly to Pylint. This linter had an influence over how I implemented my global variables. I am used to working with pandas data frames, so I set up a nested dictionary to act similarly with "column names". I had originally set it up with each entry looking like this:

```
"AK": {
        "Full Name": "Alaska",
        "Capital": "Juneau",
        "Flower": "Forget-me-not",
        "Population": 31168,
},
```

However, once I had filled out the entire dictionary of states, SonarLint threw a few warnings that I should "Define a constant instead of duplicating this literal" for each of my column names. Due to this, I created a few global variables to hold the string literals for my column names.