ITSE 1359 – Lab 6 Assignment (Refer to Ch. 6 as needed):

LAB 6: This lab requires you to work on and submit two programs. Read this document carefully and comply with submission and grading criteria. Your first problem (State Capitals) has a video to help you develop it. See last page for the video link. As for the second program (BMI Dictionary), use the comments/pseudocode provided to build your program. Focus on requirements and do not deviate. If you want to deviate do so on your own.

Program 1 – State Capitals (states_to_capitals):

Problem Definition: Create a python program that utilizes a dictionary to display four state capitals. Name your file: states_to_capitals.py. Do not forget general and specific comments.

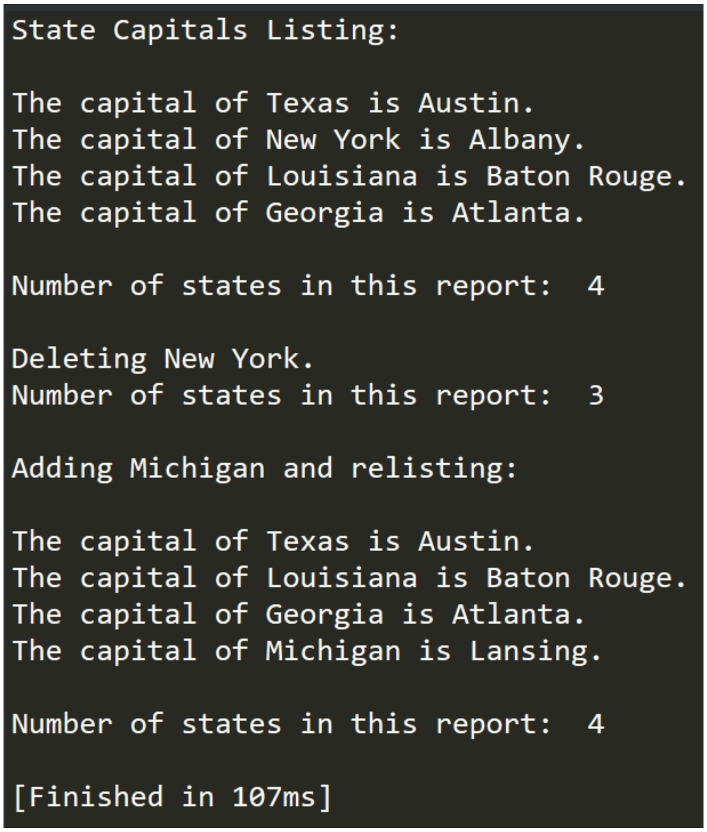
You can use states and capitals I have provided or use your own. Create your dictionary using key-value pairs for state and capital, both of which will have to be entered into the dictionary in lower case.

Use a for in Loop with the items() method to traverse the dictionary and build the output seen in the screenshot below. Your sentence structure must match that in screenshot with only state and capitals varying. After you show your state capitals, show number of states using the len() function. Delete one state (New York – in my example) and show number of states again. Add another state (Michigan – in my example) and relist and show number again.

Specific Comments/Pseudocode:

```
# print report header – State Capital Listing
# create dictionary of key-value pairs of state and capital (lower case)
# use for in Loop to show state and capital detail line (title case)
# show number of states using len() function
# delete one state (new york) and show number of states again
# add one state (michigan) to the dictionary
# relist dictionary using for in Loop
# show number of states again
```

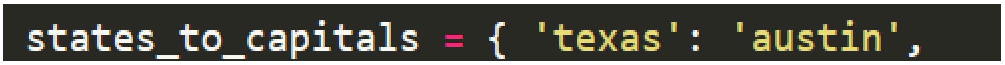
Screenshot of Output:



```
State Capitals Listing:

The capital of Texas is Austin.
The capital of New York is Albany.
The capital of Louisiana is Baton Rouge.
The capital of Georgia is Atlanta.

Number of states in this report:  4

Deleting New York.
Number of states in this report:  3

Adding Michigan and relisting:

The capital of Texas is Austin.
The capital of Louisiana is Baton Rouge.
The capital of Georgia is Atlanta.
The capital of Michigan is Lansing.

Number of states in this report:  4

[Finished in 107ms]
```

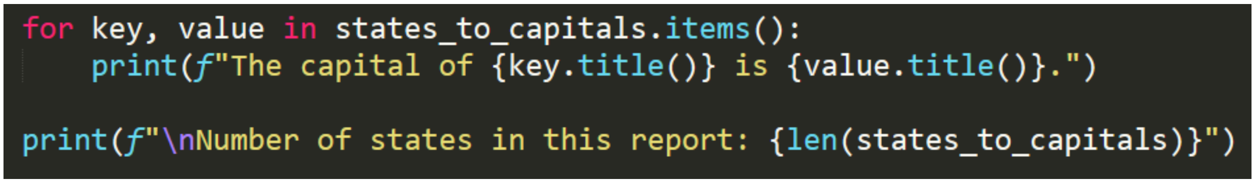
Code Examples (States Capitals):

How to create dictionary? (note key-value pair entered lower case)



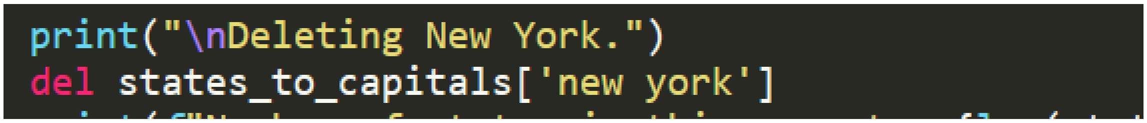
```
states_to_capitals = { 'texas': 'austin',
```

How to loop the dictionary? (note title case)



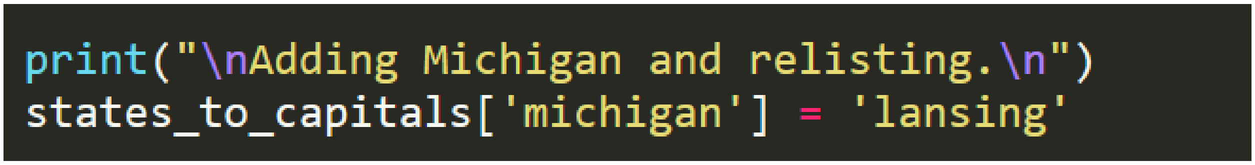
```
for key, value in states_to_capitals.items():
    print(f"The capital of {key.title()} is {value.title()}.")

print(f"\nNumber of states in this report: {len(states_to_capitals)}")
```

How to delete a state? (note del statement used to delete a state)



```
print("\nDeleting New York.")
del states_to_capitals['new york']
```

How to add a state? (assignment statement; use print() or \n for blank line)



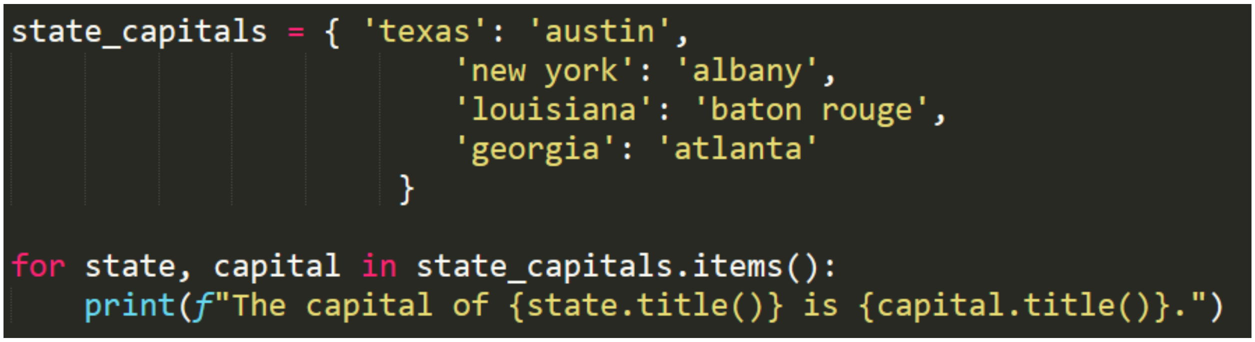
```
print("\nAdding Michigan and relisting.\n")
states_to_capitals['michigan'] = 'lansing'
```

Second Thoughts (Optional update):

After I did the video on this States to Capitals lab assignment, I think the file name would be better called “state_capitals.py” and the dictionary be called the same – “state_capitals”.

I also think the loop should use “state” instead of “key” and “capital” instead of “value”. This would match the conventions in the book better. For example, look at the “favorite_languages.py” program in your book. Note that the book says that using “descriptive names” rather than the words “key” and “value” is preferred.

Optional Update:



```
state_capitals = { 'texas': 'austin',
                        'new york': 'albany',
                        'louisiana': 'baton rouge',
                        'georgia': 'atlanta'
                     }

for state, capital in state_capitals.items():
    print(f"The capital of {state.title()} is {capital.title()}.")
```

So, you can do it either way – as in the video or as recommended here. You will be using better coding style if you make these changes.

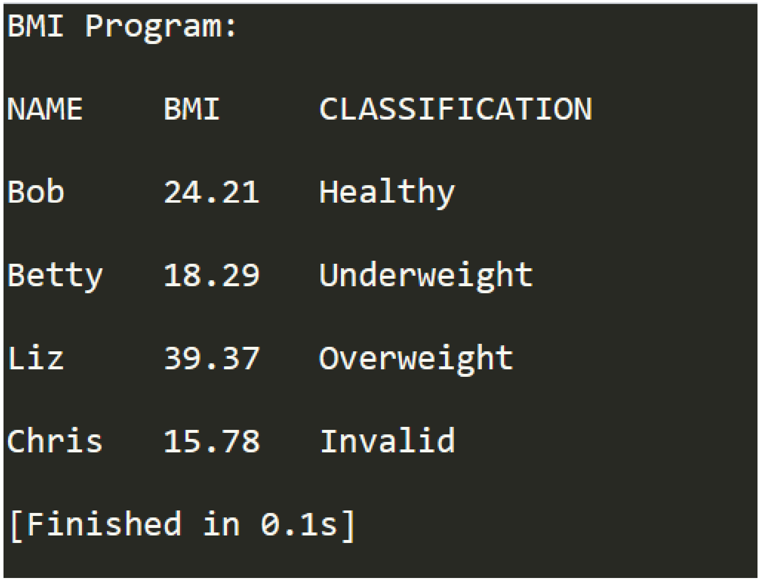
Program 2 – BMI with List and Dictionaries(bmi_dict.py):

Problem Statement: This program will start with your lab 5 program – bmi.py. Save this file as bmi_dict.py. In this program you are to use a list and dictionaries to get the same output as before. Refer to that problem statement to review general purpose and design of program.

Coding: First, define four patient dictionaries with the key-value pairs of name, height, and weight. Use the same data as before for testing purposes. Name these dictionaries: patient_0, patient_1, patient_2, patient_3. Store these dictionaries in a list called patients. Print the program name and report header line. Loop through the list, calculate bmi, find classification, and print the detail line showing patient name, bmi, and classification.

Output: Your output should match the output from lab 5, except instead of using Java style parallel lists, you will solve this problem using Python dictionaries and lists. Refer to chapters 3, 4, and 6 for discussion of lists and dictionaries. You may also want to refer to the internet for syntax examples. W3shools.com offers excellent examples of Python coding structures.

Screen shot (bmi_dict.py):



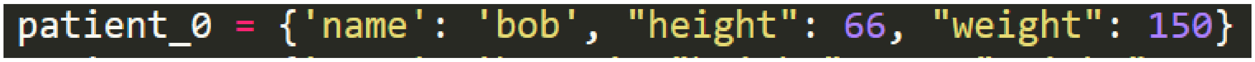
```
BMI Program:

NAME    BMI     CLASSIFICATION

Bob     24.21   Healthy

Betty   18.29   Underweight

Liz     39.37   Overweight

Chris   15.78   Invalid

[Finished in 0.1s]
```

Comments/Pseudocode (bmi_dict.py):

```
# define & initialize 4 dicts to represent patients
# store the dicts in a list
# print program name and report header line
# loop through list
    #calc bmi by accessing list of dicts
    #find classification using elif structure
    #print detail line showing patient name and classification
```

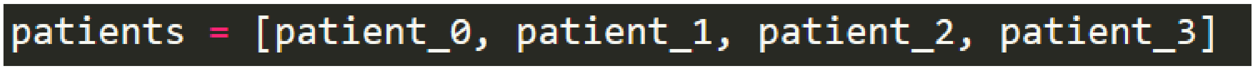
Code Examples (bmi_dict.py):

How to define and initialize a dict?



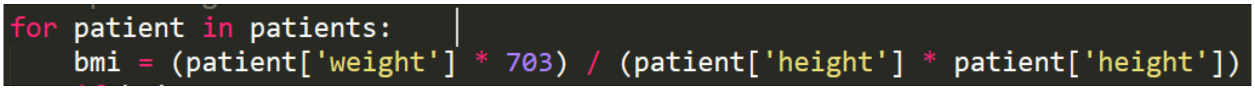
```python
patient_0 = {'name': 'bob', "height": 66, "weight": 150}
```

How to store the dicts in a list?



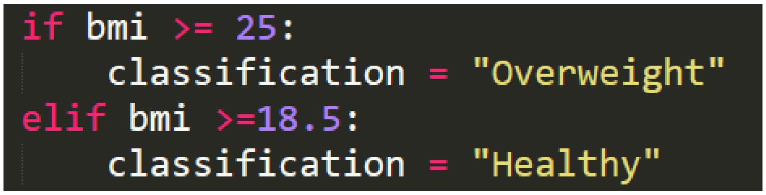
```python
patients = [patient_0, patient_1, patient_2, patient_3]
```

How to loop through list and calc bmi?



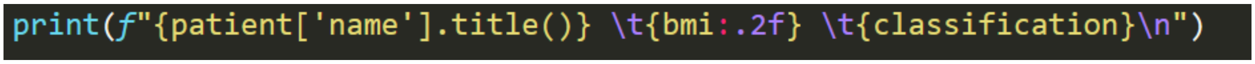
```python
for patient in patients:
    bmi = (patient['weight'] * 703) / (patient['height'] * patient['height'])
```

How to find classification? – partial listing:



```python
if bmi >= 25:
    classification = "Overweight"
elif bmi >=18.5:
    classification = "Healthy"
```

How to print detail line showing patient name and classification?



```python
print(f"{patient['name'].title()} \t{bmi:.2f} \t{classification}\n")
```

Submit your lab assignment:

Using the Canvas assignment tool, upload your completed work (2 files) to the lab 6 assignment. Attach the first file (state_capitals.py) and then the second (bmi_dict.py) and then submit.

All lab assignments must be submitted using the CANVAS assignments tool. Lab Assignments will not be accepted any other way**.** Make sure you submit your work to the right lab assignment number otherwise you will not get credit.

Grading Criteria:

- ✓ You must use the provided video to do the States Capitals program.
- ✓ Don’t forget general comments.
- ✓ Specific comments are optional.
- ✓ Use white space to make your code easy to read.
- ✓ Comply with PEP-8 conventions for variable names, file names, etc.
- ✓ Satisfy the problem definition and other grading standards.
- ✓ Your work should not have syntax errors.
- ✓ Your work must be your own.
- ✓ Match your output screen to screenshot provided.
- ✓ Any deviations from lab specifications will result in points off.
- ✓ If you want to vary, do so on your own.
- ✓ Any deviations from lab specifications will result in points off.
- ✓ If you want to vary, do so on your own.

VIP Videos:

State capitals program (step-by-step): Lab 6 state capitals Video

Lectures and other videos: VIEW MY PYTHON PLAYLIST

Closing:

If you have questions about this lab send me a message using canvas inbox or attend zoom office hours. See Unit 0 for link for my office hours. Programming tutors are also available and listed in your Canvas class as announcements.

By-the-way, a great way to get ready for your lab assignments (and exams) is do the Try It Yourself problems in your book. Most of the solutions are on the authors website and I also discuss them in my YouTube video lectures.

Another way to get ready for your lab assignments is to review my lecture notes for each chapter. In Unit 0, find the link for my chapter notes.

Warning: Labs will increase in complexity with each lab assignment. A beginner may need 5 to 15 hours to complete each lab. Please start early because there will be no extensions.

Consider doing this lab over a three-day period:

- ✓ Day 1 – Get confused and run out of time.
- ✓ Day 2 – Research and debug errors.
- ✓ Day 3 – Polish, double-check everything, and submit.

Have an exception free day!

Prof. Benavides