

### 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.

**Be sure to develop and test your Python code in the AWS Cloud9 IDE provided for the class.**

You should continue to use the PEP Python Style guide mentioned in the book and found here:

<https://www.python.org/dev/peps/pep-0008/>

Some examples of Python Coding Style best practices include:

- Limit all lines to a maximum of 79 characters.
- Imports are always put at the top of the file, just after any module comments and before module globals and constants.
- Use 4 spaces for indentation.

**Submission requirements for this project include 3 files. (Zipping them into one file is acceptable and encouraged):**

- Python State Capital and Bird List Application Code
- Python Math Set Application Code
- Word, Excel or PDF file containing your test results

### Python Applications for Lab3: (total 100 points):

This lab consists of two parts.

The first exercise **(40 points)** allows a user to display, sort and update as needed a List of U.S States containing the State Capital and State Bird. The Internet provides multiple references with these lists. For example:

[https://www.crestcapital.com/tax/us\\_states\\_and\\_capitals](https://www.crestcapital.com/tax/us_states_and_capitals)

<http://www.thewebmap.com/FiftyStates/>

You will need to embed the State data into your Python code. The user interface will allow the user to perform the following functions:

1. Display all U.S. States in Alphabetical order along with Capital and Flower
2. Search for a specific state and display the appropriate Capital and Bird
3. Update a Bird for a specific state
4. 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. Use comments to document your code

For the second exercise **(40 points)** determine the union, intersection and difference of the square and cube of integers ranging from 1 to 100. Sets are clearly the way to go here. You can use Math functions to generate the sets for the square and cube for the Integers.

The following functionality should be available for the user via a simple interface:

1. Display the Square and Cube for Integers ranging from 1 to 100
2. Search the sets for a specific Integer and display the Square and Cube values
3. Display the Union of Square and Cube sets
4. Display the Intersection of Square and Cube sets
5. Display the Difference of Square and Cube sets
6. Exit the program

If an Integer is not found an appropriate message should be displayed

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

Hints:

1. Use Sets and their associated union(), intersection() and difference() methods
2. Create and use functions as often as possible
3. Use comments to document your code

3. Document your results for each application within the AWS Cloud9 classroom environment. Provide screen captures and descriptions for each test cases you provide for each of your applications. Be sure to go through each possible user interface combination in your test cases. **(20 points)**

**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.**