CS1026: Assignment 2 – Volume Calculator

Due: October 25th 2017 at 9:00pm
Weight: 8%

Learning Outcome:
By completing this assignment, you will gain skills relating to

- using loops,
- using functions,
- and using lists in Python.

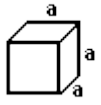
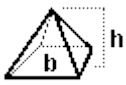
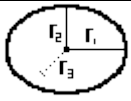
Task:
In this assignment, you will write a **complete** program in Python that computes the volume for a number of different shapes. Your program is expected to prompt the user for input and validate it before computing the volume. In addition, your program should keep track of each volume that is calculated and at the end display the volume for all the shapes. Your program should make use of functions, loops, and lists.

Functional Specifications:

1. Your program should first ask the user for the shape they are interested in, check to make sure that their input is valid; if the input is not valid, keep on asking them until valid input is given. **Valid input options are**: "cube", "pyramid", "ellipsoid", "quit"; you should accept the input in any combination of upper and lower case letters.

2. Based on the users input, prompt them for the necessary dimensions; you may assume that the user enters positive integer values.

3. Next your program should calculate the volume for the specified shape given the dimensions, round to 1 decimal place, and display the volume to the screen. You do not need to display the unit of measurement. In addition, your program should also store the volume.

Shape	Volume
cube	 $volume = a^3$ where a is the length of a side
pyramid	 $volume = \frac{1}{3}b^2h$ where b is base length and h is height
ellipsoid	 $volume = \frac{4}{3}pi * r_1 * r_2 * r_3$ where pi is π and r is used to represent each radius

Sample output

```
The volume of a pyramid with a base of 10.0 and height of 12.0 is 400.0
```

4. Your program should **continue** to compute volumes as specified by the user until the user enters "quit". At this point in time, your program should print the volumes based on their shape and in ascending order.
   If no calculations are done for a specific shape then a statement should be printed. For instance if there were no pyramid calculations the output would be as shown below.

   Sample final output

```
You have come to the end of the session.
The volumes calculated for each shape are shown below
Cube: 8.0, 27.0, 29.7, 35.9, 216.0
Pyramid: No computations for this shape
Ellipsoid: 45.1, 251.3
```

   If the user enters ***quit*** option before actually calculating any volumes show the output below.

```
You have come to the end of the session.
You did not perform any volume calculations.
```

5. In order to develop modular code, you **MUST** encapsulate the prompting for input and calculation of volume for each shape within a function. That is, you MUST have at least 3 functions, one for cube, another for pyramid, and a third for ellipsoid.

Non-functional Specifications:

1. Include brief comments in your code identifying yourself, describing the program, and describing key portions of the code.
2. Assignments are to be done individually and must **be your own work**. Software may be used to detect cheating.
3. Use Python coding conventions and good programming techniques, for example:
   - Meaningful variable and function names
   - Conventions for naming variables and constants
   - Use of constants where appropriate
   - Readability: indentation, white space, consistency

The name of the file you submit should be your UWO userid_Assign2.py. For instance, my assignment would be oola_Assign2.py. Make sure you attach your python file to your assignment; DO NOT put the code inline in the textbox.

Make sure that you develop your code with Python 3.6 as the interpreter. TAs will not endeavor to fix code that uses earlier versions of Python.

What You Will Be Marked On:

- Functional specifications:
    - Does the program behave according to specifications?
    - Does the program handle invalid input?
    - Do you use functions effectively?
    - Is the output according to specifications?
- Non-functional specifications: as described above
- Assignment submission: via the OWL, though the assignment submission in OWL.