W02 Team Activity: Can Efficiency

Instructions

Each lesson in CSE 111 contains a team activity that is designed to take about one hour to complete. You should prepare for all team activities by completing the preparation material and the individual checkpoint assignment before starting a team activity. The goal of the team activities is for students to work together and teach and learn from each other. As your team completes a team activity, instead of moving through it as quickly as you can, you should help everyone understand the concepts.

As your team works through a team assignment in a synchronous video meeting, be certain that each student can see the code that is being typed. When your team completes the video meeting, share the final Python file with everyone on your team.

Purpose

Strengthen your understanding of user-defined functions, parameters, arguments, and local variable scope by writing a program that has three or more functions.

Problem Statement

In many countries, food is stored in steel cans (also known as tin cans) that are shaped like cylinders. There are many different sizes of steel cans. The *storage efficiency* of a can tells us how much a can stores versus how much steel is required to make the can. Some sizes of cans require a lot of steel to store a small amount of food. Other sizes of cans require less steel and store more food. A can size with a large storage efficiency is considered more friendly to the environment than a can size with a small storage efficiency.

The storage efficiency of a steel can is computed by dividing the volume of a can by its surface area.

$$storage\_efficiency = \frac{volume}{surface\_area}$$

In other words, the storage efficiency of a can is the space inside the can divided by the amount of steel required to make the can. The formulas for the volume and surface area of a cylinder are:

$$volume = \pi \, radius^2 \, height$$

$$surface\_area = 2\pi \, radius \, (radius + height)$$

- π is the constant PI, the ratio of the circumference of a circle divided by its diameter (use **`math.pi`**)
- *radius* is the radius of the cylinder
- *height* is the height of the cylinder

Helpful Documentation

- The preparation content explains how to write functions.
- The preparation content local variable scope.
  - The Python **`math`** module contains mathematical constants and functions including **`math.pi`**.

Assignment

Work as a team to write a Python program named **`can_sizes.py`** that computes and prints the storage efficiency for each of the following 12 steel can sizes. Then visually examine the output and answer this question, "Which can size has the highest storage efficiency?"

Name	Radius (centimeters)	Height (centimeters)	Cost per Can (U.S. dollars)
#1 Picnic	6.83	10.16	$0.28
#1 Tall	7.78	11.91	$0.43
#2	8.73	11.59	$0.45
#2.5	10.32	11.91	$0.61
#3 Cylinder	10.79	17.78	$0.86
#5	13.02	14.29	$0.83
#6Z	5.40	8.89	$0.22

Name	Radius (centimeters)	Height (centimeters)	Cost per Can (U.S. dollars)
#8Z short	6.83	7.62	$0.26
#10	15.72	17.78	$1.53
#211	6.83	12.38	$0.34
#300	7.62	11.27	$0.38
#303	8.10	11.11	$0.42

If you separate your program into functions, this problem will be much easier to solve than if you don't separate it into functions. You are free to write any functions that you choose in your program, but we *strongly* suggest that your program include at least these three functions:

- **`main`**
- **`compute_volume`**
- **`compute_surface_area`**

Core Requirements

1. Your program must compute the volume of all 12 can sizes.
2. Your program must compute the surface area of all 12 can sizes.
3. Your program must compute and print the storage efficiency of all 12 can sizes.

Stretch Challenges

If your team finishes the core requirements in less than an hour, complete one or more of these stretch challenges. Note that the stretch challenges are optional.

1. Add another function named **`compute_storage_efficiency`** to your program. This function should call the **`compute_volume`** and **`compute_surface_area`** functions and then compute and return the storage efficiency of a steel can size. Replace code in the **`main`** function with a call to the **`compute_storage_efficiency`** function as appropriate. Did adding and calling the **`compute_storage_efficiency`** function reduce the number of lines of code in your program?
2. The table of can sizes that appears in the Assignment section above includes a column that contains the cost per can of each steel can size. Add another function to your program named **`compute_cost_efficiency`** that computes and returns the volume of a steel can divided by its cost. Write code to call the **`compute_cost_efficiency`** function and print the cost efficiency for each can size.

Then visually examine the output and answer this question, “Which can size has the highest cost efficiency?”

3. If you remember how to use lists and a for loop from CSE 110, rewrite your **`main`** function so that it uses a list or lists that contain the can size names and dimensions. Then write a loop that processes the values in the list.
4. Add **`if`** statements inside the loop to automatically determine which can size has the best storage efficiency and which can size has the best cost efficiency.

Testing Procedure

Verify that your program works correctly by following each step in this testing procedure:

1. Run your program and verify that it prints the correct results, rounded and formatted as shown below.

```
> python can_sizes.py
#1 Picnic 2.04
#1 Tall 2.35
#2 2.49
#2.5 2.76
#3 Cylinder 3.36
#5 3.41
#6Z 1.68
#8Z short 1.80
#10 4.17
#211 2.20
#300 2.27
#303 2.34
```

Sample Solution

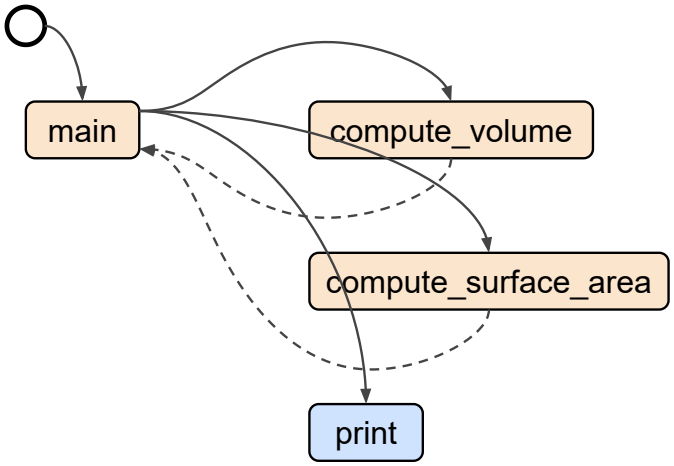
Please work diligently with your team for the one hour meeting. After the meeting is over, please compare your solution to the sample solution or the stretch solution. Please do not look at the sample solution until you have either finished the program or diligently worked for at least one hour. At the end of the hour, if you are still struggling to complete the assignment, you may use the sample solution to help you finish.

Call Graphs

The following call graph shows the function calls and returns in the sample solution for this assignment. From this call graph we see the following function calls:

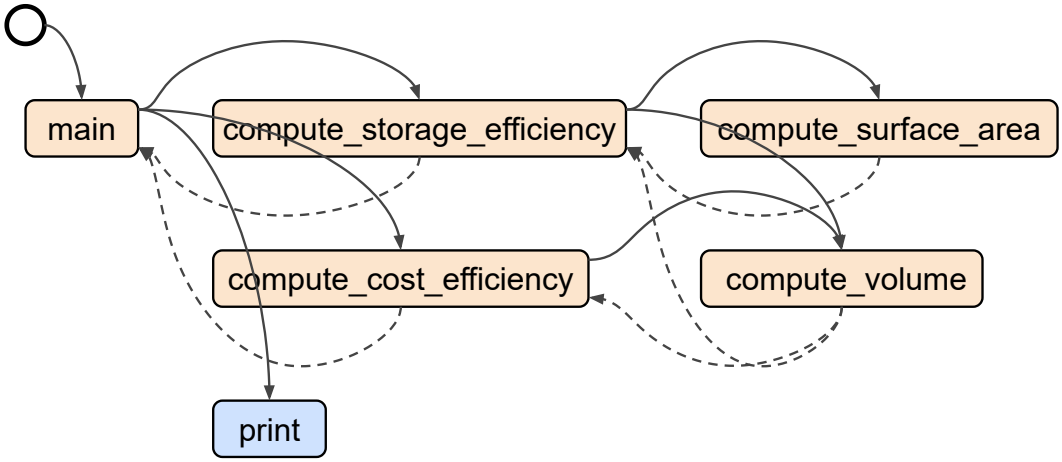
1. The computer starts executing the sample program by calling the **`main`** function.

2. While executing the **`main`** function, the computer calls the **`compute_volume`**, **`compute_surface_area`**, and **`print`** functions.



The next call graph shows the function calls and returns in the sample solution for the stretch challenges of this assignment. From this call graph we see the following function calls:

1. The computer starts executing the stretch program by calling the **`main`** function.
2. While executing the **`main`** function, the computer calls the **`compute_storage_efficiency`** function which calls the **`compute_surface_area`** and **`compute_volume`** functions.
3. While still executing the **`main`** function, the computer calls the **`compute_cost_efficiency`** function which calls the **`compute_volume`** function.
4. Finally, while still executing the **`main`** function the computer calls the **`print`** function.



Ponder

After you finish this assignment, congratulate yourself because you wrote a Python program with at least three user-defined functions. As you wrote your program, what did you learn about organizing a program into functions?

Submission

When you have finished the activity, please report your progress via the associated Canvas quiz. When asked about which of the requirements you completed, feel free to include any work done during the team meeting or after the meeting, including work done with the help of the sample solution, if necessary. In short, report on what you were able to accomplish, regardless of when you completed it or if you needed help from the sample solution.

Submission

When you have finished your team meeting, you are welcome to continue working on your own. Feel free to include that additional work when you report on your progress in Canvas.

When you are finished:

- Return to Canvas to take the quiz.

Useful Links:

- Return to: Week Overview | Course Home