

CS 1803

Individual Homework 1 – Python Practice

Due: Wednesday, January 26th, before 6 PM

Out of 100 points

Files to submit: 1. HW1.py

This is an **INDIVIDUAL** assignment!

Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 1803, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others.

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TA's or use T-Square Forums

Notes:

- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
 - **Do not wait until the last minute** to do this assignment in case you run into problems.
-

Part 1 – Simple Functions

You will write a few python functions for practice with the language. In your HW1.py file, include a comment at the top with your name, section, GTID/Email, and your collaboration statement. Also include each of the following functions.

1. kilometersToMiles
2. sphereVolume
3. timeCleanUp
4. bodyMassIndex

Function Name: **kilometersToMiles**

Parameters:

None

Return Value:

None

Description:

1. Write a user-interactive function to convert kilometers to miles.
2. Get the number of kilometers from the user; make sure to use a descriptive prompt so the user knows what to enter (e.g. "Enter number of kilometers"). You may assume that the user will enter a valid float (e.g. 34.4) and do not have to do

- error checking.
3. Convert the number entered by the user to miles, using the fact that 1 mile = 1.609344 kilometers
 4. Print the calculated number of miles, be sure to add a label to the display value so the user knows what the value means (e.g. display “35 miles” instead of just 35)

Function Name: **sphereVolume**

Parameters:

None

Return Value:

None

Description:

Write a user-interactive function to calculate the volume of a sphere.

1. Get the length of the radius in inches from the user; make sure to use a descriptive prompt so the user knows what to enter
2. Calculate the volume of a sphere with the radius entered by the user; note the volume of a sphere is calculated using the formula $V = (4 / 3) * \pi * r^3$; note, Python has a built-in constant variable for pi, named math.pi (you will have to “import math” before you use this constant).
3. Print the calculated volume; be sure to add a label to display value so the user knows what the value means (e.g. display “Volume of the sphere is 35 inches-cubed” instead of just 35)

Function Name: **timeCleanUp**

Parameters:

None

Return Value:

None

Description:

Write a user-interactive function to convert seconds to the equivalent number of hours, minutes, and seconds.

1. Get the number of seconds as an integer from the user; make sure to use a descriptive prompt so the user knows what to enter.
2. Calculate the total number of hours, minutes, and seconds represented by the original number of seconds answered, using the following hints:
 - There are 60 seconds in a minute.
 - There are 60 minutes in an hour.
 - The mod (a.k.a. remainder) operator in python is % and will show the remainder left after an integer division. It IS useful for this problem!
3. Print the calculated number of hours, minutes, and seconds on **one line**; be sure to add appropriate labels to the display values so the user knows what the value means (e.g. display “5 hours, 2 minutes, 33 seconds”)

Function Name: **bodyMassIndex**

Parameters:

None

Return Value:

None

Description:

Write a user-interactive program to calculate a person's BMI, body mass index.

1. Ask for the person's weight in pounds; make sure to use a descriptive prompt so the user knows what to enter.
2. Ask for the person's height in inches; again, use a descriptive prompt.
3. Calculate the person's BMI using the information they provided and the formula below (you will need to convert the formula into a legal python equation):

$$\text{BMI} = \frac{\text{weight}(\text{lb})}{\text{height}^2 (\text{in}^2)} \times 703$$

4. Print the result. The value must be formatted (both the wording and the number of decimal places) EXACTLY as follows: "Your BMI is 24.9." (where the value for BMI represents whatever you calculated based on the inputs). *You may have to use string formatting to remove extra decimal places from your calculation.*

Part 2 – Complex Functions

Now you will write a few python functions that return data for practice with the language. Also include each of the following functions.

Function Name: kilometersToMilesRedux

Parameters:

1. kilo - a number representing the number of kilometers to be converted; note, may be a float or an int, depending on what value is sent in when the function is called

Return Value:

A floating point number representing the number of miles calculated.

Test Cases:

1. kilometersToMilesRedux(243) --> 150.99319971367214

Description:

Write a function that takes in a number of kilometers, calculates the corresponding number of miles, and *returns* the result. Note: 1 mile = 1.609344 kilometers.

Function Name: calcGas

Parameters:

1. dist – a number representing desired travel distance in miles; note, it may be a

float or an int

2. mpg – an integer representing miles per gallon efficiency of vehicle
3. ppg – a floating point number representing price of gallon of gas in US dollars

Return Value:

A floating point number representing the total cost of gas.

Test Cases:

1. calcGas(64,27,2.65) --> 6.281481481481481
2. calcGas(64,50,2.65) --> 3.392

Description:

Write a function calcGas that will calculate and return the amount of money you will spend on gas given the distance travelled in miles, the miles per gallon of the vehicle being driven, and the price per one gallon of gas.

Function Name: **makeSmores**

Parameters:

1. grahamCrackers - an integer representing the number of graham crackers
2. marshmallows - an integer representing the number of marshmallows
3. chocolate - an integer representing the number of bars of chocolate

Return Value:

An integer representing the amount of smores you can make.

Test Cases:

1. makeSmores(6, 2, 3) --> 2
2. makeSmores(1, 2, 2) --> 0
3. makeSmores(11, 20, 20) --> 5

Description:

Write a function that returns the number of smores you can make based on the number of graham crackers, marshmallows, and chocolate specified as inputs. Assume that it takes the following to make 1 smore:

- 2 graham crackers
- 1 marshmallow
- 1 bar of chocolate

Note that the number of smores must be a whole number; you cannot have 1 1/2 of a smore. For example, if you have 12 graham crackers, 4 marshmallows, but only 3 bars of chocolate, you can only make 3 smores (despite the fact that you have enough graham crackers and marshmallow to make 4 smores). Likewise, if you have 3 graham crackers, 100 marshmallows, and 300 bars of chocolate, you can still only make 1 smore.

Hint: Python has a built-in function called min that takes in a comma separate list of numbers and returns the minimum value. E.g. min(5, 3, 7) returns the number 3. This may be useful in coming up with your solution.

You can solve this problem without using conditionals!

Grading

You will earn points as follows for each function that works correctly according to the specifications.

Simple Function

kilometersToMiles	5
sphereVolume	10
timeCleanup	20
bodyMassIndex	15

Complex Functions

kilometersToMilesRedux	15
calcGas	15
makeSmores	20

If your function works, but prints a value when it is supposed to return it, or returns it when you are supposed to print it, the TA will **deduct half of the points** for that function.