

CS 1301

Individual Homework 1 – Python Practice & Face

Due: Friday, September 3rd, before 6 PM

Out of 100 points

Files to submit: **1. HW1.py**
 2. face.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 1301, 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.*

Graphics Intro

So you're sitting there, looking at your Python window, and you find it to be, well, impersonal. So why don't we add a little personality to it? How about...a face!



Hey! That looks much friendlier. And you can do it all with python! Specifically, that infamous myro thing you've installed. Myro contains a multitude of commands for

programming your robot and making graphics. See the documentation here:

http://wiki.roboteducation.org/Myro_Reference_Manual

For this assignment, we will be focusing on the graphics aspect of myro, so you won't need your robot. Both the link given above, as well as the PDF for the Myro Graphics Reference will be rather helpful.

Part 2 – Making the Face (50 Points)

Files to submit: *face.py*

Your mission, should you choose to accept it (and we recommend that you do), is to make a face in Python. You need to create a program that initializes a graphics window, and draws the face. When you are done, save your program as “face.py”. **Make sure that your filename matches this exactly, or you will lose points!** Don't forget to include any necessary import statements needed to use myro functions.

The face should contain:

- A Head
- 2 Eyes
- A Nose
- A mouth
- At least 2 different colors

Try to keep things in a general face alignment here (Sorry to all you budding Picassos out there), but the shapes can be whatever you want. Be creative!

Grading Rubric

Grading

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

Simple Function	50 points total
kilometersToMiles	5
sphereVolume	10
timeCleanup	20

bodyMassIndex

15

Face

50 points total

50 points

File named correctly (face.py) – 5 points

Creates a graphics window – 10

Draws a head – 6

Draws 2 eyes – 6

Draws a nose – 6

Draws a mouth – 6

Uses at least 2 different colors - 6

The facial features are properly aligned – 5

Total for Face: 50 points

You can earn up to 3 points bonus [discretion of the TAs] for extra creativity/general awesomeness, for a possible total of 103/100.