CS 1301

Individual Homework 1 – Python Practice & Face

Due: Friday August 29th, before 11:55 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. machToFPS
2. sqPyramidVolume
3. makeChange
4. PPIIndex

Function Name: **machToFPS**
Parameters:
None
Return Value:
None
Description:

1. Write a user-interactive function to convert speeds in mach to feet per seconds (FPS).
2. Get the speed in mach(s) from the user; make sure to use a descriptive prompt so

the user knows what to enter (e.g. “Enter the speed in mach”). 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 speed entered by the user to FPS, using the conversion factor below:
   1 mach = 1116.4370079 feet / second

4. Print the converted speed, be sure to add a label to the display value so the user knows what the value means (e.g. display “38405.433071 feet/second” instead of just 38405.433071)

Function Name: **sqPyramidVolume**
Parameters:
None
Return Value:
None
Description:
Write a user-interactive function to calculate the volume of a square pyramid.

1. Get the length of one side of the base in inches from the user; make sure to use a descriptive prompt so the user knows what to enter.
2. Get the height in inches; again, make sure to use a descriptive prompt so the user knows what to enter.
3. Calculate the volume of a square pyramid with the base length and height entered by the user; note the volume of a square pyramid is calculated using the formula below:

   Volume = (Base x Base x Height) / 3

4. Print the calculated volume; be sure to add a label to the display value so the user knows what the value means (e.g. display “Volume of the square pyramid is 35 inches-cubed” instead of just 35)

Function Name: **makeChange**
Parameters:
None
Return Value:
None
Description:
Write a user-interactive function to convert any number of cents to the equivalent number of dollars, quarters, dimes, nickels, and pennies.

1. Get the number of cents 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 dollars, quarters, dimes, nickels and pennies represented by the original number of cents , using the following hints:

-There are 100 cents in a dollar.
-There are 25 cents in a quarter.
-There are 10 cents in a dime.

-There are 5 cents in a nickle.
-The modulo (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 dolars, quarters, dimes, nickles and pennies on **one line;** be sure to add appropriate labels to the display values so the user knows what the value means (e.g. if the user enters 12342 cents, you should display “123 dollars, 1 quarter, 1 dime, 1 nickel and 2 pennies”)

Function Name: **PPIIndex**
Parameters:
None
Return Value:
None
Description:
Write a user-interactive program to calculate a person's corrected PPI (pounds per inch) ratio.

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 PPI using the information they provided and the formula below:

PPI = Users Weight / Users Height (inches) * 1.125

4. Print the result. The value must be formatted (both the wording and the number of decimal places) EXACTLY as follows: “Your corrected PPI is 2.2.” (where the value for PPI represents whatever you calculated based on the inputs, and their is a single digit after the decimal place. Note the period after the last digit). *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!

Specifically, using the Graphics library built into Calico. The graphics library contains a multitude of commands for making graphics. See the documentation here:

http://calicoproject.org/Calico_Graphics

For this assignment, we will be focusing on the graphics aspects, so you won't need your robot. Both the link given above, as well as the PDF for the 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 the Graphics library or Myro functions and a comment at the top with your name, section, GTID/Email, and your collaboration statement

The face should contain:

- A Head
- 2 Eyes
- A Nose
- A mouth
- At least 2 different colors
- A hat (any design that combines more than one shape)

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

Grading Rubric

Grading

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

Simple Function — 50 points total

machToFPS	**5**
sqPyramidVolume	**10**
makeChange	**20**
PPIIndex	**15**

Face — 50 points total

50 points

- File named correctly (face.py) – 5 points
- Creates a graphics window – 10
- Draws a head – 5
- Draws 2 eyes – 5
- Draws a nose – 5
- Draws a mouth – 5
- Uses at least 2 different colors - 5
- The facial features are properly aligned – 5
- Draws a hat by combining more than one shape – 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.