CS 1301
Pair Homework 2 – Conversions/Complex Functions

Due: Friday, September 9th, before 11:55 PM
Out of 100 points
Files to submit: 1. hw2.py

You will be writing several functions, but they will all be saved in one file: hw2.py. Please save all of the functions in this one file or you will lose points. **This is a pair programming assignment!** You are expected to work with the person you have been pared with in class, and you are both responsible for submitting the exact same code to T-Square. Follow good pair-programming practices by working together at a single computer and switching the driver/navigator role frequently.

Your pair may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for other pairs. Collaboration at a reasonable level will not result in substantially similar code. For pair programming assignments, you and your partner should turn in identical assignments.

For Help:
      - TA Helpdesk – Schedule posted on class website.
      - Email TAs
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.
   • If you find a significant error in the homework assignment, please let a TA know immediately.

Part 1 – Conversions (50 pts)

Function name: mass (15 pts)

Write a function called **mass()** which takes in a numerical value in ounce and coverts that value into pounds. For reference, there are 0.0625 pounds (yes, that's a real unit!) in an ounce. Please **print** the value, including a label of "lbs". Save this function in your **hw2.py** file as well. Note that this function should NOT return a value.

Sample Output:
```
>>> mass(11)
0.6875 lbs
>>> mass(20)
1.25 lbs
```

Function name: volume (15 pts)

In this function, you will need to convert volumes. There are 4.227 US cups in one liter.
Write a function called **volume()** that takes in a numerical value in liters and converts it to US cups. Since we want to know what we're dealing with, **print** the value out with a nice little sentence explaining the conversion that just happened. Please print the values to 3 decimal places. See the sample output below. Save this function in your **hw2.py** file as well.

Sample Output:
```
>>> volume(5)
There are 21.135 US cups in 5.000 liters
>>> volume(10)
There are 42.270 US cups in 10.000 liters
```

Function name: puppy (15 pts)

Just like the previous two problems, it's conversion time. One puppy is equal to 1093.236234 boxes of cuteness. Seriously. Write a function called **puppy()** that takes in a number of puppies and **prints** out the resultant conversion into boxes of cuteness. This time, we want you to specify how many decimal places to round to as a second **integer parameter**. See the sample output for more clarification. Save this function in your **hw2.py** file as well.

Sample Output:
```
>>>puppy(6,3)
There are 6559.417 boxes of cuteness in 6.000 puppies
>>>puppy(10, 1)
There are 10932.4 boxes of cuteness in 10.0 puppies
>>>puppy(3, 4)
There are 3279.7087 boxes of cuteness in 3.0000 puppies
```

Source for the conversion factor: http://www.weirdconverter.com/
You should totally go there and play around with conversions that don't make any sense. Impress your professors! Entertain your friends! Dazzle your family!

Part 2 – tipCalculator() (55 points)

Parameters:
1. Chocolate chip waffle - an integer representing the number of Chocolate chip waffles
2. Hashbrowns - an integer representing the number of hashbrowns

So you've been up all night playing WOW with your friends. You were so involved in playing WOW that you forgot to have dinner. It being nearly 3 am, the only place is the great institution of Waffle House. You and your friends order some chocolate chip waffles and hashbrowns. The price of 1 chocolate chip waffle is $3.35. The price of 1 plate of hashbrowns is $1.45.

However, you have spent a night of constant awesomeness, and you realize that you just won't be able to compute the appropriate bill and tip amount for the waitress in your head. For some reason, though, the part of your brain that writes Python functions is super focused and ready to go, so you decide to

whip out your laptop and write a Python program to handle the tip calculations for you. (Yes, you brought your laptop to Waffle House with you. You are a Tech student, after all.)

-Write a function called **tipCalculator()** that prompts the user for the percent that they want to leave for the tip. **Do not include the dollar sign or percent sign when entering the numbers!**

-Calculate the bill (without the tip), tip (percent of bill amount) and the total bill. **Do not worry about the tax. The bill (without the tip) should consist of only the total price of chocolate chip waffles and hashbrowns.**

-Remember to divide the percentages by 100 when multiplying. Here, it is important to remember how Python handles division.

-You are expected to round the your total payment to the nearest dollar by use of the math.ceil() function. Basically, math.ceil() always rounds the number up to the next integer. **Do not forget to import math when using functions from the math module!**

For example:
>>> math.ceil(5.67)
6.0
>>> math.ceil(9.1)
10.0
>>> math.ceil(3.4)
4.0

-Round the tip to the nearest cent by use of string formatting. Remember, you're working with money, so how are dollar amounts usually presented?

Sample output:
>>>tipCalculator(2,3)
Calculated bill before tip: 11.05
What percent tip do you want to leave? 10
A 10 % Tip is $1.11,
Total bill is $12.16, rounded up to $13

Grading Rubric
Part 1 – Conversions – 50 points

    mass –   15 points

        Function named correctly (mass) –   5 points

        Performs correct conversion –   5 points

        Prints the value –   5 points

    volume –   15 points

        Function named correctly (volume) –   5 points

        Performs correct conversion –   5 points

        Print statement with values to 3 decimal points – 5 points

    puppy –   20 points

        Function named correctly (puppy) –   5 points

        Performs correct conversion –   5 points

        Print constrans values to specified decimal points – 10 points

Part 2 – Tip Calculator – 50 points

    Function named correctly (tipCalculator) –   5 points

    Bill before the tip is calculated correctly –   10 points

    Gets tip input from user –   5 points

    Tip is calculated correctly –   5 points

    Total is calculated correctly –   5 points

    Uses math.ceil() as needed –   5 points

    Outputs are correct –   10 points

        (5 each for tip and total)

    Print statement containing values to 2 decimal points –   5 points