Homework 1 - Functions & Expressions

CS 1301 - Intro to Computing - Fall 2021

Important

- Due Date: **Tuesday, August 31st, 11:59 PM**.
- This is an individual assignment. High-level collaboration is encouraged, **but your submission must be uniquely yours.**
- Resources:
  - TA Helpdesk
  - Email TA's or use class Piazza
  - How to Think Like a Computer Scientist
  - CS 1301 YouTube Channel
- Comment out or delete all function calls. Only import statements, global variables, and comments are okay to be outside of your functions.
- **Read the entire document before starting this assignment.**

The goal of this homework is for you to practice and understand how to write functions and evaluate expressions. The homework will consist of 5 functions for you to implement. You have been given a `HW01.py` skeleton file to fill out. However, below you will find more detailed information to complete your assignment. Read it thoroughly before you begin.

Hidden Test Cases: In an effort to encourage debugging and writing robust code, we will be including hidden test cases on Gradescope for some functions. You will not be able to see the input or output to these cases. Below is an example output from a failed hidden test case:

```
Test failed: False is not true
```

Written by Craig Nyazema (cnyazema3@gatech.edu) & Fareeda Kasim (kasimfareeda@gatech.edu)

Helpful Information To Know

String Formatting

A concept that will be very helpful for this homework is string formatting. String formatting allows you to manipulate strings using variables so that string values can change based on whatever information is stored in the variables. To explore this concept, let's look at an example where a user inputs a name and age, and the code prints out the corresponding information:

```
name = input("What is your name?")
age = input("How old are you?")
print("Your name is {} and you are {} years old!".format(name, age))
```

Anywhere in a string, you can put `{}` to indicate a placeholder for a variable. After the end quotation marks of the string, you write `.format()`, and inside the parentheses will be the variables that you want to include. The variables inside the parentheses must be in the order that you want them to be included in the string.

Rounding Numbers

Python has a built-in function that allows you to round numbers. For example:

```
>>> rounded_number = round(3.1415926, 4)
>>> print(rounded_number)
3.1416
```

Inside the parentheses of the `round()` function, put the number you want to round, followed by a comma and the number of decimal places you want to round the number to.

Club Points

Function Name: clubPoints()
Parameters: N/A
Returns: None
Description: This semester, Georgia Tech SCPC is giving out special gifts to students who join the most clubs and get other people to join them too. For each club you join, you get 30 points, and for each friend you get to join a club, you get 50 points. Write a function that asks the user how many clubs they have joined and how many friends they have gotten to join any club. Then, print the total number of points they have won.

All inputs will be positive integers.

```
>>> clubPoints()
How many clubs have you joined? 5
How many friends have you gotten to join clubs? 2
You have won a total of 250 points!!
```

```
>>> clubPoints()
How many clubs have you joined? 1
How many friends have you gotten to join clubs? 7
You have won a total of 380 points!!
```

Move-In

Function Name: moveIn()
Parameters: N/A
Returns: None
Description: You're an RA assisting with move-in this semester and you want to know how much time you will spend supporting students with move-in today. You can help each freshman move into their dorm in 35 minutes, each sophomore in 20 minutes, and each junior in 15 minutes. Write a function that asks the user how many freshmen, sophomores, and juniors they plan to move-in today. Then, print a response with how much time (hours and minutes) it will take to help all those students with move-in.

You may assume all inputs will be integers.

```
>>> moveIn()
How many freshmen do you plan to assist? 6
How many sophomores do you plan to assist? 2
How many juniors do you plan to assist? 4
It will take 5 hours and 10 minutes to help 12 students with move-in today.
```

```
>>> moveIn()
How many freshmen do you plan to assist? 1
How many sophomores do you plan to assist? 1
How many juniors do you plan to assist? 6
It will take 2 hours and 25 minutes to help 8 students with move-in today.
```

Tire Area

Function Name: tireArea()
Parameters: N/A
Returns: None
Description: You're rushing to class, and you can't make it on time unless you take a bike. It's a well-known secret that the smaller the area of your bicycle tire, then the faster you ride. Write a function that asks the user what the radius of their bicycle tire is, and then calculate the area of the tire with the given dimension.

The inputs can be floats. Round your answer to two decimal points.

Note: Use `3.14` as the value for pi.
Note: The area of a circle is **A = pi * r^2**, where r is radius.

```
>>> tireArea()
What is the radius of the tire? 3
The area of the tire is 28.26.
```

```
>>> tireArea()
What is the radius of the tire? 0.7
The area of the tire is 1.54.
```

Dining

Function Name: dining()
Parameters: N/A
Returns: None
Description: On campus, you can buy a Panda Express meal for $6, a Rising Roll meal for $8, and a Chick-fil-A meal for $9. Since you are starting a new semester and you have loads of dining dollars to spend, you've decided to go all out! Write a function that asks the user how many meals they would like to order from each restaurant, and the percentage they want to tip the GT Dining worker. Then, print the total price you pay, and the amount you tipped.

All inputs will be positive integers. Round all outputs to two decimal places.

```
>>> dining()
How many meals do you want to order from Panda Express? 5
How many meals do you want to order from Rising Roll? 2
How many meals do you want to order from Chick-fil-A? 10
What percent would you like to tip? 10
You paid $149.6 and tipped $13.6.
```

```
>>> dining()
How many meals do you want to order from Panda Express? 1
How many meals do you want to order from Rising Roll? 0
How many meals do you want to order from Chick-fil-A? 3
What percent would you like to tip? 20
You paid $39.6 and tipped $6.6.
```

Weekly Budget

Function Name: weeklyBudget()
Parameters: N/A
Returns: None
Description: You decide that you want to invest some money, so you make a weekly budget to help you minimize your spending. Each week, you save a percentage of your budget in your account. You also spend $13.50 on a scooter and $21.40 on Uber Eats per week. After accounting for all this, how much do you have left to invest per week? Write a function that asks the user what their weekly budget is and what percentage of their budget they want to save. Then, print out how much money they'll have left over at the end of the week after accounting for savings and fees on scooters and Uber Eats.

All inputs will be positive integers. Round all outputs to two decimal places.

```
>>> weeklyBudget()
How much is your budget this week? 200
What percentage do you want to save? 50
You have $65.1 left after savings and fees.
```

```
>>>weeklyBudget()
How much is your budget this week? 1500
What percentage do you want to save? 35
You have $940.1 left after savings and fees.
```

Grading Rubric

Function	Points
clubPoints()	20
moveIn()	20
tireArea()	20
dining()	20
weeklyBudget()	20
Total	**100**

Provided

The `HW01.py` skeleton file has been provided to you. This is the file you will edit and implement. All instructions for what the functions should do are in this skeleton and this document.

Submission Process

For this homework, we will be using Gradescope for submissions and automatic grading. When you submit your `HW01.py` file to the appropriate assignment on Gradescope, the autograder will run automatically. The grade you see on Gradescope will be the grade you get, unless your grading TA sees signs of you trying to defeat the system in your code. You can re-submit this assignment an unlimited number of times until the deadline; just click the "Re-submit" button at the lower right-hand corner of Gradescope. You do not need to submit your `HW01.py` on Canvas.