

Homework 1 - Functions & Expressions

CS 1301 - Intro to Computing - Fall 2024

Important

- Due Date: **Thursday, August 29th, 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 Ed Discussion
 - [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.**

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: Alice Heranval, Avinash Palliyil, & Paige Holland

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.

Have We Met Before?

Function Name: personalizedMessage()

Parameters: N/A

Returns: None (`NoneType`)

Description: During Week of Welcome on campus, overwhelmed by meeting numerous people, you develop a Python program to record and organize individuals' names, years, majors, hometowns, and the dates of initial encounters.

Write a function called `personalizedMessage()` that asks asks the user the name, year, major, hometown, and meeting date of their new friend. Finally, print out a string of the form: `{name} is a {year} year from {hometown} majoring in {major}. We met on {date}.`

```
>>> personalizedMessage()
Enter their name: Avinash
Enter their year: 4th
Enter their hometown: Marietta, GA
Enter their major: CS
Enter the date today: 08/12/2024
Avinash is a 4th year from Marietta, GA majoring in CS. We met on 08/12/2024.
```

```
>>> personalizedMessage()
Enter their name: Harshith
Enter their year: 4th
Enter their hometown: Cairo, Egypt
Enter their major: Computer Science
Enter the date today: 08/14/2024
Harshith is a 4th year from Cairo, Egypt majoring in CS. We met on 08/14/2024.
```

Canvas Conundrum

Function Name: gradeCalculator()

Parameters: N/A

Returns: None (`NoneType`)

Description: The fall semester for the 1301 team is off to a great start, but there's a problem with Canvas grade calculations that requires your Python expertise to fix.

Write a function called `gradeCalculator()` that asks the user to enter the averages for each assignment category. The function should use the following percentage distributions to calculate a final score:

Assignment	Weight (%)
Participation	5
Homework	20
Exams	45
Labs	10
Final	20

Finally, print out a string of the form: `"Your final score: {rounded grade}%"`, with the final score rounded to **two decimal places**.

```
>>> gradeCalculator()
Enter your participation average: 100
Enter your HW average: 90
Enter your exam average: 80
Enter your lab average: 95
Enter your final exam score: 85
Your final score: 85.5%
```

```
>>> gradeCalculator()
Enter your participation average: 85
Enter your HW average: 95
Enter your exam average: 90
Enter your lab average: 85
Enter your final exam score: 90
Your final score: 90.25%
```

Lunchtime Dash

Function Name: lunchTime()

Parameters: N/A

Returns: None (`NoneType`)

Description: In all of your struggles to get into your classes, you didn't even realize how terrible your schedule is! You are walking past Brittain (RIP) and want to see if you have time to get lunch before your next class.

Write a function called `lunchTime()` that prompts the user to input the name of your next class, its distance from Brittain Dining Hall (in kilometers), and how long you have until the next class (in minutes). **Assume that it takes you 10 minutes to walk 1 kilometer.** Finally, print out a string of the form: `You have {minutes} minutes to eat lunch before you have to run to {class}!`, with the minutes left as an **integer**. Minutes may be 0 or negative if you don't have enough time to eat lunch.

```
>>> lunchTime()
What is your next class? CS1301
How far is it from Brittain? 0.7
How long do you have until it starts? 15
You have 8 minutes to eat lunch before you have to run to CS1301!
```

```
>>> lunchTime()
What is your next class? Linear Algebra
How far is it from Brittain? 2
How long do you have until it starts? 50
You have 30 minutes to eat lunch before you have to run to Linear Algebra!
```

Monthly Budget

Function Name: snackBudget()

Parameters: N/A

Returns: None (`NoneType`)

Description: You've been needing more and more small treats to get you through the semester, so you decide to create a monthly budget for yourself.

Write a function called `snackBudget()` that prompts the user to input a total budget in dollars and the number of midterms they have this month. First, you want to set aside 20% of your **total** budget on coffee and 45% on going out to lunch with friends. Then, for every midterm you want to let yourself spend \$5.00 on a smoothie to cheer yourself up. The function should calculate the amount of money you'll be spending on coffee and the amount left over in your budget. Finally, print out a string of the form: `You will spend ${coffee total} on coffee and have ${remainder} left after the month is over!` , with the total and remainder rounded to **two decimal places**.

```
>>> snackBudget()
What is your budget for this month? 100
How many midterms will you have? 5
You will spend $20.0 on coffee and have $10.0 left after the month is over!
```

```
>>> snackBudget()
What is your budget for this month? 35
How many midterms will you have? 2
You will spend $7.0 on coffee and have $2.25 left after the month is over!
```

Free Time

Function Name: freeTime()

Parameters: N/A

Returns: None

Description: Now that you are a college student, you have way more studying to do than before! Unfortunately, there are still only 24 hours in a day.

Write a function called `freeTime()` that asks the user for the number of credit hours they are taking and how many hours of sleep they would like to get each night. Each credit hour results in 30 minutes of studying each day. Finally, print out a string of the form `"Relax for {numHrs} hours and {numMin} minutes before bed."` . Be sure that the number of hours and minutes are left as **integers**.

```
>>> freeTime()
How many credit hours are you taking? 21
How many hours of sleep do you need each night? 10
Relax for 3 hours and 30 minutes before bed.
```

```
>>> freeTime()
How many credit hours are you taking? 18
How many hours of sleep do you need each night? 8
Relax for 7 hours and 0 minutes before bed.
```

Grading Rubric

Function	Points
personalizedMessage()	20
gradeCalculator()	20
lunchTime()	20
snackBudget()	20
freeTime()	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 “Resubmit” button at the lower right-hand corner of Gradescope. You do not need to submit your `HW01.py` on Canvas.