Homework 2 - Conditionals

CS 1301 - Intro to Computing - Fall 2022

Important

- Due Date: **Tuesday, September 6th, 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 that implement conditionals. The homework will consist of 5 functions for you to implement. You have been given HW02.py skeleton file to fill out. Please read this PDF thoroughly as you will find more detailed information to complete your 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 Audrey Cho (acho45@gatech.edu) & Lasya Sreenivasan (lsreenivasan6@gatech.edu)

Helpful Information To Know

Print vs Return

Two concepts that may be difficult for beginner programmers to differentiate between are the print function and the return statement. While it may appear that they do the same thing, it is important to note that they are quite different. The purpose of the print function is to display information to the user. You cannot save what you print. The return statement, on the other hand, is part of a function definition. All functions have a return value, whether you explicitly write a return statement or not. Functions that do not explicitly have a return statement always return the value None. The return statement is useful because it allows you to give a value to a function. This allows you to either save it for later, use it in a more complex expression, or print it for human consumption.

For example, let's say we have the following two functions below in a file called `file.py`:

```
def printFunc():
    print(2)

def returnFunc():
    return 2
```

This is what would happen if we ran the file and typed the following into the Python shell:

```
>>> a = printFunc()
2
>>> print(a)
None
```

Notice that although the number '2' is printed to the screen, the variable we assigned the function call to, a, has the value None (`NoneType`).

```
>>> b = returnFunc()
>>> print(b)
2
```

When we call returnFunc() and assign it to a variable, nothing is printed to the screen because there are no print statements inside the function. However the variable, b, now holds the value 2 (`int`).

Surprise Party

Function Name: availableDate()
Parameters: date (`int`), isWeekend (`bool`)
Returns: availability (`str`)
Description: You are planning a surprise birthday party for your friend and want to figure out their availability. You know that they have work on odd numbered dates, but are free on even numbered dates and weekends. Write a function that takes in the date and a bool representing whether or not it is a weekend.

If your friend is available, return a string in the following format:

```
'Available on {date}!'
```

If not, return `'Not available :('`

Note: Assume the date passed in can be any integer from 1 to 31 .

```
>>> availableDate(12, False)
'Available on 12!'
```

```
>>> availableDate(29, False)
'Not available :('
```

Game Shopping

Function Name: buyGame()
Parameters: gameTitle (`str`), budget (`float`), cost (`float`), positiveRating (`float`)
Returns: None (`NoneType`)
Description: Your friend Sasuke is looking to buy a new video game, but wants to look at reviews before purchasing. Write a function that takes in the title of the game, his budget, the cost of the game, and the percent of positive reviews given as a decimal.

If the cost of the game exceeds his budget, Sasuke cannot buy the game. Print out a string in the following format:

```
'{gameTitle} is over ${budget}!'
```

If a game is in his budget, and has a positive rating percent greater than or equal to 70%, print out a string in the following format:

```
'Sasuke will buy {gameTitle}!'
```

If a game is in his budget, but has a positive rating percent less than 70%, print out a string in the following format:

```
'Let's find another game.'
```

```
>>> buyGame("Splatoon 3", 80.0, 59.99, .87)
'Sasuke will buy Splatoon 3!'
```

```
>>> buyGame("Elden Ring", 9000.0, 9001.0, .9)
'Elden Ring is over $9000.0!'
```

Time for Food

Function Name: foodTime()
Parameters: restaurant (`str`), time (`int`),
Returns: howLong (`float` or `int`)
Description: You decide you want to get something to eat before your next class starts. Assumming it takes you 10 minutes to eat and you walk at a speed of 80 meters per minute, write a function that takes in the name of a restaurant you want to go to and the time you have in minutes before your next class starts, then return how many minutes you will have left if you are able to make it back in time for class, or return **-1** if you are not able to make it back in time.

Note: Assume your starting location and the location of your next class is the same.

Restaurant	Distance (m)	Wait Time (mins)
Cafe Leblanc	620	5
The Roost	700	10
Lumpy Pumpkin	850	12
The Milk Bar	1200	3

```
>>> foodTime("Cafe Leblanc", 50)
19.5
```

```
>>> foodTime("Lumpy Pumpkin", 40)
-1
```

Restaurant Reservation

Function Name: restaurantReservation()
Parameters: total (`int`), toSave (`int`), average (`int`)
Returns: canReserve (`bool`)
Description: Jake is trying to reserve a seat at a restaurant for his anniversary with Amy; however, they're trying to save money. Given their total amount of money, and how much they want to save, and the average cost of the food in the restaurant, return True if they have enough money leftover to make the reservation and False if they don't.

Note: There are two people, and the average represents the price for one person.

```
>>> restaurantReservation(400, 250, 50)
True
```

```
>>> restaurantReservation(200, 180, 15)
False
```

Halloween Heist

Function Name: halloweenHeist()
Parameters: num1 (`int`), num2 (`int`), name (`str`)
Returns: message (`str`)
Description: Amy and Rosa are on the same team for the Halloween Heist this year. They're planning to communicate through code. Create a function that takes in two numbers and calculates whether their difference is positive or negative. If the difference is positive, return the following string:

```
'{name} has the package.'
```

If the difference is negative, return the following string:

```
'{name} does not have the package.'
```

If the difference is 0, return the following string:

```
'{name} has taken over.'
```

Note: Always subtract num2 from num1.

```
>>> halloweenHeist(20, 16, 'Jake')
"Jake has the package."
```

```
>>> halloweenHeist(13, 25, 'Holt')
"Holt does not have the package."
```

Grading Rubric

Function	Points
availableDate()	20
buyGame()	20
foodTime()	20
restaurantReservation()	20
halloweenHeist()	20
Total	**100**

Provided

The `HW02.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 `HW02.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 `HW02.py` on Canvas.