2016-09-12 Lab

Part 1: Debugging with `print()`

As you write longer and more complicated programs, you are likely going to make mistakes. These mistakes in code are called *bugs*, a terminology that dates back at least 1947, when computers were just circuits and a moth caused a short-circuit in a computer.

For example, imagine you are writing a function that calculates the quadratic formula. (Also, this is the most math you will see in this course.) The function might look something like this:

```
def quadratic(a,b,c):
    # calculate the quadratic formula
    root=sqrt((b*b)-(4*a*c))
    # calculate and print the first root
    numerator=-b+root
    x=numerator/2*a
    return x
```

A copy of this program is on Moodle. Download it and open it in PyCharm.

1. The program is a little hard to read because everything is close together with no breathing space. Reformat the code by going to the Code menu, then click on Reformat Code. Additionally, add blank lines before each comment so you can see distinct blocks of code. That's much better, isn't it? When you write your own programs, try to make the code just as readable.
2. Note the `if __name__ == "__main__":` at the bottom. We will talk about this in lecture, but the important thing for now is that your testing code should go here. Instead of typing in everything with `input()` - which is annoying when you're running a program mulitple times - we will just print out the return value of the function with specific arguments.
3. Read over the program and make sure you roughly understand what is going on. For a = 2, b = 0, c = -1, the correct (rounded) answer should be 0.707107.
4. Run the program. Is the program correct? (Hint: No)
5. One way to figure out what went wrong is to print out information about the variables. For example, it's possible that the error was in calculating the root. So after the assignment to `root`, print out its value. For a = 2, b = 0, c = -1, the root should be 2.828. Is `root` correct?
6. So it's not the root. Maybe it's the numerator? After the assignment to `numerator`, print out its value. It should be 2.828 - is `numerator` correct?
7. That really only leaves one option - it's the assignment to `x` that is incorrect. Do you see the problem?
8. Fix the program, then submit it to the autograder to make sure it's correct

What you just did is called *debugging*, or the process of removing bugs from your program. Printing out the values of variables in different parts of your program is one of the most basic, but also one of the most pervasive, ways of locating the problem. Throughout this semester, whenever you come to me for help, one of the first questions I will ask you is, "What could you print to help you figure out what is going on?"

Notice how at each step, **you need to know what each variable should contain**. You cannot debug a program unless you know what the program is supposed to do. This means that understanding what you're doing is paramount - good programmers spend more time thinking through the problem than they do writing code. As an example, the autograder took me two months to plan out, and only two weeks to write.

I should also say that there are more powerful tools - called *debuggers* - but printing is easier and often good enough.

Part 2: Cupcakes

Consider the following recipe for making 12 cupcakes:

> **Ingredients**
>
> For the cupcakes:
>
> - 1 1/2 cups all-purpose flour
> - 1 cup granulated sugar
> - 1 1/2 teaspoons baking powder
> - 1/2 teaspoon table salt
> - 1/2 cup unsalted butter (1 stick), room temperature
> - 1/2 cup sour cream
> - 1 large egg, room temperature
> - 2 large egg yolks, room temperature
> - 1 1/2 teaspoons vanilla extract
>
> Additionally, for the vanilla buttercream frosting:
>
> - 1 cup (8 ounces) unsalted butter, at room temperature
> - 2 1/2 cups powdered sugar
> - 1 tablespoon (3 teaspoons) vanilla extract
>
> **Cupcakes**
>
> 1. Adjust oven rack to middle position; heat oven to 350 degrees F. Line standard muffin/cupcake tin with paper or foil liners.
>
> 2. Whisk together flour, sugar, baking powder, and salt in a bowl. Add butter, sour cream, egg (with yolk), and vanilla; beat for about 30 seconds until smooth and satiny. Scrape down sides of bowl with rubber spatula and mix by hand until smooth and no flour pockets remain.
>
> 3. Divide batter evenly among cups. Bake about 20 to 24 minutes until cupcake tops are pale gold and a toothpick inserted into the center comes out clean. Transfer the cupcakes from the tin to a wire rack; cool cupcakes to room temperature before frosting.
>
> **Frosting**
>
> 1. Whip the butter on medium-high speed for 5 minutes, stopping to scrap the bowl once or twice.
>
> 2. Reduce the speed to low and gradually add the powdered sugar. Once all the powdered sugar is mixed in, increase the speed to medium-high and add the vanilla. Whip for about 2 minutes until light and fluffy, scraping the bowl as needed.

Your task for this lab is to create a program that can calculate how much the ingredients would cost for a specified number of cupcakes. Your program should first ask the user how many people wants cupcakes, assuming one cupcake per person, and 12 cupcakes in a batch. **You cannot make a partial batch of cupcakes**. You may assume that the user will input a whole number. After printing the number of batches needed, your program should print out the shopping list, then followed by the expected cost (to two decimal places). Here's an example run:

```
How many people do you need to serve? 13

You need to make: 2 batches of cupcakes

Shopping List for Vanilla Cupcakes
----------------------------------
1 bag of flour
1 bag of granulated sugar
2 boxes of butter
1 tub of sour cream
1 dozen eggs
1 bag of powdered sugar
1 bottle of vanilla extract

Total expected cost of ingredients: $20.34

Have a great party!
```

Creating the shopping list requires you to convert between the amounts of each ingredient required by the recipe, to the sizes those ingredients are sold in. Here's the size and cost of each ingredient, together with the unit conversion:

- 5 lb (20 cups) bag of flour costs $3.09
- 5 lb (10 cups) bag of granulated sugar costs $2.98
- 1 lb (2 cups) box of butter costs $2.50
- 8 oz (1 cup) tub of sour cream costs $1.29
- A dozen (12) eggs cost $2.68
- 2 lb (5 1/2 cup) bag of powdered sugar costs $1.18
- 2 fluid oz (12 teaspoons) bottle of vanilla extract costs $4.12

You may assume that the user already has baking powder and salt, so they don't go on the shopping list. **You can only buy these ingredients in these amounts at this cost.** Since the conversions and the cost don't change in this program, you should consider them constants.

Getting Started

Download `starter.py` from Moodle. In the file, you will find three functions that may help you. You do not have to understand how the function works, but you should read the comments to understand what the functions do.

We will be pair programming for this lab, which means that **there should only be one laptop between each pair**. One student will be writing code, while the other student will be observing and watching for errors. Switch roles every 15 minutes or so.

Take however long you need to fully understand the problem. You should be able to calculate, by hand, how much of an ingredient you will need to buy and how much it will cost. **Do not start writing code until you can do this**, or you will likely get very lost very quickly. Remember: focus on solving the problem first, before working on telling the computer.

You need to turn in two things:

- **Both** partners need to submit a copy of the code to the autograder. It can be the same file.
- Both partners need to fill out the peer evaluation for this lab.