Programming Problems

Each of these problems should be solved in the starter file available on the course website. They should be submitted to the Gradescope assignment Hw1 - Programming to be autograded. Make sure to check the autograder feedback after you submit!

For each of these problems (unless otherwise specified), write the needed code directly in the Python file, under the comment and print statement that correspond to the problem. Do not delete the provided print statements- we're using them to autograde.

If you find yourself struggling to get your code to work, remember your resources! Office hours in particular are useful for debugging problems.

#1 - Data Types - 10pts

Write Python code at the top level of the file to do the following:

1. Assign the integer 15 to the variable **a**.
2. Assign the float 3.14 to the variable **b**.
3. Assign the string "20" to the variable **c**.
4. Assign the boolean True to the variable **d**.
5. Reassign the variable **a** to hold the value 45.
6. Add "21" to **c** and assign the result to variable **e**. Don't change the value in **c**.

Feel free to print any of these variables to check your work.

#2 - Printing - 10pts

Write code at the top level of the file to match the following algorithm.

1. Assign the string Kelly to **prof**.
2. Assign a string holding your 110 TA's name to **ta**. If you have two TAs, choose one.
3. Write a single print statement that greets both Prof. Kelly and your TA by name. The statement must use the variables **prof** and **ta**, as well as at least one additional string.

#3 - Functions - 15pts

Write a function that implements the following algorithm, which finds the slope of a line between two points.

- The function's name is **slope**
- The function takes four arguments: **x1**, **y1**, **x2**, and **y2**.
- The function should follow this algorithm:
    a. First, compute the difference in y values (using subtraction) and assign it to the variable **diffY**.
    b. Second, compute the difference in x values and assign it to the variable **diffX**.
    c. Third, compute the slope (**diffY** divided by **diffX**) and assign it to the variable **m**.
- The function should return the variable **m**.

After the function definition, write a print call that directly calls slope on four different numbers and displays the result.

#4 - Graphics - 10pts

Add code to the function `drawFence(canvas)` so that it draws a simple version of the CMU Fence on the canvas. If you have not yet heard of the Fence, learn more here: www.amusingplanet.com/2014/09/the-fence-of-carnegie-mellon-university.html

Note: because graphics require set-up code, we have included graphics functions in the starter file for you. You should add your graphics code to **drawFence,** not `makeCanvas` or the top level of the file.

Your Fence must meet the following basic requirements, but otherwise you may customize it as much as you like.

- The Fence must have at least two columns, with one column ending the left side of the Fence and one ending the right
- The Fence must have two cross-boards that connect all the columns
- There must be a gap above the top cross-board, between the cross-boards, and below the bottom cross-board

You can find bonus instruction on graphics here: www.cs.cmu.edu/~110/slides/week2-graphics.pdf Feel free to 'paint' your Fence too!

Here's an example of a simple Fence that meets the requirements:
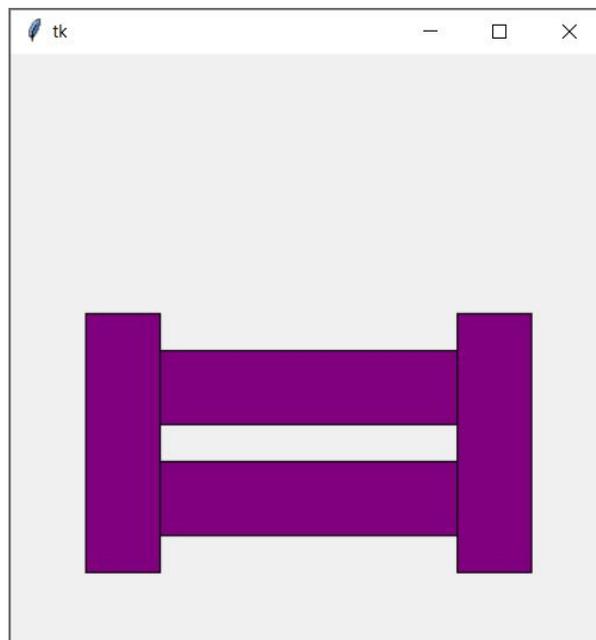
#5 - Conditionals - 10pts

Finally, write a function that takes a number as a parameter and returns a string representing its sign.

- The function's name should be **numSign**, and its single parameter should be **x**.
- You are guaranteed that the function will only be called on ints and floats.
- The function should return `"positive"` if the number is positive, `"negative"` if it is negative, and `"zero"` otherwise.

After the function definition, assign a variable **num** to a random number between -100 and 100. You should use the `random` library to do this. Then print both the number and the number's sign by calling `numSign` on the number.

Before you submit: uncomment the last line of code (`makeCanvas(400, 400)`) and click 'Run File as Script' to make sure your code runs without raising an error message. Any syntax or runtime errors left in the code will result in a deduction on the assignment grade. You should do this for all future programming assignments as well.