CSCE100 Introduction to Informatics
Fall 2023

Programming Assignment 4: Hello Functions: Making a Program Modular

Points: 100 points. Assignment Date: October 5, 2023 Due Date: October 12, 2023

Objectives

1. To familiarize with writing and running Python programs and the Python environment
2. To familiarize with designing programmer-defined functions
3. To familiarize with problem decomposition
4. To familiarize with parameter passing with functions
5. To familiarize with the use of online documentations on Python
6. To practice problem decomposition

Problem

Use the solution that you created for Programming Assignment 3 as the basis. Convert the solution by moving lines of code into functions. Define your functions accordingly. Here are some additional requirements:

- Your new program is required to perform *exactly* the same way as your solution in Programming Assignment 3, in terms of what users see when running the program. (5 points)
    1. *Exception: If you do not have a working solution for Programming Assignment 3, please contact the instructor.*
- Your program is required to have *at least* three programmer-defined functions. (10 points)
- You are also required to turn in a separate report on:

    1. You are required to describe, in a Table, each function that you have in your solution (10 points). For example, for Programming Assignment 3, it might be:

Name	**Input Parameters**	**Returns**	**Purpose**
readFile	1. The filename in string 2. The variable list to store the data	The number of rows as an integer	To open a file and read row-by-row the content of the file, storing it in the variable list.
writeFile	1. The filename in string 2. The variable list to store the data	Nothing	To write the data to an output file.
computeStatistics	…	…	….
modifyImage	…	…	…

    2. Why do you choose those specific lines of code to convert into functions? (5 points)
    3. How general are the functions? How applicable are your functions to other problems? (5 points)
    4. Compare your functions to built-in functions in Python: which ones are more general? (5 points)
    5. How do the functions improve your code's Modularity? Readability? Maintainability? (10 points)
- You must document your program (see https://devguide.python.org/documenting/).
    1. Name, Date, Affiliation, a description of the program, what inputs does it need, what outputs does it generate (5 points)
    2. Inline comments in the program (5 points)

Example Input/Output: None

Handin

1. The submission deadline for all handins is October 11, 2022, 11:00 AM. **Late handins will *not* be accepted or graded.**
2. You are required to handin a screen capture of your "testing session" using your program. (5 points)
3. You are required to handin all program files. (5 points)
4. You are required to handin all input and output files. (5 points)
5. You are required to handin the report. (5 points)

Think About

Now, think about the usefulness of functions. Defining a function improves modularity, maintainability, and reusability of our program. Furthermore, imagine a function that gets called numerous times at the different occasions in a program. If that function has 25 lines of code, how would the overall program code look like? Does it mean that if the function is called 4 times, then that means we would have to see 25 + 25 + 25 + 25 = 100 lines of repeated code in the program. Is that bad? Why? (*Hint*: Reusability of a function.)

Think about when we approach a problem to find a solution. One useful approach is to break the problem down to smaller subproblems. Why? And think about how the solution to a subproblem and a function are related. Indeed, programmers often think about how to break a problem down by identifying the functions that need to be designed first, before they start programming. This is especially so when the solution could involve many lines.