**Background**

Complete the following exercises. Your answers must be submitted via Canvas. Answers to Questions 01, 03, 04, and 05 may be typed in this document or handwritten and scanned. Answers to Questions 02 and 06 should be submitted **in their own .py file which is named as specified in the question**.

Note that for each code section, you may assume that the in-code documentation is accurate and reflects the both the correct approach and desired design for the piece of software.

**Section 1: Duplication Reduction**

1. Consider the function in Question01.py which is attached to the Canvas assignment. Note each aspect in the file that is redundant or duplicated by line numbers and/or including the section of code and how it could be resolved. You may consider comments as part of this problem. (Hint: It may be worthwhile to consider what aspects of this code are “derived values” and how could you remove them?)

Lines 17-43 could be placed in a function that returns the date string in the proper format.

Lines 45-102 and lines 103-161 could be replaced by a function that handles file processing.

Lines 58-68 and 116-126, 73-83 and 131-141, 88-98 and 146-156 are functionally the same and can be placed in a separate function which can process a given section.

Lines 161-164 are redundant due to being able to do the calculations anywhere.

Lines 174-186 and 190-202 can either both be written on one line or can use a date processing function to make sure its in the right format.

Lines 204-221 could be removed and placed in a function or loop that would do the same thing

1. Okay, now resolve it. Implement the changes discussed in your answer to Question 01, using good programming practices and naming conventions, in its own Python file. You may consider comments as part of this solution but keep in mind that the file still have sufficient comments so another developer could understand the function and purpose of the code within. **The file containing your solution to this question should be named Answer02.py and points will be deducted if it is not.**

**Section 2: Rooting out Run-Time Errors**

1. Consider function in Question03.py which is attached to the Canvas assignment. While we have not covered why the above approach is improper and how to fix it, it should be obvious that you could have some problems. Given that, when run, the console prints “There has been an exception”, we are going to have a VERY hard time understanding what went wrong with our code. You need to analyze the code to identify the various aspects that could break unpredictably. Note the various sections of code which would cause a problem by line numbers and/or including the section of code that could cause problems. Explain why each section of code could be a problem.

**(Note 1: You do not have to give the technical name of the exception, just a general description of the problem is sufficient.)**

**(Note 2: This code cannot and will not actually run so you will have to analyze this file by sight using principles that we learned in class.)**

**(Note 3: There are MANY problems with this code. None of them are syntactical. Simply pointing out only one or two issues will not be sufficient for any meaningful credit for this question.)**

Line 17: File opening could fail

Line 31-33: One of the sensors could not have a valid int in the text

Line 38: The socket could not be able to connect

Line 39: The data from the sensor could not have a valid int value

Line 44: Could not be able to get values due to connection/invalid value issues or the sum of the sensor counts may be 0

Line 47: Database might not be able to connect

Line 51-53: Database connection issues, the wrong datatype of parameters were given, the procedure doesn’t exist, the procedure was provided with the wrong number of parameters

1. Continuing to consider the code in Question 03, what steps would you take to determine which line (or lines) are causing the problem. Note that this flawed design was chosen for some reason, so you should both consider wholesale redesigns as well as ways to debug it while keeping the general structure as is. Descriptions of your solutions are sufficient but be sure that the description is detailed enough that I understand what your solutions are.

How I would determine which line/lines were the cause of the issue is to break up the code block into smaller statements since it is all under a single try statement so breaking the code up with multiple code blocks that would allow for more accurate exceptions to be provided. One way to do it would be to split lines 14-28, 37-40, and 47-57 into their own statements which are where most of the computation/connections are established so it would allow for better understanding of where exactly a problem occurred and would not be too different from the original code structure and design.

**Section 3: Bug Identification & Fixes**

1. Debug the function in Question05.py which is attached to the Canvas assignment. For each bug, list the line numbers and/or the section of code, what the bug is, and the type of bug that occurs.

Line 15: Extra Presentences at end of statement, causes Syntax error for invalid syntax

Line 18, 21, 24, 27, 30, and 33: Missing colon at end of statement, causes Syntax error for invalid syntax

Line 18: If statement tries to check >= to for two different datatypes, those being str and int. Is a logic error because code does not change the userInput variable to an integer so it can be compared to.

Line 21: Loop does not start at the number the user stated, counts from 0-100 instead. Is both a logic and control flow error.

Line 24: or should be and in the if statement. Is a logic error.

Line 27: The if statement is not related to the previous if statement, should be elif instead. Is a Logic error.

Line 33: The equation should be “==0” instead of !=0 and the ‘or’ should be changed to ‘and’. Is a logic error

Line 24-34: The flow of the condition checking is incorrect. A way to solve the issue is by switching around the if statements where starting with checking for Flippity Floppity first and then checking for both Flippity and Floppity respectively, finally landing on printing out the current number. Is a Control Flow error.

1. Fix the code from Question 05, using good programming practices and naming conventions, in its own Python file. **The file containing your solution to this question should be named Answer06.py and points will be deducted if it is not.**