CSC/CPE 101: Fundamental of Computer Science I

LAB2

Practice on Function

For this lab you will explore perhaps the single most important building block in programming: the function. The lab requires you to implement and test multiple functions on simple data types (e.g., integers, floating points, booleans, and characters).

1. In CPE101 directory that you created for Lab1, create a directory as LAB2.
2. Inside this directory, Create a file named: **funcs.py**
    a. Write the header for this file:
    ```
    #Lab 2 Functions
    #Name:
    #Section:
    ```
    b. This file will have multiple functions you implement.

3. The test cases will be placed in the provided **lab2_funcs_tests.py**.
4. You must provide at least two test cases for each of these functions. The testing separated into multiple functions.
5. This part will be executed with: **python3 lab2_funcs_tests.py**
6. For each function, you need to write the function purpose statement and its signature.
7. You need to implement the following functions:
    a. Write a function name math_func1 with the given mathematical definition:

$$math\_func1(x, y) = \frac{x^3 + y^3}{5x + 7}$$

    b. Write a function name math_func2 with the given mathematical definition. For computing square root, you will need to use the math.sqrt function from the math library, so be sure to import math at the top of your source file. :

$$math\_func2(a, b, c) = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

    c. Write a function to compute the Euclidean distance between two points.

$$d(x1, y1, x2, y2) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

    d. Write a function, named is_negative, that takes a single number as an argument and that returns True, when the argument is negative and False otherwise. You must write this function using a relational operator and without using any sort of conditional (i.e., if); the solution without a conditional is actually much simpler

than one with. Your test cases should use assertTrue and assertFalse as appropriate.

e. Write a function, named dividable_by_5, that takes a single number as an argument and that returns True, when the argument is dividable to 5 and False otherwise. You must write this function using a relational operator and without using any sort of conditional (i.e., if). Your test cases should use assertTrue and assertFalse as appropriate.

8. Demonstrate the test cases as part of the lab to your instructor to have this lab recorded as complete. In addition, be prepared to show your instructor the source code for functions in funcs.py.

9. **Submit the funcs.py and lab2_funcs_test.py in the PolyLearn.**

The lab2_funcs_tests.py can be downloaded from PolyLearn.