CSC/CPE 101: Fundamental of Computer Science I

Spring 2018 (LAB-3)

This lab requires the implementation of functions that use boolean logic and functions that use conditional (if) statements.

Download lab3 files from PolyLearn in your CPE101\Lab3 directory. These files are **logical_tests.py** and **conditional_tests.py** (corresponding to the different parts of the lab).

1) Boolean Logic

Create a file named **logic.py**. This part of the lab requires that you implement and test multiple functions that compute Boolean values (similar to is_positive from the previous lab). You should develop these functions and their tests one at a time. The function implementations must be placed in **logic.py**. The test cases will be placed in the provided **logical_tests.py**.

You **must** write each of the following functions **without** using any sort of conditional (if) statement.

You must provide at least **three test cases** for each of these functions though you should really provide enough test cases to ensure that the function works even for edge cases.

This part will be executed with: **python3 logical_tests.py**

is_even

Write a function, named `is_even`, that takes a single argument assumed to be an integer and that returns `True` when the integer is even.

There are many ways that one can implement this function; as one example, you should explore the remainder/modulus operator (`%`).

in_an_interval

Write a function, named `in_an_interval`, that takes a single number argument and returns `True` when the argument falls in one of the following intervals (recall that a square bracket indicates inclusivity whereas a parenthesis indicates exclusivity; e.g., the interval [-2,9) includes -2 but not 9). The intervals are [-2,9), (22,42), (12, 20], and [120,127].

This function is meant as an exercise without any clear real-world analog. Write test case for the following values: -10, 0, 9, 20, 122, 127.

2) Functions with If Statements

This part of the lab is similar to the previous part, but the functions for this part will use conditional (if) statements.

In the conditional directory create a file named **conditional.py**. Place your test cases in the provided conditional_tests.py file.

You must provide at least two test cases for each of these functions, but you should really provide enough test cases to ensure that every path through the function is properly tested.

This part will be executed with: **python3 conditional_tests.py**

max_101

Write a function, named **max_101**, which takes two numbers as arguments and returns the larger of the two values. (Note, this function is actually already provided in Python as max. Do not use the provided function; reason through the logic yourself).

max_of_three

Write a function, named **max_of_three**, which takes three arguments of type float and returns the largest of the three values. You should write this using if statements for the practice, but give consideration to how you might write this using the max_101 function above. (Again, do not use the built-in max function.)

rental_late_fee

Write a function, named **rental_late_fee**, which takes a single argument representing the number of days late a rental item is returned. This function will return the number of dollars (represented as an integer in this example) for the assessed late fee as defined by the following table.

Days	Fee
≤ 0	0
≤ 9	5
≤ 15	7
≤ 24	19
> 24	100

Demo Only: Demo your work and submit .py files to the PolyLearn.