Assessment content

Objectives:

Learn the basics of Python programming
Study some of the key differences between C/C++ and Pythons

Setup:

All files needed for this lab were created by doing the first task in lab 0. If you didn't do lab 0, do the first task to make your lab repository.

Part A Coding:

This part of the lab will introduce you to Python programming basics. You are not allowed to use any library functions for this lab unless it is specifically stated in the specification
Write the following functions in lab1.py
The function names must be exactly as specified. The number and order of arguments must be the same.

Function 1:

Name: wins_rock_scissors_paper
Parameters: 2 strings
Return: a boolean value
Description: This function passes two strings. Each of the two strings is going to be one of 3 values:

rock
paper
scissors

The strings may have any casing. Rock, ROCK, and roCK are all possible and valid.
The first string represents what the player threw in a game of rocks, paper, and scissors. The second string represents what the opponent threw. This function returns true if the player won, and false if it was a tie or a loss.
In a game of rock, paper, scissors, each player chooses to "throw" one of the 3 items. The winner is determined by the following rules

rock beats scissors
paper beats rock
scissors beats paper

Function 2:

Name: factorial

Parameters: a number (int)
Return: a number (int)
Description: This function passes a non-negative integer, which we will call n in this description. the function returns n! (pronounced n factorial). n! = n * (n-1) * (n-2)... * 1 Thus, 3! = 3 * 2 * 1. Note that 0! is 1 by definition.

Function 3:

Name: Fibonacci
Parameters: a number (int)
Return: a number (int)
Description: This function passes a non-negative integer, which we will call n in this description. the function returns the nth Fibonacci number in the Fibonacci sequence.

The nth Fibonacci number is the sum of the 2 previous Fibonacci numbers.

Function 4:

Name: sum_to_goal
Parameters: list of numbers and a goal (number)
Return: a number
Description: This function finds the two numbers in the list that sum up the goal value. The function returns the product of the two numbers(the product is the result of multiplying the two numbers together). If there are no two numbers that, when summed, results in the goal state, the function returns 0

Python Objects

In lab1.py
Create a Python class called UpCounter.
An UpCounter keeps track of a number. a fixed step size can increase the number
When a counter is initialized, it is given a step size. If no step size is given, the step size is assumed to be 1. The counter always starts at a count value of 0
The counter class has three functions:

- count() - returns the current counter value
- update() - increases the counter value by stepsize

Derive a Python class call DownCounter , which is inherited from UpCounter
A DownCounter is initialized with the same arguments as an UpCounter. The only difference is that when the update() function is called, the counter decrements the counter value by stepsize.

Part B Reflection:

Write about your experience programming in Python for this lab. Your reflection should include comments on the following:

What did you like/not like about Python
Was there anything that behaved differently than you expected in Python?
Based on what you wrote in your lab, write something about the similarities and differences between Python and C/C++ and how that affects how you write your program.

Submitting your lab

To get any marks for this lab, you must submit:

working code for the functions described in part A.

Please make sure you follow the steps listed below fully

push an updated version of the files you altered into your GitHub repo
Do not alter the structure of the files in your lab repositories in any way, as the build process will fail.
Do not add extra folders.
Do not add extra files.
Do not upload an entire VS solution.
Check that your code has passed testing by looking in the actions tab.

When you are happy with the state of your files, submit a link to your repo's lab1 folder on Blackboard.

Note: Submitting a link to Blackboard indicates that your lab is ready for grading in the current state. Only submit a link once you are prepared to be graded.