

CEG 2170

Laboratory 5

The purpose of this lab is to practice creating loops for repeating tasks and for validating input. You will create one project with two functions. The first function uses **while** loops to process user input and the second function uses a **for** loop to perform similar processing.

Turn In: Upload your project to the Dropbox on Pilot. Be sure your program follows the guidelines given on the Style Requirements document (provided on Pilot) with respect to commenting, variable naming, indenting, etc.

Part 1 – Lab requirements and the `main` function

Create your new project that includes `main` and two functions as specified in this lab handout. Be sure to test your program after each step.

1. In `main`, call your `testWhileLoop` and `testForLoop` functions directly. The `main` function does nothing other than print an information message before each function call and invoke each function (see Sample Output for information message). Make sure to include your function prototypes.

Part 3 - Using a counter-controlled while loop and an input validation loop

1. Write a function named `testWhileLoop` that uses a counter-controlled while loop to process a collection of integers entered by the user. Set the counter and loop condition so that the loop iterates 10 times. The function should initially just get a number from the user and echo it to the console (output) within the loop. Test and run your program. Verify that it prompts for and displays exactly 10 integers.
2. Modify your `testWhileLoop` function so that it calculates the total of the numbers entered, and displays that total at the end of the program. (If you enter the integers 1-10, the total should be 55)
3. Modify your `testWhileLoop` function that asks the user to enter an integer so that it asks the user to enter an integer between 5 and 20 (inclusive on both ends). Add an input verification loop immediately after the input statement to ensure that the user input is valid before processing the value. The loop test should be true if the value is NOT valid. The body of the loop should output an error message stating that the input value was invalid. Print the error message and allow the user to re-enter the value. Test your function by entering values that are not in the expected range. Be sure to enter several consecutive values that are incorrect. Your function should still allow the user to input 10 correct values (the incorrect values should not be counted).

Part 3 - Using a for loop

Create a second function named `testForLoop` that completes the following steps.

1. Copy and paste your function from Part 2 to create a new `testForLoop` function. Remove all code pertaining to step 3 above. The program should now accept any integer input value.
2. Change the counter-controlled while loop to a **for** loop. Verify that your program still works correctly and outputs the total of all numbers entered.

Sample Output

```
"C:\Users\Public\Documents\CEG 2170\Lab\Lab 5 - loops\Hutchison_Lab5_Part1\bin\Debug\Hutchison_Lab5_Part1.exe"
While Loop Test:
Enter an integer between 5 and 20: 8
You entered 8.
Enter an integer between 5 and 20: 10
You entered 10.
Enter an integer between 5 and 20: 12
You entered 12.
Enter an integer between 5 and 20: 1
Error: You must enter a value between 5 and 20. Please re-enter: 2
Error: You must enter a value between 5 and 20. Please re-enter: 3
Error: You must enter a value between 5 and 20. Please re-enter: 4
Error: You must enter a value between 5 and 20. Please re-enter: 5
You entered 5.
Enter an integer between 5 and 20: 6
You entered 6.
Enter an integer between 5 and 20: 18
You entered 18.
Enter an integer between 5 and 20: 19
You entered 19.
Enter an integer between 5 and 20: 20
You entered 20.
Enter an integer between 5 and 20: 21
Error: You must enter a value between 5 and 20. Please re-enter: 22
Error: You must enter a value between 5 and 20. Please re-enter: 23
Error: You must enter a value between 5 and 20. Please re-enter: 24
Error: You must enter a value between 5 and 20. Please re-enter: 10
You entered 10.
Enter an integer between 5 and 20: 15
You entered 15.

Total of all numbers entered: 123

For Loop Test:
Enter an integer: 1
You entered 1.
Enter an integer: 2
You entered 2.
Enter an integer: 3
You entered 3.
Enter an integer: 4
You entered 4.
Enter an integer: 5
You entered 5.
Enter an integer: 6
You entered 6.
Enter an integer: 7
You entered 7.
Enter an integer: 8
You entered 8.
Enter an integer: 9
You entered 9.
Enter an integer: 10
You entered 10.

Total of all valid numbers entered: 55

Process returned 0 (0x0)   execution time : 32.365 s
Press any key to continue.
```