CIS 150 – Lab 04

**Submission of Your Work**

You need to prepare and submit ONE SINGLE MS Word document to Canvas (in your lab section) as LastName\_FirstName\_Lab04.doc. It must contain:

* Your NAME
* For each question:
  + Specify the question number.
  + After reading the question requirements, but before beginning any coding, create the test case table, below, through column Expected Output. Write your program then complete the **test table** with actual output results and include in your report.
  + Copy/Paste your completed source code. You must include standard “header” even if code is provided.
  + Paste in a snippet of output showing results for **every listed test case**, labeled with test case #

Test Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Actual Output | Test Pass / Fail |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

* Add / delete rows from Test Table as necessary
* Modify column widths as necessary
* Test both valid and invalid input
* Test for every output expected
* If failure is an expected output and it happens then that test Passes
* Any test that fails means the program must be fixed so that it passes the test

### **Question 1**

The following program asks the user for the current temperature (in Fahrenheit). Add a series of if statements (if with multiple alternatives i.e. if/else if/else) so that **one** of the following messages is printed based on the temperature value:

Temperature (F) Message

>90 “Go swimming.”

<=90, >80 “Turn on air conditioning.”

<=80, >70 “Do nothing.”

<=70, >55 “Turn on heat.”

<=55, >30 “Wear a heavy coat.”

<=30, >0 “Wear gloves.”

<=0 “Stay inside, make a fire.”

// Include ALL the compiler directives you need in the program

/\* to be filled in\*/

using namespace std;

int main()

{

int temperature;

cout << “Please enter the current temperature (F): “;

cin >> temperature;

// add cascading if-else statements to complete the program

/\* to be filled in by student \*/

system(“pause”);

return 0;

}

Provide a total of 4 valid test cases (one for each message). Do not provide invalid test cases.

**Question 2**

Use a **switch statement** to write the following program:

The program prompts the user for a letter grade (of type char). The list of valid letter grades is:

A B C D E F

The program should consider both lower and upper case

The program will then display the following messages:

For grade ‘A’: display “Excellent”

For grade ‘B’: display “Good”

For Grade ‘C’: display “Average”

For grade ‘D’ or ‘E’: display “Below Average”

For Grade ‘F’: display “Fail”

For any other letter grade entered by users: display “Invalid Grade”

Your test plan should include tests for each possible output, invalid input, and lowercase letters.

Provide a total of 4 test cases:

* You can choose a mix of test cases with uppercase inputs A, B, C, D, E, F or lowercase inputs a, b, c, d, e, f.
* 1 invalid test case (provide a letter that is not within [A,F] or [a,f])

**Question 3**

Run the following code and explain, using your own words, what is happening with each variable (w,x,y,z) and what is the difference between ++x and w++ (is there any influence on the result between prefix or postfix ++ operator?). Provide a snippet of the output showing the result.

int main()

{

int w = 20, x = 20;

int y = 5, z = 5;

y = y + (++x);

z = z + (w++);

cout << "x=" << x << " and y=" << y << endl;

cout << "w=" << w << " and z=" << z << endl;

system("pause");

return 0;

}

Provide a paragraph that answers question 3.

For the code provided in question 3, the postfix and prefix function using the ++ operator has two cases: using ++ **before** a variable name (++variable) tells the compiler to increment the variable by a value of 1 (variable+1) **before** it is used in any other expression – this is why it is called a **prefix**.

When the ++ operator comes **after** the variable (variable++), it says to use the variable’s currently stored value, and afterwards execute the function of incrementing the variable by one (variable +1). This is why it is called a **postfix**.

So, for the code provided above, whenever ++comes before a variable (in this case for x), the compiler executed incrementing the value of x by 1, and then proceeding to add y [y = y + (++x); is the same as: y = y + (x+1);].

For variable w, where ++ comes after, the value of w is first added to z, and only then after that variable w is used in that expression is it incremented as w = w +1

[z = z + (w++); is the same as z = z + w; and w = w + 1; 🡪 executed as separate statements].

Provide 4 valid test cases for (w, x, y, z).

**Question 4 –need 2 .cpp files**

Use a for loop to write a program that calculates the sum of N integers entered by the user. N is a computer generated random number between 5 and 15 (both inclusive).

1. Repeat the program using a while loop
2. Test plan for question 4 is only done once, for any of the three loops (pick one loop type and run the test cases – you need to make sure you’re code works correctly for all 2 loops, so you still need to test them).

Provide 4 valid test cases (ALL SCREENSHOTS only for the for loop). No invalid test cases. (DO ADDITIONAL TESTING FOR WHILE LOOP, BUT DO NOT NEED TO PUT IN SCREENSHOTS; JUST NEED TO UPLOAD AND TURN IN THE CODE FOR BOTH FOR AND WHILE LOOP)

**Note:** Use the following libraries for rand(),srand(), and time() support.

#include <cstdlib> (random numbers library)

#include <ctime> (used for time function)

Use this for setting a seed: srand(time(NULL))

Use this for generating a random number: rand() (restrict random number 5-15 inclusive) (loop rand() val number of times)

**Question 5**

Prompt the user for a character and the height of a right triangle. Then, print a triangle of that height using the symbol entered. The ith line of the triangle should contain i copies of the symbol. Validate that the height entered is > 0.

Hint: Use **nested** loops

|  |  |
| --- | --- |
| **Example Input** | **Corresponding output** |
| **Enter a character : #**  **Enter a height : 4** | **#**  **##**  **###**  **####** |

Provide 4 valid test cases. No invalid test cases.

SEE CANVAS FOR HINTS!!!

FOR IF AND ELSE, IF YOU USE TOO MUCH NESTING THEN YOUR MEHTOD IS LIKELY CONVOLUTED AND YOU’RE MAKING IT HARDER AND LESS CONCISE