**University of Michigan – Dearborn**

**CIS 150 – Computer Science 1**

**Lab# 4**

Demetrius Johnson

[meech@umich.edu](mailto:meech@umich.edu)

February 5, 2020

**Table Content**

Contents

[Question 1 3](#_Toc32370836)

[Test Cases 3](#_Toc32370837)

[Source Code – see LAB4Q1.cpp 3](#_Toc32370838)

[Screenshots 3](#_Toc32370839)

[Question 2 6](#_Toc32370840)

[Test Cases 6](#_Toc32370841)

[Source Code – see LAB4Q2.cpp 6](#_Toc32370842)

[Screenshots 6](#_Toc32370843)

[Question 3 9](#_Toc32370844)

[Test Cases 9](#_Toc32370845)

[Source Code – see LAB4Q3.cpp 9](#_Toc32370846)

[Screenshots 10](#_Toc32370847)

[Question 4 13](#_Toc32370848)

[Test Cases 13](#_Toc32370849)

[Source Code – see LAB4Q4--1 – forLoop.cpp & LAB4Q4--1 – whileLoop.cpp 13](#_Toc32370850)

[Screenshots 13](#_Toc32370851)

[Question 5 15](#_Toc32370852)

[Test Cases 15](#_Toc32370853)

[Source Code – see LAB4Q5.cpp 16](#_Toc32370854)

[Screenshots 16](#_Toc32370855)

# Question 1

## Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Actual Output | Test Pass / Fail |
| 1 | Valid | See if correct message will display | Temperature = 74 | “Do nothing.” | Pass |
| 2 | Valid | See if correct message will display | Temperature = 5.5 | “Not a valid temperature input” | Pass |
| 3 | Valid | See if correct message will display | Temperature = 24 | “Wear gloves.” | Pass |
| 4 | valid | See if correct message will display | Temperature = 55 | “Wear a heavy coat.” | Pass |

## Source Code – see LAB4Q1.cpp

## Screenshots

Test 1

![A screenshot of a computer

Description automatically generated]()

Test 2

![A screenshot of a computer

Description automatically generated]()

Test 3

![A screenshot of a computer

Description automatically generated]()

Test 4

![A screenshot of a cell phone

Description automatically generated]()

# Question 2

## Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Actual Output | Test Pass / Fail |
| 1 | Valid | See if correct message will display | studentLetGrad = ‘A’ | “Excellent” | Pass |
| 2 | Valid | See if correct message will display | studentLetGrad = ‘b’ | “Good” | Pass |
| 3 | Valid | See if correct message will display | studentLetGrad = ‘d’ | “Below Average” | Pass |
| 4 | valid | See if correct message will display | studentLetGrad = ‘F’ | “Fail” | Pass |
| 5 | valid | See if correct message will display | studentLetGrad = ‘s’ | “Invalid Grade” | Pass |

## Source Code – see LAB4Q2.cpp

## Screenshots

Test 1![A screenshot of a computer

Description automatically generated]()

Test 2

![A screenshot of a cell phone

Description automatically generated]()

Test 3![A screenshot of a computer

Description automatically generated]()

Test 4

![A screenshot of a computer

Description automatically generated]()

Test 5

![A screenshot of a computer

Description automatically generated]()

# Question 3

## Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Actual Output | Test Pass / Fail |
| 1 | Valid | Test to see if postfix and prefix work as anticipated | w= 30  y= 150  x= 25  z= 155 | w= 31  y= 176  x= 26  z= 185 | Pass |
| 2 | Valid | Test to see if postfix and prefix work as anticipated | w= 10  y= 15  x= 10  z= 15 | w= 11  y= 26  x= 11  z= 25 | Pass |
| 3 | Valid | Test to see if postfix and prefix work as anticipated | w= 20  y= 20  x= 5  z= 5 | w= 21  y= 26  x= 21  z= 25 | Pass |
| 4 | Valid | Test to see if postfix and prefix work as anticipated | w= 3  y= 1  x= 2  z= 1 | w= 4  y= 4  x= 3  z= 4 | Pass |

## Source Code – see LAB4Q3.cpp

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).

## Screenshots

Test 1

![A screenshot of a computer

Description automatically generated]()

Test 2

![A screenshot of a computer

Description automatically generated]()

Test 3

![A screenshot of a computer

Description automatically generated]()

Test 4

![A screenshot of a computer

Description automatically generated]()

# Question 4

## Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Actual Output | Test Pass / Fail |
| 1 | Valid | Test to see if correct number of integers will be output and summed correctly | NumIntToGen  =  5 | “7, 9, 8, 5, 10,”  randIntSum = 39 | Pass |
| 2 | Valid | Test to see if correct number of integers will be output and summed correctly | NumIntToGen  =  6 | “12, 5, 7, 15, 10, 7”  randIntSum = 56 | Pass |
| 3 | Valid | Test to see if correct number of integers will be output and summed correctly | NumIntToGen  =  7 | “7, 13, 9, 8, 12, 11, 14”  randIntSum = 74 | Pass |
| 4 | Valid | Test to see if correct number of integers will be output and summed correctly | NumIntToGen  =  8 | “15, 15, 6, 15, 7, 7, 14, 11”  randIntSum = 90 | Pass |

## Source Code – see LAB4Q4--1 – forLoop.cpp & LAB4Q4--1 – whileLoop.cpp

## Screenshots

Test 1

![A screenshot of a computer

Description automatically generated]()

Test 2

![A screenshot of a cell phone

Description automatically generated]()

Test 3

![A screenshot of a cell phone

Description automatically generated]()

Test 4

![A screenshot of a cell phone

Description automatically generated]()

# Question 5

## Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Actual Output | Test Pass / Fail |
| 1 | Valid | See if correct triangle with N characters and N lines appears as expected | userChar = a  triHght = 5 | Correct - See screenshot | Pass |
| 2 | Valid | See if correct triangle with N characters and N lines appears as expected | userChar = #  triHght = 10 | Correct - See screenshot | Pass |
| 3 | Valid | See if correct triangle with N characters and N lines appears as expected | userChar = \*  triHght = 8 | Correct - See screenshot | Pass |
| 4 | Valid | See if correct triangle with N characters and N lines appears as expected – added a mini note at end after I updated code for height <= 0 “Invalid height” message | userChar = $  triHght = 0 | Correct - See screenshot | Pass |

## Source Code – see LAB4Q5.cpp

## Screenshots

Test 1

![A screenshot of a computer

Description automatically generated]()

Test 2

![A screenshot of a computer

Description automatically generated]()

Test 3

![A screenshot of a computer

Description automatically generated]()

Test 4

![A screenshot of a computer

Description automatically generated]()