**University of Michigan – Dearborn**

**CIS 150 – Computer Science 1**

**Lab# 6**

**Professor Bacha**

Demetrius Johnson

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

February 19, 2020

**Table Content**

Contents

[Question 1 3](#_Toc33610492)

[Test Cases 3](#_Toc33610493)

[Source Code – see LAB6Q1.cpp 3](#_Toc33610494)

[Screenshots 3](#_Toc33610495)

[Question 2 6](#_Toc33610496)

[Test Cases 6](#_Toc33610497)

[Source Code – see LAB6Q2.cpp 6](#_Toc33610498)

[Screenshots 6](#_Toc33610499)

[Question 3 – ADDENDUM AT END OF SCREENSHOTS 9](#_Toc33610500)

[Test Cases 9](#_Toc33610501)

[Source Code – see LAB6Q3.cpp 9](#_Toc33610502)

[Screenshots 9](#_Toc33610503)

# Question 1

## Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Actual Output | Test Pass / Fail |
| 1 | VALID | See if input values work correctly | userN = 0 | Sum = 0  Average = 0 | PASS |
| 2 | VALID | See if input values work correctly | userN = 1 | Sum = 1  Average = 0.5 | PASS |
| 3 | VALID | See if input values work correctly | userN = 27 | Sum = 378  Average = 13.5 | PASS |
| 4 | VALID | See if input values work correctly | userN = 101 | Sum = 5151  Average = 50.5 | PASS |
| 5 | VALID | Test exit value | userN = -99 | “you have deiced to exit program…” | PASS |

## Source Code – see LAB6Q1.cpp

## Screenshots

TEST 1

![A screenshot of a cell phone

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 computer

Description automatically generated]()

TEST 5

![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 | Test character/wdth/hgt of ‘C’ output | userChar = #  width = 4  height = 5 | See screenshot | PASS |
| 2 | VALID | Test character/wdth/hgt of ‘C’ output | userChar = %  width = 6  height = 7 | See screenshot | PASS |
| 3 | VALID | Test character/wdth/hgt of ‘C’ output | userChar = ^  width = 9  height = 9 | See screenshot | PASS |
| 4 | VALID | Test character/wdth/hgt of ‘C’ output | userChar = !  width = 12  height = 12 | See screenshot | PASS |
| 5 | VALID | Test exit value | Option B | “you have selected to exit…” | PASS |

## Source Code – see LAB6Q2.cpp

## Screenshots

TEST 1

![A screenshot of a cell phone

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]()

TEST 5![A screenshot of a cell phone

Description automatically generated]()

# Question 3 – ADDENDUM AT END OF SCREENSHOTS

## Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Actual Output | Test Pass / Fail |
| 1 | VALID | Test the three separate functions calls on two integers | Num1: 4  Num2: 4 | See screenshot | PASS |
| 2 | VALID | Test the three separate functions calls on two integers | Num1: 3  Num2: 4 | See screenshot | PASS |
| 3 | VALID | Test the three separate functions calls on two integers | Num1: 33  Num2: 55 | See screenshot | PASS |
| 4 | VALID | Test the three separate functions calls on two integers | Num1: 0  Num2: 0 | See screenshot | PASS |

## Source Code – see LAB6Q3.cpp

## Screenshots

TEST 1

![A screenshot of a cell phone

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 computer

Description automatically generated]()

Addendum:

The most important aspect about each function to consider is the difference between call by Reference using the ‘&’ character on the variable declaration in the function, versus call by Copy in the function declaration of the variable. When the variable uses call by Reference, the variable in the function manipulated is directly associated with the memory allocation of the value for the variable in Main. For example, if Main has a variable called int1, and the function has a variable &int2, that means int1 and int2 both use the same memory block, so whatever you do to one variable (let’s say, int2), it affects the other variable directly since they both point to the same block of memory to store their variable’s value.

For call by copy, the variable in MAIN is copied into the function, so that the variable in the function has its own memory block independent of the MAIN variable, and thus any manipulations of either variable is independent of each other and do not affect each other as they point to different blocks of memory to store its values.