**Programming 1 – Fall 2020**  Student Name: \_\_\_\_\_\_Liam Martell\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**MP1 Scoresheet – 10 September 2020**

**Source Code Validation – to be turned in with your source code and screen shots**

**Source Code (i.e., the .cpp file) is to be submitted and placed in your shared Google Drive Folder.**

1. Validation testing (50 points – 10 points deduction per failed test. Must “demo” via Screenshots of output):

Number of regular donuts ordered: ***0***

**Test #1: First Attempt Correct?** Y | N

Incorrect value: owes | dollars | quarters | dimes | nickels | pennies

Incorrect display: text | dollars | quarters | dimes | nickels | pennies

Number of fancy donuts ordered: ***1***

Customer owes $0.91

Customer pays $***1.00***

Change owed is $0.09 – 1 nickel, 4 pennies.

**Test #2: First Attempt Correct?** Y | N

Incorrect value: owes | dollars | quarters | dimes | nickels | pennies

Incorrect display: text | dollars | quarters | dimes | nickels | pennies

Number of regular donuts ordered: ***1***

Number of fancy donuts ordered: ***0***

Customer owes $0.81

Customer pays $***0.81***

Exact payment received – no change owed.

Number of regular donuts ordered: ***1***

**Test #3: First Attempt Correct?** Y | N

Incorrect value: owes | dollars | quarters | dimes | nickels | pennies

Incorrect display: text | dollars | quarters | dimes | nickels | pennies

Number of fancy donuts ordered: ***1***

Customer owes $1.72

Customer pays $***5.00***

Change owed is $3.28 - 3 dollars, 1 quarter, 3 pennies.

Number of regular donuts ordered: ***12***

**Test #4: First Attempt Correct?** Y | N

Incorrect value: owes | dollars | quarters | dimes | nickels | pennies

Incorrect display: text | dollars | quarters | dimes | nickels | pennies

Number of fancy donuts ordered: ***12***

Customer owes $17.72

Customer pays $***20.22***

Change owed is $2.50 - 2 dollars, 2 quarters.

Number of regular donuts ordered: ***12***

**Test #5: First Attempt Correct?** Y | N

Incorrect value: owes | dollars | quarters | dimes | nickels | pennies

Incorrect display: text | dollars | quarters | dimes | nickels | pennies

Number of fancy donuts ordered: ***4***

Customer owes $12.24

Customer pays $***12.65***

Change owed is $0.41 - 1 quarter, 1 dime, 1 nickel, 1 penny.

Number of regular donuts ordered: ***35***

**Test #6: First Attempt Correct?** Y | N

Incorrect value: owes | dollars | quarters | dimes | nickels | pennies

Incorrect display: text | dollars | quarters | dimes | nickels | pennies

Number of fancy donuts ordered: ***35***

Customer owes $54.35

Customer pays $***55.55***

Change owed is $1.20 – 1 dollar, 2 dimes.

**Second Chance (to be initialed by instructor/Lab TA):**

Test #1 correct: \_\_\_\_\_\_ Test #2 correct: \_\_\_\_\_\_ Test #3 correct: \_\_\_\_\_\_ Test #4 correct: \_\_\_\_\_\_ Test #5 correct: \_\_\_\_\_\_ Test #6 correct: \_\_\_\_\_\_

**Testing/Validation Subtotal: \_\_\_\_\_\_\_\_\_ / 50**

**Readability Subtotal: \_\_\_\_\_\_\_\_\_ / 30**

**Documentation Subtotal: \_\_\_\_\_\_\_\_\_ / 20**

**TOTAL: \_\_\_\_\_\_\_\_\_ / 100**

**Programming 1 – Fall 2020**  Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**MP1 Scoresheet – 10 September 2020**

**Source Code Analysis Rubric (this will be filled out by the graders during grading)**

**Readability – 30 points total**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Meets Expectations - 5** | **Needs Minor/Major Improvement – 4/3/2** | **Unacceptable/Missing - 0** |
| Organization  Score (x1): \_\_\_\_\_\_\_ | Code is broken down into clear, recognizable, well thought out sections of functional units; blank lines and comments used to establish visual structure. |  |  |
| Separation  Score (x1): \_\_\_\_\_\_\_ | Spaces used as appropriate to help differentiate distinct elements within each coding statement. |  |  |
| Alignment  Score (x1): \_\_\_\_\_\_\_ | Indentation emphasizes the body of an iterative or a conditional statement; braces and parentheses follow appropriate standards; start of comments are column-aligned as warranted. |  |  |
| Consistency  Score (x1): \_\_\_\_\_\_\_ | Similar coding constructs regularly use the same format regarding indentation and alignment; similar or related variable names follow an established pattern. |  |  |
| Nomenclature  Score (x2): \_\_\_\_\_\_\_ | All variables, save for common exceptions, have meaningful and informative names without being verbose; use of upper and lower case clearly differentiates variables, constants, and classes. |  |  |

**Documentation – 20 points total**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Meets Expectations - 5** | **Needs Minor/Major Improvement – 4/3/2** | **Unacceptable/Missing - 0** |
| Header Comments  Score (x1): \_\_\_\_\_\_\_ | Every program starts with a header comment that, at a minimum, contains the name of the file, the date of its writing, the full name of its author, and a description of what the program does. |  |  |
| Section Comments  Score (x2): \_\_\_\_\_\_\_ | Each functional section of code includes a comment describing the goal or purpose that that section is trying to accomplish without being either verbose or parroting. |  |  |
| Code Comments  Score (x1): \_\_\_\_\_\_\_ | Line-oriented comments are used to clarify meaning and/or provide elaboration as needed. |  |  |