**Scoring Rubric for Project 1 : Introduction to C++ Programming**

*Due 9/12/2019 @ 5 pm*

|  |
| --- |
| Student Name: Raniery Mendes |

|  |  |  |
| --- | --- | --- |
|  | **Score** | **Maximum** |
| **Execution (50 pts):** | | |
| Program compile without errors (warnings are okay). | 50 | **50** |
| **Implementation (40 pts):** | | |
| Repeatedly asks the user to guess location on matrix. (all or nothing) | 10 | **10** |
| Prints out an error message if the guessed location is outside of the bounds of the matrix dimension. (all or nothing) | 10 | **10** |
| Uses random number generator to choose location on 3x3 matrix. (all or nothing) | 0 | **5** |
| Correctly updates and prints out matrix with each user input (try all until battleship sunk; all or nothing). | 0 | **5** |
| Outputs message for correct guess. (all or nothing) | 5 | **5** |
| Outputs correct number of guesses. (all or nothing) | 5 | **5** |
| **Style (5 pts):** | | |
| The driver is easy to follow based on the use of comments. | 3 | **3** |
| Easily identifiable variable names. | 2 | **2** |
| **Submission Organization (5 pts):** | | |
| Github repository named in format project-1-username, with C++ driver source code for program named Battleship.cpp (-1 for each incorrect case) | 5 | **5** |
| **Total (100 pts):** | 90 | **100** |

Notes:

You should have previous guesses marked on the board every time the user enters a guess. Calling rand() % 2 returns a number between 0 – 1, so you never consider the third row/column of the matrix as a possible location for the battleship.