**Scoring Rubric for Project 4 : TicTacToe**

*Due 10/09/2019 @ 8 pm*

|  |
| --- |
| Student Name: Nathan Carter |

|  |  |  |
| --- | --- | --- |
|  | **Score** | **Maximum** |
| **Execution (50 pts):** | | |
| Program compiles without errors (warnings are okay). | 50 | **50** |
| **Implementation (45 pts):** | | |
| Uses the given class without modifications. | 5 | **5** |
| Asks for a choice of a one or two-player game. | 5 | **5** |
| Asks the player(s) to choose a location (all or nothing). | 5 | **5** |
| Prints out a message if the chosen location is outside of the bounds of the board dimension and asks for another location (all or nothing). | 5 | **5** |
| Prints out a message if the chosen location has been taken and asks for another location (all or nothing). | 5 | **5** |
| In one-player game, uses random number generator to choose a location on the board or implements a smarter approach (all or nothing). | 5 | **5** |
| In a one-player game, randomly determines whether the player or the computer goes first. | 5 | **5** |
| Correctly updates and prints out board with each player’s input (all or nothing). | 5 | **5** |
| The game successfully ends with one of the two players winning or with a draw. | 5 | **5** |
| **Style (5 pts):** | | |
| The driver and functions are easy to follow based on the use of comments | 3 | **3** |
| Easily identifiable variable names | 2 | **2** |
| **Total (100 pts):** | 100 | **100** |

Notes:

Great work! However, you can do better than having a for-loop that “repeats virtually infinitely many times” to check if user input valid. Use a while-loop instead, because that actually will loop infinitely many times if conditions are not met. It might not seem like a big deal for this small lab, but imagine a bot is trying to hack your program; it can easily input an invalid entry more than a few hundred times.