**Self-Evaluation Report for Programming Assignments**

|  |
| --- |
| **Section I: Basic Information**   1. **Programming assignment # 3** 2. **Name of the author: Connor Bacon** 3. **Name of the peer reviewer if any: Kenneth Chen** 4. **Number of hours spent in programming**: **12+ hours** |
| **Section II: Integrity Review**  **Integrity rules for regular programming assignments**   * **Peer discussion**: Peer discussion of code shown on a screen or board is acceptable for explanation of ideas and for debugging purpose. Such discussion may help to cultivate an open learning environment in the class, but you should carefully read the guidelines below to avoid any dishonest behavior and never step over the guidelines explicitly described in the following. * **Don’t use any code (i.e. C++ statements, segments of a program or an entire program) written by others (except for examples in our textbooks or reading)**: Any copy-and-paste of code from other people’s programs or from websites is viewed as cheating and you will get 0 points for the assignment. * **Don’t circulate your code around**: You should never pass around your code (electronically or on paper) to others except for the TA and the instructor. Violating this rule is viewed as cheating in the class and the provider will receive 0 points for the assignment. * **Don’t provide false or exaggerated results of test cases**: You need to report results of test cases in the self-evaluation report together with all your source code files for each assignment**.** Providing false or exaggerated results of test cases in the report is viewed as cheating and you will receive 0 points for the assignment. * **Demonstrate the credibility of your authorship of the work**: When you submit your code as your own work for points, you should make sure that you are able to explain your code and reconstruct your code from scratch without any outside help when requested. If you are not able to do that on your own when requested, you will get 0 points for the assignment and there will be an investigation. * **Consequence of cheating in the class**:Cheatings end in 0 points for the assignments followed by discipline actions described in the student handbook.  1. **Have you received any code written by others? No** 2. **Have you passed any code you wrote to others? No** 3. **Have you used any code written by others? No** |
| **Section III: Test cases and peer review**  **Note: To get all the points, you should have a peer reviewer watch the behavior of your program before you submit the work. You should prepare your own test cases and have your reviewer see the results when you run your program over the test cases. Optionally, you may also have the reviewer run your program through the reviewer’s own test cases to see whether your program works correctly.**   1. **Compile and run your code using Visual C++ 2019 as the testing environment and describe the test cases used and the results you and the reviewer have observed**:   All functions work as expected.   1. **Description of bugs or other problems discovered by you or the peer reviewer if any:**   No bugs discovered   1. **Have you implemented everything required by the programming assignment? If not, describe what are missing.**   Everything has been implemented |
| **Section IV Self-evaluation: Points you think you deserve \_\_\_\_6 i tried rly hard\_\_\_\_\_\_\_\_\_**  *Deduct* ***two points total*** *if you submit the work after the due date but before it’s closed.*  **Grading scale for each problem (max 5 points per problem)**:   1. Source code fails to compile successfully. 2. Source code does something required, but has serious bugs or miss key features. 3. Source code does most of the required features, but has many minor bugs or miss a key feature. 4. Source code does all the features required, only a few minor bugs. 5. Source code does all the features required, fully functional, no bugs.   (**+1 point**) In addition to the points received above, get one more point if   * 1. (0.5 point) the self-evaluation report provides description of test cases used and the test cases were verified by a peer reviewer.   2. (0.5 point) the code is well indented and commented to make it visually readable. |