**Scoring Rubric for Project 7: Linked Lists, Stacks, Queue, Sorted Linked Lists, and Inheritance**

*Due 11/14/2019 @ 11:59 pm*

|  |
| --- |
| Student Name: Henry Evans |

|  |  |  |
| --- | --- | --- |
|  | **Score** | **Maximum** |
| **Execution (50 pts):** | | |
| Program compiles without errors (warnings are okay) | 50 | **50** |
| **Implementation (45 pts):** | | |
| Writes a brief description of your dataset and why you chose to use it for this project. Implements the Data class with at least 4 data members, overloads the output stream operator, and overloads the comparison operators (< or >). | 4 | **5** |
| Implements the Stack class inheriting from LinkedList class; includes a default constructor, a member public function named “push\_head”, a member public function named “pop\_head”. | 10 | **10** |
| Implements the Queue class inheriting from LinkedList class; includes a private Node \*tail, a default constructor, a member public function named “enqueue\_tail”, a member public function named “dequeue\_head”. | 7 | **10** |
| Implements the SortedLinkedList class inheriting from LinkedList class and includes a member public function named “insertSorted” | 7 | **10** |
| Reads from CSV file and writes the content of the Stack, Queue, and SortedLinkedList classes into stacked, queued, sorted text files. | 10 | **10** |
| **Style (5 pts):** | | |
| The driver and functions are easy to follow based on the use of comments | 0 | **3** |
| Easily identifiable variable names | 2 | **2** |
| **Total (100 pts):** | 90 | **100** |

Notes:

No description included about why you chose your data.

You didn’t write any comments.

There’s an issue with your insertSorted function because your sorted.txt doesn’t include all the players. If two players have the same number of points, then one of them isn’t included in the file.

There’s an issue with your enqueue\_head function because your queued.txt only has one player.