**Scoring Rubric for Project 6: Insertion Sorting a Linked List**

*Due 10/31/2019 @ 3:30 pm*

|  |
| --- |
| Student Name: Alexa Brown |

|  |  |  |
| --- | --- | --- |
|  | **Score** | **Maximum** |
| **Execution (50 pts):** | | |
| Program compiles without errors (warnings are okay) | 50 | **50** |
| **Implementation (45 pts):** | | |
| Implements the LinkedList class to be a friend of the Node class and contain only one pointer “head” | 5 | **5** |
| Implements a member function of LinkedList to do the insertion sort: implementation designs can vary, but some operations should be encapsulated in helper functions and it should be quadratic time | 7.5 | **15** |
| Implements the “big three”: copy constructor, copy assignment operator, and destructor for the LinkedList class | 15 | **15** |
| Plots the execution time vs. N for the vector and linked list | 5 | **5** |
| Describes the similarity or difference observed in the performance of the InsertionSort algorithm for a vector vs. Linked List | 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):** | 92.5 | **100** |

Notes:

Your insertion sort algorithm sorts most of the list but seems to get confused about some values, especially repeating values, and sticks them at the end of the list. You might be keeping track of a pointer incorrectly that is causing this to happen.