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

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

|  |
| --- |
| Student Name: Sarah Ruth Nicholls |

|  |  |  |
| --- | --- | --- |
|  | **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 | 15 | **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 | 0 | **3** |
| Easily identifiable variable names | 2 | **2** |
| **Total (100 pts):** | 97 | **100** |

Notes:

You should have code that belongs within a class inside your main function (you are basically writing an append function for the linked list in your main). Instead, this function should be in your LinkedList class and you can just call list.append() in your main.