**Keeping Things in Order**



picture from http://therocktulsa.com/blog/

**Objectives:**

* To practice doubly linked lists

**Background:**

Linked lists are great, but as we have seen in the last lab (and in class), a lot of time is spent traversing the list because there is no way to back up. Using a doubly linked list solves this problem.

**Assignment:**

Download the following from myCourses:

* + ListNode2.py
  + grader.py

You will need to create a DoublyLinkedList class that uses the Node class provided and is used in turn by the grader program (grader.py) provided. The DoublyLinkedList class has the following methods:

\_\_init\_\_() – Constructs the object by assigning the head and rear references to None.

\_\_len\_\_() – Returns the number of nodes in the linked list

\_\_str\_\_() – Returns a string that contains all of the items stored in the list with spaces in between the items

in\_order\_insert() – Takes the item passed and places it in the list so that the items in the list are **in order** from lowest to highest.

delete\_item() – Takes an item that is passed to it, finds the first instance of it in the linked list, and removes that item. If the item is not found, raise a ValueError with the message “This item is not in the list”.

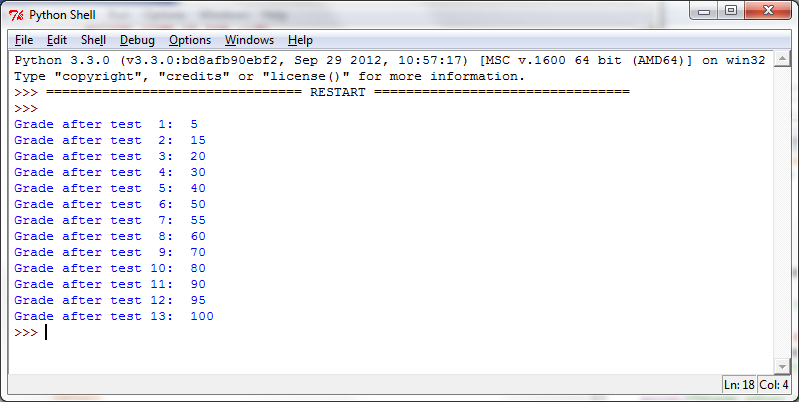
total\_list() – Adds the items stored in the list and returns the total.

get\_first() – Returns the value stored in the first node

get\_last() – Returns the value stored in the last node

Once the class is completed, run the grader program.

**Sample Execution:**



**Program Requirements:**

* Your program should use the correct comment block at the very beginning of your program
* Your program should made good use of whitespace and comments

**Deliverables:**

* Source Code: Be sure to submit the source code electronically in myCourses. You may submit the file more than once before the deadline if you find and correct an error after you submitted it the first time. You may not submit a correction after the deadline however.

**Grading:**

Grade is printed by the driver program.