**Aaron Williams**

**Assignment 2 (Homework Unit 3)**

**30 August 2017**

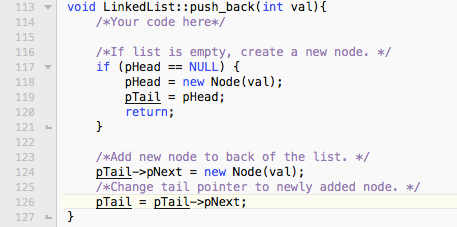
**Assignment Description**

The purpose of this assignment is to begin with source code for a single linked-list (SLLlist.cpp) and make the following changes: 1) Complete an additional method called push\_back(int) that will add an integer node to the end of the linked-list and 2) Modify the Node class and LinkedList class so that the parent node can be accessed (create a double linked-list). Additionally, some minor changes were implemented as necessary.

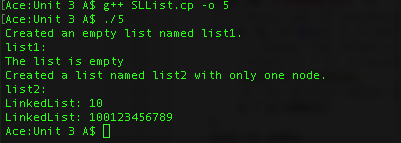
**Logic and Outputs**

1) The first change made to the source code was the completion of the push\_back(int) method. In order to accomplish this, the logic employed including first checking if the list was empty. If the list was empty, a node was created based on the int input value and was assigned as both the head and tail node. If the list already contained nodes, a new node was added by changing the value of the tail node next field. Once the node was added, the tail pointer was adjusted to the new node.

push\_back(int) Method:



Current Main Output:

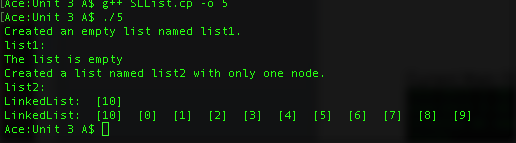


Next, the traverse\_and\_print() was updated to provide a cleaner output by adding square brackets around each node in a linked-list.

traverse\_and\_print() Method Changes:



Current main() Output:

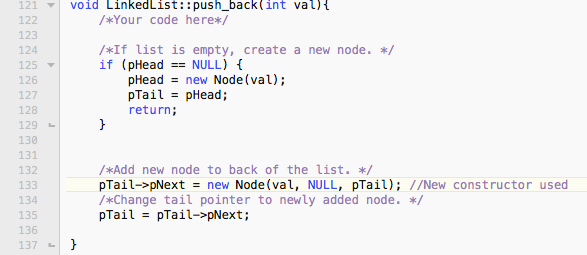


2) Secondly, the Node and LinkedList classes were updated so that the parent node of each node could be accessed, essentially creating a double linked-list. The Node class was edited first to include a pPrev pointer for each node. The necessary changes were also made to the constructors and getters.

Updated Node Class:

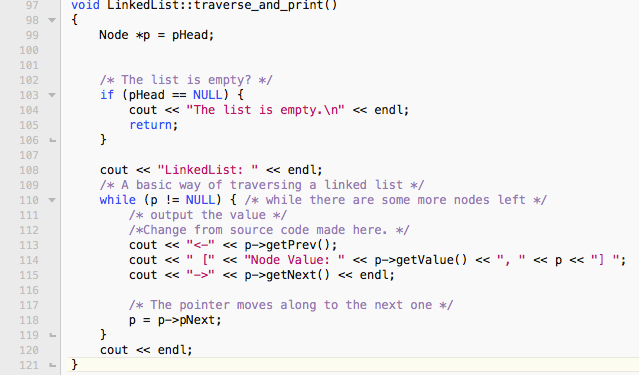
The LinkedList class methods traverse\_and\_print() and push\_back() were then edited. The push\_back() method was edited first to account for the assignment of the pPrev pointer to each newly created noted. This was accomplished by changing the constructor used from part one.

Updated push\_back() Method:



Next, the traverse\_and\_print() method was updated to show the next and previous links for each node.

Updated traverse\_and\_print() Method:



Lastly, the main function was updated to show the reflected changes and results of the final code.

Final main() Output:

