## CIS 22B Assignment 6

*Topics:* Linked Lists

Use the same format for problem and function headings as assignment 1.

## Problem 6.1 (use the data in cardata61.txt as shown below)

Copy the solution from problem 5.2.

You will change the implementation of the StringOfCars class, but keep the public interface. This implementation will use a linked list, rather than an array or vector to hold the cars.  
Keep all the public function prototypes in the StringOfCars.  
Discard all the private data and the implementation of the functions; you will rebuild all of these.

**Do not change anything outside the StringOfCars class**.

**Build a new class called Node:**

You will build the linked list using Node objects, linked together to make a list.

In the private data of the Node class put two pointers:   
One with type Node \* and name next, which will point to the next node in the linked list,  
The second with type Car\* and name data, which will point to the Car data associated with this node.

Also in the private area create a default constructor that sets the next and data pointers to zero. Because the constructor is private, only friends can use this class.

In the public area of the Node class, make StringOfCars a friend class.

The order of the following three things is important:

1. Declare the StringOfCars class with:   class StringOfCars;
2. The Node class
3. The StringOfCars class

This is needed because the Node class uses the StringOfCars and the StringOfCars class uses the Node class.

**In the StringOfCars class implementation:**

Replace the private data with two pointers of type Node \*, and nothing else. Name these two pointers **head** and **tail**.

Change the StringOfCars default constructor to set the head and tail pointers to zero.

Rebuild the push function, with the same function heading.   
Declare a local pointer variable of type Car \* named currentCarPtr.   
Declare a local pointer variable of type Node \* named currentNodePtr.   
Use **new** to get space in the heap for a Node and save the address of the space in the heap in currentNodePtr   
Use new get space in the heap for a new Car that is a copy of the car parameter of the push function and save the address of the space in the heap in currentCarPtr

Set the data pointer in this new Node object to point to the new Car object.

If the head pointer is zero   
  set the head and the tail pointer to the value of currentNodePtr   
else   
  set the next pointer of the Node object pointed to by the tail pointer to the value of   currentNodePtr   
  set the next pointer to the value of the currentNodePtr

**Do not write a pop function**.

Rebuild the output function, with the same function heading.   
Declare a local pointer variable of type Node \* named currentNodePtr - it will point to the Node you are currently working on.

if the head pointer is zero   
  print: NO cars   
else   
  set the currentNodePointer to the value of the head pointer   
  while the currentNodePointer is not zero   
      print the Car pointed to by the currentNodePointer   
      set the currentNodePtr to the next pointer in the Node pointed to by   
      the currentNodePtr, which now makes the next Node the current Node

Rebuild the copy construtor.   
Declare a local pointer variable of type Node \* named currentNodePtr - it will point to the Node in the oldStringOfCars that you are currently working on.   
Set the head and tail pointers in the StringOfCars being constructed to zero.   
If the oldStringOfCars head pointer is not zero:   
  loop while the currentNodePointer is not zero,   
      push the Car pointed to by the data pointer in the current Node, which is pointed to by the currentNodePointer.   
      set the currentNodePtr to the next pointer in the currentNodePtr so we now make the next Node the current Node

Use the data in cardata61.txt to test your program. The data is shown below:

cardata61.txt

Car car1 SP 819487 maintenance false NONE

Car car2 NP 46877 business true Portland

Car car3 NS 157 tender true Saint Louis

FreightCar car4 PVT 123457 tank false Fort Worth

FreightCar car5 MP 98767 box true Saint Louis

FreightCar car6 SP 567897 flat true New York

FreightCar car7 NSF 7877 hopper true Texarkana

PassengerCar car8 KCS 7897 chair true Kansas City

PassengerCar car9 ATSF 147 sleeper true Tucson

PassengerCar car10 B&O 747 combine false Winslow

Submit your code (.cpp file) and your data (.txt file)

# There is no assignment 6.2