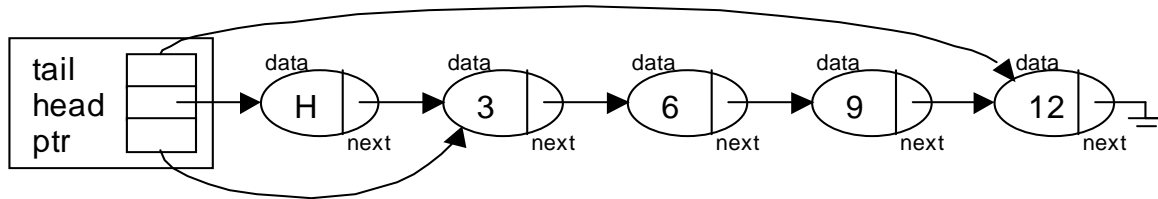


**MS-3832**  
**Advanced C++ Program #1**  
**Data Structures**

**Date Assigned:**      **Friday, September 9, 2005**  
**Date Due:**            **Friday, September 16, 2005**

A “singly-linked” list class is characterized in the diagram below:



The implementation for this “singly-linked” list appears on the course web page for CS-183 (linked from the MS-3812 course page), along with a “templated” version and an “iterator” version. The assignment is two-fold, and should be done in this order:

- 1) Modify the implementation to produce a “circular, doubly-linked” lists. A circular, doubly-linked list is depicted below. This entails adding more functionality to the list so that “previous” values can also be retrieved from the list structure. In other words, make it easy for the end user to go either left-to-right or right-to-left through your list structure. Also notice that circular lists have a “trailer” node as well as a “header”.
- 2) Make your list implementation more compatible with the STL’s list implementation by “templating” it and adding an “iterator” class to go with your new container class. Your iterators need to be bi-directional (i.e. support prefix and postfix ++ and --) and also must support the necessary iterator operations (\*, ==, !=, etc.). Also implement the idea of a reverse\_iterator (one that goes right to left on the ++ operation).

As always, a well commented source code file set is required by the above due date. Hopefully, most of this assignment is a refresher of material already understood from MS-3811. If not, please be sure to ask for help with this lab in class.

A “circular, doubly-linked” list is shown in the following diagram:

