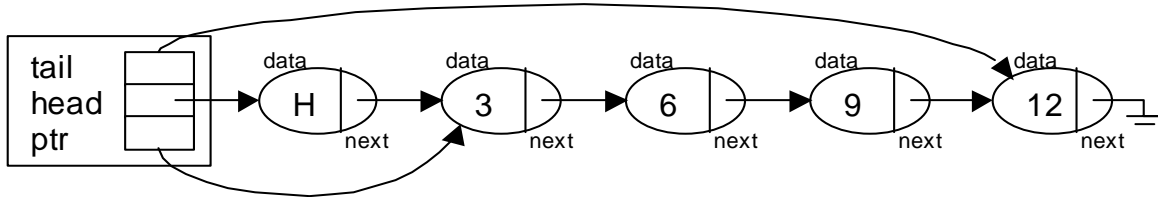


**CS-183**  
**C++ Program #4**  
**Data Structures**

**Date Assigned:**      **Friday, April 19, 2002**  
**Date Due:**            **Monday, April 29, 2002**

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



The implementation for this “singly linked” list appears on the course web page, along with a “templated” version and an “iterator” version. The assignment is two-fold:

- 1) Modify the implementation to produce “doubly linked” lists. A doubly linked list is depicted below. This entails adding more functionality to the list so that “previous” values can also be retrieved 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.
- 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 and a written report that externally documents the implementation and testing of your code is required by the above due date. Take this opportunity to add whatever additional functionality you wish to make your lists more like the STL’s lists (you can check my CS-285 web page for links to the STL online descriptions).

A “doubly linked” list is characterized in the following diagram:

