

Emphasis on: **Implementing Linked Lists and Templates**

Write a complete **LinkedList** class template using the ideas described in 8.6 of our textbook. For basic operations the class should have a constructor, destructor and the basic list operations: *empty*, *insert*, and *delete*. Also include a linear *search* operation to search the linked list for a given item, returning a pointer to a node containing the item in its data part, or a null pointer if it is not found, and an operator to *display* the list (extra credit if you overload the operator<<).

Requirements

- 1) You must use a class template, so that you **LinkedList** class may be used to hold a list of any data type.
- 2) You must implement the linked list using pointers (no arrays) so that your insert and delete operations are efficient.
- 3) All basic operations given above should be supplied as public member functions of your **LinkedList** template class.
- 4) Provide a test driver where you demonstrate the ability to handle a list of int and another of char, showing the insertion, deletion, display and empty operators on your list class.

Extra Credit

Extra credit will be given for implementing a list *traversal* mechanism. This mechanism should work similar to iterators of the STL. Basically there should be a method that resets the list traversal to the beginning of the list, then a method that, when called, will return consecutive items in the list, returning NULL when the end of the list is reached.