**The Assignment:**

You will start with your sequence class that uses a linked list to store the items. You will convert this class to a template class.

**Purposes:**

Ensure that you can convert a container class to a template class. In fact, any time that you write a container class, it is a good idea to start by writing an ordinary container class. After the ordinary class is debugged, convert it to a template class.

**Before Starting:**

Read all of Chapter 6.

**Due Date:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Files that you must write:**

1. sequence4.h: The header file for the new Sequence template class. Actually, you don't have to write much of this file. Just start with a copy of your header file for the sequence class that uses a linked list. Change the documentation at the top to indicate that the class is now a *template* class. Make sure that you delete the part of the documentation that refers to the typedef (because you no longer have a typedef!). You can use the documentation at the top of page 280 as a guideline.

Also, at the bottom of the header file, change the sequence class definition to a template class. When you do this change, follow the pattern in Step 1 on page 277. You will also need to change link1.h (the linked list toolkit) to link2.h (the template version of the linked list toolkit), and change each of your Node\* member variables to a Node<Item>\* member variable.

Finally, at the bottom of the header file you will need the include statement:

#include "sequence4.template"

The reason for this include statement is explained in Step 3 on page 279.

1. sequence4.template: The implementation file for the new sequence template class. Notice that the name of this file ends in ".template" rather than ".cxx". This is to remind you that template implementation files are never compiled on their own.

To implement this file, start with a copy of the implementation from your ordinary sequence class and make the changes described in Step 2 on page 277. Some further clarifications are given on pages 278-286 and 295-296 (using the Bag class as an example).

**Other files that you may find helpful:**

1. [sequence\_test.cxx:](http://www.cs.colorado.edu/~main/chapter3/sequence_test.cxx) This is the same interactive test program that you used with the earlier sequences. If you want to use it with the new sequence, then copy it to your directory and open it with your editor. Then change the statement   
   #include "sequence1.h"   
   to  
   #include "sequence4.h"   
   Also change each sequence to a sequence<double>, and change the namespace to main\_savitch\_6B.
2. [sequence\_exam4.cxx:](http://www.cs.colorado.edu/~main/chapter6/sequence_exam4.cxx) A non-interactive test program that will be used to grade the correctness of your new sequence class.
3. [node2.h](http://www.cs.colorado.edu/~main/chapter6/node2.h) and [node2.template:](http://www.cs.colorado.edu/~main/chapter6/node2.template) Copy these files to your hw06 subdirectory. They contain the template version of the linked sequence toolkit from Section 6.4. You may use these files without changing them.

**The Sequence Template Class Using a Linked List   
Discussion of the Assignment**

Most of your work consists of following the pattern for converting a container class to a template class, as shown on pages 277-279. Also, you must use the template version of the linked list toolkit (link2.h instead of link1.h), and each Node\* variable will be changed to a Node<Item>\* variable. The conversion of the Bag (in Section 6.5) may serve as a good example to follow.