

## 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.

## Files that you must write:

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 bottom of page 304-308 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 301.

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

```
#include "sequence4.tpp.h"

// .tpp.h is a .cpp file with templates!
```

The reason for this include statement is explained in Step 3 on page 303.

sequence4.tpp: The implementation file for the new sequence template class. Notice that the name of this file ends in ".tpp" 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 302. Some further

clarifications are given on pages 301-304 and 305-308 (using the Bag class as an example).

## Other files that you may find helpful:

1. [sequence\\_text.cxx](#)
2. [Download sequence\\_text.cxx](#)
3. : 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 project and open it with your editor. Then change the statement  
Also change each sequence to a `sequence<double>`
4. [sequence\\_exam4.cxx](#):
5. [Download sequence\\_exam4.cxx](#):
6. A non-interactive test program that will be used to grade the correctness of your new sequence class. Hint: It is using a namespace of `main_savitch_6B`.
7. [node2.h](#)
8. [Download node2.h](#)
9. and [node2.tpp.h](#)
10. [Download node2.tpp.h](#)
11. Copy these files to your hw06 project. 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 301-304. Also, you must use the template version of the helper functions 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.

## HINT

This line should not exist in your newly created sequence4.h nor you sequence4.tpp.h file:

```
typedef item value_type;
```

Every place you see

```
value_type
```

should be templatized.