## Programming Assignment 3:  Revise the sequence Class to Use a Dynamic Array (Chapter 4)

**Modified from projects of Chapter  4  at www.cs.colorado.edu/~main/projects/**   
**of Data Structures and Other Objects Using C++  by** [**Michael Main**](http://www.cs.colorado.edu/%7Emain/) **and** [**Walter Savitch**](http://cseng.awl.com/authordetail.qry?AuthorID=355)

**The Assignment:**

You will implement and test a revised sequence class that uses a dynamic array to store the items.

**Purposes:**

Ensure that you can write a small class that uses a dynamic array as a private member variable.

**Before Starting:**

Read all of Chapter 4.

**Due Date:**

Monday, June 26th, 1:30 PM.

**Files that you must write and turn in (Please do not turn in other files!!):**

1. sequence2.h: The header file for the new sequence class. Actually, you don't have to write much of this file. If some of your member functions are implemented as inline functions, then you may put those implementations in this file too. By the way, you might want to compare this header file with your first sequence header file sequence1.h, the new version no longer has a CAPACITY constant because the items are stored in a dynamic array that grows as needed. But there is a DEFAULT\_CAPACITY constant, which provides the *initial* size of the array for a sequence created by the default constructor.
2. sequence2.cpp: The implementation file for the new sequence class. You will write all of this file, which will have the implementations of all the sequence's member functions.

**Other files that you may find helpful (but you do not need to turn in):**

1. sequence\_test.cpp: This is in fact the same interactive test program that you used with the earlier sequence. 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 "sequence2.h"

And change the statement   
using namespace main\_savitch\_3   
to   
using namespace main\_savitch\_4

1. seq\_ex2.cxx: A non-interactive test program that will be used to grade the correctness of your new sequence class.

### The sequence Class Using a Dynamic Array Discussion of the Assignment

Your sequence class for this assignment will differ from the previous sequence in the following ways:

* The number of items which may be stored in the sequence should only be limited by the amount of memory available on the heap. When new items are added to a sequence which is at capacity, the size of the data array in which items are stored should be automatically enlarged.
* Because you are dynamically allocating memory within your sequence class, you will need to define a copy constructor, an assignment operator, and a destructor.
* The constructor should have a default argument which allows the user to set the initial capacity of the sequence.
* There should be a resize function that allows the user to explicitly set the capacity of the sequence.

Start by declaring the new sequence's private member variables in sequence2.h. This should include the dynamic array (which is declared as a pointer to a value\_type). You will also need two size\_type variables to keep track of the number of items in the sequence and the total size of the dynamic array. After you've declared your member variables, write an invariant for the top of sequence2.cxx.

Many of the features of this class are similar to the **bag** class from Section 4.3, so start by thoroughly reading Section 4.3 and pay attention to new features such as how the sequence differs from a bag (see page 124). Also, the implementation of some of the functions are almost the same as in Project 2. Once again, do your work in small pieces. For example, my first version of the sequence had only a constructor, start, insert, advance, and current. My other member functions started out as stubs.

Use the interactive test program and the debugger to track down errors in your implementation. If you have an error, *do not start making changes until you have identified the cause of the error.*

When a member functions needs to increase the size of the dynamic array, it is a good idea to increase that size by at least 10% (rather than by just one item).

### The sequence Class Using a Dynamic Array \*Optional Part of the Assignment (for extra 10% points )

You may wish to provide some additional useful member functions, such as   
(1) Operators + and +=.. For + operator, x+y contains all the items of x, followed by all the items in y. The statement x += y appends all the items in y to the end of what's already in x.   
(2) Operator []. For a sequence x, we would like to be able to refer to the individual items using the usual C++ notation for arrays. For example, if x has three items, then we want to be able to write x[0], x[1] and x[2] to access these three items. The use of the square brackets is called the subscript operator. The subscript operator may be overloaded as a member function, with the prototype shown here as part of the sequence class:

class sequence   
{   
public:   
    ...   
    value\_type operator[](size\_type index) const;   
    ...   
};

The only parameter is the index of the item we want to retrieve. The implementation of this member function should check that the index is valid, and then return the specified item.