The Assignment:

You will implement and test a revised sequence class that uses a dynamic array to store the items. A complete definition is here:

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.

Files that you must write:

1. sequence2.h
2. Download sequence2.h
3. :The header file for the new sequence class. Actually, you don't have to write much of this file. Just start with our version and add your name and other information at the top. 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
4. Download sequence1.h
5. . 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.
6. sequence2.cxx: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. NO using copy(). Please create your own for loops to move data around.

Other files that you may find helpful:

1. sequence_test.cxx

Download sequence_test.cxx

: This is 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 also change the namespace to main_savitch_4.`

2. sequence_exam2.cxx

Download sequence_exam2.cxx

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

Submission:

Include required information in the .cpp file comments and in the output (Name and Class). The output should be from running your code with the "...exam2.cxx" file.

Copy the output results to a separate file (like exam2Out.txt). Note you will need more than one run of the program with different input values. Copy each of those into one file or submission.

Upload the files to Canvas

Upload as .cpp, .cxx, .docx, .h, .hpp, .rtf or .txt Only!