Programming Assignment #1

1. **This program tests the concepts of:**
    1. Container classes
    2. Pointers
    3. Dynamic Arrays

2. **Program Objective:**
   Implement a sequence container class that utilizes a dynamic array. Reference chapter 3 and 4 of the textbook for a detailed explanation of this class. Note that the supplied header file is a modification of the authors original sequence2.h.

   Sequence class notes:

    1. The major difference between the authors bag class and the authors sequence class is that ORDER matters

    2. 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.

    3. Because you are dynamically allocation memory within your sequence class, you will need to define a copy constructor, an assignment operator, and a destructor.

    4. The constructor should have a default argument which allows the user to set the initial capacity of the sequence.

    5. There should be a resize function that allows the user to explicitly set the the capacity of the sequence.

    6. 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 an Item). You will also need two size_t variables to keep track of the number of items in the sequence and the total size of the dyamic array. After you've declared your member variables, write an invariant for the top of sequence2.cxx.

    7. 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.

    8. 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.

    9. 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).

    10. Do not modify the sequence2 header file since your project will be compiled and graded using the original version.

   The header has already been completed for you as well as the test file.

Programming Assignment #1

File	Format	Example
sequence2 Class Implementation	lastname_sequence2.cpp	miller_sequence2.cpp

3. **Record Layouts:** None.

4. **Special Calculations:** None.

5. **Specific Processing:** None.

6. **Output Layout:** None

7. **Other:**

- See Blackboard for the rubric (A scoring *rubric* is an attempt to communicate expectations of quality around a task) used to grade this assignment.
- For full credit your code will have to fully pass the authors sequence_exam test program.
- For full credit your code should have zero compiler warnings.
- No modifications to the header files should be done, doing so may render your code unable to be compiled by the grader
- Your code will be checked for memory leaks. Memory leaks will be verified via a tool called "valgrind" as well as visually.
- Usage of any STL containers will result in no credit
- Usage of recursion will result in no credit
- Avoid all STL items except std::copy (algorithm), std::endl, assert (cassert), size_t