COEN 79L - Object-Oriented Programming and Advanced Data Structures

Lab 3

- **Sequence**

A `sequence` class is similar to a bag—both contain a bunch of items, but unlike a bag, the items in a sequence are arranged in an order. **In contrast to the bag class, the member functions of a sequence will allow a program to step through the sequence one item at a time**. Member functions also permit a program to **control precisely where items are inserted and removed** within the sequence.
Our `sequence` is a class that depends on an underlying `value_type`, and the class also provides a `size_type`.

Three member functions work together to enforce the **in-order retrieval** rule:

```
void start( );
value_type current( ) const;
void advance( );
```

- After activating `start`, the `current` function returns the first item
- Each time we call `advance`, the current function changes so that it returns the next item in the sequence

Provide some additional useful member functions, such as:

1. `insert_front`: insert a new value at the front of the sequence. This new item should now be the current item.
2. `remove_front`: remove the value at the front of the sequence. The new front item should now be the current item.
3. `attach_back`: insert a new value at the back of the sequence. This new item should now be the current item.
4. `end`: The last item in the sequence should now be the current item.
5. `operator+` and `operator+=`: These operators should have the precondition that the sum of the sizes of the two sequences being added is smaller than the `CAPACITY` of a sequence.

Subscript 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. This 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;
...
```

As you can see, the `operator[]` is a member function with one parameter. The parameter is the index of the item that we want to retrieve. The implementation of this member function should check that the index is a valid index (i.e., index is less than the sequence size), and then return the specified item.

For this project, specify, design, and implement this new subscript operator for the sequence.

More information about the sequence class is available in *Slide Set 3*.

→ **The documentation of the sequence class has been provided for you in the `sequence1.h` file.**

- Make sure you include the invariants of the class on top of the implementation (*.cpp) file.

Submission Guideline

The names of the files that you submit should be `sequence1.h` and `sequence1.cpp`. When you upload your solution to Camino, please make sure that your sequence is set up to hold the `value_type` of `double` and has a `capacity` of 50.
Test your program using the `official_seq_tes.cpp` code provided.