CS 2401 – Fall 2021

Project 2 – Sequence Classes and Dynamic Arrays

Due: 11:59 PM Friday, September 24th

The idea of a *sequence* class is that the programmer can choose where an item is stored in the list, and that sequence, or order, of the items remains the same, even when things are deleted. In this project we are going to pass that capability on to the user allowing them to order their Facebook friends in any way they choose. We are just maintaining a list here, not working with your actual Facebook account, and you are allowed to have fictitious friends. Of course, some people have hundreds of Facebook friends, while others have only a few, so we are going to implement this using a *dynamic array.*

Begin this project by copying the `main.cc, date.h, date.cc, MyFriend.h and fbfriends.h` files that I have provided on Blackboard.

`MyFriend.h` is a header file for a class called MyFriend (note that we cannot call this class friend, since the word friend is reserved). This class has two private variables, one for the friend's name and the other for their birthday. The birthday is of type Date which you also used for the last project. You are to write the implementation of this class (`MyFriend.cc`), including overloaded insertion, extraction, = =, and != operators. Two friends are "equal" only if they have the same name and the same birthday.

Test this class by writing a main of your own that declares two friends, lets you put the information into both, outputs them to the screen and compares them for equality.

Now, in the main that I have given you, you will find that the application allows the user to:

- Add a new friend to the beginning of the list
- View all the friends in the list
- Walk through the list, viewing one friend at a time
- Remove a friend from the list (de-friending someone)
- Insert a new friend at some spot in the middle of the list, which includes at the back end
- Sort all the friends by their birthdays
- Find a friend using their name

There is also a file backup mechanism that uses the person's username for the name of the file.

I have given you the header file for the container class that makes all of this possible**,** `fbfriends.h`. **You** are to write the implementation of this (`fbfriends.cc`). The private variables for this class consist of a pointer of type MyFriend as well as variables for capacity, used, and current_index. The constructor will begin by allocating a dynamic array of type MyFriend with the capacity to hold five friends (this will save on memory space for those users with few friends). When the action of adding an additional friend to the list happens you should check if used == capacity, and if it does, do a resize operation that increases the size of the array by five.

This container also has an internal iterator, as the author illustrated in section 3.2 of the text, which will require that you write the functions

- start
- advance
- is_item
- current
- remove_current
- insert
- attach

You will also need to implement show_all, bday_sort, find_friend, and is_friend.

Because this is a dynamic array you will need to write a resize function and the Big 3 (destructor, copy constructor, and assignment operator). Also, because we're providing file backup, we will need to have functions for load and save.

Your submission should include a data file of at least four friends, although they can be fictitious. It would probably be smart to have this file hold at least six friends, since the initial size of the array is five, and the resizing will happen automatically when the sixth person is added.

For grading purposes, I will be using my own file of friends, and then testing all the different branches of the menu. It would behoove you to do the same with your list of at least six friends.

Submit to Blackboard MyFriend.cc and fbfriends.cc as well as .txt versions of those files (MyFriend.txt and fbfriends.cc). Also, be sure that you submit the individual files instead of a zipped version of the files.