

## Project 2 – Sequence Classes and Dynamic Arrays

Due: 11:59 PM Friday, February 17<sup>th</sup>

The idea of a *sequence* class is that the application 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 the profiles they follow on Instagram in any way they choose. We are just maintaining a list here, not working with your actual Instagram account, so you don't have to import the people you really follow (if you have an Instagram account). Of course, some people follow thousands of people on Instagram, while others follow 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`, `profile.h` and `instafollows.h` files that I have provided on Blackboard. The only thing you should change in any of these files is you need to add Doxygen comments for each function in the header files once you figure out what they should do.

`profile.h` is a header file for a class called Profile. This class has two private variables, one for the profile'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 (`profile.cc`), including overloaded insertion, extraction, `=`, `!=` operators. Two profiles 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 profiles, 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 profile to the beginning of the list
- View all the profiles in the list
- Walk through the list, viewing one profile at a time
- Remove a profile from the list (unfollowing someone)
- Insert a new profile at some spot in the middle of the list, which includes at the back end
- Sort all the profiles by their birthdays
- Find a profile 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 this possible, `instafollows.h`. **You** are to write the implementation of this (`instafollows.cc`). The private variables for this class consist of a pointer of type `Profile` as well as variables for `capacity`, `used`, and `current_index`. The constructor will begin by allocating a dynamic array of type `Profile` with the capacity to hold five profiles (this will save on memory space for those users with few profiles they follow). When the action of adding an additional profile 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 like the one I described in class. To do this you will need to implement the functions:

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

You will also need to implement `show_all`, `bday_sort`, `find_profile` (to identify a profile by its name and return the whole `Profile` object), and `is_profile` (to determine if a `Profile` is in the array or not).

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 six profiles, although they can be fictitious. The format of the data file should be:

```
name1
bday1
name2
bday2
etc...
```

For grading purposes, I will be using my own file of profiles, 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 profiles.

**Submit to Blackboard** `profile.h`, `profile.cc`, `instafollows.h` and `instafollows.cc`. Also, be sure that you submit the individual files instead of a zipped version of the files.