Santa Clara University
School of Engineering

CSEN 79L: OOP and Adv. Data Structure

Lab 3.1 Roster using Pointers

Switch the storage mechanism from the previous lab to pointer.

In the previous lab, the class “Roster” use a fixed-sized array to store records of “Student” objects, instantiated from data reading from stdin. In this exercise, we change “Roster” to use a pointer to Student type instead.

1. The constructor should initialize the “roster” member, now declared as a pointer to Student, to nullptr. Add members to track the current “capacity” and how many have been consumed.
2. Change the “insert” function to allocate memory blocks when the existing capacity has been exhausted. Since the storage was initially nullptr, the first call to “insert” will allocate the first chunk.
   Every time insert needs to increase the capacity, it should allocate new memory, copy over the old record, and de-allocate the old. Of course, it should also update various tracking members.
3. Write the destructor for the Roster class to deallocate whatever previously allocated.

The program, therefore, behaves as if there is infinite capacity and should be able to handle arbitrary number of data records. Examine “erase” and iterator member functions to see if they need to change or not.

4. Revise your test plan to cover the new design.