Review and Discussion

- Operator syntax: member functions vs. friends
- Returning objects vs. returning references to objects
- A little more about the str class, version 1 using vectors.
  - The toughest part is `operator>>`. We will come back to this at the end of lecture today if we have time, but you are not responsible for knowing it.

Reading

- Chapter 12 should be accessible now.
- Chapter 11 will make more sense after today’s class and even more after Monday’s

Week 9 Homework

- Use the first version of the str class, from Monday’s lecture, as an example. Combine this with what you’ve learned in lecture and lab about operators.
- Note: you must use a vector or a list.
- Write and test one or two functions at a time.

Today: Str Class Without Use of Vector

This is a good intermediate example before diving into the `Vec` class discussed in Chapter 11.

- Classes with dynamically allocated memory
- The big three: copy constructor, destructor and assignment operators.
- Growing and shrinking the string
- Iterators as typedefs on pointers
Classes With Dynamically Allocated Memory

- Each object must do its own dynamic allocation
- We must be careful to keep the memory of two objects separate
  - Default or poorly written copy constructors can ruin this.
  - Default or poorly written assignment operators can ruin this.
- Dynamically memory must be released when an object is finished with it. This is done through what’s called a destructor.

str Member Variables

- The array: arr_
- The number of chars in the str: size_
- The number of chars in the allocated array: alloc_

The latter two are different to allow efficient dynamic growth and shrinkage. This is covered in detail below.

Copy Constructor and Assignment Operator

- These two functions must allocate a new char array for the current object (*this) and copy the contents of the passed object.
- The assignment operator must also
  - Make sure that this is not a self-assignment. This is absolutely crucial.
  - Delete the current array before copying.
- The actual copying is done in a private member function called copy, which we will write during lecture.
- We will consider implications of incorrect constructors and assignment operators during lecture.
Destructor

- Called implicitly when an automatic object goes out of scope or a dynamic object is deleted.
- Must delete the dynamic memory owned by the class.
- Syntax is a bit weird, at first.

Growth

operator+=

- Adds to the end of the array.
- What should it do when there is no more room in the array?
  - Allocate a new, larger array and copy the contents of the current array.
  - The best strategy is generally to double the size of the current array.
- We will write the details in class

Other Functions

- erase — using str iterators
- substr — constructing a new object from inside a current object
- operator< and operator== — comparing two str objects
- Iterators are using just like char array pointers, which is exactly what they are.

The complete code for the two different versions of the str class will be posted on the web.
Looking Ahead

• Monday (3/25):
  – Templated classes, an introduction
  – Design example: Game of Life

• Wednesday lab (3/27):
  – Vec class
  – Debugging practice

• Thursday (3/28):
  – Design example: Game of Life
  – Review

• Monday (4/1): Test 2