CSI33 Data Structures

Sharon Persinger

Fall 2019

Day 22 November 20

Topics

Dynamic Arrays

Dynamic Array class

Dynamic Arrays

C++ built-in arrays have a fixed size, determined at compile time, or in newer C++ implementations at run-time.

If we want to be able to make an array larger than its original size, we need to use dynamically allocated arrays.

In C++ this is done with pointers.

Example array1.cpp

Dynamic Arrays

How do we go about resizing an array in C++ if we need to make it larger? (Python lists do this automatically.)

Basically we allocate new memory for our larger array, copy the existing array into it, deallocate the earlier existing array, and update the array name.

Example array2.cpp

C++ Class List

This List class will have the functionality of a Python list. The items are stored in a C++ array. The array is allocated dynamically using a pointer.

The List class will have these public methods:

- Indexing with [ ]
- Assignment operator=
- Append for adding one item to the end of a list
- Operator += for adding one list onto the end of another

And these private methods:

- copy to create a deep copy of the list
- resize to copy the array into memory with a bigger capacity

Dynamic Memory Classes

A class that uses dynamically allocated memory must have these methods:

- A destructor method for deallocating memory
- A copy constructor to construct a deep copy of an existing object
- An assignment operator= to assign an Independent copy of an object to an existing object variable.

Why are these methods needed?

List class

The data members of the C++ List class are:

int size_; // the number of items in the array

int capacity_;//the number of items the array can hold

int* data_; // pointer to an array of ints

List class

List4.h has implementation of the three methods needed by any class with dynamically allocated memory.

The methods are implemented inline, so there is no separate List4.cpp file.

Review the implementation of these methods.

Run the test program test_list1, test_list2, test_listp to see where the methods are called.

Complete List class

Full implementation of the List class is in List.h and List.cpp

Look at implementation of

- Indexing with [ ]
- resize
- append

New terminology

size_t is an unsigned int type used to represent the size of an object in bytes. It is defined in stlib.h and other places.

this is a pointer to the object that calls a member method. It is an implicit parameter for the definition of every member method. It is a pointer, so to access an instance variable or method of the object, use the operator ->