CSI33 Data Structures

Sharon Persinger

Fall 2019

Day 21 November 18

Topics

More Operator overloading

C++ Pointers:

- address operator &
- dereference operator *
- new operator
- pointers and objects, -> operator

Dynamic Arrays

Another operator overloading example

Look at latest version of Rational.h and Rational.cpp

For overloading of < as a stand-alone function

RationalNum5.cpp demonstrates these.

C++ pointers

Example p1.cpp

Declaring a pointer variable:
- int *b;
- int x;

Assigning a value to the pointer:
- b = &x

Unary operator & computes the address of its operand. &var is the address where var is stored. This operator can be applied to any variable.

Unary * operator is used to dereference a pointer If b is a pointer, then *b accesses the data at the address where b points. *b can be used to retrieve the data and to change the data at that location.

Swap function using pointer arguments

Swap.cpp

new operator

New operator allocates new memory dynamically for a variable of a specified type.

The memory is allocated from the memory heap and the starting address is returned.

When the memory is no longer needed it must be deallocated using the delete statement.   If it is not deallocated, your program has a memory leak.   The program consumes more memory than is necessary, and so the program is not as efficient as it could be.

Example p2.cpp

Using pointers to objects of a class

You can declare a pointer to an object of a class.

You can use the pointer to refer to member methods of the class.

There are two ways to do this.

See the examples in RationalNum .cpp

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