

Details of Classes

1-25-2001

Opening Discussion

- What did we talk about last class? Do you have questions about the quiz? I have posted answers on the web.
- Did anyone else write anything on the Connect-4 game?
- I have posted a description of your first assignment on the web. The design is due a week from today and the code is due 2 weeks from today.

Questions from Students

- Public or Private?
 - When should methods and members be public and when should they be private?
- Dynamic memory
 - What is the syntax for allocating and deallocating dynamic memory? More importantly, what does it mean? When should we be doing it?

The this Pointer

- One thing that can be helpful to be able to do in a method is to have access to the object the method is being called for. This is done through the this pointer.
- You have access to the this pointer in all methods and can use it like any pointer to an object of that type. In many ways it is like an unspecified first argument to all of those calls.

Type Conversions

- Last class we briefly mentioned how single argument constructors are used by C++ to do type conversions.
- Conversions are not transitive. An implicit cast will only be done if there is a "direct route".
- You can also overload typecasting operators so that an object can be cast to another type.

Operator Overloading

- Another ability you have with classes in C++ is that of overloading operators.
- This allows you to write expressions involving objects just like you would write expressions with primitives.
- It can be done in two ways, either by writing operator methods, or by writing stand alone operator functions. These functions are often friends of the class.

Input and Output Operators

- You can also overload the bit-shifting operators that C++ uses with the iostream libraries.
- Doing this allows you to place objects of that class type in standard iostream expressions and have it work.
- Note that all of this is for convenience, it isn't required for you to be able to output a class.

Avoiding Friends

- While often the easiest way to write stand alone function operators for a class is to make that function a friend of the class, it is better to try to reduce the number of friend classes used at much as possible.
- Putting appropriate accessors and other functions in the class to help non-friend operators is often a better route to take.

Exceptions

- Exceptions are a way to signal errors in code that can't be easily ignored. The fact that it can't be easily ignored makes it somewhat better than other methods of communicating errors.
- You throw an exception with the throw operator. Unfortunately they aren't handled well in C++. The book will deal more with exceptions later as will we.

Minute Essay

- One thing that should be clear at this point is that using classes and objects in C++ properly is not a trivial matter. What features of classes in C++ do you find interesting or think will be helpful? What features do you think cause more problems than they are worth.
- I'll be playing basketball from noon until about 2pm.
