Basic Data Structures and some more C++

9-9-2003

Opening Discussion

- Do you have any questions about the project? Have you made any progress on assignment #1. How are you approaching the web spider? What class(es) in the libraries are helpful?
- What do you remember about stacks, queues, and linked lists?

Arrays

- Arrays in C++ are just like arrays in C. You can declare static arrays that go on the stack or dynamically allocate them. The only difference is that you dynamically allocate them with new and must later use delete.
- There is no bounds checking on them and they don't know their size. In C++ you will only use these in small, select situations.

Vectors

- This is something like a dynamically sized array that is fast and easy to use. It is part of the more general STL. The [] is overloaded but doesn't to bounds checks.
- Here are some methods of vector.
  - int size(), type at(int), void push_back(type), type pop_back(), iterator begin(), iterator end(), void insert(iterator,type), iterator erase(iterator), void resize(int), void clear()

Stacks and Queues

- Hopefully you recall from last semester that stacks and queues are very simple data structures that only let you add an item, remove an item, and check if they are empty. You have no choice of how items are added or removed.
- A stack if LIFO and a queue is FIFO.
- Good implementations of these are O(1) for all functions.

Linked Lists

- A more complex ADT is a list. One way of implementing a list is with a linked list. In a linked list each element knows about one or more of its neighbors, but no one knows where to find the entire list.
- To get to an element we have to walk the list, but it is very easy to add and remove elements by altering links.
- In C++ the links are pointers.

Inheritance in C++

- Classes in C++ can inherit from other classes much like in Java. There are no explicit interfaces. Inheritance is denoted as class Sub:public Super {};
- There is also private inheritance, but you will never need it in this class. That is the default though so don't forget to put public.
- As expected, Sub is a subtype of Super and inherits all methods and members from it.
- C++ supports multiple inheritance but we won't get into that unless required.

Virtual Methods

- By default methods are not virtual (this means they are statically linked and won't do what you expect if you override them in the subclass). A non-virtual method represents something that is uniform across all subtypes of that type.
- To make a method virtual just put the virtual keyword in front of the declaration.
- A virtual method is abstract if you put =0 at the end.

Pass by Value -> No Polymorphism

- In C++ you can pass objects by value. When you do this it copies the data for the exact static type. You don't get polymorphism this way.
- In order to get inclusion polymorphism in C++ you need to be dealing with a pointer or a reference.
- Let's review pointer syntax from C real quickly.

References in C++

- C++ also adds reference types. These store a reference to another variable.
- A reference variable is declared with the & in the place you would put a * for a pointer.
- It must be set on creation and all further uses act as if they were on the thing that it references.
- Never return a reference to a stack variable and rarely to dynamic memory.

The explicit Keyword

- Constructors that take a single argument normally become implicit type casts. If you don't want this, which is often the case, then the keyword explicit should go in front of the declaration.
- C++ does a lot more implicit conversions than Java does which weakens the type system a bit and allows odd bugs. Using explicit can help with this.

Uses of the const Keyword

- const is a type modifier.
- const variables can't be changed after initialization. Use these instead of #define.
- const arguments to functions should be used if a function doesn't change the value of the argument.
- const methods are methods that don't change an the object they are called on. The const goes at end of declaration.

STL

- The Standard Template Library contains many classes that you can use with virtually any datatype. Vector is an example of this. In generally, vector is safer and easier to use than doing your own arrays.
- We'll talk more about other parts of the STL throughout the semester.

Sharing Files with CVS

- The greatest challenge of working in groups is how to keep the code consistent between multiple people. Pair programming can help you with this, but there are also tools like CVS that can aid you. Eclipse works with CVS.
- One member should create a CVS repository. Create the directory first then run cvs init -d directory to get things going.

Minute Essay

- Write a stack of doubles. Use the vector class to store the data (remember the push_back and pop_back methods).
- Thursday 2:30-4:30+ open lab in 340.
- Assignment #1 is due in a week.