Design Patterns 2

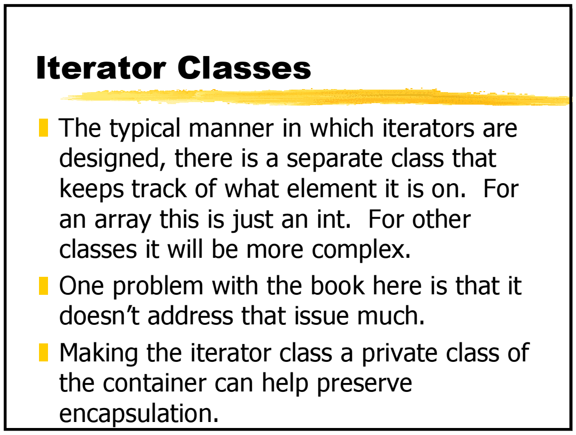
3-4-2002

Opening Discussion

- Do you have any questions about the test?
  - (Median=87, Mean=84.9, RMS=10.8)
  - (As : 7, Bs : 4, Cs and lower : 4)
- What did we talk about last class? Do you have any questions about assignment #3?
- What were the design patterns from the reading for today?

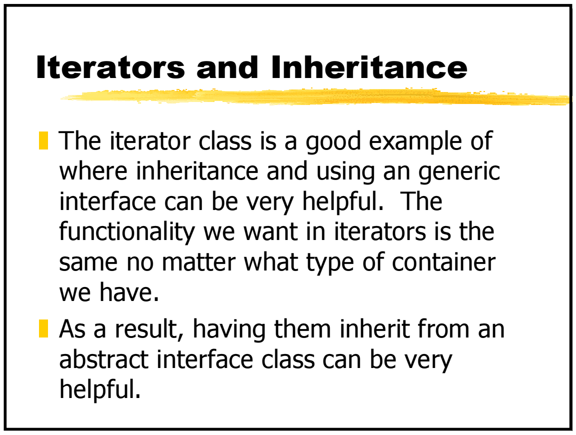
Iterator Pattern

- When we have containers, one of the standard things that we want to do is to be able to loop through the elements of that container. This is often called iteration.
- A basic iterator class only need the functionality for the components of a loop: initialization, conditional check for the end, and iteration.



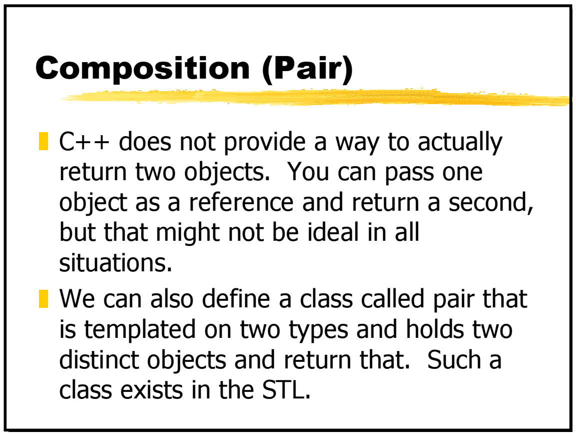
Iterator Classes

- The typical manner in which iterators are designed, there is a separate class that keeps track of what element it is on. For an array this is just an int. For other classes it will be more complex.
- One problem with the book here is that it doesn't address that issue much.
- Making the iterator class a private class of the container can help preserve encapsulation.



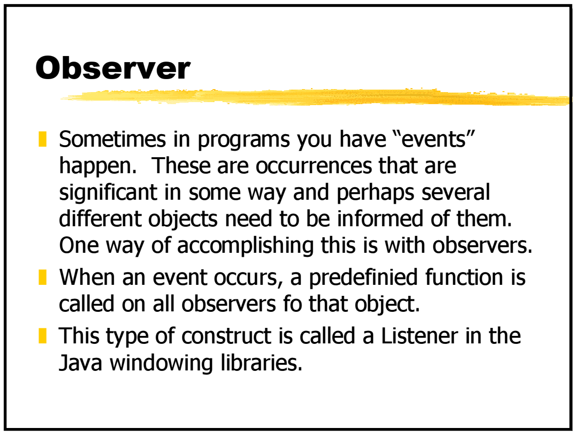
Iterators and Inheritance

- The iterator class is a good example of where inheritance and using an generic interface can be very helpful. The functionality we want in iterators is the same no matter what type of container we have.
- As a result, having them inherit from an abstract interface class can be very helpful.



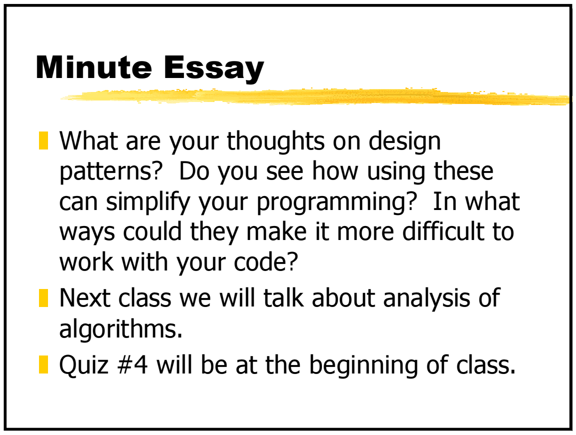
Composition (Pair)

- C++ does not provide a way to actually return two objects. You can pass one object as a reference and return a second, but that might not be ideal in all situations.
- We can also define a class called pair that is templated on two types and holds two distinct objects and return that. Such a class exists in the STL.



Observer

- Sometimes in programs you have “events” happen. These are occurrences that are significant in some way and perhaps several different objects need to be informed of them. One way of accomplishing this is with observers.
- When an event occurs, a predefinied function is called on all observers fo that object.
- This type of construct is called a Listener in the Java windowing libraries.



Minute Essay

- What are your thoughts on design patterns? Do you see how using these can simplify your programming? In what ways could they make it more difficult to work with your code?
- Next class we will talk about analysis of algorithms.
- Quiz #4 will be at the beginning of class.