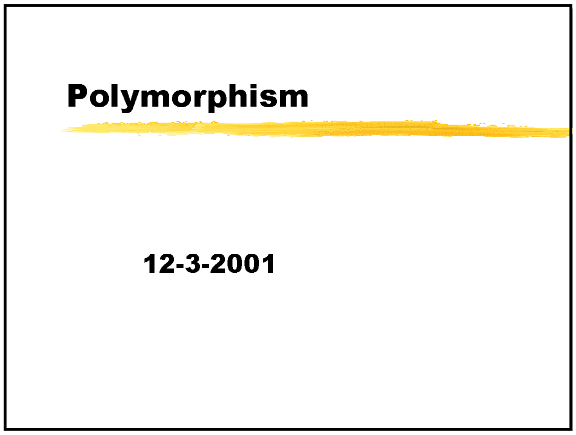
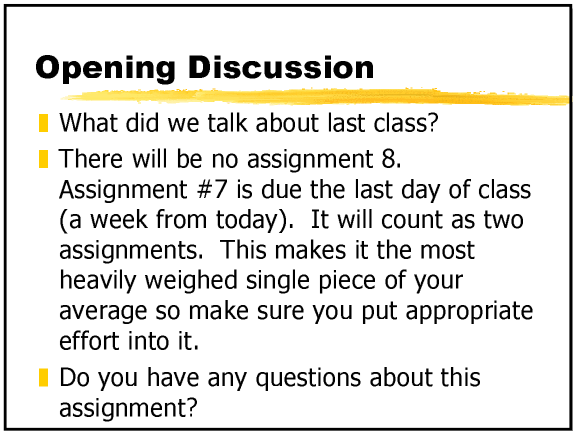


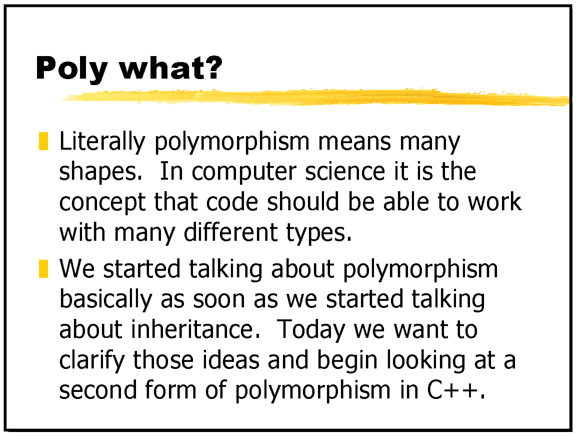
Polymorphism

12-3-2001



Opening Discussion

- What did we talk about last class?
- There will be no assignment 8. Assignment #7 is due the last day of class (a week from today). It will count as two assignments. This makes it the most heavily weighed single piece of your average so make sure you put appropriate effort into it.
- Do you have any questions about this assignment?



Poly what?

- Literally polymorphism means many shapes. In computer science it is the concept that code should be able to work with many different types.
- We started talking about polymorphism basically as soon as we started talking about inheritance. Today we want to clarify those ideas and begin looking at a second form of polymorphism in C++.

Motivation

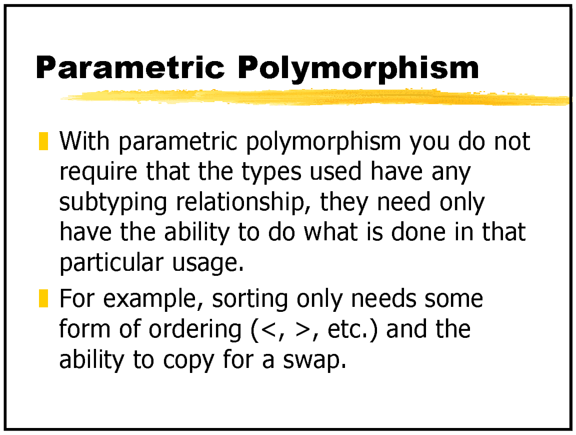
- Polymorphism is basically the ultimate form of making code reusable. The basic idea is that we only want to write code once and have it work in as many cases as possible.
  - Polymorphic functions - For a function like a sort we don't want to have to write new versions for every type of things we want to sort.
  - Polymoprhic containers - Instead of writing hundreds of linked lists (or other things), write on and have it work with hundreds of types.

Types of Polymorphism

- Ad hoc - works with a limited number of types
  - Overloading
  - Coercion - forcing the type to change
- Universal - works with a potentially infinite number of types.
  - Inclusion Polymorphism
  - Parametric Polymorphism

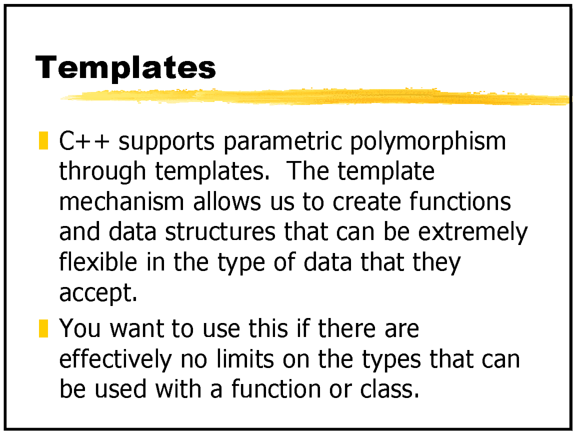
Inclusion Polymorphism

- This type of polymorphism you have seen. It has been developed the previous 3 class meetings.
- This is basically polymorphism by subtyping. Any place you use the supertype you can use a subtype. You have to have virtual functions to get the real power of this.
- Any thing you would do with the supertype can be done with a subtype.



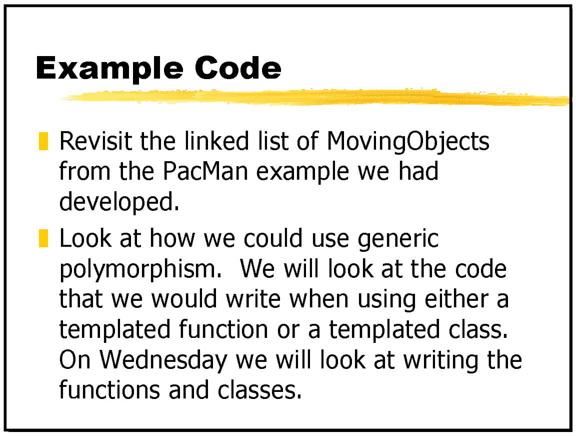
Parametric Polymorphism

- With parametric polymorphism you do not require that the types used have any subtyping relationship, they need only have the ability to do what is done in that particular usage.
- For example, sorting only needs some form of ordering (<, >, etc.) and the ability to copy for a swap.



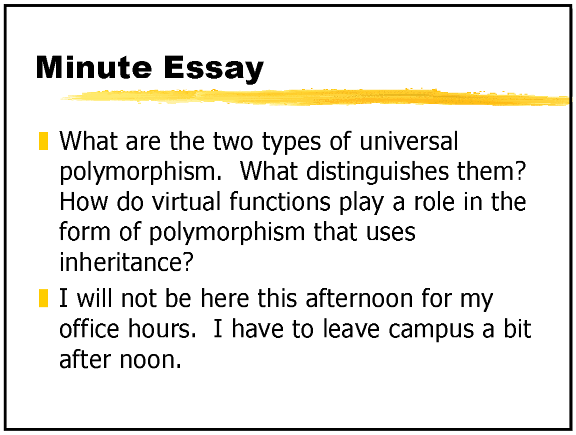
Templates

- C++ supports parametric polymorphism through templates. The template mechanism allows us to create functions and data structures that can be extremely flexible in the type of data that they accept.
- You want to use this if there are effectively no limits on the types that can be used with a function or class.



Example Code

- Revisit the linked list of MovingObjects from the PacMan example we had developed.
- Look at how we could use generic polymorphism. We will look at the code that we would write when using either a templated function or a templated class. On Wednesday we will look at writing the functions and classes.



Minute Essay

- What are the two types of universal polymorphism. What distinguishes them? How do virtual functions play a role in the form of polymorphism that uses inheritance?
- I will not be here this afternoon for my office hours. I have to leave campus a bit after noon.