More Inheritance

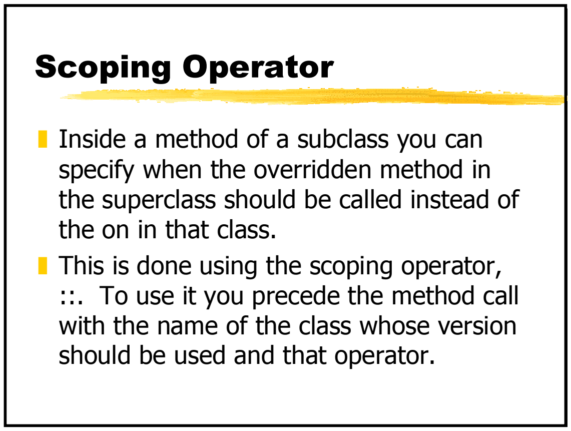
2-13-2002

Opening Discussion

- What did we talk about last class? Do you have any questions about assignment #2?
- Once again I remind you to read the chapter. I'm not discussing all the nuances on inheritance so you will need to look at that for some of the information. If people ask questions about it in class I will be more than happy to discuss it though.

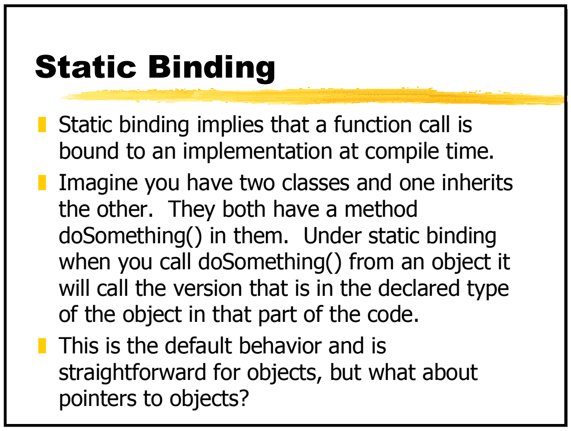
Overriding Methods

- So there is an interesting question about what happens when you put a method in an subclass with the same name are argument types as one in the superclass. This is called overriding and the result depend on the exact nature of the fuinction declaration and how it is called.
- This is what we will look at today.



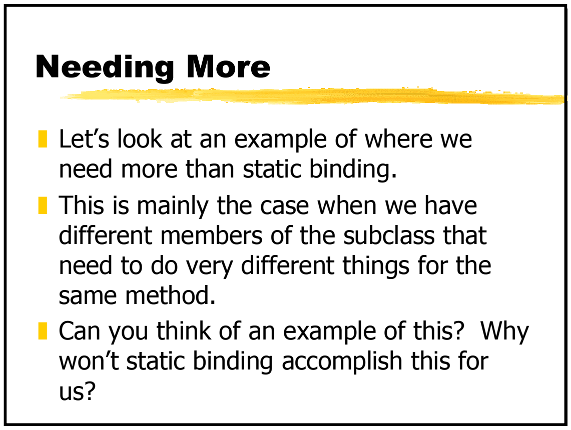
Scoping Operator

- Inside a method of a subclass you can specify when the overridden method in the superclass should be called instead of the on in that class.
- This is done using the scoping operator, ::. To use it you precede the method call with the name of the class whose version should be used and that operator.



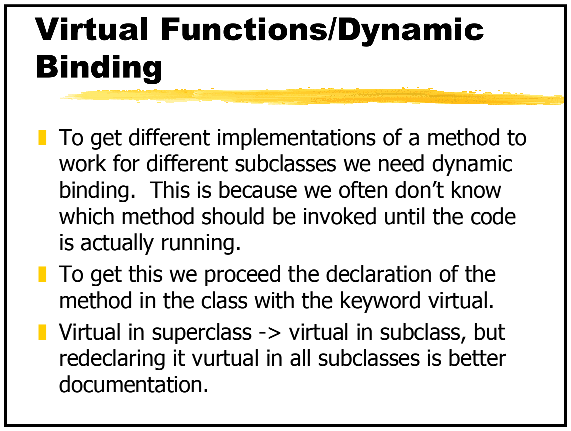
Static Binding

- Static binding implies that a function call is bound to an implementation at compile time.
- Imagine you have two classes and one inherits the other. They both have a method doSomething() in them. Under static binding when you call doSomething() from an object it will call the version that is in the declared type of the object in that part of the code.
- This is the default behavior and is straightforward for objects, but what about pointers to objects?



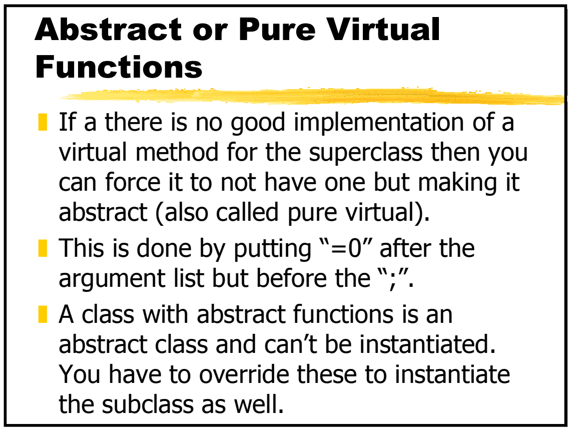
Needing More

- Let’s look at an example of where we need more than static binding.
- This is mainly the case when we have different members of the subclass that need to do very different things for the same method.
- Can you think of an example of this? Why won’t static binding accomplish this for us?



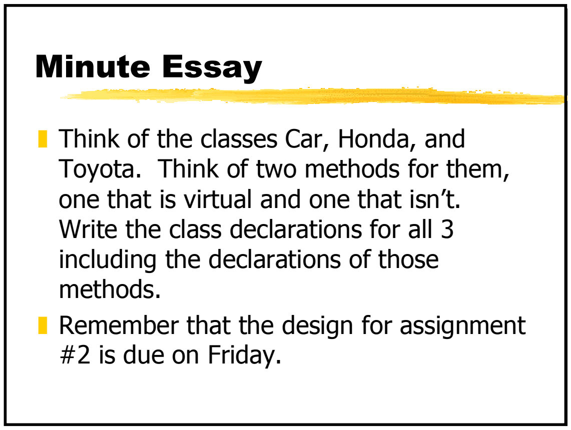
Virtual Functions/Dynamic Binding

- To get different implementations of a method to work for different subclasses we need dynamic binding. This is because we often don't know which method should be invoked until the code is actually running.
- To get this we proceed the declaration of the method in the class with the keyword virtual.
- Virtual in superclass -> virtual in subclass, but redeclaring it vurtual in all subclasses is better documentation.



Abstract or Pure Virtual Functions

- If a there is no good implementation of a virtual method for the superclass then you can force it to not have one but making it abstract (also called pure virtual).
- This is done by putting "=0" after the argument list but before the ";".
- A class with abstract functions is an abstract class and can't be instantiated. You have to override these to instantiate the subclass as well.



Minute Essay

- Think of the classes Car, Honda, and Toyota. Think of two methods for them, one that is virtual and one that isn't. Write the class declarations for all 3 including the declarations of those methods.
- Remember that the design for assignment #2 is due on Friday.