



# Inheritance in Java

1/25/2007



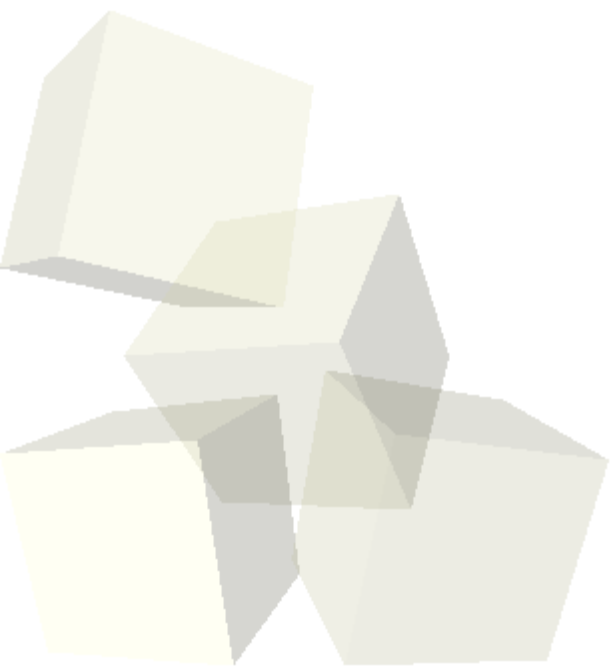


# Opening Discussion

- Do you have any questions about the assignment? Let's look at the description and talk about it just a bit, then look at how you will submit your code.
- What is inheritance? What two things does inheritance provide?
- What does it mean when we say one type is a subtype of another type?
- What does it mean for a method to be virtual?
- What is an abstract method?
- What are final methods in Java?
- What is the common subtype for all classes in Java?



- Go to your account main and have it make a second account. Deposit 200 into that account. Then print the balance on both accounts.
- Does it make sense what is happening?





- Now I want you to construct some code that uses inheritance and polymorphism.
- The reading runs you through the classic example of a shape so We are going to do something a bit different. I want to use the example of a simple function of one variable as our supertype and then create subtypes for specific types of functions.
- We'll make an interface called MathFunction. What can you do with one of these? What subtypes of this function could we consider creating?
- Once we have the interface and some subtype of it, we can look into making some polymorphic functions that use them.



# Java 5.0 Features

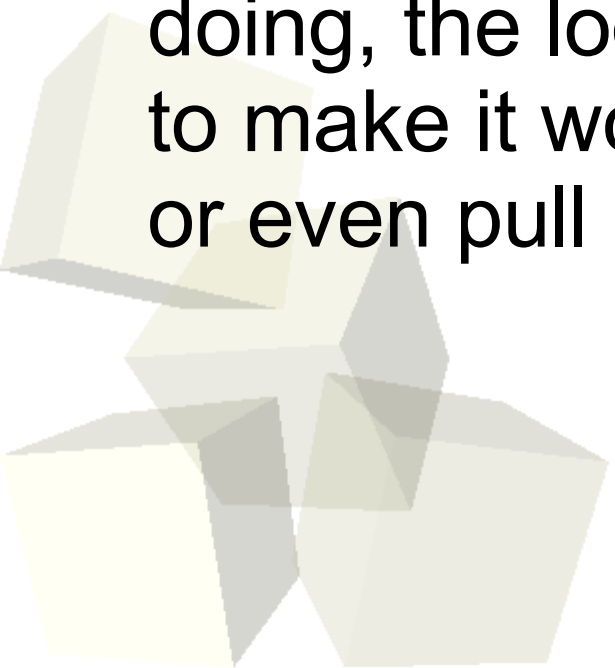
- Java 5.0 adds quite a few features to the Java language that make it easier to write real code in Java. Most of class today we will focus on two big ones, generics and enums. There are a few others worth mentioning.
- Autoboxing – primitive types are not objects, but sometimes you need to treat them that way. This automates the process.
- Varargs – Java now has syntax for passing a variable number of arguments. We won't use this in this class.
- Foreach loop – A handy shortcut for when you want to do something with all elements of a container.



- The most significant feature added to Java 5.0 was that of generics. Generics provide a form of parametric polymorphism, typically for code that can take any type, but might be limited to a specific type for one instance.
- The most common use of this is for containers as container classes typically should be able to hold anything, but any one container is generally intended to hold only one type.
- In practice, generics give you extra type safety and prevent you from doing a lot of type casts.



- Generics play a big role in the project and appear in the first assignment so I want to show you a bit of the syntax.
- I'm going to write some code that uses generics so show the syntax. I'd rather you not try to follow along with this coding at the end of class and instead just try to pay close attention to what I'm doing, the logic behind it, and the syntax required to make it work. You can see the code on the web or even pull it down so you can play with it later.





# Minute Essay

- What is a potential pitfall of inheritance? Name something you shouldn't do with inheritance.
- Remember to turn in the first assignment by tonight.
- We have our first quiz at the beginning of next class. It will be over everything that we have done so far and can include information from all the readings.

