

CS61B Summer 2006
Instructor: Erin Korber
Lecture 2, 27 June
Reading for tomorrow: Ch. 3 (if not done already), Ch. 4 (pages 71-79 only), Ch. 5 and 6

1 Using Objects

- State
 - Things the object *knows*
 - these are *instance variables*
- Behavior
 - Things the object *does*
 - these are *methods*

2 Classes

A class is a blueprint for an object. We can use the same class to make many objects (*instances*), each of which has its own values for the instance variables.

- Everything in Java goes in a class.
- But what about `main`?
 - to test your real class
 - to launch your application
- What about global variables?
 - This is possible, but it's a very special case.
 - Will learn how to do this later.
- A Java program
 - A bunch of classes, one of which has a `main` method
 - At runtime: objects interacting with each other

3 Creating Objects

- Simplest constructor: `Thing t = new Thing();`
- `t` is an *object reference* variable
 - A way to access the object - a “remote control”
 - Does NOT hold the object itself.
 - Can refer to nothing - `null`

4 Primitives

- The only things you will see in Java that are not objects.
- 8 types of primitives:
 - 4 integers - `long`, `int`, `short`, and `byte`
 - 2 floating point - `double` and `float`
 - 2 others - `boolean` and `char`

5 Variables

- 2 main kinds
 - 1. To hold primitives - size depends on what kind it is
 - 2. To hold object references - all the same size
- Must have a name and a type, which cannot change.
- The type of a reference variable - what kind of object it points to.
- The *value* - what’s “in the container” can change.

6 Arrays

- Arrays are objects (regardless of what they hold).
- Elements of an array are just variables. So anything you could put in a variable of that type can be put in an array element of that type.