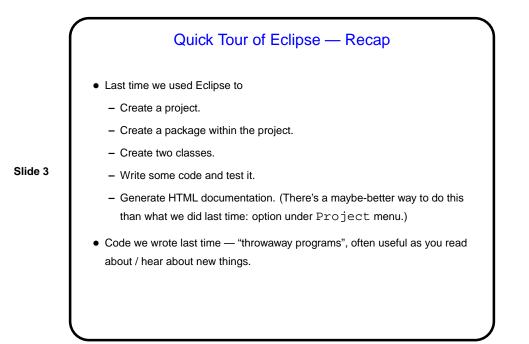


## A Little More About Homework 1

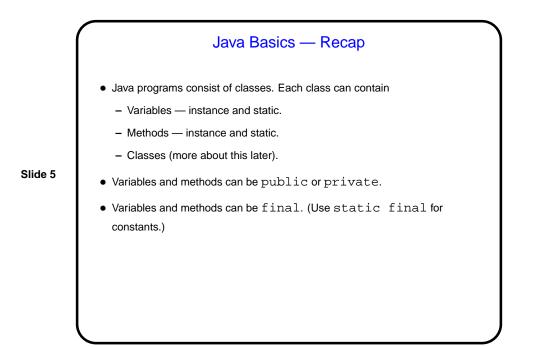
• You're not committing yourself to anything at this point, but try to be as detailed as you can — so I can try to spot potential trouble. Also good to think in terms of a basic design (not too ambitious) plus extras. Keep in mind that what you do has to fit into an existing framework. (That's actually one of the pedagogical goals.)

Slide 2

• What you will actually turn in is HTML documentation of your planned game's main class — put it in your Local/HTML-Documentation and send me mail saying "ready to be graded". (Complete instructions in homework writeup.)



## Compiling and Running Programs — Java Versus C/C++ With C/C++, your program ("source code") is transformed by a compiler into ... "object code" (different for different processors), which is combined with library object code to produce ... an "executable" (different for different operating systems) that can be run like other applications. With Java, your program (source code) is transformed by a compiler into ... "byte code" (same on any processor), which is executed by ... "Java virtual machine" (which has access to library byte code).

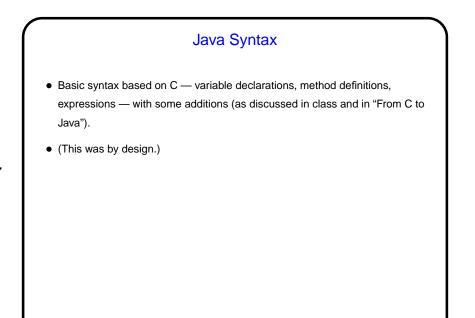


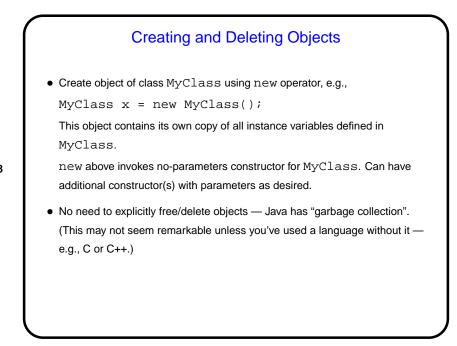
Variables
Primitive types provided for efficiency (not purely object-oriented):

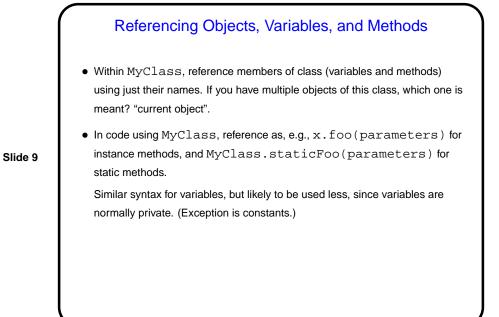
boolean, short, int, long, float, double are pretty much as in C.
char is 16-bit Unicode.
byte is 8-bit byte.

All other variables are *references to objects*, similar to pointers:

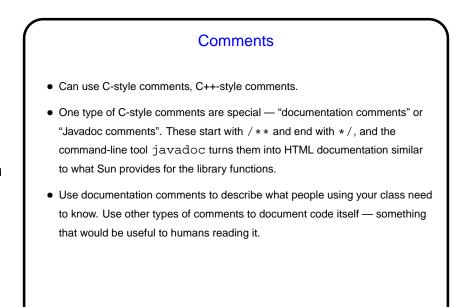
MyClass x creates a *reference*, not an object — use new to create objects.
Type of x is MyClass (just as type of an int variable is int).
Value of null means it doesn't point to anything.







**Passing Parameters** • Syntax is like C. • Everything is passed by value -- but for reference variables, copying just creates two pointers to the same object, and the called method can change the object. Slide 10 (More about this later.)



## Control Structures Most control structures are the same as C — if, while, do, switch, for, etc. Also a simplified for, as of Java 5.0 (a.k.a. 1.5). More about it later. Also have "exceptions" — a way to deal with unusual or error conditions, break out of current flow of control. Can be "thrown" and "caught" (or not caught, in which case the program crashes). More about them later.

