Administrivia

- Comments about Homework 1 design mailed today.

- Reminder: Homework 1 code due today.

- Reminder: Quiz 2 Thursday.

- Homework 2 design due Thursday, code next Tuesday.

Slide 1

- Office hours revised slightly (see my home page).

Strings in Java

- In C, "strings" are just arrays of characters, terminated by a null character. Simple, but many potential problems (such as trying to read more characters from input than will fit into allocated space).

- In Java, there's a library class, `String`.

Slide 2

- To see what's available, look at the API . . .

String Class, Continued

- In general, no operator overloading in Java, with one exception — "+" for strings. Non-string objects converted using (their) `toString` method. Primitives converted in the "obvious" way.

- To compare two strings, "==" is rarely what you want. Instead, use `equals`.

Slide 3

- Strings are "immutable" — once created, can't be changed. (Why? allows them to be safely shared.) Methods you would think might change the value return a new string.

- Use `StringBuffer` if you need something you can change, or for efficiency.

- Let's do some examples . . .

Sidebar — Immutable Objects

- `String` is an example of a class that's "immutable" — once created, objects can't be changed. If you look at the API for `String`, you notice that methods that "change" the string actually return a new one.

Slide 4

- This sounds inconvenient, right? What advantages might it have? (Remember that "object" variables in Java are really references. So two variables may both refer to the same object.)

Arrays in Java

Slide 5

- Arrays are objects — unlike in C/C++, where they're basically pointers.

- Declaring (references to) arrays — denote by putting brackets after type.

- Creating arrays — use `new`, e.g.,

  `new int[10]`

  `new String[n]`

  (Notice that the second one only creates *references*.)

- All arrays have `length` variable.

- Otherwise, syntax is same as C/C++; indices start at 0.

- Java runtime does automatic bounds-checking — unlike in C/C++, get "exception" rather than random problems.

Multidimensional Arrays

Slide 6

- "Arrays of arrays", e.g.,

  `int[][] x = new int[10][100];`

  declares an array of 10 arrays of 100 `int`s.

- Reference elements with row, column indices, e.g.,

  `x[row][col] = 10;`

- Both dimensions accessible:

  `x.length = ?`

  `x[0].length = ?`

Slide 7

Minute Essay

- None — sign in.