Slide 1

Administrivia

- How many of you plan to take CSCI 1323 (Discrete Structures) in the spring?
- Reminder: Homework 3 due today. Okay to turn in tomorrow.
- Midterm moved to October 18. Next quiz next time.

Slide 2

Arrays and Lists — Overview

- With what we've done so far we have enough tools to compute anything we want to compute.
- However, some things are awkward (repetition), and we don't yet have a convenient way to store many values — something similar to subscripted values in math.
- Most programming languages give you a way to represent *collections*. Exactly what you get depends on the language — e.g., C gives you only something quite primitive (but close to what the hardware can do), Java gives you something more abstract/useful, and Scala goes even further.

Slide 3

Arrays and Lists in Scala

- Scala provides two ways of representing a "sequence" (ordered list of elements), `Array`s and `List`s. From the outside they look very similar, but behave fairly differently:

  An `Array` has a fixed number of elements, but the values of the elements can be changed.

  A `List` cannot be changed at all, but there are easy and efficient ways to build lists.

- Both are "parameterized types", which means you can specify the type of the elements.

Slide 4

Arrays in Scala

- Two syntaxes for creating an `Array`. Examples:

```
val a1 = Array(1,2,3,4)       // four elements, initial values as given
val a2 = new Array[Int](10)   // ten elements, all zero
```

- Syntax for referencing element uses name of array plus index in parentheses. Indexes range from 0 through length minus 1. Examples:

```
println(a1(1))
a2(2) = 20
```

Slide 5

Arrays in Scala, Continued

- Length of `Array` can be obtained with `.length` or `.size`.
- That gives us enough to write some simple functions using recursion . . .

Slide 6

Minute Essay

- TBA