Slide 1

Administrivia

- Reminder: Homework 2 due next Tuesday.

Slide 2

Minute Essay From Last Lecture

- Several people commented that we/I go kind of fast in class. Yes. If you don't keep up, examples will mostly be online later, or ask after class / in office hours.
- One person wrote:

  "It is interesting though how long it takes to write such a simple program which humans can do in their heads in seconds."

  So maybe we really are smarter than computers, at least in some ways? If you wonder then what the point is — the payoff is if you can use a program to get the computer to do something that would be slow or difficult for humans, such as lots of calculations.

Slide 3

Example — Review/Revisited

- Last time we wrote a program to count out change. Some calculations were somewhat cumbersome, partly because we used only `val` variables — a so-called *functional style*, after math functions. Has its benefits but isn't always/necessarily best.
- We could write it another way, using a `var` variable for the starting number of cents. Would be closer to how we solve the problem as humans, and might be shorter.

Slide 4

Conditional Execution

- So far all our programs have executed the same statements every time, just maybe with different numbers.
- Often, though, we want to be able to do different things in different circumstances — for example, print an error message and stop if the input values don't make sense (such as a negative number for the program to count out change).
- So, Scala (like most languages) provides some constructs for *conditional execution*. Before we talk about them, we need . . .

Slide 5

Boolean Expressions

- A *Boolean value* is either *true* or *false*; a *Boolean expression* is something that evaluates to true or false.
- We can make simple examples in Scala using familiar math comparison operators (except that the ones for which the keyboard doesn't have a symbol require more than one character). Examples:
  - `x > 10`
  - `y <= 5`
  - `x == y` (**NOTE** the use of `==` and not `=`.)

Slide 6

Boolean Expressions, Continued

- *Boolean algebra* defines some operators on these values; the most important for us now are written in Scala as
  - `!` — "not", true if the operand is false.
  - `&&` — "and", true if both operands are true.
  - `||` — "or", true if either operand is true (or both are).
- Can use these to build up complex expressions. As with arithmetic expressions, use parentheses when in doubt. Examples:
  - `(x >= 0) && (x <= 10)` (What if we just write `0 <= x <= 10`?)
  - `!(x == y)` (though we could also just write `x != y`).

Slide 7

Boolean Expressions in Scala

- Scala has a type for boolean values (`Boolean`) with the obvious values.
- One thing to know is that the operators for "and" and "or" do not evaluate both operands if the value of the first operand determines the result.

Slide 8

Conditional Execution — `if/else`

- To execute a statement if an expression evaluates to true, use `if`:

```
if (x > 0)
    println("greater than zero")
```

- To execute one statement if an expression is true, another if it's false, use `if` and `else`:

```
if (x > 0)
    println("greater than zero")
else
    println("not greater than zero")
```

Slide 9

`if/else` in Expressions

- Similar rules apply within expressions, e.g.,

```
if (x < 0) -1 else 1
```

  has the value -1 if X is less than zero, 1 otherwise.

- Many programming languages have a similar construct but express it differently; in C and Java the equivalent expression is

```
(x < 0) ? -1 : 1
```

Slide 10

`if/else` and Blocks

- To execute a group (“block”) of statements rather than just a single statement, use curly braces for grouping:

```
if (x > 0) {
    println("greater than zero")
    println("and that is good")
}
else {
    println("not greater than zero")
    println("and that is bad")
}
```

Slide 11

`if/else`, Continued

- What happens if you forget the braces? The program may still run, but it probably won't do what you meant.
- Several styles for where to put the curly braces. Which is best? Some people care; I say pick one that's readable and stick with it.

Slide 12

Example — Calculating Grades

- As a simple example, consider a program to calculate numeric and letter grades as described in the syllabus. (Next class.)

Slide 13

Minute Essay

- None — quiz.