Slide 1

Administrivia

- Reminder: Homework 1 due today, 11:59pm.
- Homework 2 on the Web; due next week. First programming homework!

Slide 2

More Administrivia

- When turning in homework, please do identify in the subject line both the course and the assignment. I ask for this so it's easy for me to drop it into the right "folder" for grading — I'll be getting assignments by e-mail for four courses this semester.
- As with the minute essays, if there's an urgent question in what you're sending, put "question" or "urgent" in the subject line.
- If you need/want to mail homework to me when you're logged in remotely (or if you just want to know one more thing you can do from the command line!), see the `mail-files` script on the "Sample programs" page.

Minute Essay From Last Lecture

- Many people said what we had done in C so far looked at least a little familiar, either to Matlab or to something else (C++ or Java/Arduino – in both cases, no accident as these both use C's syntax for basic things).
- A couple of people seemed unclear on the distinction between C (a programming language) and Linux (an operating system). "Hm!"?

Slide 3

Example — "Counting Change"

- Problem statement: Given a number of pennies, show how to represent it with minimum number of coins (pennies, nickels, etc.).
- First define the problem, possibly doing some examples without a computer. Might be a good time to also come up with a short list of sample inputs/outputs that can be used for testing later.
- Next figure out a strategy for solving it using the tools you have.
- Finally turn that into source code. Good idea to start by writing *comments*, because . . .

  When writing source code you are writing for two audiences! the compiler, yes, but also (usually) for human readers.

Slide 4

Slide 5

Example — “Counting Change”, Continued

- Turns out that there are two basic strategies we could try:
- One starts by first figuring out whole dollars, then quarters, etc.
- Another goes the other direction — first pennies, then nickels, etc. (But trying a couple of examples — this one doesn’t work well.)

Slide 6

Defining Named Constants with Preprocessor Directives

- Sometimes it makes sense to use numeric constants in programs — e.g., in “count change” example.
- But sometimes it’s more readable, for humans, to give these constants a name. Can do this with `#define`. Examples (somewhat contrived):

  ```
  #define NICKEL_CENTS 5
  ```

  Then when you write `NICKEL_CENTS`, compiler (strictly speaking, its preprocessor) replaces it with 5.

Slide 7

Minute Essay

- Have you tried writing and running any programs yet?
- Any questions?