Slide 1

Administrivia

- Reminder: Homework 4 due next week.
- Homework 3 grades mailed. Most people who turned something in did well, and only a few didn't remember about checking for input errors!

  (A significant number didn't turn anything in. If that's you, it's not too late to get part credit. Maybe we should talk, though?)
- Sample solution for Homework 3 posted.

Slide 2

Minute Essay From Last Lecture

- No clear consensus on how to get seconds per hours, etc. — some people calculated, some used Web search, and a few . . .
- A few did the calculation in the program. To me this is better! more readable, less chance of getting it wrong.

Slide 3

Homework 3 Essays

- Several people weren't happy with their solutions to the first problem. Indeed, solutions *can* be long and complicated. Look at the sample solutions for what I think are several about-as-simple-as-possible solutions.
- A few people mentioned that the second problem was straightforward — just turn the math into code — which, yes, that's one of the reasons to use a recursive function!

Slide 4

`main()` Revisited

- `main()` should (for now) be declared

  `int main(void)`

  (There's an alternate form used when you need access to command-line arguments. A topic for next week!)

  Note that in C this is subtly different from

  `int main()`

- Return value from `main()` should be zero if the program "worked" (whatever that means), something else if it didn't. Appropriate values for the "something else" are somewhat implementation-dependent. If you want to do a really good job of this, `#include stdlib.h` and use

  `EXIT_SUCCESS` and

  `EXIT_FAILURE`.

Slide 5

Recap of Video Lectures

- A little about "random" numbers, plus a digression about TEX.
- An introduction to pointers — perhaps the most difficult topic this course addresses, and the one that matters most in preparing you for Data Abstraction.
- Questions?

Slide 6

Homework 4

- The second problem gives many students trouble.
- Outline of what you're supposed to do:

  Generate N "samples" using `srand()` and `rand()`.

  Map each of them to a range from 0 to $B-1$, where B is a number of "bins", and count how many fall into each "bin".

  The starter program has code to do both mappings, so you should not need to struggle with that.

Slide 7

Practice Problem

- In CSCI 1312 I use as an example of somewhat complex logic a program to find the real roots of a quadratic equation

$$ax^2 + bx + c = 0$$

  using the formula

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- I like to structure the program with a function to compute the roots and one to "demo" it for selected cases.
- I've put starter code and sample output in `/users/bmassint/TEMP1120/Pgms/`. Your mission is to fill in as much as you can in the time we have. "Turn in" answers as previously.

Slide 8

Minute Essay

- I've mentioned before that how I'm doing this class (video lectures, class time used for something else) is something of an experiment for me. I'm hoping that using class time for practice problems is useful but — what do you think?
- Any questions — about pointers, anything else?