Slide 1

Administrivia

- Reminder: Homework 5 due Friday.
- Review sheet for midterm on the Web. Mostly about exam format but also has list of topics.

  Aside: In general my idea is that students who have kept up reasonably well with reading and homeworks won't have to spend a lot of time preparing for exams. The goal is to test whether you understand the material rather than whether you can memorize!

Slide 2

Numerical-Integration Example — Recap

- Program written in class does two things: illustrates numerical integration and also looping until convergence.
- To make things clearer(?), I made a simpler version also that doesn't loop until convergence. Also I put in code to prompt for inputs and to compare the computed value(s) to the best-available library value for π. (Surprisingly, there's a library constant `M_PI`, but it isn't standard, so use `acos()` to compute.)

Slide 3

A Little About "Random" Numbers

- Among the C library functions discussed briefly in the textbook chapter on functions are `srand()` and `rand()`. A few words about what they do . . .
- First, what we mean by "random" is (I think!) an interesting question with no obvious answer. What's often wanted is something that can't be predicted, and it's not clear we can get that with a system that's deterministic. Further, even if we could, we might not want that, since we often want to be able to repeat a test.
- So, often what we really want is a "pseudo-random number generator" — something that generates a sequence of numbers that looks random but is repeatable given some reproducible starting point.

Slide 4

Pseudo-Random Number Generators

- Mathematically interesting topic; classic reference is in one of the volumes of Donald Knuth's *The Art of Computer Programming*.

  (Aside: Some of you may know Knuth as the inventor of the typesetting system TeX. It's an extreme example of a "side project" that turned into something much more?)
- Early researchers apparently thought more-complex algorithms would give better results, but — not necessarily. Very simple algorithms can give quite good results. For example, one reasonable one (not the best, but good) computes each element of the sequence in terms of the previous one:

  $x_{n+1} = (ax_n + b) \bmod M$

  for carefully selected values of a, b, and M.
- Uses in programming include simulating various things in the physical world. Textbook examples often involve simulating rolling dice, shuffling cards, etc.

Slide 5

Pseudo-Random Number Generators in C

- C library includes functions `srand()`, `rand`. `srand()` uses a “seed” to initialize some behind-the-scenes variables, after which you call `rand()` repeatedly to generate a sequence of “random” numbers. If you do this more than once with the same seed you get the same sequence; using different values of the seed gives different sequences.
- “Monte Carlo” algorithms are based on “random” numbers. An example is a program to estimate π by simulating throwing darts into a board containing a quarter circle. (Example program.)

Slide 6

Character Data

- As mentioned previously, in C we can represent characters as type `char`.
- Simplest way to input/output a single character is with `getchar` and `putchar`. Note that `getchar` returns an `int`; this is so there can be a “special” value (`EOF`) for “end of file”. (For input from a terminal, signal with something system-dependent, control-D on Linux machines.)
- Functions in `ctype.h` classify characters as alphabetic, digits, etc. Functions `toupper()` and `tolower()` do what their names suggest.

Slide 7

Files and C

- Why files? You probably already know: Things stored in memory vanish when you turn the computer off; to preserve them, usually save them as *files*.
- We know one way for a C program to get its input from a file, or write its output to a file — I/O (input/output) redirection. But this makes it difficult to get input from more than one source, or save output in more than one place.
- So C (like many other programming languages) provides ways to work more generally with files. (To be continued!)

Slide 8

Minute Essay

- None — quiz.