

Slide 1

Administrivia

- Reminder: Homework 8 accepted without penalty through Monday (December 6). Homework 9 accepted without penalty through final deadline. Late penalties for Homeworks 7, 8, and 9 reduced to 5% per working day (usually 10%).
- Not-accepted-past deadline for all homeworks December 14.
- I'll continue to keep you posted on grade status.
- The ACM tutors won't be available past the end of classes. I have good intentions of having office hours during part of the time when they're normally available, most days, but my record is not good, and it may be better to let you know early in the day whether I'll be able to have them that day. I'll ask in the minute essay.
(Be advised that my record on answering questions via e-mail is much better! not as satisfactory but something?)

Slide 2

More Administrivia

- And this is a good time to say . . .
- I really hoped that this semester would go better than last spring, when poor health was also a problem. I started the semester thinking I'd made some improvements, but they didn't last once the normal academic-year stresses started to build up.
- Note, however, that in my thinking most learning in this course takes place as you work through assignments rather than in class.
- And curiously enough, I've kept up with grading far better than usual!

Slide 3

Quotes of the Day/Week/?

- From a key figure in the early days of computing:
“As soon as we started programming, we found to our surprise that it wasn’t as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent finding mistakes in my own programs.” (Maurice Wilkes: 1948)
- From someone in a discussion group for the Java programming language:
“Compilers aren’t friendly to anybody. They are heartless nitpickers that enjoy telling you about all your mistakes. The best one can do is to satisfy their pedantry to keep them quiet :)”

Slide 4

Course Topics — Recap

- Basic C programming, for people who already know how to write programs in some other language. Especially useful (I think!) for those who start in a very abstract/high-level language.
- Review of the Linux/UNIX command-line environment and command-line development tools.
- Review of basics of computer arithmetic and data representation. A little more about floating-point representation.

Why Learn C? (For Java/Python/Scala Programmers — Recap)

Slide 5

- Scala and Python (and Java, less so) provide a programming environment that's nice in many ways — lots of safety checks, nice features, extensive standard library. But they hide a lot about how hardware actually works.
- C, in contrast, has been called “high-level assembly language” — so it seems primitive in some ways compared to many other languages. What you get (we think!) in return for the annoyances is more understanding of hardware — and if you do low-level work (e.g., operating systems, embedded systems), it may well be in C. (Performance *may* also be better, though “measure and be sure”.)

Course Evaluations

Slide 6

- Used for two purposes:
 - By those up the chain of command as one measure of my performance.
 - By me as I reflect on this semester and plan for future semesters.
- I find the numbers less useful than the prose, so please do more than just the numbers if you can!
- URL is the same as for your other courses, but I'll send it via e-mail just in case. Password shared in the recording.

Minute Essay

- About office hours, I was thinking an hour or two most days during the hours the ACM tutors have been available. Does that work for you?
- Any administrative questions?
- Best of luck finishing the semester, and best wishes for a pleasant (and safe!) winter break!

Slide 7