

Slide 1

Administrivia

- Reminder: Homework 7 due today at 5pm. Not accepted past Thursday at 5pm.
- Solutions to programming problems on Web soon.
- Grades on programming assignments will be e-mailed when available.
- Review sheet for final on Web.
- Extra-credit problems are a possibility — if provided, would be on Web soon, due early next week.
- Office hours this week to be announced by e-mail.
- Should we have a review session? (Yes. Friday at noon. Room TBA by e-mail.)

Slide 2

Course Recap

- Recall from first class — course is an “introduction to programming.”
- Ideally, a first course would focus more on ideas of programming than details — except that, in the words of a colleague
“Programming is not a spectator sport.”
so we have to choose a programming language, and an environment, and then it's difficult *not* to get caught up in the details.

Course Recap, Continued

Slide 3

- Course was originally designed to meet the needs of two audiences:
 - Prospective CS majors/minors.
 - Majors in other fields where programming is a useful skill (math, engineering, sciences).
- More recently, course was added to common curriculum.
- Meeting the needs of all these diverse groups — may not even be possible, at best one must compromise.

Course Recap, Continued

Slide 4

- Many possible choices for a first programming language/environment.
- In this course, we use C and simple tools because we think this works best for our original target audience — helps you understand what's going on, at a fairly low level.
- Once you know *one* programming language, the next one is easier, and the next one easier still. (Caveat: "Easier" is relative to how similar language is to one you already know. Programming languages come in groups — procedural, functional, object-oriented, etc.)

What I Hope You Got From This Course

Slide 5

- A basic understanding of what programming is — expressing a problem and its solution as “an algorithm” and turning that into code.
- A good enough foundation in *some* programming language to let you write programs using whatever tools/environment are appropriate to your field.
- If you never write another program — now you know what “source code” is, and you’ve done something most people don’t have any idea how to do, and you’ve done it using tools that are really not novice-friendly!
- (If you found this course difficult — many people do! Subject is not easy, and a lot of material to cover.)
- (Look briefly at an example of code that does something more complicated / interesting?)

Minute Essay

Slide 6

- How did the course compare to your expectations/goals? Did you learn what you hoped to learn?