Slide 1

Administrivia

- Reminder: Final reports, code, and individual evaluations (of the others in your group) due today, or ASAP.
- Syllabus lists again what your grade is based on. If your group turned everything in, and you attended class regularly, you will likely make an A.

Slide 2

A Little History / Rationale

- Impetus for the course — in part, difficulty seniors had with senior software project, where they were asked for the first time to do requirements analysis, high-level design, group work, etc. Idea was to give students some prior practice / experience.
- Prior to most recent catalog change, all majors except first-year took some section of P/E/D. So common sessions could work as a kind of department seminar / way to build community among majors. Senior P/E/D, though, seemed a bit redundant, and we wanted to add CSCI 1194 (survey course), so it was dropped.
- So now we have only sophomores and juniors. Goals?

Slide 3

What We Hoped This Course Would Teach You

- The name gives some hints:
  - “Professional” — a little about computer science as a profession.
  - “Ethics” — a little about ethics as it relates to computer science.
  - “Design” — a little about high-level design.
- But it’s only a one-unit course . . .

Slide 4

“Professional”

- Goal — give you some exposure to computing professionals outside academia, tell you a little about CS as a profession.
- Mostly we do this via outside speakers; this year we had several, from different companies.

Slide 5

“Ethics”

- Goal — review “codes of ethics” laid out by professional bodies (ACM, IEEE), think about how they apply to sample scenarios.
- Mostly we do this by telling you a little about these codes of ethics (lecture by Dr. Howland), then providing some sample scenarios and making you think about them.

Slide 6

“Design”

- Goal — give you some exposure to requirements analysis and high-level design; that is, what to do when you’re given a not-very-well-defined “problem” and asked to come up with a computing-based “solution”.

  Not clear that this can be taught except by asking you to try it, hence the “design problem”. Possibly you also learn from observing juniors’ and seniors’ presentations.
- Another goal — to provide some practice with ways to represent / formalize this process.

  Hence the lectures on use cases and UML diagrams. Some overlap here with Software Engineering course.
- Yet another goal — give you some practice working in groups.