

### Administrivia

- Did you get a “this is a test of the class mailing list” e-mail message? If not, come talk to me.

Slide 1

### Syllabus / More Administrivia

- One purpose of the syllabus is to spell out policies, especially about:
  - Course requirements and grading.
  - Late work.
  - Academic integrity.
- Most other information will be on the Web, either on my home page (office hours) or the “course Web page”.
- Part of my job is to answer your questions outside class. E-mail usually works well if office hours don’t.

Slide 2

### A Few Words About Computer Use in Class

Slide 3

- Checking your e-mail when you first get here is okay.
- Taking notes online is okay.
- Surfing the Web or playing games during lecture is not okay — fun, but distracts you and maybe your neighbors.
- Remember that I can lock all screens, project what's on one student's screen, etc. — and I will if need be. But I'd rather start by assuming you're all responsible people who will do the right thing!

### Why Are Design Skills Important?

Slide 4

- Large (i.e., “interesting”?) programming projects more or less require good design to have much hope of succeeding.
- Even if you don't plan to program for a living, the ability to analyze a problem and think about how it could be solved using computer systems is a valuable skill. We think this kind of analysis and design work is relatively difficult to “outsource”.

### Design Project — Introduction

Slide 5

- Most of you have had the experience of “collaborative programming” using a single computer — two or more people clustered around a machine, with the ability to
    - Edit, compile, and run code, and view the results.
    - Communicate with each other — verbally and by drawing pictures on a paper or a whiteboard.
- This has many advantages in all phases of program design and implementation, including debugging.
- Your mission for this course is to design an environment that supports this kind of collaborative programming among people who are *not* all clustered around a single machine — i.e., an environment for distributed interactive collaborative programming.

### Design Project — Constraints

Slide 6

- You have some flexibility in deciding exactly what functionality to provide — that’s part of the design problem.
- You also have the following constraints:
  - Your solution should be as cross-platform and portable as possible — e.g., no Windows-only or Linux-only solutions.
  - Your solution should not require spending money — e.g., if you use existing products/programs, they must be public-domain / free.

### Design Project — Major Phases

Slide 7

- First you need to think through what functionality your environment needs to provide. Ultimately you will produce a UML use-case analysis, but first you will probably want to research existing programs/products that provide this kind of functionality. It might also be helpful to research IDEs.
- You will then design an implementation of this functionality and code a prototype. The prototype should focus on showing how your environment would look to a user; you can “fake” parts that are hidden.

### Minute Essay

Slide 8

- What are your goals for this course?