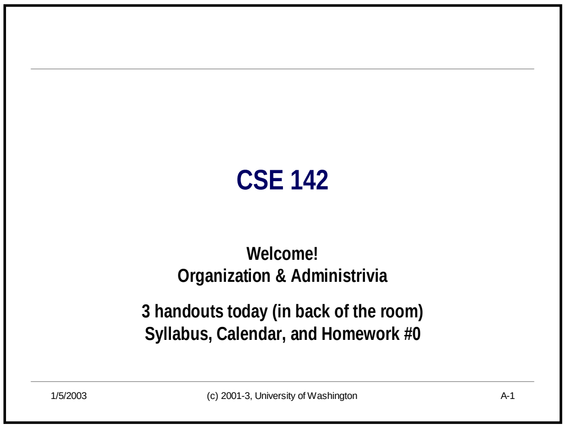
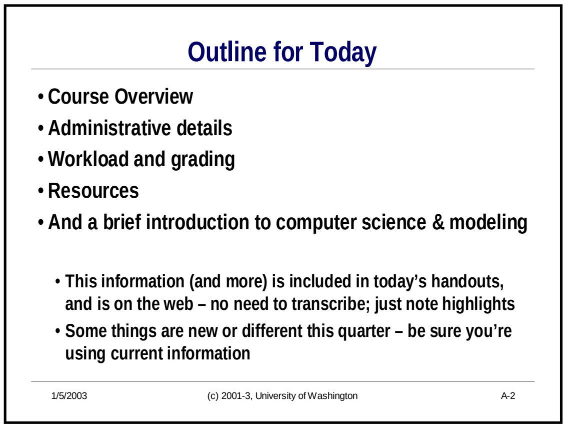


CSE 142

Welcome!
Organization & Administrivia

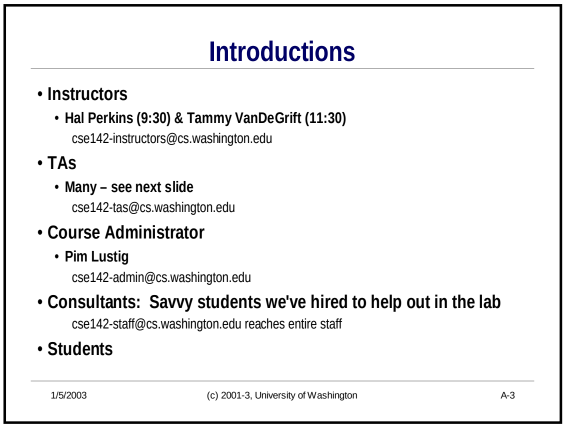
3 handouts today (in back of the room)
Syllabus, Calendar, and Homework #0



Outline for Today

- Course Overview
- Administrative details
- Workload and grading
- Resources
- And a brief introduction to computer science & modeling

  - This information (and more) is included in today's handouts, and is on the web – no need to transcribe; just note highlights
  - Some things are new or different this quarter – be sure you're using current information



Introductions

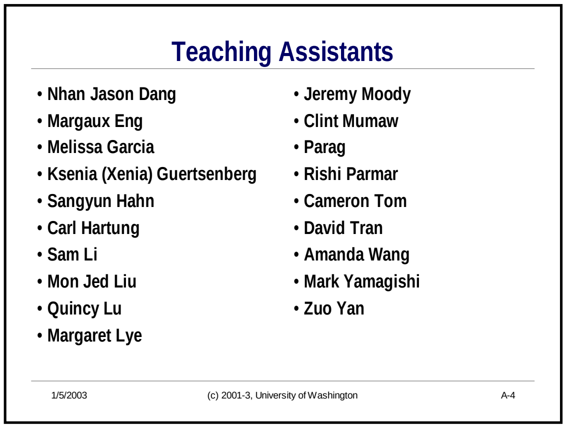
- Instructors
  - Hal Perkins (9:30) & Tammy VanDeGrift (11:30)

    cse142-instructors@cs.washington.edu
- TAs
  - Many – see next slide

    cse142-tas@cs.washington.edu
- Course Administrator
  - Pim Lustig

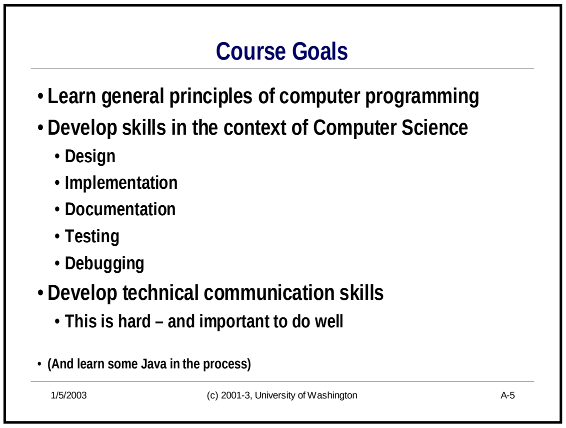
    cse142-admin@cs.washington.edu
- Consultants: Savvy students we've hired to help out in the lab

  cse142-staff@cs.washington.edu reaches entire staff
- Students



Teaching Assistants

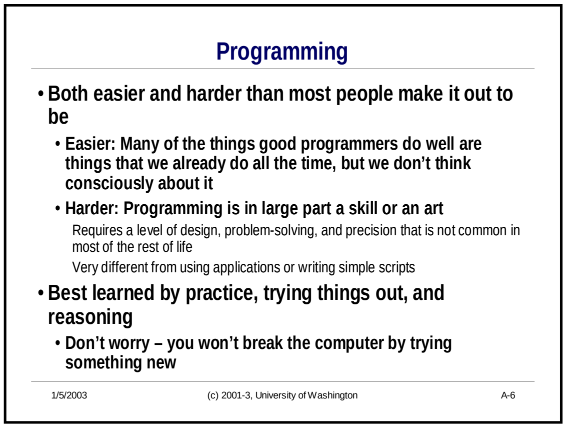
- Nhan Jason Dang
- Margaux Eng
- Melissa Garcia
- Ksenia (Xenia) Guertsenberg
- Sangyun Hahn
- Carl Hartung
- Sam Li
- Mon Jed Liu
- Quincy Lu
- Margaret Lye
- Jeremy Moody
- Clint Mumaw
- Parag
- Rishi Parmar
- Cameron Tom
- David Tran
- Amanda Wang
- Mark Yamagishi
- Zuo Yan



Course Goals

- Learn general principles of computer programming
- Develop skills in the context of Computer Science
  - Design
  - Implementation
  - Documentation
  - Testing
  - Debugging
- Develop technical communication skills
  - This is hard – and important to do well

- (And learn some Java in the process)

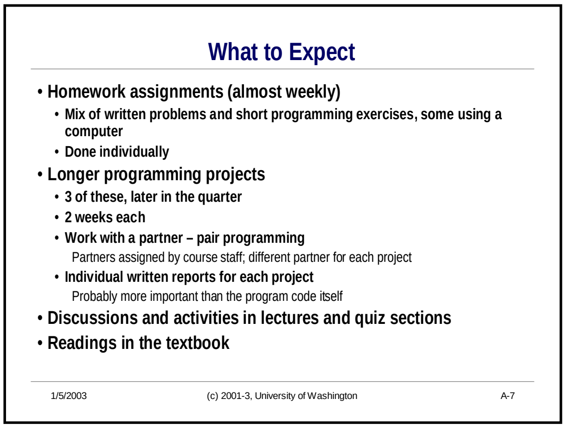


Programming

- Both easier and harder than most people make it out to be
  - Easier: Many of the things good programmers do well are things that we already do all the time, but we don't think consciously about it
  - Harder: Programming is in large part a skill or an art

    Requires a level of design, problem-solving, and precision that is not common in most of the rest of life

    Very different from using applications or writing simple scripts
- Best learned by practice, trying things out, and reasoning
  - Don't worry – you won't break the computer by trying something new



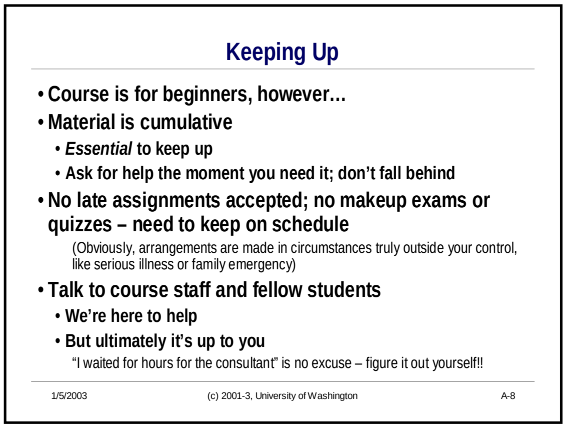
What to Expect

- Homework assignments (almost weekly)
  - Mix of written problems and short programming exercises, some using a computer
  - Done individually
- Longer programming projects
  - 3 of these, later in the quarter
  - 2 weeks each
  - Work with a partner – pair programming

    Partners assigned by course staff; different partner for each project
  - Individual written reports for each project

    Probably more important than the program code itself
- Discussions and activities in lectures and quiz sections
- Readings in the textbook

1/5/2003 (c) 2001-3, University of Washington A-7



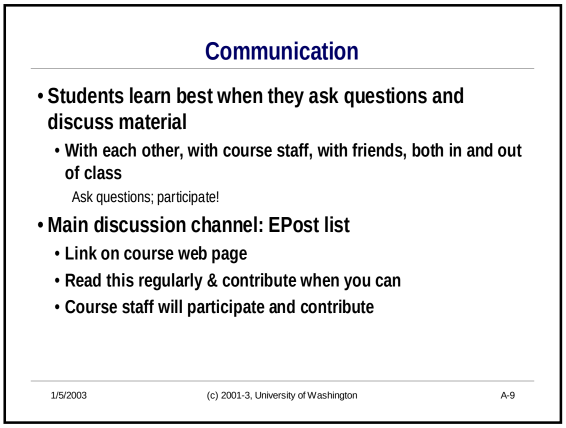
Keeping Up

- Course is for beginners, however…
- Material is cumulative
  - *Essential* to keep up
  - Ask for help the moment you need it; don't fall behind
- No late assignments accepted; no makeup exams or quizzes – need to keep on schedule

  (Obviously, arrangements are made in circumstances truly outside your control, like serious illness or family emergency)
- Talk to course staff and fellow students
  - We're here to help
  - But ultimately it's up to you

    "I waited for hours for the consultant" is no excuse – figure it out yourself!!

1/5/2003 (c) 2001-3, University of Washington A-8

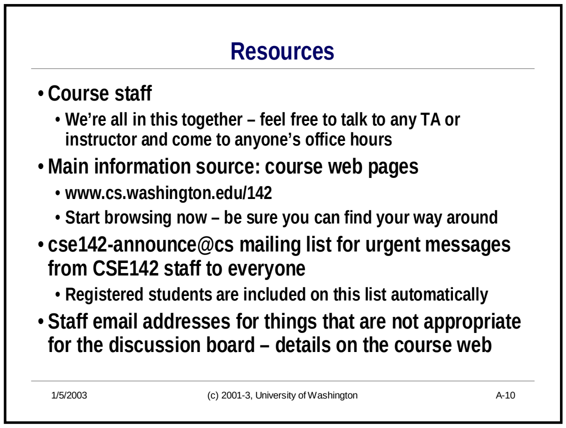


Communication

- Students learn best when they ask questions and discuss material
  - With each other, with course staff, with friends, both in and out of class

    Ask questions; participate!
- Main discussion channel: EPost list
  - Link on course web page
  - Read this regularly & contribute when you can
  - Course staff will participate and contribute

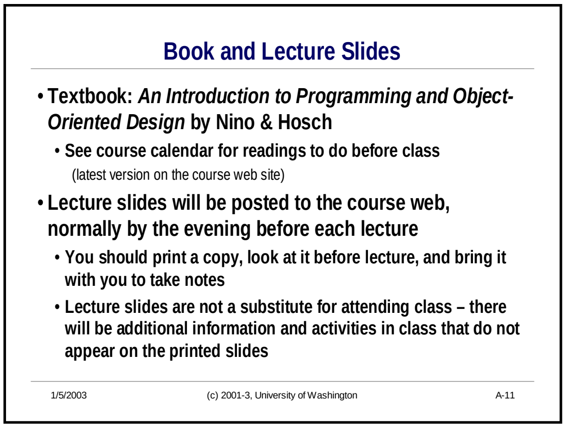
1/5/2003 (c) 2001-3, University of Washington A-9



Resources

- Course staff
  - We're all in this together – feel free to talk to any TA or instructor and come to anyone's office hours
- Main information source: course web pages
  - www.cs.washington.edu/142
  - Start browsing now – be sure you can find your way around
- cse142-announce@cs mailing list for urgent messages from CSE142 staff to everyone
  - Registered students are included on this list automatically
- Staff email addresses for things that are not appropriate for the discussion board – details on the course web

1/5/2003 (c) 2001-3, University of Washington A-10

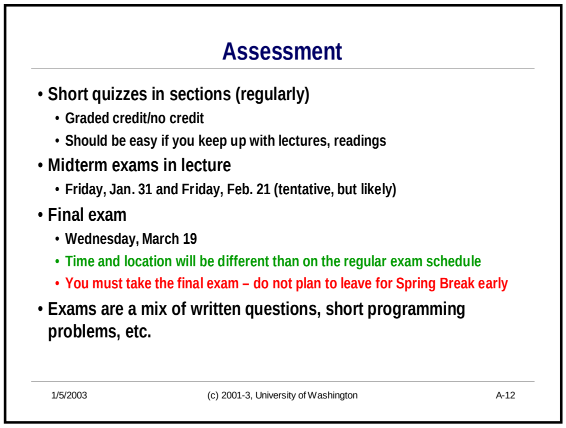


Book and Lecture Slides

- Textbook: *An Introduction to Programming and Object-Oriented Design* by Nino & Hosch
  - See course calendar for readings to do before class

    (latest version on the course web site)
- Lecture slides will be posted to the course web, normally by the evening before each lecture
  - You should print a copy, look at it before lecture, and bring it with you to take notes
  - Lecture slides are not a substitute for attending class – there will be additional information and activities in class that do not appear on the printed slides

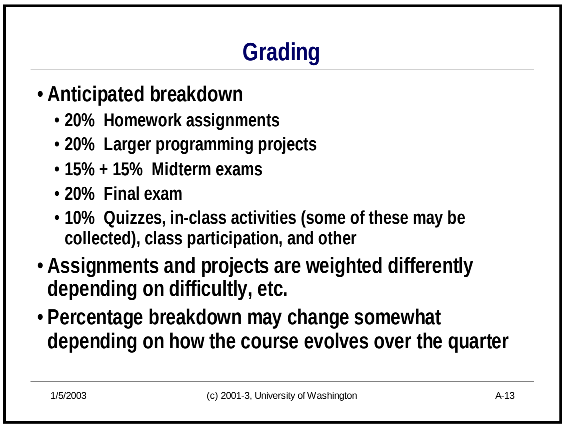
1/5/2003 (c) 2001-3, University of Washington A-11



Assessment

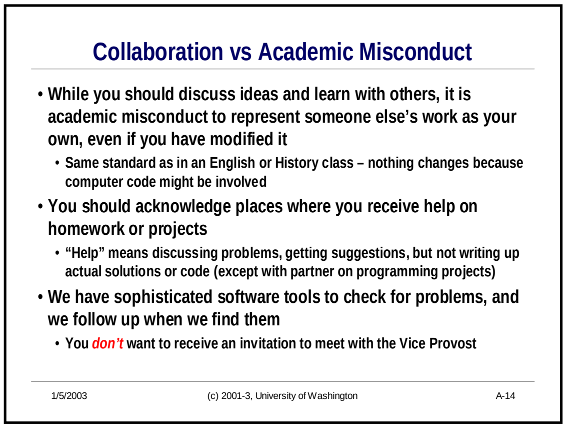
- Short quizzes in sections (regularly)
  - Graded credit/no credit
  - Should be easy if you keep up with lectures, readings
- Midterm exams in lecture
  - Friday, Jan. 31 and Friday, Feb. 21 (tentative, but likely)
- Final exam
  - Wednesday, March 19
  - Time and location will be different than on the regular exam schedule
  - You must take the final exam – do not plan to leave for Spring Break early
- Exams are a mix of written questions, short programming problems, etc.

1/5/2003 (c) 2001-3, University of Washington A-12



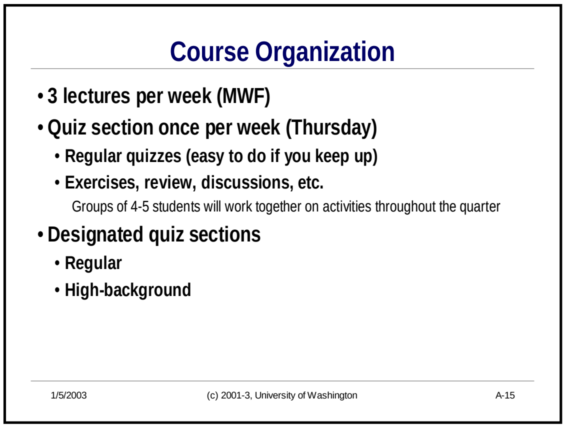
Grading

- Anticipated breakdown
  - 20% Homework assignments
  - 20% Larger programming projects
  - 15% + 15% Midterm exams
  - 20% Final exam
  - 10% Quizzes, in-class activities (some of these may be collected), class participation, and other
- Assignments and projects are weighted differently depending on difficultly, etc.
- Percentage breakdown may change somewhat depending on how the course evolves over the quarter



Collaboration vs Academic Misconduct

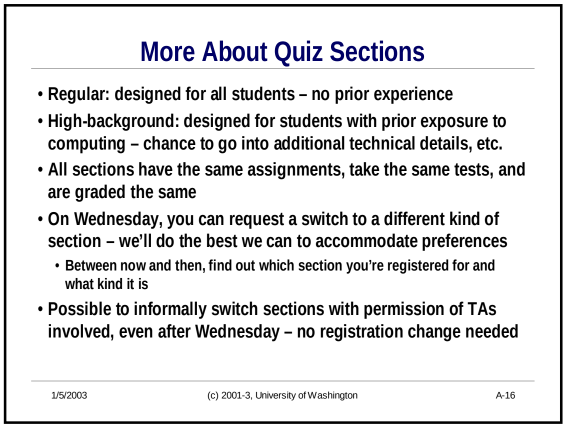
- While you should discuss ideas and learn with others, it is academic misconduct to represent someone else's work as your own, even if you have modified it
  - Same standard as in an English or History class – nothing changes because computer code might be involved
- You should acknowledge places where you receive help on homework or projects
  - "Help" means discussing problems, getting suggestions, but not writing up actual solutions or code (except with partner on programming projects)
- We have sophisticated software tools to check for problems, and we follow up when we find them
  - You ***don't*** want to receive an invitation to meet with the Vice Provost



Course Organization

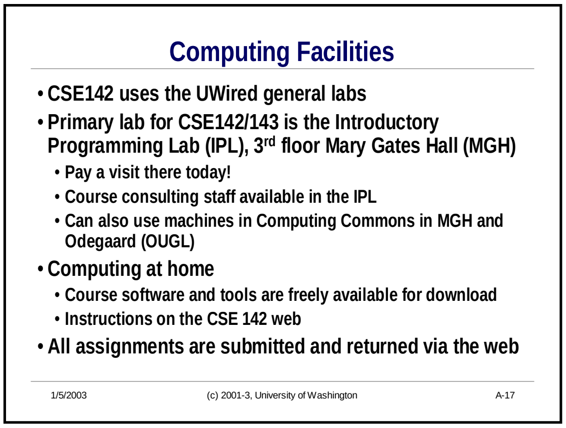
- 3 lectures per week (MWF)
- Quiz section once per week (Thursday)
  - Regular quizzes (easy to do if you keep up)
  - Exercises, review, discussions, etc.

    Groups of 4-5 students will work together on activities throughout the quarter
- Designated quiz sections
  - Regular
  - High-background



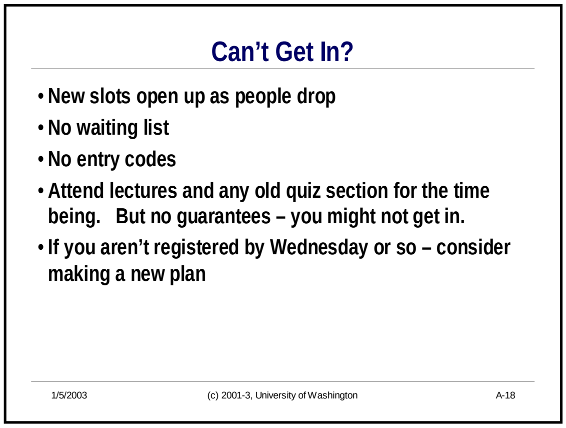
More About Quiz Sections

- Regular: designed for all students – no prior experience
- High-background: designed for students with prior exposure to computing – chance to go into additional technical details, etc.
- All sections have the same assignments, take the same tests, and are graded the same
- On Wednesday, you can request a switch to a different kind of section – we'll do the best we can to accommodate preferences
  - Between now and then, find out which section you're registered for and what kind it is
- Possible to informally switch sections with permission of TAs involved, even after Wednesday – no registration change needed



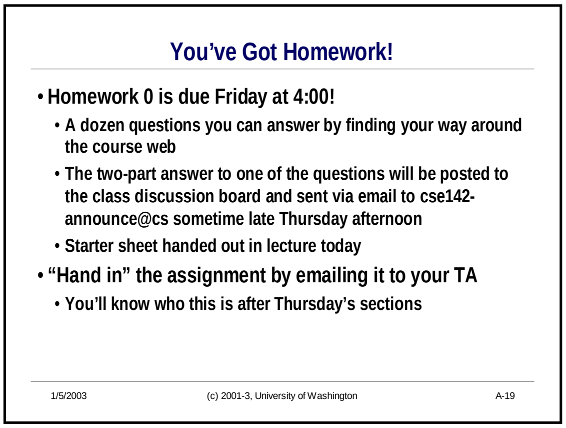
Computing Facilities

- CSE142 uses the UWired general labs
- Primary lab for CSE142/143 is the Introductory Programming Lab (IPL), 3rd floor Mary Gates Hall (MGH)
  - Pay a visit there today!
  - Course consulting staff available in the IPL
  - Can also use machines in Computing Commons in MGH and Odegaard (OUGL)
- Computing at home
  - Course software and tools are freely available for download
  - Instructions on the CSE 142 web
- All assignments are submitted and returned via the web



Can't Get In?

- New slots open up as people drop
- No waiting list
- No entry codes
- Attend lectures and any old quiz section for the time being.   But no guarantees – you might not get in.
- If you aren't registered by Wednesday or so – consider making a new plan



You’ve Got Homework!

- Homework 0 is due Friday at 4:00!
  - A dozen questions you can answer by finding your way around the course web
  - The two-part answer to one of the questions will be posted to the class discussion board and sent via email to cse142-announce@cs sometime late Thursday afternoon
  - Starter sheet handed out in lecture today
- “Hand in” the assignment by emailing it to your TA
  - You’ll know who this is after Thursday’s sections