CMPS 12B
Introduction to Data Structures

Professor Ethan Miller
elm@cs.ucsc.edu
http://www.soe.ucsc.edu/~elm/

Welcome!

- Lectures: 2–3:10 PM, Classroom 001 (you already knew that!)
- Sections
  - Mon 3:30PM–5:30PM (Baskin 105)
  - Tue 3:30PM–5:30PM (Baskin 105)
  - Wed 10:00AM–noon (Baskin 105)
  - Thu 6:00PM–8:00PM (Crown Computer Lab 201)
    - Note time change!
  - Fri 11:30AM–1:30PM cancelled
    - Please talk to me if this was the only section you could attend
- Office hours
  - Professor: 1:30-3:00 Tuesday, 3:30–5:00 Wednesday
  - Other hours in a bit…
Course staff

- TAs
  - Andy Pohl (aamp@soe.ucsc.edu)
  - Yue Zhou (zhou@soe.ucsc.edu)
- Tutors / readers
  - Darren Fitzgerald (darrenf@cats)
  - Mark Pauley (markap@cats)
  - Casey Robinson (casey@soe)
  - Donna Wong (dwsmurf@cats)
- Use cmps12b-w02-staff@cats to send email to all course staff
  - Please email questions and concerns to this address
  - Faster response time: anyone can answer the question

Computers & CMPS 12B

- Nearly everything is online (on the Web)
  - Home page: http://www.soe.ucsc.edu/~elm/Classes/12b/w02/
  - Syllabus, schedule, assignments, slides, etc.
- Assignments handed in online
- Grades returned online
- Most communication via email
  - Please use your CATS account rather than a Yahoo mail or Hotmail account
- Bottom line: you need a CATS account now
  - First assignment due in two weeks, so don’t delay
  - You need your own account; you may not borrow someone else’s for any reason (violation of CATS policy)
Course requirements

- Two exams
  - Midterm (planned for Monday, February 11th in class)
  - Final exam (official time: Saturday, March 16th 4–7PM)
- Programming assignments
  - 5 projects during the quarter
  - About 2 weeks per project
  - Hand in online
- Quizzes / class participation
  - 6 pop quizzes during the quarter (unannounced)
  - Quizzes cover material from the last week or two of class
  - Simple: 15 minutes max for 2–4 short questions
  - Graded on a 0–10 scale
    - Minimum of 1 point if you actually take the quiz
  - Lowest two scores are dropped
  - No makeup quizzes!

Grading

- Final grades based on
  - Programming assignments: 45%
    - All assignments weighted equally
  - Exams: 45%
    - Midterm 15%
    - Final 30%
  - Quizzes & class participation: 10%
- To pass the class you must
  - Take both exams
  - Hand in a reasonable attempt at each programming assignment
  - Pass at least two of the three areas (programming, exams, quizzes)
    - Passing two of three areas doesn’t guarantee that you’ll pass the class!
- Curves
  - Programming assignments & quizzes will not be curved
  - Exams may be curved (but not a guarantee)
Programming assignments

- Five programming assignments
  - About two weeks each
  - Late assignments receive a 0
- Assignments will use material learned in class
  - Concepts covered in class
  - Sections will cover assignments in more detail
- Topics for assignments may include
  - Linked lists
  - Stacks & queues
  - Hash tables
  - Trees
  - Sorting

Programming assignment logistics

- For each assignment, turn in
  - Design document (more on this in a few days)
  - Code files (.java or .c)
  - Other files as required by the assignment
- Files submitted online
  - Use `submit` program to turn your files in
    - May resubmit files as often as you like
    - Each `submit` overwrites the previous one
  - Use `peek` program to see what files you’ve turned in
  - Details on the Web
- Programming may be done anywhere you like
  - Code must work on CATS systems using standard tools
  - Try out your code before handing it in
Is CMPS 12B a difficult class?

- Yes, CMPS 12B is a challenging class
  - Last quarter, more than 25% of the class didn’t receive a passing grade (failed or withdrew)
  - Programming can be difficult for many people
- Is it worth taking CMPS 12B?
  - Yes! You need it if you want to be a CS, CE, or ISM major (and perhaps others)
- Why do we have to take it?
  - It’s crucial to understand the basic data structures used in computer programs
  - Big programs are composed of little pieces, and little pieces are composed mostly of structures that you’ll learn about in 12B
  - Programming style, design & debugging techniques, and documentation are also very important!

Getting help

- Don’t be afraid to ask for help!
  - The only dumb question is one you didn’t ask
- Attend lecture & section
  - Higher priority for help goes to those who don’t ask us to give the lecture during office hours
  - One section is sufficient (don’t go to all four!)
- ACE section available (7:30–9PM in Baskin 105)
- Office hours
  - Over 20 hours per week!
  - Many held in a computer lab (Crown or Baskin 105)
- Newsgroup: ucs.class.cmps012b
- Email: cmps12b-w02-staff@cats.ucsc.edu
  - Please send email to this address for help
  - Concerns about course staff can go to elm@cs.ucsc.edu
- Anonymous feedback available from course Web page
Getting too much help (cheating)

- Cheating is
  - Unethical
  - Against university and department policy
  - A really bad thing to have on your record
- What constitutes cheating?
  - Representing someone else’s work as your own
    - Directly copying
    - Having someone else design or debug your program
  - Getting help from course staff is OK
- We aggressively pursue cheaters
- What happens if you cheat?
  - At a minimum, 0 on the assignment for all parties involved
  - Failing grade in the class
  - Report to your college and the university
- When in doubt, ask before doing it!

What you should already know

- Basic computer usage
  - Email
  - Web browsing
  - Reading news (not essential, but helpful)
- Basic Unix commands
  - Editing and compiling Java files
  - We’ll go over more Unix stuff in the first week
- Basic Java programming
  - Variables
  - Conditionals & loops
  - Arrays
  - Objects & classes (declaring and using them, methods)
  - Good programming & documentation style
- Bottom line: all the stuff you learned in 12A…
What you’ll learn in CMPS 12B

- Algorithms and data structures
  - Linked lists
  - Queues & stacks
  - Hash tables
  - Binary searches & binary trees
  - Sorting algorithms & data structures
  - “Big O” notation
- Unix
  - Makefiles
  - Debugging
  - AFS
- C programming
- Software design & documentation (the basics)