CSE 143
Computer Programming II

Welcome!
Course Overview and Administrivia

Pick up a handout as you come in

Outline for Today

- Course Overview
- Goals
- Administrative details
- Workload and grading
- Resources
- Background

This information is largely included in today's handout, and is on the web – no need to transcribe, but do take notes about important items!

Introductions

- Instructor: Hal Perkins
  cse143-instructor@cs.washington.edu, perkins@cs.washington.edu
  Allen Center CSE 548, office hours TBA
- TA's: Kurtis Heimerl, Eugene Hwang, Jimmy Malone, Lincoln Ritter, William Rossiter, Rebecca Stecker, David Tran, Pat Tressel, Kasia Wilamowska
  cse143-ta@cs.washington.edu (goes to all TAs and the instructor)
- IPL Consultants: Savvy former students who help out in the lab
  - Once we get their hours worked out, we'll post a schedule
- Course administrator: Pim Lustig
  cse143-admin@cs.washington.edu
- Everyone on the course staff:
  cse143-staff@cs.washington.edu (reaches entire staff)
- You!

Can't Get In?

- Still a few slots left! Tell your friends to sign up
- Historically, openings appear during the first week, but no guarantees
- No waiting list/entry codes
  (Based on past experience this is hard to organize fairly and isn't really needed)
- Non-matriculated students, grad students, registration problems – please see Pim Lustig (Sieg 114)

Are You Ready?

- Course is a direct continuation of CSE 142 Java
- Must have a firm grasp of Java basics
  - including statements, expressions, methods, parameters, arrays, basics of classes and objects, JavaDoc, etc.
    concepts and terminology as well as being able to use in programs
  - No systematic review
  - Look at old CSE 142 web pages – you should be able to handle those assignments and exams
- What if you took the C version of CSE 142? Or took 142 elsewhere?
  - Let's talk about that now
  - Not sure?
    Sit in on both for a few days
    Try the first 143 assignment
    We'll help you switch to 142 if that's your decision

Java!

A modern approach to programming including
- Objects everywhere; classes, interfaces, polymorphism
- Exceptions
- Streams and networking support
- Garbage collection
- Specifications, design by contract support
- Rich set of standard libraries
- Documentation tools and standards, on-line library documentation
- We'll use Sun's Java SDK 1.4.2 (windows, *nix, OS X)
  - 1.3 will *not* do; 1.4.1 is OK if that's the latest you can get (OS X Jaguar)
  - J++ (Microsoft) will *not* do (Java 1.1!)
  - Please update your software!
  - Details: *Computing at Home* page on course web

Content Overview (1)

Programming language and libraries
- Classes, class relationships (inheritance), interfaces, types
- Debugging and systematic testing (JUnit)
- Graphical user interfaces & event-driven programming (Swing, event handling, model-view-controller design)
- Exceptions
- Stream I/O and files
- Recursion

Content Overview (2)

Data structures and algorithms
- Basics of lists, stacks, queues, trees, dictionaries
- Implementation techniques: arrays, linked data structures, hash tables
- Comparing implementations: basic complexity theory
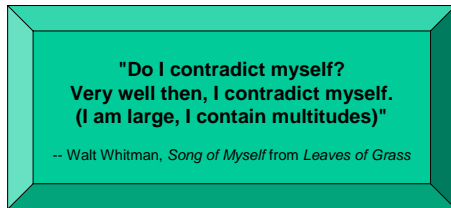- Divide and conquer algorithms: sorting and searching
- **And Much Much More!**

Course Objectives

- This *is* a programming course
- This is *not* a programming course

> **"Do I contradict myself?**
> **Very well then, I contradict myself.**
> **(I am large, I contain multitudes)"**
>
> -- Walt Whitman, *Song of Myself* from *Leaves of Grass*

Is it or Isn't it?

- This *is* a programming course
  - The key goal is learning to program *well*, not just getting stuff to run
    - Good design, good organization, good style
    - Good algorithms, meaningful efficiency
- This is *not* a programming course
  - Lots of Java features won't be covered
    - See Java reference books & JavaDoc for full descriptions of the Java language & libraries
    - We cover the essential parts of Java that support good programming
  - Many important computer science topics
    - Some related to programming, but broader than Java
    - Data structures, algorithms, complexity analysis, software engineering…
- Fact: writing programs that work perfectly isn't enough to get a perfect grade (!)

My Goals for You

- 5 things you should be able to do after CSE143
  - Be able to design and implement abstractions (classes) using modern programming language features and techniques
  - Be able to test and systematically locate and remove errors in programs
  - Be able to learn and use new libraries using standard documentation (no training wheels)
  - Be able to evaluate tradeoffs between different implementations of an abstraction and pick suitable ones
  - Be able to communicate technical concepts literately

My Expectations for You

- Responsibility
  - Keep up, know what's happening
  - Meet deadlines, budget your time, make backups
  - Take responsibility for your own code and debugging
- Respect
  - For others in the class (people sitting around you in lecture, members of your quiz section, partners on programming projects, ...)
  - For the course staff
  - For yourself

My Goals For Myself

- Be an advocate for your learning (credit to Prof. Mary Pat Wenderoth for this notion)
  - Help all of you learn
  - Keep the course on track
  - Make the homework projects interesting
  - Make lectures and sections events you look forward to!
- Keep in touch with what's happening
  - Office hours – please drop by if just to chat
    (you're not being sent to the Principle's office!)
  - "Muddiest" concept of the day, informal evaluations, etc.

Course Organization

- 3 lectures per week (MWF)
- Quiz section twice per week (T & Th)
  - Exercises, review, discussions, etc.
- Frequent quizzes
  - To keep you up with reading, lectures, and assignment instructions
    (Quizzes should be easy if you've reviewed previous material covered in lectures and sections)
  - To test mastery of current material
  - To provide TAs and me with feedback

Assignments

- Typically (but not always!) due Wed. night 9pm (electronic) and/or in sections or lecture Thursday or Friday (written)
  - Written assignments often collected all day in CSE office
- Primarily fairly substantial programming projects with written reports
- Maybe some shorter problems and programming drills
- Assignments will be more complex than in CSE142
      Assignment directions, too!
- No late assignments accepted
      But be sure to talk with your TA about problems truly beyond your control like illness or family emergency so we know what happened

Academic (Mis)conduct

- Goal: balance the following
  - *Learning:* each student must do the work to learn effectively
  - *Cooperation:* people learn best when they can cooperate with others
  - *Fairness* and *honesty:* Nobody should ever represent the work of someone else as their own or try to claim credit for it

Academic (Mis)conduct

- Policy
  - You must do assignments by yourself or with your assigned partner (unless explicitly stated otherwise in an assignment)
  - You may discuss general approaches and ideas with others, but
  - You *may not ever* give code to or receive code from others
- We check this and act when trouble is discovered
- Use your common sense and ask first if unclear
  - Rule of thumb: *any activity you engage in for the purpose of earning credit while avoiding learning, or to help others do so, is likely to be an act of academic misconduct* (from CSE dept. policy – see link on the web)

Exams & Quizzes

- Exams
  - 2 midterm exams in class; probable dates: Monday Jan. 31 and Wednesday Feb. 23
  - Final exam: Wednesday, Mar. 16, 2:30 pm (time set by the university, probably here but might be in a different room)
- The exams will not be given on any other days. Don't make plans which would take you away!
- Format: mixture of short answer, short essay, multiple choice, programming (both short and longer problems)

Grading

- Grade distribution (subject to change)
  - 30% homework assignments and projects
    - Weights vary depending on difficulty of each assignment
  - 16% + 16% midterm exams
  - 25% final exam
  - 8% quizzes
  - 5% participation, service, citizenship
- Class is "curved"
  - Median of final course grades is around 3.0
    - Maybe a bit higher when there are a lot of drops
    - Definitely higher if everyone does a great job (but statistically unlikely)
  - Why?

Grading

- Project and quiz grading will be very coarse
- No partial points
- Typical scale: 4, 3, 2, 1, 0 for projects and written reports
  - Mastery || Good Job! || On the Right Track || Honest Effort, but... || Really, Now!
  - Intermediate turnins typically 3, 2, 1, 0 (all is well, some problems, serious problems, not credible)
  - Separate scores for program operation/code quality
    - i.e., Yes! Clarity, readability, style matters
  - Written reports count as much as the actual code (being able to communicate what you do is a crucial skill)
- Other assignments, typically 2, 1, 0 per question or question part
- Quiz question grading: check (1), check minus (1, but you should have been more on top of things), 0

Resources to Help You Succeed

- Course staff
  - Your TA is your primary contact, but please feel free to talk to any of us
    - *Especially*: don't leave me lonely in office hours once they're set!
  - Consultants in the IPL
    - A limited resource!

More Resources

- Help each other! Form study groups, spend time on the discussion list, etc.
- Undergraduate advisors, for general questions about the CSE programs (Sieg 114)
- College of Engineering has some special resources for women and minorities
- Other university resources

For Reading and Study

- Lecture slides and course notes
  - Alert! Not all lecture material is on the slides!
  - Slides used will be posted on the web generally by previous evening – print out and bring to class to take notes
    - NOT distributed in lecture
- Textbook: Next slide
- Other Material
  - Possibly handouts
  - All e-mail announcements, assignment descriptions, etc. should be considered required reading. They could even be tested on!

Textbooks

- Textbook (recommended): Niño & Hosch, *An Introduction to Programming and Object-Oriented Design using Java,* 2nd edition, 2004.
  - First edition is fine if you've got it.
  - Alert! We may not follow the book very closely!
  - Will not always match our way of doing things, or our order!
    - But does provide a complementary view

Communicating Electronically

Course web site
- www.cs.washington.edu/143/
- Discussion Board: linked from Web site
  - UWNetID required
  - Open discussion – please contribute!
  - Course staff monitors and contributes as needed
- Email to us for things not appropriate for public discussion
  - Addresses on the web
  - Email works better for some things than other (e.g., very bad for trying to debug code)
- E-mail from us: cse143-announce
  - Sent directly to your UWNetID account
  - We'll try to keep the spam to a minimum, but... you must read and heed what we do send!

Computing Facilities

- Introductory Programming Lab (IPL)
  - Mary Gates Hall 334
  - CSE 143 consulting staff in IPL
    Hours posted on the web
    Goal is to provide quick help when you're stuck and have already tried to diagnose and fix the problem
- Computing at home
  - Java software and tools are freely available for download
    Java version MUST be 1.4+. Install entire SDK (Windows, Linux), or run software update (Mac OS X). Java 1.5 is still a prerelease (Windows, Linux), but is probably OK.
    You're free to use any Java development environment
       Recommended: DrJava (powerful but pretty simple), Eclipse (industrial strength)
    See Computing At Home page for links and details
- Even if you plan to compute at home, learn your way around the UW labs

Your First CSE143 Assignment

- Required reading:
  - syllabus, academic conduct policy page.
  - Do this before quiz section tomorrow!
  - Get a copy of the textbook unless you're fairly certain you won't need it
  - Review your CSE142 textbook
- Review rest of web (still somewhat incomplete)
  - Find the first day's slides
  - Visit the discussion board and find the announcements archive (two separate things!)
- Install the needed software on your home computer
  - and/or visit a campus lab and locate the software
- (After tomorrow) *memorize* your quiz section # and TA's name