Topic 1
Course Introduction, Syllabus, and Software Tools

Chapman: I didn't expect a kind of Spanish Inquisition.

Cardinal Ximinez: NOBODY expects the Spanish Inquisition! Our chief weapon is surprise...surprise and fear...fear and surprise.... Our two weapons are fear and surprise...and ruthless efficiency.... Our **three** weapons are fear, surprise, and ruthless efficiency...and an almost fanatical devotion to the Pope.... Our **four**...no... **Amongst** our weapons.... Amongst our weaponry...are such diverse elements as fear, surprise....

Mike Scott, Painter Hall 5.68,
scottm@cs.utexas.edu
www.cs.utexas.edu/~scottm/cs307

Who Am I?

- Lecturer in CS department since 2000
- Undergrad Stanford, MSCS RPI
- US Navy for 8 years, submarines
- 2 years Round Rock High School
- Wife (Kelly) is a nurse.
  - 2 daughters, Olivia and Isabelle

Rensselaer

What We Will Do Today

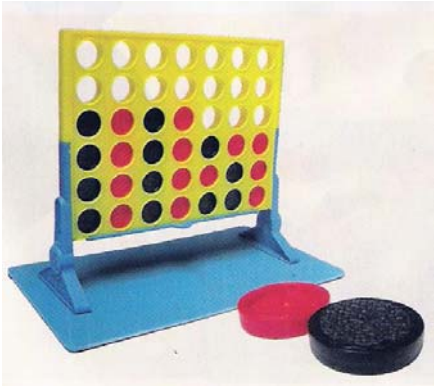
- Discuss
  - course content
  - procedures
  - tools

Formal Prerequisites

- One year of programming in high school, a grade of at least C in CS303E or CS 305J or consent of instructor (very rarely given).
- Credit or registration for Mathematics 408C, 408K, or 408N, or a score of at least 520 on the SAT Mathematics Level 1 or Level 2 test

Are you in the right place? Required Programming Knowledge and Experience for 307 – (Informal Prerequisites)

- variables and data types
- expressions, order of operations
- decision making (if statements)
  - including boolean logic and boolean expressions
- loops (fixed and variable repetition)
- procedures or functions
- parameters (reference and value parameters, local variables, scope, problem generalization)
- structures or records or objects
- arrays (vectors, lists)
- top down design (breaking big rocks into little rocks)
  - algorithm and data design
  - create and implement program of at least 200 - 300 loc
  - could you write a program to let 2 people play connect 4?



What We Will Do in 307

- A second course in programming with a focus on canonical data structures, algorithms on those data structures, and object oriented programming
- Java Basics and Review (1 week)
- Object Oriented Basics (2 weeks)
  - classes and objects, encapsulation, inheritance, polymorphism
- Fundamental of programming (2 weeks)
  - algorithm analysis, recursion, sorting and searching
- Introduction, application, and implementation of basic abstract data types (4 weeks)
  - lists, iterators, stacks, queues, trees, sets (hash tables, maps/dictionaries, heaps)

Course Materials and Procedures

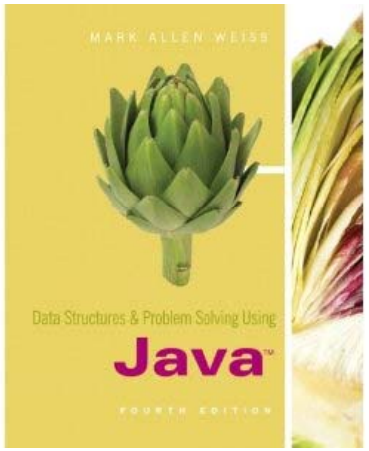
- If you are new to university level classes, you may be surprised by how much of the responsibility for knowing what to do in a class is up to you.
- You are responsible for a great number of things!

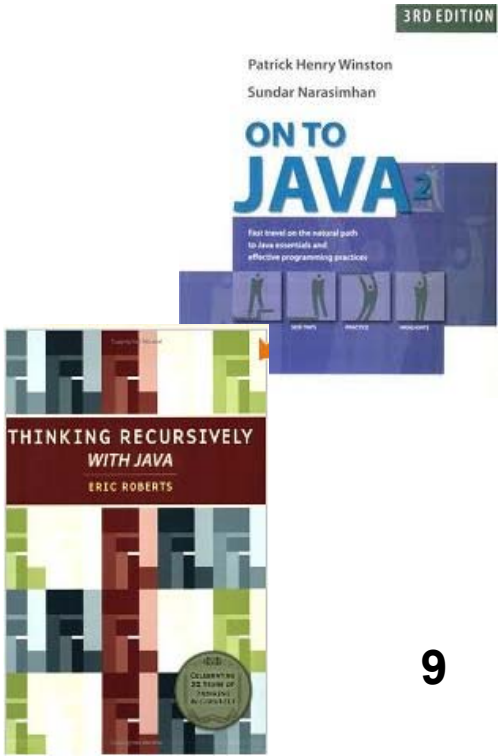
Course Materials and Procedures

- web site
  - www.cs.utexas.edu/~scottm/cs307/
    most materials you need are on the web site
  - links, assignments, schedule, coding samples, study materials, section problems
- schedule
  - on the web site
  - schedule of topics
  - required readings, many from the web
  - links to the slides I use in class
    - Slides are a reference only.
    - We will diverge from the slides on many occasions.
  - due dates

Course Materials and Procedures

- syllabus
  - very important
  - like a contract between instructor and students
  - policies for the course
  - online with links to more information
- books
  - books are recommended not required
  - Weiss book -> data structures
  - On to Java-> Java reference
  - Thinking Recursively in Java -> recursion





Course Materials and Procedures

- Lecture
  - lecture / discussion with instructor
  - not just lecture, I ask questions of you and I encourage you to ask questions of me
  - iClicker questions
- Discussion Section
  - with graduate teaching assistant
  - coding quiz at the start of each, similar in nature to test questions
    - quizzes cannot be made up
  - your chance to ask questions on the assignments
  - cover materials from section handouts which are available on the class web site

Attendance Question 1

Which of these best describes you?

A. First year at UT

B. First year at UT, transferring from another college or university.

C. In second year at UT.

D. In third year at UT

E. More than 3 years at UT

Attendance Question 2

Which computer programming language are you most comfortable with?

A. Java

B. C or C++

C. Python

D. PHP

E. Other

Course Materials and Procedures

- Class Discussion Group
  - www.piazzza.com
  - CS 307 under University of Texas at Austin
  - post questions about class, assignments, material, concepts
  - set up email alerts!
  - answer your classmates questions
  - updates and information from me will come via the discussion group and the UT group email system
  - no large chunks (> 3 lines) of solution code
  - additional test cases are okay

Graded Course Components

- Attendance, 25 lectures, 3 points each: 75 points total
- Discussion section quizzes, 9 quizzes, 10 points each: 90 points total
- Programming projects, 9 projects, 20 or 30 points each: 210 points total
- Midterm: 200
- Final: 460 points

- 75 + 90 + 210 + 200 + 460 = 1035
- Attendance, Quizzes, Programming Assignments capped at 340 points.
- 35 points of “slack” among those 4 components
- No points added! I divide by 1000 to get final average instead of 1035.

Grades and Performance

- Final grade determined by final point total and a 900 – 800 – 700 – 600 scale
  - Will be adjusted with plusses and minuses if within 25 points of cutoff: 875 – 899: B+, 900 – 924: A-

- Last semester **192** students enrolled in the course.
  - **138** students got a C or better. (53 As, 53 Bs, 32 Cs)
  - **38** students got a D or F.
  - **16** students dropped or withdrew.

- The majority of students getting Ds or Fs missed 1 or more exams without an excuse and / or had a failing average on non exam components. (programming assignments, attendance, and quizzes)

Course Materials and Procedures

- Assignments
  - where ~80% of your learning will take place
  - constant feedback -> good news / bad news
  - for learning, not evaluation -> low point value
  - posted to class web site
  - see assignment page for general guidelines
  - creating programs using Java
  - usually creating *parts* of programs based on provided code
  - sometimes a complete program
  - some assignments done as individual, some can be done with a partner

Course Materials and Procedures

- More on assignments
  - some test cases provided
  - some provided test cases may have errors
  - use class listserv to discuss and resolve errors in provided test cases
  - create your own test cases
  - graded on correctness, style, efficiency, generality, comments, testing
    - not graded on a linear scale or on effort
  - program must work, compile errors / runtime errors lose all correctness points

Course Materials and Procedures

- Still more on assignments
  - ***VERY IMPORTANT***: must get account for CS department labs -> see syllabus for procedure
  - turn in assignments to your lab account via the turnin program – DEMO
  - ***turn in the right thing, in the right format, with the right name! Failure to do so will lead to a loss of slip days and/or points.***
  - slip days, 6 total for the semester
  - no provisions other than slip days and “slack" in grading scheme for late / missed assignments
  - slip days and “slack” are for emergencies!

Course Materials and Procedures

- And yet more on assignments
  - graded by teaching assistant and proctor
  - scores posted to grade center on Blackboard-> link on class web site
  - individual assignments are just that, individual
  - copying solution code or giving code to someone else is **CHEATING** -> F in the course
  - solutions checked with plagiarism detection software
  - sharing test cases okay and encouraged
  - read the portion of the syllabus regarding cheating and collaboration

Course Materials - Exams

- Midterm in class on Friday, July 15

- Final Exam: Monday, August 15, 7 – 10pm

More on Exams

- old tests on line – study materials
- tests consist of short answer questions and coding questions
- test emphasize problem solving, algorithm implementation, some syntax
- tests scores curved up if instructor feels necessary.

Succeeding in the Course

- Randy Pausch,
  CS Professor at CMU said:

- *"When I got tenure a year early at Virginia, other Assistant Professors would come up to me and say, 'You got tenure early!?!?! What's your secret?!?!?' and I would tell them, 'Call me in my office at 10pm on Friday night and I'll tell you.' "*

- Meaning: Some things don't have an easy solution.
- Some things simply require a lot of hard work.

Succeeding in the Course

- do the readings
- start on assignments early
- get help from the teaching staff when you get stuck on an assignment
- attend lecture and discussion sections
- participate on the listserv
- do Javabat problems (http://codingbat.com/java)
- do the extra section problems
- study for tests using the old tests
- study for tests in groups
- ask questions and get help when needed

Course Materials and Procedures

- Software
  - can work in CS department microlab, 5th floor of Painter Hall
  - login via CS account name and password
  - can work at home if you wish
  - Java.
    - Free.
    - Web page has details under Software. - JDK 6.0
  - Optional IDE.
    - Recommended IDE is Eclipse, also free