

Topic 1

Course Introduction, Syllabus, and Software Tools

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

Cardinal Ximenez: 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

How Does This Course Fit the Discipline of Computer Science

- ▶ Most introductory CS classes, both at the high school and college level, focus on teaching students how to program a computer.
- ▶ ... and yet, computer science and computer programming are not the same thing!
- ▶ So what is Computer Science?

What is Computer Science?

- ▶ Poorly named in the first place.
- ▶ It is not so much about the computer as it is about *Computation*.
- ▶ *“Computer Science is more the study of managing and processing information than it is the study of computers.”*
 - Owen Astrachan, Duke University
- ▶ learn to program
 - programming a key tool in later courses



Learning Programming

- ▶ Programming = Algorithms + Data Structures
 - Niklaus Wirth



- ▶ "Learning computer programming essentially involves two things: computer algorithms and how data types are defined, designed, and used to solve problems."
 - Chenglie Hu

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 M408C or M408K, or a score of at least 520 on the SAT II Math Level 1 or Math 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

- ▶ Java Basics and Review (2 weeks)
 - How to do the stuff on previous slide in Java
- ▶ Object Oriented Basics (3 weeks)
 - object oriented design, classes and objects, encapsulation, inheritance, polymorphism
- ▶ Fundamental of programming (3 weeks)
 - algorithm analysis, recursion, classic algorithms including sorting and searching
- ▶ Introduction, application, and implementation of basic abstract data types (7 weeks)
 - lists, stacks, queues, trees, sets (hash tables, maps/dictionaries, heaps)

Course Materials and Procedures

- ▶ This information is important!
- ▶ 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.
- ▶ If you have a question about something in the course first, try to find the answer yourself, then ask someone such as a classmate, me, the teaching assistant, or the proctor
 - Making mistakes is okay!
 - That is how you learn.

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
- includes readings, many from the web
- includes links to the slides I use in class
 - Slides are a reference only.
 - We will diverge from the slides on many occasions.
- includes due dates

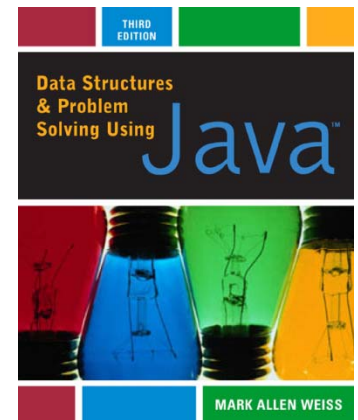
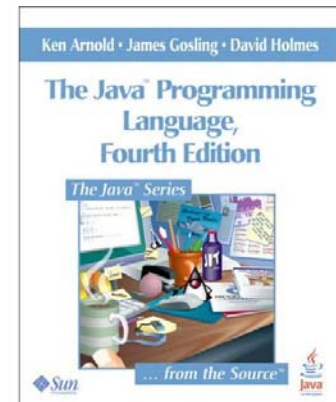
Course Materials and Procedures

▶ syllabus

- very important
- like a contract between instructor and students
- policies for the course
- online with links to more information

▶ books

- both are recommended not required
- Weiss book -> data structures
- Java Programming Language, 4th Edition -> Java reference



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
- attendance via iClicker questions

▶ Discussion Section

- with graduate teaching assistant
- coding quiz at the start of each, similar in nature to some 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 semester at college, recent high school grad.
- B. First semester at UT, transferring from another school.
- C. Have been at UT for a year. (Starting second year)
- D. Have been at UT for 2 or more years

Attendance Question 2

Which computer programming language are you most comfortable with?

- A. Java or C#
- B. C or C++
- C. Python
- D. Visual Basic
- E. Other

Course Materials and Procedures

- ▶ class listserv
 - sign up for the listserv, procedure in syllabus and on assignment 1
 - learn to set up a filter in your email client
 - post questions about class, assignments, material, concepts
 - answer your classmates questions
 - updates and information from me will come via the listserv
 - no large chunks (> 3 lines) of solution code on the listserv
 - additional test cases are okay

Graded Course Components

- ▶ Attendance, 40 lectures, 1 point each, 40 points total
 - ▶ Discussion section quizzes, 13 quizzes, 5 points each, 65 points total
 - ▶ Javabat problems, 9 problem sets, 5 points each, 45 points total
 - ▶ Programming projects, 12 projects, 10 or 20, 220 points total
 - ▶ Midterm 1: 175 points
 - ▶ Midterm 2: 200 points
 - ▶ Final: 300 points
-
- ▶ Attendance, Quizzes, Javabat, Programming capped at 325 points.
 - ▶ 45 points of “slack” among those 4 components

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
 - you should use class listserv to discuss and resolve errors in provided test cases
 - you must 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! (source code now, jar files later, correct name)**
 - 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 egradebook -> link on class web site
 - individual assignments are just that, individual
 - copying solution code 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

Javabat Problems

- ▶ Small scale problems
- ▶ 9 sets
- ▶ create account, grant access to TA
- ▶ <http://javabat.com/>

Course Materials - Exams

- ▶ Midterm 1: 6 – 8 pm, Wed. October 7
- ▶ Midterm 2: 6 – 8 pm, Wed. November 18
- ▶ Final Exam: TBD. Usually evening first 3 days of finals.
 - Only up, never down
- ▶ If you have a conflict with midterm email instructor ASAP.
 - Make ups possible early same day or next morning.

More on Exams

- ▶ old tests on line to aid is studies
- ▶ tests consist of short answer questions and coding questions
- ▶ test emphasize problem solving, algorithm implementation, some syntax
- ▶ tests scores curved if instructor feels necessary.

Succeeding in the Course

- ▶ Randy Pausch,
CS Professor at CMU



- ▶ *"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 the Javabat problems
- ▶ do the extra section problems
- ▶ study for tests using the old tests
- ▶ study for tests in groups
- ▶ ask questions and get help when needed

Grades and Performance

- ▶ Final grade determined by final point total and a 900 – 800 – 700 – 600 scale
 - Will be adjusted with plusses and minuses at instructors if within 25 points of cutoff:
 - 875 – 899: B+, 900 – 924: A-
- ▶ Last semester **165** students enrolled in the course.
 - **125** students got a C or better.
 - **29** students got a D or F.
 - **11** students dropped or withdrew.
- ▶ The majority of students getting Ds or Fs missed 1 or more exams without an excuse, had a failing homework average, and a failing quiz average.

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