

# Topic 1

## CS314 Course Introduction

**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....

**In class: please close laptops and put away mobile devices.**

Mike Scott, Gates 6.304  
scottm@cs.utexas.edu  
www.cs.utexas.edu/~scottm/cs314/



## Who Am I?

- ▶ Professor of Instruction (lecturer) in CS department since 2000
- ▶ Undergrad Stanford, MSCS RPI
- ▶ US Navy for 8 years, submarines
- ▶ 2 years Round Rock High School



Rensselaer



CS314

Course Overview

## The Elephant in the Room

- ▶ In light of the ongoing global pandemic:
- ▶ Per UT Guidance: "Masks are strongly recommended indoors regardless of vaccination COVID-19 status."
- ▶ Per UT Guidance: If you are sick stay home. Lectures are recorded and may be watched later if you are sick.
  - lectures shall be recorded for later viewing
  - office hours and discussion sections shall have online options as circumstance warrant

CS314

Course Overview

3

## The Elephant in the Room

- ▶ Proactive Community Testing (PCT) is available on campus and "provides a quick and painless saliva or painless nasal-swab COVID-19 test for asymptomatic UT Austin students and employees."
- ▶ If you are interested in obtaining a Covid-19 vaccine please see this page:  
<https://uthealthaustin.org/patient-resources/covid-19-updates/covid-19-vaccination>

CS314

Course Overview

4

## What We Will Do Today

- Discuss
  - course content
  - procedures
  - tools
- For your TO DO list:
  - complete items on the startup page

[www.cs.utexas.edu/~scottm/cs314/handouts/startup.htm](http://www.cs.utexas.edu/~scottm/cs314/handouts/startup.htm)

## Course Goals

- Analyze algorithms and code for efficiency
- Be able to create and use canonical data structures: lists (array and linked), stacks, queues, trees, binary search trees, balanced binary search trees, maps, sets, graphs, hash tables, heaps, tries
- Know and use the following programming tools and techniques: object oriented programming (encapsulation, inheritance, polymorphism), interfaces, iterators, sorting, searching, recursion, dynamic programming, functional programming

## Course Goals

- After CS314 you can design and implement medium size programs (several 100s of lines of code, split between multiple classes) to solve interesting problems
- Recall, the three core areas of the UTCS undergrad degree:  
Programming, Theory, Systems
- After this class your instructors shall expect you to be able to write complex programs given a specification or problem statement.

## Prerequisites

- Formal: CS312 with a grade of C- or higher
- Informal: Ability to design and implement programs in Java using the following:
  - variables and data types
  - expressions, order of operations
  - Conditionals (if statements)
    - including boolean logic and boolean expressions
  - iteration (loops)
  - Methods (functions, procedures)
  - Parameters
  - 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 two people play connect 4?



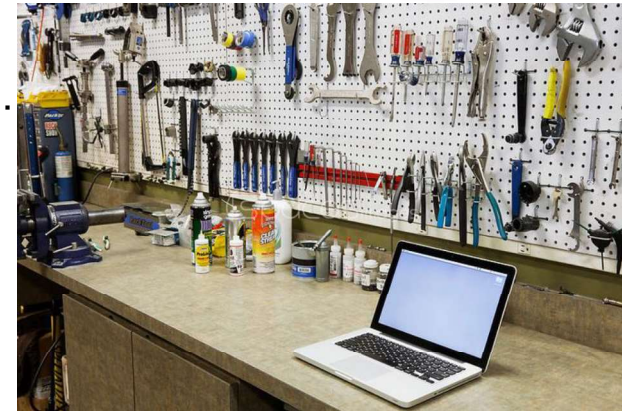
# CS314 Topics

- |                            |                             |
|----------------------------|-----------------------------|
| 1. Introduction            | 14. Searching, Simple Sorts |
| 2. Algorithm Analysis      | 15. Stacks                  |
| 3. Encapsulation           | 16. Queues                  |
| 4. Inheritance             | 17. Fast Sorting            |
| 5. Polymorphism            | 18. Trees                   |
| 6. Generics                | 19. Binary Search Trees     |
| 7. Interfaces              | 20. Graphs                  |
| 8. Iterators               | 21. Hash tables             |
| 9. Abstract Classes        | 22. Red-Black Trees         |
| 10. Maps, Sets             | 23. Huffman Code Trees      |
| 11. Linked Lists           | 24. Heaps                   |
| 12. Recursion              | 25. Tries                   |
| 13. Recursive Backtracking | 26. Dynamic Programming     |
|                            | 27. Functional Programming  |

## Data Structures



- ▶ simple definition:
  - variables that store other variables
- ▶ We will learn a toolbox full of data structures ...
- ▶ ... and how to build them ...
- ▶ ... and how to use new ones.



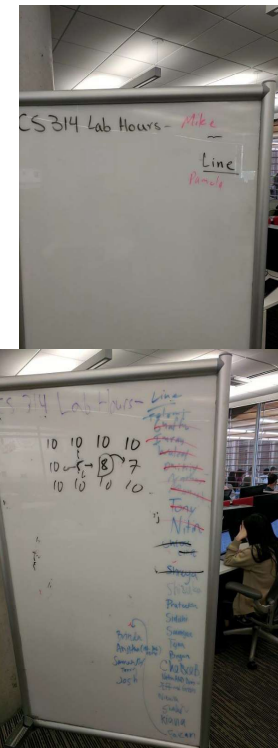
CS314

## Clicker Question 1

- ▶ Which of the following is a data structure?
- A. a method  
B. a try / catch block  
C. a double  
D. an array  
E. more than one of A - D

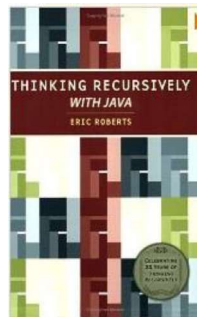
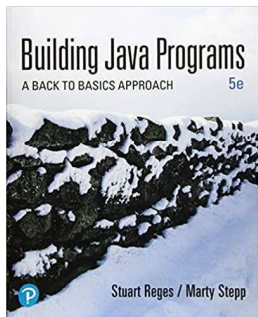
## Resources

- ▶ Class web site – most course material
- ▶ Class discussion group – Piazza
- ▶ Canvas -> Grades, Program Submissions, Access Perusall, Zoom Links, Recorded Lectures



## Books

- books are recommended, not required
- free alternatives on the web, see schedule
  - BJP (CS312 book) ***strongly recommended***
  - Thinking Recursively in Java - recursion



CS314

Course Overview

13

## Clicker Question 2

Which of these best describes you?

- A. First year at UT and first year college student
- B. First year at UT, transferring from another college or university
- C. Second year at UT
- D. Third year at UT
- E. Other

CS314

Course Overview

14

## Graded Course Components

- Academic Integrity Quiz, **5 points** (all correct or 0, multiple attempts)
- Programming projects
  - 11 projects, 20 points each, **220 points total**
- Perusall
  - 10 reading assignments, 4 points per, **40 points total**
- Exams: Outside of class
  - Exam 1, Wednesday 9/29, 6:45 – 9:15 pm, **250 points**
  - Exam 2, Wednesday 11/3, 6:45 - 9:15 pm, **250 points**
  - Exam 3, Final Exam period, time and date TBD, **250 points**
- Extra credit: Background survey **10 points**  
Course Instructor Evals **10 points**
- **5 + 220 + 40 + 250 + 250 + 250 + 10 + 10 = 1035**
- Non exam points capped at 250 pts
  - 35 points of “slack” among those non exam components
- No points added! Grades based on 1000 points, not 1035
- final points = min(250, sum of non exam) + e1 score + e2 score + e3 score

CS314

Course Overview

16

## Grades and Performance

- Final grade determined by final point total and a 900 – 800 – 700 – 600 scale
  - plusses and minuses if within 25 points of cutoff:  
A: 925 - 1000      A-: 900 - 924      B+: 875 - 899
- My CS314 Historical Grades
- **80% C- or higher:**
  - 28% A's, 35% B's, 17% C's
- **10% D or F**
- **10% Q or W (drop)**
- **ON CIS WORK LOAD EVALUATED AS HIGH**

CS314

Course Overview

16



## Assignments

- ▶ Non trivial programming projects
- ▶ Individual – do your own work
- ▶ **Programs checked automatically with plagiarism detection software (MOSS)**
- ▶ Turn in the right thing - correct name, correct format or you will lose points / slip days
- ▶ Graded on Correctness AND program hygiene  
"Code is read more often than it is written."  
- Guido Van Rossum
- ▶ Slip days: 8 for term, max 2 per assignment, don't use frivolously

## 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.' "*
- ▶ *"A lot of people want a shortcut. I find the best shortcut is the long way, which is basically two words: work hard."*

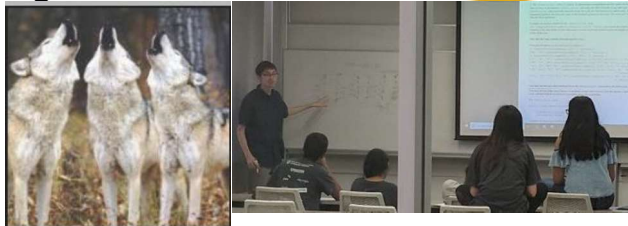
18

## Succeeding in the Course - Meta

- ▶ "Be the first penguin"
  - Ask questions!!!
  - lecture, section, Piazza, lab hours
- ▶ "It is impossible to be perfect"
  - Mistakes are okay.
  - That is how we learn.
  - Trying to be perfect means not taking risks.
  - no risks, no learning



- ▶ "Find a Pack"
  - Make friends.
  - Study with them!



## How to Get Help

- ▶ Piazza Post
- ▶ Help Hours
- ▶ Class examples
- ▶ Examples from book
- ▶ Discuss with other students at a high level

20

## Succeeding in the Course - Concrete

- ▶ Former student:
  - "I really like the boot camp nature of your course."
- ▶ do the readings
- ▶ start on assignments early
- ▶ get help from the teaching staff when you get stuck on an assignment
- ▶ attend or watch lecture and discussion sections
- ▶ participate on the class discussion group
- ▶ **do extra problems** - <http://tinyurl.com/pnzp28f>
- ▶ study for tests using the old tests
- ▶ study for tests in groups
- ▶ ask questions and get help

## Software

- ▶ Java - Oracle or OpenJDK, limit ourselves to Java 8
- ▶ IDE such as IntelliJ or Eclipse
- ▶ SSH into CS machines to test your programs
  - CS department account
  - SSH keys
  - Ability to transfer files and login remotely (WinSCP, Putty, Cyberduck, Filezilla, ...)
- ▶ A zip tool (create zip files to turn in)
- ▶ Zoom (for remote options)
- ▶ Chrome and Proctorio (if circumstances warrant)

## Clicker Question 3

Which computer programming language are you most comfortable with?

- A. Java
- B. C or C++
- C. Python
- D. Javascript
- E. Other

See: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>  
and <http://lang-index.sourceforge.net/>