CS303e Course Introduction

Chapman: I didn't expect a kind of Spanish Inquisition.
Cardinal Ximinez[Palin]: 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
scottm@cs.utexas.edu
www.cs.utexas.edu/~scottm/cs303e

Agenda

- Overview of: this course
- the elements of computing program
- Course logistics including: how to get help
- the schedule
- tips for success

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

My Path to CS
Intro to Programming

- Learn to design and implement computer programs to solve problems.
- I assume you have NEVER written a line of code.

1. output
2. identifiers
3. errors (syntax, runtime, logic)
4. reserved words
5. variables, operators, computations
6. constants
7. built in math functions
8. conditional execution
9. boolean logic
10. iteration
11. programmer defined functions
12. strings
13. lists
14. 2d lists (matrices)
15. files
16. exceptions
17. dictionaries
18. objects and classes (programmer defined data types)
19. sorting and searching

Programing and CS

- A tool for doing the cool stuff in CS
- You can't create a self driving vehicle without the software to control the vehicle.

Startup

- If you have not already done so ...
- ... complete the items on the class start-up page
- [http://www.cs.utexas.edu/~scottm/cs303e/handouts/startup.htm](http://www.cs.utexas.edu/~scottm/cs303e/handouts/startup.htm)
**Book**

- book is required
  - we follow it quite closely
- programming assignments, limited to features from the book up to a given chapter
- suggested exercises

**Letter Grades**

- Final grade determined by final point total and a
  - $\geq 925 \rightarrow A$
  - 900 - 924 $\rightarrow A-$
  - 875 - 899 $\rightarrow B+$
  - 825 - 874 $\rightarrow B$
  - 800 - 824 $\rightarrow B-$
  - 775 - 799 $\rightarrow C+$
  - 725 - 774 $\rightarrow C$
  - 700 - 724 $\rightarrow C-$
  - 675 - 699 $\rightarrow D+$
  - 625 - 674 $\rightarrow D$
  - 600 - 624 $\rightarrow D-$
  - $\leq 599 \rightarrow F$

**Graded Course Components**

- Programming projects
  - 13 projects, 20 points: **260 points**

- Exams (online, Canvas, Chrome, Proctorio)
  - Midterm, Tuesday, 7/6, 5 - 9 pm window, roughly 2 hour exam, **250 points**
  - Final, Thursday, 7/29, 9 am - 12 noon window, **500 points**

- Extra credit
  - practice exam, Monday, 6/28, **5 points**
  - eCIS completion, **5 points**

\[ 260 + 250 + 500 + 5 + 5 = 1020 \]

- Programming Assignments capped at 250 pts
- No points added! Grades based on 1000 points, not 1020

**Assignments**

- Start out simple but get more challenging
- 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
- Slip days
  - 6 for term, max 1 per assignment
  - don’t use frivolously
- Graded on correctness and program hygiene (style, best practices), typical 12 / 8 split
Getting Help

- **Post to Piazza.**
  - can make anonymous to other students
  - can post to instructors only
  - do not post more than 2 lines of code on a public post

- **Help Hours**
  - check schedule
  - sign up at web site
  - join Zoom session
  - Zoom links on Canvas.

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.**"

Succeeding in the Course - Meta

- “Be the first penguin”
  Randy Pausch
  - Ask questions!!!
  - lecture, Piazza, lab hours

- “It is impossible to be perfect”
  Captain Symons
  - Mistakes are okay.
  - That is how we learn.
  - Trying to be perfect means not taking risks.
  - no risks, no learning

Succeeding in the Course - Concrete

- Whole course is cumulative!
  - Material builds on itself
    - failure to understand a concept leads to bigger problems down the road, so ...
  - do the readings
  - watch the videos per the schedule
  - start on assignments early
  - get help from the teaching staff when you get stuck on an assignment
  - participate on the class discussion group
  - ask questions and get help when needed
  - **DO MORE PRACTICE PROBLEMS** -> Book, CodingBat, Professor Bulko’s Site
Succeeding in the Course

- Cannot succeed via memorization.
- The things I expect you to do are not rote.
  - programming is a skill
  - you cannot memorize your way through the material and the course
- Learn by doing.
- If you are brand new to programming or have limited experience I strongly recommend you do lots and lots of practice problems.