

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

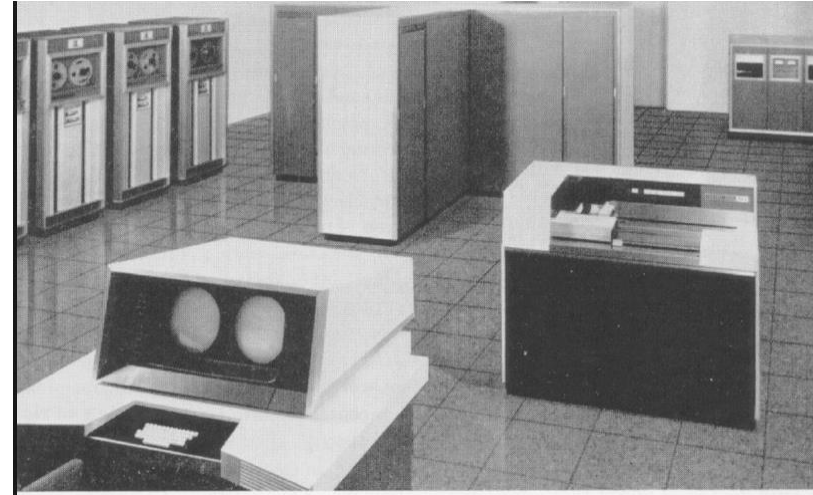
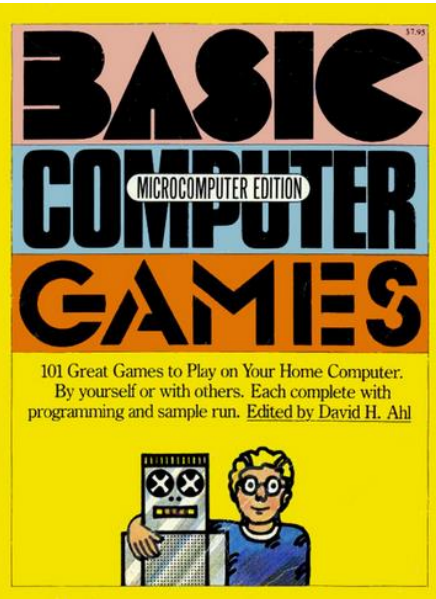
- ▶ Lecturer in CS department since 2000
- ▶ Undergrad Stanford, MSCS RPI
- ▶ US Navy for 8 years, submarines
- ▶ 2 years Round Rock High School



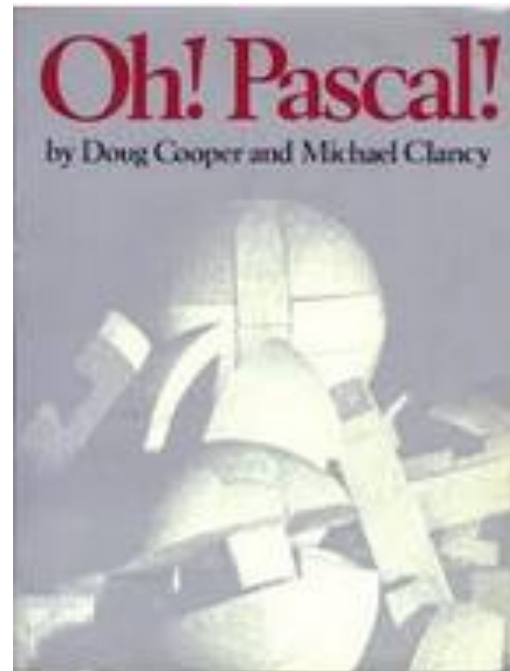
Rensselaer



My Path to CS



```
10 INPUT "What is your name: "; U$
20 PRINT "Hello "; U$
25 REM
30 INPUT "How many stars do you want: "; N
35 S$ = ""
40 FOR I = 1 TO N
50 S$ = S$ + "*"
55 NEXT I
60 PRINT S$
65 REM
70 INPUT "Do you want more stars? "; A$
80 IF LEN(A$) = 0 THEN GOTO 70
90 A$ = LEFT$(A$, 1)
100 IF (A$ = "Y") OR (A$ = "y") THEN GOTO 30
110 PRINT "Goodbye ";
120 FOR I = 1 TO 200
130 PRINT U$; " ";
140 NEXT I
150 PRINT
```



Intro to Programming

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

1. output, fstrings
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, repetition
11. programmer defined functions
12. Strings
13. lists
14. lists of lists (matrices)
15. files
16. exceptions
17. dictionaries
18. objects and classes (programmer defined data types)
19. recursion
20. 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



Programming

- ▶ Start simple
- ▶ ... but get complex by end of the class

```
>>>
>>>
>>>
= RESTART: C:/Users/scottm/Documents/303e/_Su 20/example programs
/assignments/Initials.py

MDS

MMM      MMM      DDDDDDD      SSSSSSSSS
MMM      MMM      DDDDDDDDD      SSSSSSSSS
MM MM   MM MM    DD      DDD      SSSS
MM MM   MM MM    DD      DD       SSSS
MM MM   MM MM    DD      DD       SSSSS
MM      MMMM   MM  DD      DD       SSSSS
MM      MMMM   MM  DD      DD       SSSS
MM      MM      MM  DD      DDD      SSSS
MM      MM      MM  ..  DDDDDDDDD  ..  SSSSSSSSS  ..
MM      MM      MM  ..  DDDDDDD    ..  SSSSSSSSS  ..

>>>
>>>
>>>
```



```
P2
250 360
255
104 103 105 99 92 92 94 100 100
109 103 105 100 96 92 93 94 88
109 105 99 98 98 102 105 93 88
106 110 104 99 103 112 112 97 91
113 113 112 103 105 109 107 91 90
121 121 119 107 103 99 95 90 90
114 117 112 105 105 95 92 93 94
101 107 103 103 105 96 92 96 97
94 108 110 104 103 101 97 99 100
94 112 123 106 99 98 95 96 96
100 116 123 107 99 100 95 101 96
104 115 113 105 99 100 97 104 104
103 107 108 107 103 101 93 92 96
103 107 104 107 106 102 97 91 90
109 107 104 105 108 104 105 105 98
106 110 109 106 108 105 107 109 108
106 120 127 123 117 108 107 111 111
107 124 127 127 117 108 107 111 111
```



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>

Book

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

starting out with >>> **PYTHON**[®]
FIFTH EDITION



TONY GADDIS

Graded Course Components

- ▶ Programming projects
 - 13 projects, 10 or 20 points : **210 points**
- ▶ Exams
 - Midterm, In class Wednesday, July 3, 11:30 am – 1:30 pm **400 points**
 - Final, Thursday, August 1, 7 - 10 pm **400 points**
- ▶ Extra credit
 - CS background survey on Canvas. **10 points**
 - course survey completion, **10 points**

$210 + 400 + 400 + 10 + 10 = 1030$
- ▶ Programming Assignments capped at 200 pts
 - 30 points of “slack”, including extra credit
- ▶ No points added! Grades based on 1000 points, not 1030
- ▶ Final point total = **$\min(200, \text{sum of points on programs} + \text{background survey completion} + \text{instructor end of course survey}) + \text{midterm exam score} + \text{final exam score}$**

Letter Grades

- ▶ Final grade determined by final point total

$\geq 925 \rightarrow A$

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 $\rightarrow D-$

$\leq 599 \rightarrow F$

In Class Exercises - Grade Bump

- ▶ Recall: Final point total = $\min(200, \text{sum of points on programs} + \text{background survey completion} + \text{instructor end of course survey}) + \text{midterm exam score} + \text{final exam score}$
- ▶ Each lecture shall have an in-class programming exercise. 21 total. Completing these may help you get bumped to the next higher grade if you are close to a cutoff.
- ▶ 1 point added for every 2 exercises completed with reasonable effort
 - rounded up
- ▶ **For example, you end up with 893 points per the formula above. You complete 14 or more of the 21 in class exercises with a reasonable attempt. Your grade shall be bumped from B+ to A-.**

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
 - 8 for term, max 1 per assignment
 - don't use frivolously
- ▶ Graded on correctness and *program hygiene* (style, best practices), typical 60% / 40% split

Getting Help

- ▶ Post to Ed (link on Canvas).
 - 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
 - Most help hours in person in GDC 3.202
 - A few help hours via Zoom, check the Canvas course page and the Zoom tab for links

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, help 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
- ▶ come to class
- ▶ 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.***