CS312 Course Introduction

"Computers are good at following instructions, but not at reading your mind."

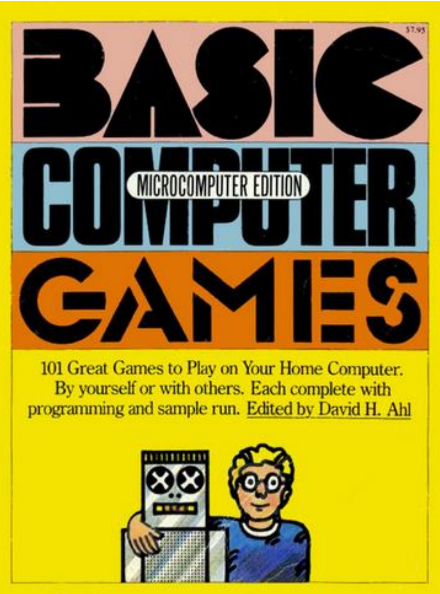
-Donald Knuth, *Tex* p. 9

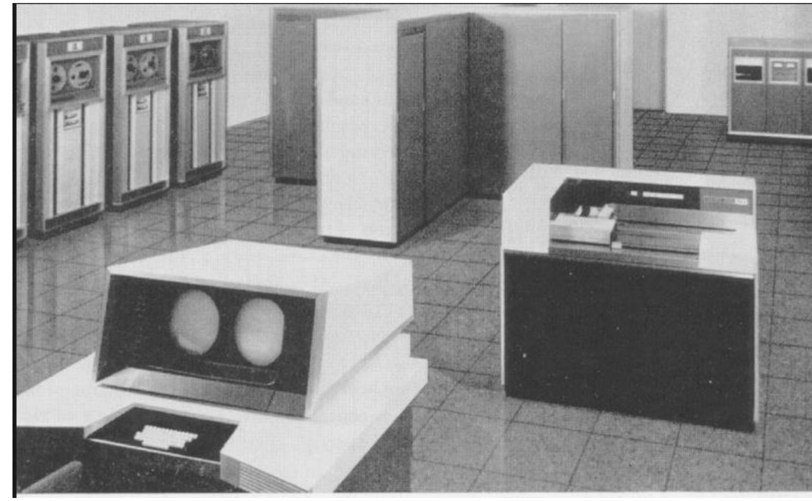
Mike Scott, Gates 6.304
scottm@cs.utexas.edu
www.cs.utexas.edu/~scottm/cs312

Who Am I

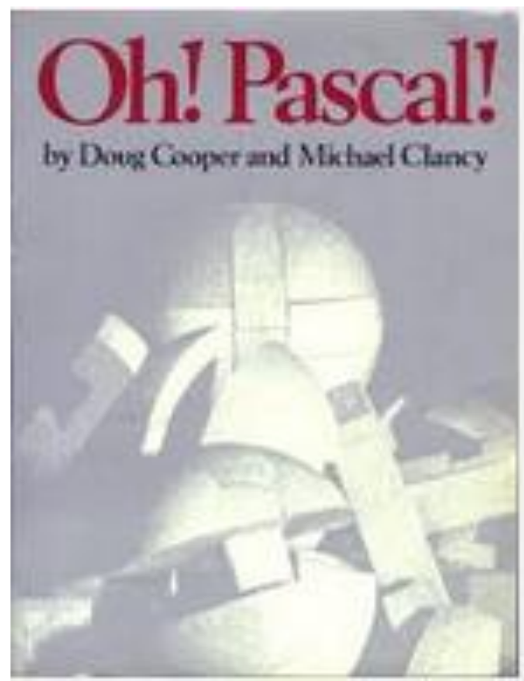
- Lecturer in CS department since 2000
- Undergrad Stanford, MSCS RPI
- US Navy for 8 years, submarines
- 2 years Round Rock High School

My Path to CS





```
 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
```



What We Will Do Today

- Introductions and administrative details
- Start Java Basics

Intro to Programming

- Learn to design and implement computer programs to solve problems.

1. course Intro
2. basic Java
3. static methods
4. expressions & variables
5. for loops
6. more loops, constants
7. parameters
8. 2d graphics
9. more graphics
10. return values, Math methods
11. conditional statements
12. cumulative algorithms
13. Strings
14. while loops
15. random numbers
16. Boolean logic
17. assertions
18. file input 1
19. file input 2
20. file input 3
21. arrays
22. more arrays
23. tallying algos
24. sorting, searching
25. more array algos
26. 2d arrays
27. classes and objects
28. methods
29. constructors
30. creating classes, Enums
31. inheritance
32. polymorphism
33. ArrayList
34. recursion

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

The Parable of Aaron D.

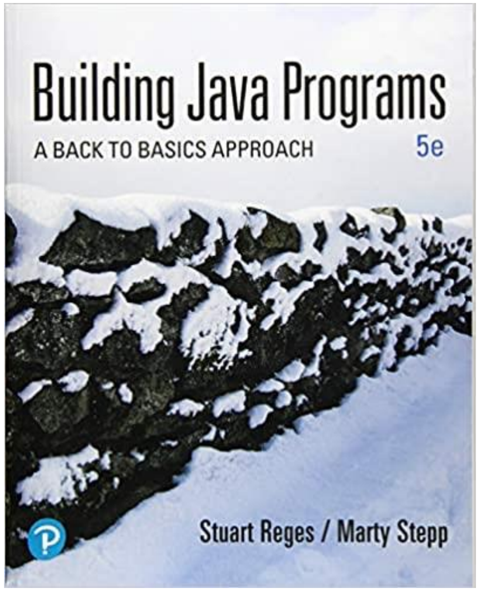
- I assume no prior programming experience.
- You are limited to what you can use on assignment to what we have covered in the book.
- I will defer questions that are well past what we are currently covering.
- Programming is not a spectator sport.
  - The only way to learn to program is to program
- Aaron D. and the 500 problems.

Startup

- If you have not already done so …
- … complete the items on the class start-up page
- http://www.cs.utexas.edu/~scottm/cs312/handouts/startup.htm

Books and software

- book is required - we follow it quite closely
- Chrome and Proctorio extension for exams
- Java for programming
- IDE for programming
- Canvas for turning in assignments, grades, UT Instapoll, section problems, exams



Clicker 1

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. in second year at UT

D. in third year at UT

E. other

Graded Course Components

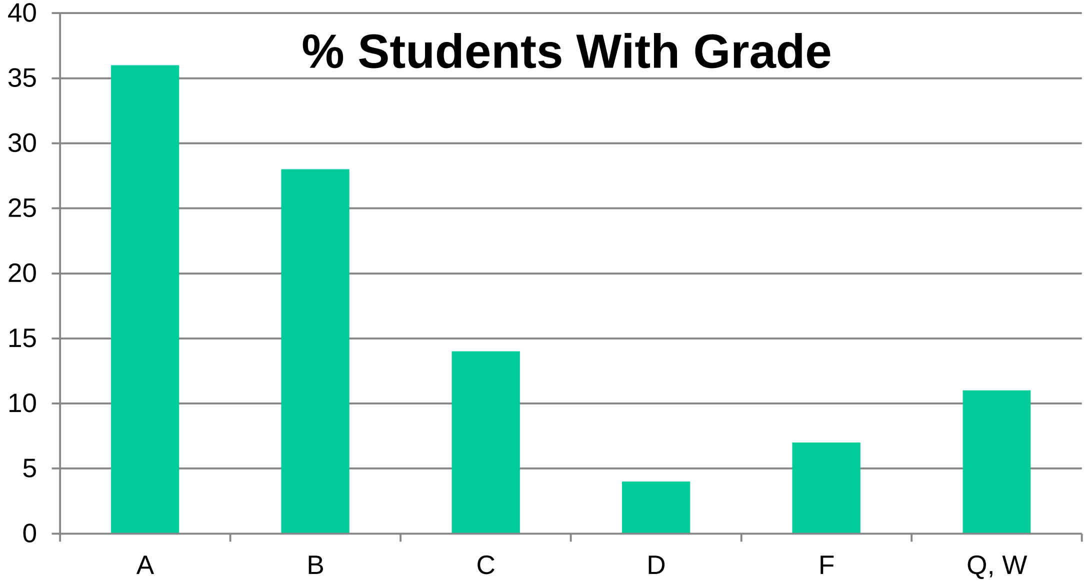
- UT Instapoll (In class questions)
  - 42 lectures with UT Instapoll, 7 dropped, **35 points**
- Discussion section problems (Go to your section. Do not refer to the course id from Canvas, same for all.)
  - 10 problems, 3 points each, 2 dropped, **24 points**
- Programming projects
  - 12 projects, 1st 10, rest 20 points each: **210 points total** (lowest grade of assignments 2 - 12 dropped)
- Exams: Outside of class
  - Exam 1, Wednesday, 9/30, approx. 6:45 – 9:15 pm, **150 points**
  - Exam 2, Wednesday, 11/11, approx. 6:45 – 9:15 pm, **250 points**
  - Exam 3, Date and time TBD (during finals week), **340 points**

35 + 24 + 210 + 150 + 250 + 340 + 5 (quiz) = 1014

- clicker, Quizzes, Programming Assignments capped at 260 pts
  - 14 points of “slack” among those 3 components
  - Extra Credit: Computing background survey +3 points, practice exam + 3 points, eCIS & TA Survey +6 points

Grades and Performance

- No points added! Grades based on 1000 points, not 1014
- Final grade determined by final point total and a 900 – 800 – 700 – 600 scale
  - plusses and minuses if within 25 points of cutoff: 875 – 899: B+, 900 – 924: A-
- historically my CS312 classes (~2000 Students)
- **80% C- or higher:**
  38% A's,
  28% B's
  12% C's
- **10% D or F**
- **10% Q or W (drop)**



Assignments

- Start out easy but get **much, much** harder
- Individual – do your own work
- Programs checked automatically with plagiarism detection software
- Turn in the right thing - correct name, correct format or you will lose points / slip days
- Slip days
  - 8 for term, max 2 per assignment
  - don't use frivolously

Succeeding in the Course

- Randy Pausch,
  CS Professor at CMU said:

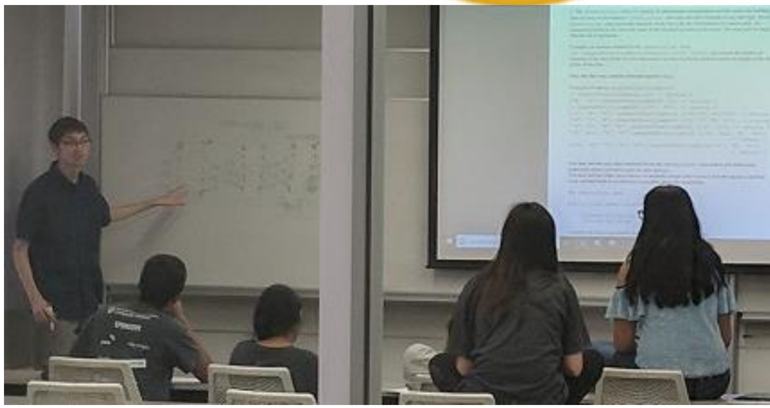
- *"When I got tenure a year early at Virginia, other Assistant Professors would come up to me and say, 'You got tenure early!?!?! What's your secret?!?!?' and I would tell them, 'Call me in my office at 10pm on Friday night and I'll tell you.' "*

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

Succeeding in the Course - Meta

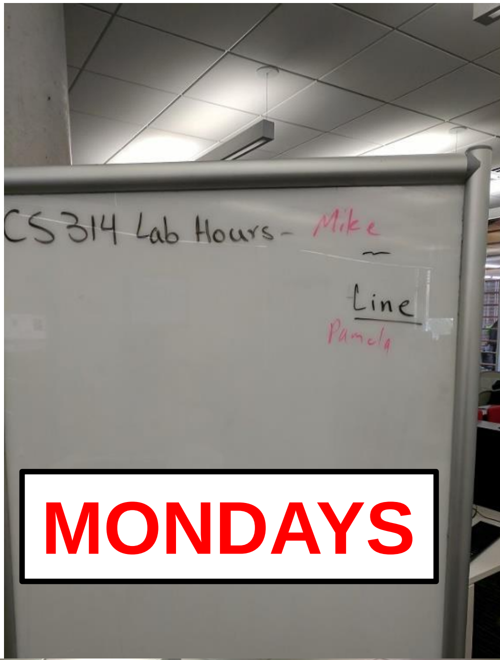
- "Be the first penguin"
  - Ask questions!!!
  - lecture, section, Piazza, lab hours

- "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!

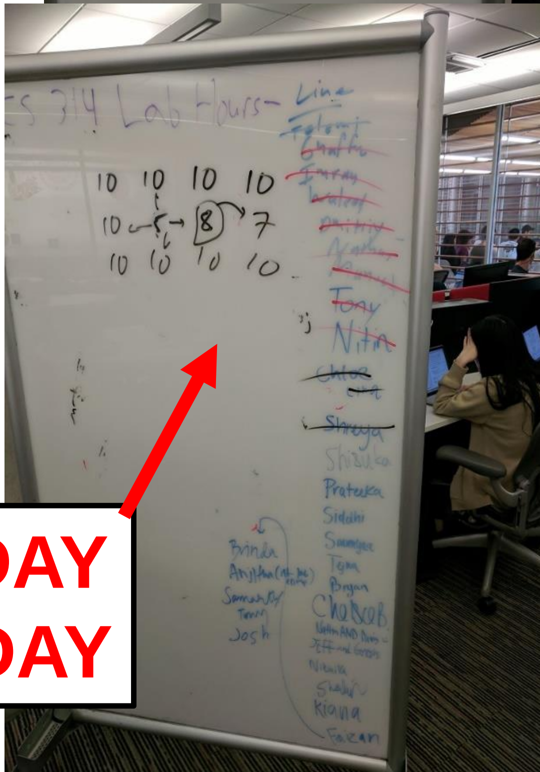


How to Get Help

- Piazza Post
- Help Hours via Zoom
- SI Sessions
- Email instructor or TAs
  – Prefer Piazza
- Class examples
- Examples from book
- Discuss with other students at a ***high level***



MONDAYS



WEDNESDAY & THURSDAY

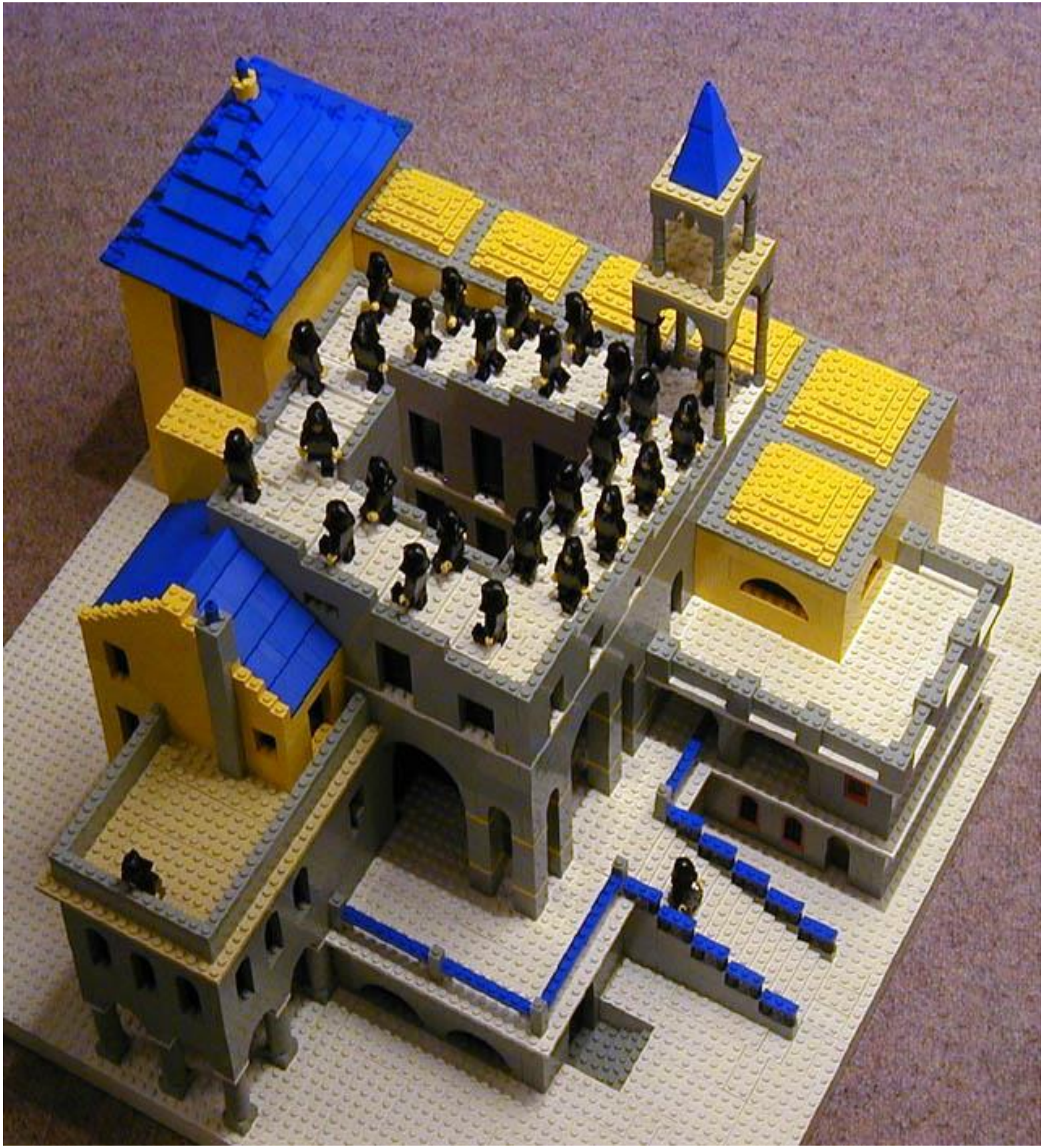
Succeeding in the Course - Concrete

- Whole course is cumulative!
- Material builds on itself
  - failure to understand a concept leads to bigger problems down the road, so …
- do the readings
- 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 class discussion group
- **do extra problems (Practice It! http://practiceit.cs.washington.edu/)**
- study for tests using the old tests
- study for tests in groups
- ask questions and get help when needed

Succeeding in the Course

- Cannot succeed via memorization.
- The things I expect you to do are **<u>not</u>** rote.
- Learn by doing.
- If you are brand new to programming or have limited experience I ***<u>strongly</u>*** recommend you do ***lots and lots of practice problems.***
  – Practice It! web site
  – JavaBat

Programming is like Legos…



Legos and Programming

- With Legos and Programming you have a small number of primitives. (basic tools or pieces)
- But you build huge, elaborate structures out of those simple pieces.

A Brief Look at Computer Science

- This class, like most first classes in Computer Science, focuses solving problems and implementing those solutions as computer programs.
  - you learn how to program
- … 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."*

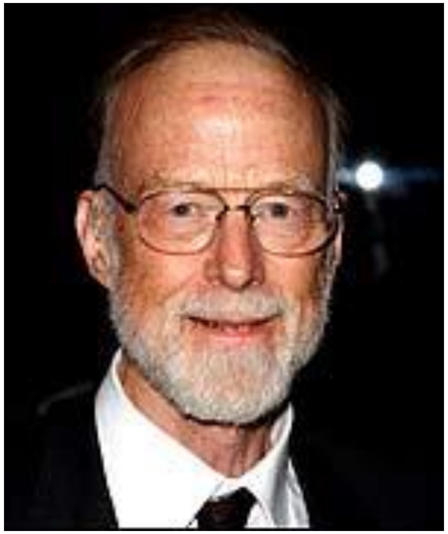
  -Owen Astrachan, Duke University
- learn to program
  - programming a key tool in later courses

Computer Programming and Computer Science

- Generally the first thing that is studied in Chemistry is stoichiometry.
  - Why? It is a skill necessary in order to study more advanced topics in Chemistry
- The same is true of problems solving / programming and computer science.

- “What is the linking thread which gathers these disparate branches into a single discipline? …it is the art of programming a computer. It is the art of designing efficient and elegant methods of getting a computer to solve problems, theoretical or practical, small or large, simple or complex.”

  - C. A. R. Hoare
- Sir Tony Hoare. Turing Award Winner. Inventor of the quicksort algorithm



- “Programming is unquestionably the central topic of computing.

  In addition to being important, programming is an enormously exciting intellectual activity. In its purest form, it is the systematic mastery of complexity. For some problems, the complexity is akin to that associated with designing a fine mechanical watch, i.e., discovering the best way to assemble a relatively small number of pieces into a harmonious and efficient mechanism. For other problems, the complexity is more akin to that associated with putting a man on the moon, i.e, managing a massive amount of detail.

  In addition to being important and intellectually challenging, programming is a great deal of fun. Programmers get to build things and see them work.. What could be more satisfying? “

  - John V. Guttag, Professor at MIT
  research in AI, medical systems, wireless networking

Computer Programming

- a skill and tool that are applied to all other areas of computer science
  - artificial intelligence, networks, cpu architecture, graphics, systems (programming languages, operating systems, compilers), security, and on and on …
- We will be using solving problems and implementing solutions in a programming language called Java
- problem solving and computational thinking are key

What do Computer Scientists do?

- Computer Scientists solve problems
  - creation of algorithms
- Some examples
  - you
  - Kurt Dresner, Intersection Control
  - Austin Villa, Robot Soccer
  - Doug and Steve, the TRIPS processor

You!

- Encryption and Decryption
- Ever entered your credit card number to a website? game company?

```
HABUQXC 3

TLQ BJJ QABQ DCY. GXTTXQ, ALPXIXC, PZQA QAX BYYZYQBTHX LS AXC SZIX
OBKEAQXCY, HLKJO BYN LT QAX YKGVXHQ, PBY YKSSZHZXTQ QL OCBP SCLD AXC
AKYGBTO BTW YBQZYSBHQLCW OXYHCZUQZLT LS DC. GZTEJXW. QAXW BQQBHNXO AZD
ZT IBCZLKY PBWY--PZQA GBCXSBHXO FKXYQZLTY, ZTEXTZLKY YKUULYZQZLTY, BTO
OZYQBTQ YKCDZYXY; GKQ AX XJKOXO QAX YNZJJ LS QAXD BJJ, BTO QAXW PXCX BQ
JBYQ LGJZEXO QL BHHXUQ QAX YXHLTO-ABTO ZTQXJJZEXTHX LS QAXZC TXZEAGLKC,
JBOW JKHBY. AXC CXULCQ PBY AZEAJW SBILKCBGJX. YZC PZJJZBD ABO GXXT
OXJZEAQXO PZQA AZD. AX PBY FKZQX WLKTE, PLTOXCSKJJW ABTOYLDX, XMQCXDXJW
BECXXBGJX, BTO, QL HCLPT QAX PALJX, AX DXBTQ QL GX BQ QAX TXMQ BYYXDGJW
PZQA B JBCEX UBCQW. TLQAZTE HLKJO GX DLCX OXJZEAQSKJ! QL GX SLTO LS
OBTHZTE PBY B HXCQBZT YQXU QLPBCOY SBJJZTE ZT JLIX; BTO IXCW JZIXJW
ALUXY LS DC. GZTEJXW'Y AXBCQ PXCX XTQXCQBZTXO.
```

After a Little Computation:

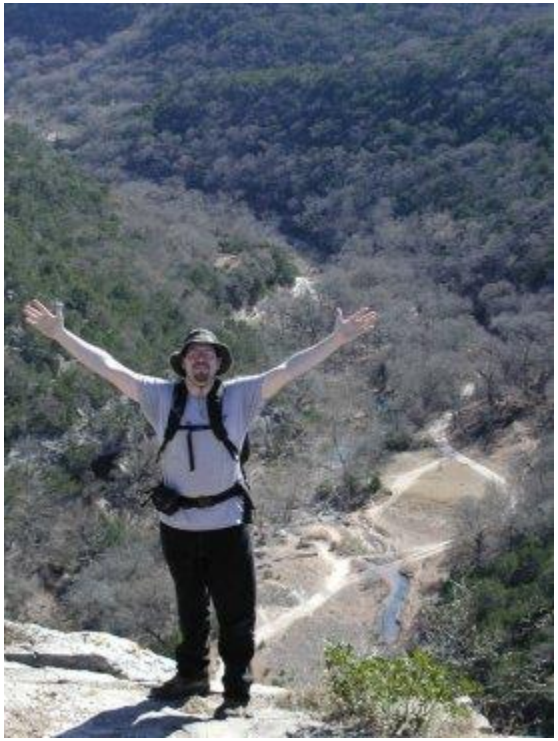
```
FSABTER 3

ONT ALL TSAT URH. PEOOET, SNMEVER, MITS TSE AHHIHTAOFE NY SER YIVE
DACGSTERH, FNCLD AHK NO TSE HCPQEFT, MAH HCYYIFIEOT TN DRAM YRNU SER
SCHPAOD AOW HATIHYAFTNRW DEHFRIBTINO NY UR. PIOGLEW. TSEW ATTAFKED SIU
IO VARINCH MAWH--MITS PAREYAFED JCEHTINOH, IOGEOINCH HCBBNHITINOH, AOD
DIHTAOT HCRUIHEH; PCT SE ELCDED TSE HKILL NY TSEU ALL, AOD TSEW MERE AT
LAHT NPLIGED TN AFFEBT TSE HEFNOD-SAOD IOTELLIGEOFE NY TSEIR OEIGSPNCR,
LADW LCFAH. SER REBNRT MAH SIGSLW YAVNCRAPLE. HIR MILLIAU SAD PEEO
DELIGSTED MITS SIU. SE MAH JCITE WNCOG, MNODERYCLLW SAODHNUE, EXTREUELW
AGREEAPLE, AOD, TN FRNMO TSE MSNLE, SE UEAOT TN PE AT TSE DEXT AHHEUPLW
MITS A LARGE BARTW. ONTSIOG FNCLD PE UNRE DELIGSTYCL! TN PE YNOD NY
DAOFIOG MAH A FERTAIO HTEB TNMARDH YALLIOG IO LNVE; AOD VERW LIVELW
SNBEH NY UR. PIOGLEW'H SEART MERE EOTERTAIOED.
```

- Apply some human smarts:

Kurt Dresner – Intersection Control

- Former PhD student in UTCS department
  - – working at Google now
- area of interest artificial intelligence
- ***Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism***
  - – how will intersections work if and when cars are autonomous?
  - – Simulator



Austin Villa – Robot Soccer

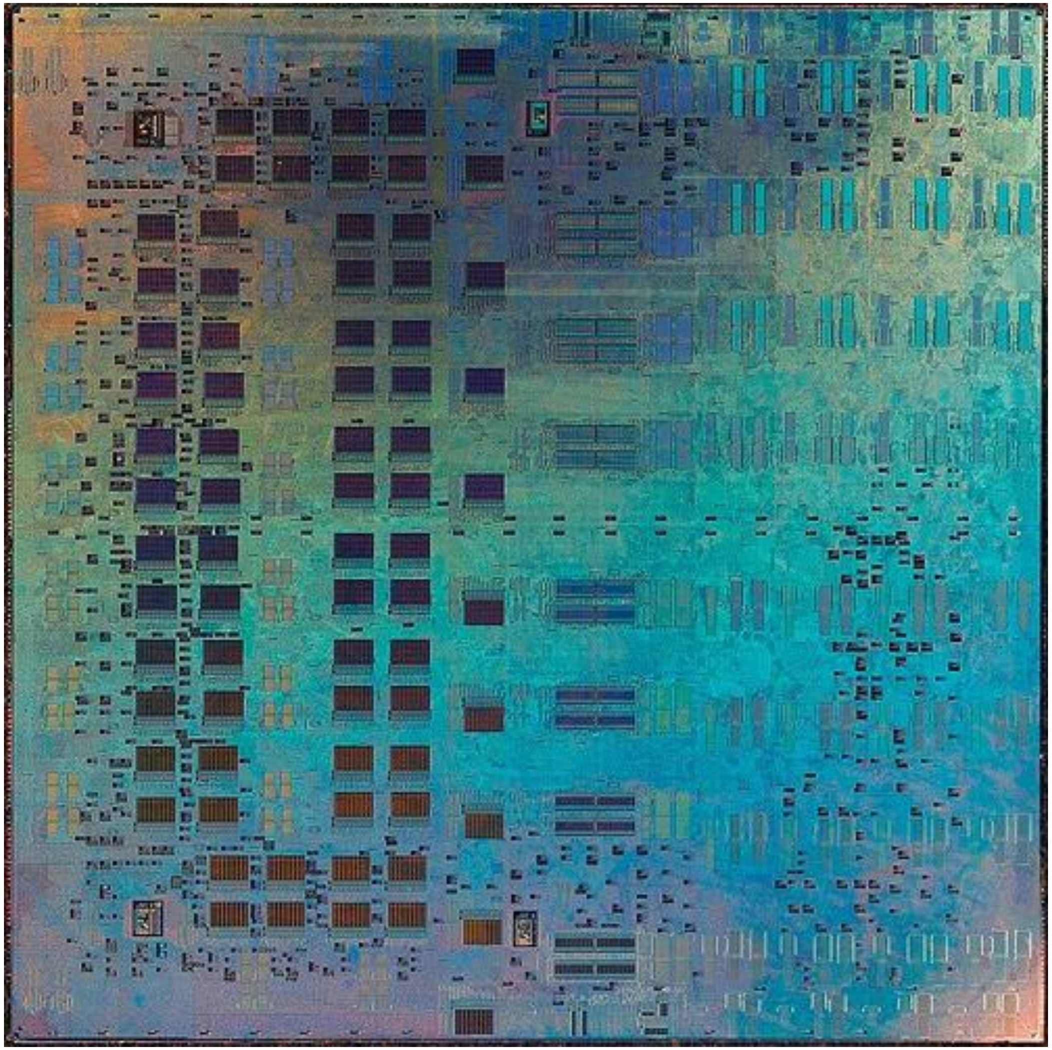
- Multiple Autonomous Agents
- Get a bunch of Sony Aibo robots to play soccer
- Problems:
  - *vision* (is that the ball?)
  - *localization* (where am I?)
  - *locomotion* (I want to be there!)
  - *coordination* (I am open! pass me the ball!)
- http://www.cs.utexas.edu/~AustinVilla/
- Video Video2

Doug and Steve

- Doug Burger and Steve Keckler
  - and many, many others ....
- TRIPS
  - what has happened to processor speeds the past 5 years?
  - what is a super computer?

  - http://www.cs.utexas.edu/users/cart/trips/

The Trips Chip Prototype



Google Trends

- http://www.google.com/trends
- Try these:
  - computer science
  - Mumford and Sons
  - computer science, Mumford and Sons
  - facebook, computer science, Mumford and Sons
  - binary search tree
  - recursion
  - linked lists, binary search tree
  - AP
  - super bowl

Goolge N Grams

- http://books.google.com/ngrams