Welcome!

- CS105: “Great Insights in Computer Science”.
- Somewhat experimental course.
- Instructional technologies
  - Using online course support (sakai)
  - iClickers for in-class interaction
  - Programs for out-of-class interaction
  - Podcasting (including music videos)
My Course Goals

• Cool facts, cool ideas. Ideas are “how to” facts. Shoot for one or two per lecture.

• People (my parents, say), don’t understand how I have a PhD in CS and can’t help them when their Windows box crashes.

• If it’s not about XP, what else is there?
   - That’s what I want to tell you...

Introduction to CS101s

• Target audience: Undergrads, as a first (possibly only) computer-science course.

• Seminar in Computers and Society: What impact have computers had on the world?

• Introduction to Computer Science: How do I learn to create my own software?

• Introduction to Computers and Their Application: What do I need to know about computing technology?
Course Goals: Questions

- What is Computer Science?
- Why is it fun/interesting?
- How is it different from software engineering?
- What are the insights that make computer science its own academic discipline?

Textbook

- Not too daunting or detailed.
- Inspiring and informative.
- *Pattern on the Stone, the simple ideas that make computers work*, Danny Hillis, Basic Books, 1998.
- Enjoyable to read; not really a textbook at all.
- I will add meat to the wonderful skeleton he creates.
Clickers

- Available at several bookstores.
- Can keep them (most popular clicker at Rutgers) or sell them back.
- We’ll use them for attendance, reading comprehension quizzes, straw polls.
- I’ve created some interactive demos, hope to expand the set.

http://www.iclicker.com/

Grading

- 20%: Class participation
- 20%: Homework
- 20%: Midterm #1 (in class)
- 20%: Midterm #2 (at final)
- 20%: Cumulative (at final)
Class Participation

- Each lecture, roughly 5 multiple choice questions.
- 1 point for each answer, .1 bonus if correct.
- May answer questions in class using iClickers
- May also download audio podcast and use sakai.
- For you benefit and mine, I’d prefer you use the podcasts as a backup (ill, out of town, lost clicker) and plan to attend lectures.

Other Grading Elements

- Homeworks (HW)
  - biweekly
  - hand in answers online via sakai
  - may need to run programs in “scratch”
- Exams
  - Exam questions modeled on HW
Syllabus: Chapters 0-2

Preface: Magic in the Stone
- What does the title mean?

Chapter 1: Nuts and Bolts
- From physics to bits.

Chapter 2: Universal Building Blocks
- From logic gates to a computer.

Syllabus: Chapters 3-5

Chapter 3: Programming
- Giving the computer instructions.

Chapter 4: How Universal Are Turing Machines?
- How many kinds of computation are there?

Chapter 5: Algorithms and Heuristics
- Solving some (hard?) problems.
Syllabus: Chapters 6-7

Chapter 6: Memory: Information and Secret Codes
  • Bits as messages.

Chapter 7: Speed: Parallel Computers
  • Computers working together.

Syllabus: Chapters 8-9

Chapter 8: Computers That Learn and Adapt
  • Can computers exceed their programming?

Chapter 9: Beyond Engineering
  • Programs that evolve.
They Are Everywhere!

1977: “There is no reason for any individual to have a computer in his home.”


Survey

• How many computers are with you right now? Laptop, handheld game, cell phone, GPS device, PDA, mp3 player, ...

A. 0
B. 1
C. 2
D. 3
E. 4 or more
Previously Unthinkable

Life After Death?
In a word, computer science is the study of:

A. Programming  
B. Communication  
C. Reduction  
D. Automation  
E. Acceleration

One-Word Summary

- If I had to summarize the intellectual contribution of computer science in one word, it would be “reduction”.
- Computer scientists solve problems by reducing them to simpler problems.
- We’ll see this same idea played out over and over again in different settings...
Levels of Complexity

- Networking (OSI Layers): application, presentation, session, transport, network, data link, physical.
- Vision (Marr): computational, algorithmic, implementation.
- Computing: application, high-level language, machine language, logic blocks, logic gates, physical.
- Storage hierarchy: offline-storage, hard disk, RAM, cache, registers.

What Does The Title Mean?

- Read the preface.
- Come back next time and tell me what you found out!
Today’s Idea

- I will start with the textbook next time, introducing bits and some simple gates.
- Please read the Preface and Chapter 1.
- But, I want to give you something to chew on to get those gears turning...

Bar Codes

- Many different styles of barcodes.
- Most common is UPC-A, in use in most North American retail stores.
- I will describe the UPC encoding.
- Many of the same ideas apply to other codes: checks, photostamps, IR remotes.
**Universal Product Codes**

- Method of identifying products at point of sale by 11-digit numbers.
- Method of encoding digit sequences so they can be read quickly and easily by machine.

**Reduction Idea**

- Each level uses an **encoding** to translate to the next level.
  - Patterns of ink.
  - Sequence of 95 zeros and ones (‘‘bits’’).
  - Sequence of 12 digits.
  - Sequence of 11 digits.
  - Name/type/manufacturer of product.
Product Name

• Ponds Dry Skin Cream
  - 3.9 oz (110g)
  - Unilever Home and Personal Care USA
• Name Badge Labels (Size 2 3/16" x 3 3/8")
  - 100 Labels
  - Avery Dennison/Avery Division

11-Digit Number

• Digit = \{0,1,2,3,4,5,6,7,8,9\}
• Sequence of 11 digits
• How many different items can they encode?
  A. 10,000,000,000 (one with 10 zeros)
  B. 100,000,000,000 (one with 11 zeros)
  C. 9,999,999,999 (10 nines)
  D. 19,999,999,999 (one with 10 nines)
Encode Name By 11 Digits

- First 6 digits: Manufacturer
  - First digit, product category:
    - 0, 1, 6, 7, 8, or 9: most products
    - 2: store’s use, for variable-weight items
    - 3: drugs by National Drug Code number
- Last 5 digits: Manufacturer-assigned ID

Examples

- Labels: 0-72782-051440
  - 0=general product
  - 72782=Avery
  - 051440=Avery’s code for this product
- Ponds: 3-05210-04300
  - 3=drug code
  - 05210=Unilever
  - 04300=National Drug Code for this product
12-Digit Number

- The UPC folks decided to include another digit for error checking. Example:
  - 01660000070   Roses Lime Juice (12 oz)
  - 04660000070   Eckrich Franks, Jumbo (16 oz)
  - 05660000070   Reese PB/Choc Egg (34 g)
  - 08660000070   Bumble Bee Salmon (14.75 OZ)

- Misread digit #2 and you turn sweet to sour.

Check Digit

1. Add the digits in the odd-numbered positions (first, third, fifth, etc.) together and multiply by three.

2. Add the digits in the even-numbered positions (second, fourth, sixth, etc.) to the result.

3. Subtract the result from the next-higher multiple of ten. The result is the check digit.
**Code and Example**

set evensum to \(d_2 + d_4 + d_6 + d_8 + d_{10}\)
set oddsum to \(d_1 + d_3 + d_5 + d_7 + d_9 + d_{11}\)
set checkdigit to \((0-3 \times \text{oddsum} + \text{oddsum}) \mod 10\)

\(01660000070\)

odd-digit sum: \(0+6+0+0+0+0=6\)
evendigit sum: \(1+6+0+0+7=14\)
odd*3+even = \(6\times3+14=32\)
subtract from mult of 10 = \(40-32=8\)

- Lime juice: \(01660000070\rightarrow 016600000708\)
- Franks: \(04660000070\rightarrow 046600000705\)
- Choc Egg: \(05660000070\rightarrow 056600000704\)
- Salmon: \(08660000070\rightarrow 086600000701\)

**Some (Mod) Math**

- \(3 \times S_{\text{odd}} + S_{\text{even}} = 0 \mod 10\)
- The sum of the odd-position digits (times 3) plus the sum of the even position digits (including the check digit) is 0 mod 10.
- Modulo math is just like regular math, except things wrap around (like an odometer). Mod 10 means we only pay attention to the last digit in the number.
- Divide by 10 and only keep the remainder.
More Modulo Math

• What’s the check digit for the code 0-000000-000000?
• What happens to the check digit if you add one to an odd-position digit?
• What happens to the check digit if you add one to an even-position digit?

Bits

• We’ve gone from a product name to an 11-digit number to a 12-digit number. Next: bits.
• abcdefghijkl→101abcdef01010 ghijk101

Digits encoded as 7-bit patterns, chosen to be:
- start with 0, end with 1
- switch from 0 to 1 twice
- no reverse complements

0: 0001101 1: 0011001 2: 0010011 3: 0111011 4: 0100011
5: 0110001 6: 0101111 7: 0111011 8: 0110111 9: 0001011

Last 6 digits have 0s and 1s reversed!
How Many Bits?

• From our 12-digit number, how many bits (zeros and ones) long is the code now?

A. 84
B. 96
C. 100
D. 25
E. 16

Finally, Ink!

• Given the long pattern of bits, we write a 1 as a bar and a zero as a space.

• Two 1s in a row become a double wide bar.

• Two 0s in a row become a double wide space.

• Never have more than 4 in a row.

• All digits equal width.

• Starts and ends with bars.

• Can read upside down.

• Can read at an angle or variable speed via ratios.
Example

• Barcode for skin cream:

• 3-05210-04300-8 (8 is the check digit)

  start: 101; 3: 0111101
  05210: 0001101-0110001-0010011-0011001-0000110
  middle: 01010
  04300: 1110010-1011100-1000010-1110010-1110010 (rev)
  8: 1001000 (rev); end: 101

• The digits underneath are for our benefit.

Close Up

10101111010001101011000100100110011001000110101010111001010111001000010111001011100101001000101
Summary

- Product name turned to 11-digit code
- 11-digit code extended to 12 digits by adding a check digit
- 12 digits become a 95-bit sequence
- 95 bits are drawn in ink 1=black, 0=white

Reverse the process to get the product!
Circuits in Silicon

Xbox Chip Layout
Computer Design

- Computers have billions of parts.
- Repeated patterns (hierarchy/reduction) make it simple!
- Ideas transcend technology.
- “Imagination machine”.