Week 0 Lecture 1 — Computer Science 50
September 15, 2008 — Scribe Notes

Intro (1:15-1:23)

- Welcome to Computer Science 50! This course will be taught by David Malan.
- Students in this course are not expected to be computer science majors or to have any familiarity with computer science
- David himself took the course as a sophomore gov concentrator several years ago
- This course will teach you to think more algorithmically (methodically) in order to complete programming exercises
- Equipped with basic programming skills, you may begin to find applications of course material in your real life
- After taking CS51, for instance, David wrote Menuboy (to check the day's offerings at HUDS), Weatherboy, and, something that you might have heard of, Shuttleboy (shuttleboy.com)
- More recently, Shuttleboy has evolved into a text messaging service, and a website showing the movement of shuttles using GoogleMaps
- But in addition to hard skills (e.g., programming in C), you will find that exercises in this course will teach you how to solve problems more efficiently

Counting Demo (1:24-1:34)

- How can we efficiently count students in the room?
- Naïve way: Have one person stand in front of the lecture hall and point to each student once, saying "1, 2, 3, …" until each student has been counted.
- This will take as many steps as there are students, say, 400
- We can do better by counting in twos in just 200 steps. But even if we count by threes or fours, the number of steps will always be proportional to the number of students
- Can we employ an algorithm that increases in speed as we increase the number of students?
- We can count the number of students in the room by having each individual do the following:
    1. Stand up
    2. Assign yourself the number 1
    3. Find someone that is standing up.
    4. Add your number to that person's number. The total is your new number.
    5. One of you sit down.
    6. If you are still standing, go back to step 3.
    7. Algorithm finishes when there is one person standing. His number is the number of students in the class.
- Employing this algorithm, we find that there are about 400 students in the room
- How efficient was this?
- If there were 2 people, it would take one iteration of the above algorithm to count
- If there were 4 people, it would take two iterations
- Similarly, for n people, it would take $\log_2 n$ iterations
- For 400 people, as we have, it should have taken 9 iterations
- Another way to think about it is that each "round" cuts the number of students still standing in half. So if there are about 400 students, it will require 9 rounds because we can halve 400 nine times.
- If we graph this, we see that it takes a relatively long time for small numbers of students, but as the number of students grows, the algorithm seems to speed up
- For 4 billion students, it would take just 32 iterations!

Phone book Demo (1:34-1:37)

- Suppose we want to find the phone number of Drew Faust in a phone book
- Naïve way: begin at front of phone book and begin flipping pages until we get to the paint store section in the yellow pages.

- Drew Faust is probably about 1/4 of the way through the book, so this would take quite a lot of page-flipping (say 250 steps for a 1000 page phone book)
- Can we do better?
- Well, if we open up to the middle, say, M, we know Faust will be in the first half. Therefore, we can rip the phone book in half and literally throw away the entire second half. This has just cut the size of the problem in half.
- Next we can open the remaining chunk to the middle and rip it in half. Now, we know that the desired page is in the second half, so throw the first half out. Continue to repeat this process, each time retaining only the necessary half.
- As long we check correctly to make sure we keep the correct half, we must eventually arrive at the desired page
- In each round of this algorithm, we cut the size of the phone book in half. So if we start with 1024 pages, we will be down to a single page in 10 rounds. This algorithm takes just 10 steps.
- Compare the two algorithms: 250 steps vs. 10 steps. The second algorithm is much more efficient.
- As with the counting example, if there were 4 billion pages, we could find the desired phone number in 32 phone book rips
- Clearly, our second algorithm has huge payoffs in efficiency. In this class, we will not just solve problems, but strive to solve them *well*, i.e., efficiently.

Lolcats (1:37-1:39)

- The website now has a Lolcat of the day
- How are we doing this?
- Lolcats can be found at icanhascheezburger.com
- We can get the daily Lolcat from this page using the RSS feed
- Our website downloads the feed, analyzes it, looks for the link to the daily Lolcat, and puts it on the page

Who Is in This Course? (1:39-1:47)

- Last year's statistics: 30% "less comfortable", 20% more comfortable, 50% in between
- Even if you are "less comfortable", you can take solace in this clip from The Daily Show
- In other words, you actually know a lot more than you think, and even if you don't have any prior programming experience, you are not alone
- Last year's statistics: 43% of students had none, 47% had a little, 10% had a lot
- If you're still unsure whether you should take the course, check out a "Should I?" session (see course website for dates)

Counting in Binary (1:47-1:57)

- Everything in computers boils down to bits
- A bit is like a light bulb: it can be in an on state (represented by 1) or an off state (represented by a 0)
- We can use bits to count in binary
- When we count in decimal, each digit represents a multiplier for the corresponding power of 10
    - Example: $123 = 1 * 100 + 2 * 10 + 3 * 1 = 1 * 10^2 + 2* 10^1 + 3 * 10^0$
- Similarly, when we count in binary, each digit represents a multiplier for the corresponding power of 2
    - Example: $1101 = 1 * 8 + 1 * 4 + 0 * 2 + 1 * 1 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0$
- To figure out what a binary number is in decimal, think of columns as 128s, 64s, 32s, 16s, 8s, 4s, 2s, and 1s in the same way we normally have 1000s, 100s, 10s, 1s:

128s	64s	32s	16s	8s	4s	2s	1s
0	0	0	1	0	1	0	0

- Using this sort of notation, we can see that 00010100 is 20 in decimal.
- We can also use binary to encode letters using ASCII. Corresponding to each lower case letter, upper case letter, symbol, and number is a decimal number between 0 and 255.
- For instance, A is 65 and a is 97.

What is CS50? (1:57 – 2:05)

- Expectations, grades, etc. refer to the slides, syllabus, and course website.
- Course website: http://cs50.net
- There are no required books, but David recommends Absolute Beginner's Guide to C (2nd ed.) by Greg Perry for "those less comfortable," Programming in C (3rd ed.) by Stephen Kochan for "those more comfortable," and How computers Work by Ron White to everyone (and Hacker's Delight by Henry S. Warren Jr. for aspiring hackers).
- Lectures are on Mondays and Wednesdays, and a (optional) lecture ("Lunch with David") on Fridays (rsvp@cs50.net).
- We have podcasts and they are available in Flash, QuickTime, and mp3 formats. This class is also available from the Harvard Extension School.
- There are three types of sections: "those less comfortable," "those more comfortable," and those somewhere in between for students of different comfort levels.
- We have Office Hours (OH's) and Virtual Office Hours (VOH's) available for those who need help.
- Workload differs from students to students, but over 60% of the class (from last year) spends between 5-15 hours a week on problem sets.

Scratch (2:05 – 2:15)

- Scratch is a program that can be used to create simple videos and games. Check out some examples on the Lectures page of the website.
- We will be using Scratch for Problem Set 0, and only later dive into the C language. We have circuit boards for those who are interested in the "hacker edition" of Problem Set 0.
- Will be released on Wednesday.

Overview of the rest of the course (2:15 – 2:30)

- See syllabus for schedule.
- If you're not sure whether you want to take CS50, feel free to attend/bring questions to "Should I take CS50?" in Emerson 105 on Monday, Wednesday, or Thursday at 3:30-5pm.
- Welcome to CS50! ☺