CS 50: Introduction to Computer Science I
Harvard College
Fall 2007

Scribe Notes
Week 10 Monday
Anjuli Kannan

Some Notes on the Survey (0:00-9:00)

- Thanks for filling out the survey!
- In response to a widespread desire to spend more time with David, David will be having lunch with students a few times during the week of 12/3. Send an email to cs50+lunch@fas.harvard.edu if you are interested in attending.
- According to the survey responses, sophomores make up the largest percentage of the class, with 31%. Otherwise, it's about evenly distributed across the classes.
- The male:female ratio is about 7:3.
- About 30% of classified yourselves as less comfortable, 20% as more comfortable, and the rest somewhere in between.
- About half of you came in with no prior experience, half with some, and 10% with a good amount of programming experience.
- The majority of you are devoting 5-15 hours to problem sets, but about 20% take more than 20 hours and about 5% less than 5.

Other Announcements (9:00-12:30)

- Sign up for seminars, as listed on the course website.
- It's time to start thinking about the next computer science course you're going to take!
- (Remember, you don't have to be a CS concentrator to take a CS course!)
- Friday, December 7, will be a sneak preview of CS 51: Intro to Computer Science II (Radhika Nagpal). This course is commonly taken directly after CS 50.
- Monday, December 10, will be a sneak preview of CS 61: Systems Programming and Machine Organization (Matt Welsh). This is a new course.
- It's also time to start thinking about CS 50 '08! We will be recruiting for CS 50 teaching staff beginning in January. If interested, stay tuned for more info.

What's Coming Up (12:30-21:30)

- More and more software these days is moving onto the web. Consider, for instance, Google Docs.
- Going along with that trend, we'd like to introduce you to web programming.
- In addition, we want you to have the experience of learning new languages. After CS 50 or 51, you will probably not be taught many more languages formally. You will learn them on your own with the help of books and online tutorials.
- To prepare you for that, we will teach you the basics of some languages used for web development, but will expect you to teach yourself the rest using resources we recommend.
- You will be exposed to HTML, XHTML, CSS, PHP, and SQL.
- See the course website for some recommended readings on those languages.
- Before you ask us questions about these languages at office hours, RTFM
- If you still can't find an answer, post on the bulletin board. The bulletin board is a great way to find out how to do something in a particular language without waiting in line for two hours in the terminal room. You can get prompt responses from fellow classmates, TFs, and David (he's got an RSS feed). And in a class of 300, chances are someone has the same question as you.

So What Is the Internet? (21:30-32:00)

- Every computer on the internet speaks a language, or protocol, called TCP/IP.
- This means that every computer has an IP address that uniquely identifies it on the Internet.
- You can find your IP address by pulling up a command prompt and typing ipconfig
- When you pull up a webpage, you don't have a direct connection from you to the webpage you're viewing. The data gets to you via a series of routers.
- Routers are machines whose purpose is to route data from point A to point B
- The data knows where to go because of IP addresses.
- You can also run a command called traceroute followed by a website to see the series of routers required to get from your computer to that website
- You can think of the Internet as a physical infrastructure that allows data to get from one point to another.
- The World Wide Web is a service that lays on top of this infrastructure and makes use of it.
- When you type in a URL, your browser sends a server a request for that file. Then the server sends back the requested information.
- How do we make sure that everyone is communicating in the same way? Web servers and browsers follow a protocol called hypertext transfer protocol (HTTP) when they communicate.
- You can think of this as a sort of "language" used for browser-server communication.
- So your computer sends the server a request for a file. The file (or webpage) returned will be written in a language called HTML or XHTML
- If your browser requests the root (for instance, http://www.cnn.com/), your browser assumes that it wants the index.html page in the root directory
- We will learn how to write pages like index.html using XHTML

Our Very First Webpage (32:00-39:00)

- HTML is a markup language, not programming language. This means it doesn't have logic like if statements and for loops and functions. It is purely aesthetic.
- HTML allows you to do something like take a word foo and make it bold
- You do this using tags. Example: <b> foo </b> would print **foo**
- All of you have space for a webpage on FAS's service. Your webpage is located at http://www.people.fas.harvard.edu/~username/
- people.fas, courses.fas and nice.fas are different web servers but they all have access to the same data
- We will work with a dummy account cs50stud
- When you first go to http://www.people.fas.harvard.edu/~cs50stud/, you'll get an HTTP 404, meaning file not found
- This is because we have not yet made a file called index.html in cs50stud's directory
- First, we SSH to nice as cs50stud and go to the home directory, as we would to start a problem set or something
- Next, we type "chmod 711 ~" to make everything in the home directory readable to people besides cs50stud
- Then we make a folder called public_html. Everything in this directory will in the directory http://www.people.fas.harvard.edu/~cs50stud/ on the Internet. That is, if we make a file called

hello.html, it can be accessed at www.people.fas.harvard.edu/~cs50stud/hello.html

- But first let's make a homepage. Typing http://www.people.fas.harvard.edu/~cs50stud/ into Firefox fetches the page http://www.people.fas.harvard.edu/~cs50stud/index.html by default, so our first task is to make a file called index.html
- We do this, as usual, by typing "nano index.html" at the prompt
- Now we type this and save:

```
<!DOCTYPE html
      PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
      <head>
            <title>hello, world</title>
      </head>
      <body>
            Hello, world
      </body>
</html>
```

- Back at the command prompt, we make the files world-readable (more details on this in the problem set)
- When we pull this up in Firefox, it's just a page that says "hello, world" (you can view it here: http://www.courses.fas.harvard.edu/~cs50/lectures/weeks/10/src/hello.html)

How We Did That (39:00-51:00)

- OK, so how did we write that page?
- Step 1: when writing in XHTML, you must always put this at the top of your page:

```
<!DOCTYPE html
      PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- In order to apply a tag to some text, the text must go in between an opening and closing tag. The opening and closing tags are identical except that the closing one has a / in it.
- All of the text in our file will go in between the tags <html> and </html>
- A webpage has two parts: the head (the title and a few other things) and the body (most of the content). We indicate the head and body with the <head> and <body> tags
- Once you know these basics, you can pick up most everything else using Google or from looking at examples.
- To see a webpage's source code, right-click it in Firefox and say View Source.
- Within tags, we can assign attributes. For instance, we can change the background color of the webpage by setting its style attribute:

```
<body style="background:yellow;">
```

- Now we introduce a div, or paragraph, and change characteristics of our text by setting the align and style attributes of the div:

```
<div align=“center” style=“color: #00ff00; font-size: 72 pt;”>
      Hello for the second time!
</div>
```

- Now let’s add an animation. We find an appropriate picture on the Internet, save it onto our home computer as image.gif, and upload it cs50stud/public_html using SFTP
- We embed it using the <img> tag. We don’t really open and close the image tag. Instead, we combine them as follows:

```
<div>
      <img src=“image.gif” />
</div>
```

- Notice that we put the image in its own paragraph using the <div> tag again
- Usually, webpages have links to other pages. We can make a link using the anchor tag <a> and we can make a bulleted list of items using the unordered list tag <ul>.

```
<div>
      <ul style=“font-size: 24 pt;” />
            <li><a href=“http://cnn.com>CNN</a></li>
            <li>The Crimson</li>
      </ul>
</div>
```

- Notice that the whole list is delimited by the <ul> and </ul> tags, and each list item is delimited by the <li> and </li> tags.
- Text that we want to make into a link is delimited by the <a> and </a> tags, with the opening tag indicating the page that should be linked to.
- Check out the final product at http://www.courses.fas.harvard.edu/~cs50/lectures/weeks/10/src/index.html
- Right click and choose view source to see the final source code