CS 380
WEB PROGRAMMING

Instructor: Nat Martin
Class

- Instructor: Nat Martin
  - Office Hours: 3:30-4:30 Monday and Wednesday
  - Email: martin@cs.rochester.edu
- Lecture Monday and Wednesday (Dewey 2162)
  - 2:00-3:15
- Lab Tuesday and Thursday (MEL 210)
  - 12:30-1:45, 2:00-3:15, 3:15-4:40
- Textbook:
  - “Web Programming Step by Step”
Lectures (Monday and Wednesday)

- Dewy 2163 2:00-3:15 (Bring your laptops)
- Sit with your scrum team
  - Read before lecture (slides, book, and online material)
  - Discuss with your group at the beginning of class
  - When all questions are answered, there will be a quiz
  - Remaining time will be dedicated to team work
- Work will be done in class to simplify scheduling.
  - This means you need to come to class.
  - Each team will report attendance
Labs (Tuesday and Thursday)

- Mel 210
  - 12:30-1:45
  - 2:00-3:15
  - 3:25-4:40

- Time to work with your team on projects
  - Tomorrow: Assign teams
  - Tuesday, 20 January: No class
Academic Honesty

1. Give credit
   - Reference everything you get from elsewhere

2. Don’t cheat
   - Cheating is stealing from your classmates

- I will follow strictly the Universities guidelines for academic honesty.
Teams

- Development will be done using the Scrum model
  - Teams of five or six work together through the semester

- Each team will turn in one program per assignment
  - Advantage: you have a group to answer questions
  - Disadvantage: I have to assign individual grades

- Team assignments, projects and the grading of these will be published on the web
Agile

- Software changes continually
  - Software must always work
  - Requirements clarified during programming

Waterfall

- Software projects are completed
  - Software works when project is done
  - Requirements completed before programming starts
Scrum = Team focused Agile

- As in Rugby, the team moves to ball forward together

- Roles
  - Product Owner: represents the customer/user
  - Developers: write and test software
  - Scrum Master: coordinates with people outside team
Scrum Process

- **Product Backlog**: Work to be done
- **Sprint**: Work period (2-4 weeks)
- **Sprint Backlog**: Work for this sprint
- **Demo**: Working software inspected at end of sprint
Scrum Teamwork

- Daily Scrum: Team members state yesterday’s progress, today’s work and barriers
- Sprint planning meeting: Team clarifies and divides work
- Demo: Team shows working software generated by sprint.
Programming Project

- You can start working on this from the first week of the class
- Design and implementation of a professional website:
  - Professional Style
  - Interactive
- You will complete the project with your team
Homework and Programming projects will be posted online on the class webpage
<table>
<thead>
<tr>
<th>Date</th>
<th>Type</th>
<th>Topic</th>
<th>Reading</th>
<th>Assignments</th>
<th>Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wednesday, January 15, 2014</td>
<td>Lecture</td>
<td>Internet</td>
<td>Week 0: The Internet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday, January 16, 2014</td>
<td>Lab</td>
<td>Form Group</td>
<td>1 Intro</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday, January 20, 2014</td>
<td>Lecture</td>
<td>MLK Day</td>
<td>Week 1: HTML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday, January 21, 2014</td>
<td>Lab</td>
<td>No Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday, January 22, 2014</td>
<td>Lecture</td>
<td>Basic HTML</td>
<td>2 HTML</td>
<td></td>
<td>AWS</td>
</tr>
<tr>
<td>Thursday, January 23, 2014</td>
<td>Lab</td>
<td>Assignment 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday, January 27, 2014</td>
<td>Lecture</td>
<td>Basic CSS</td>
<td>3 CSS 1</td>
<td></td>
<td>Github</td>
</tr>
<tr>
<td>Tuesday, January 28, 2014</td>
<td>Lab</td>
<td>Assignment 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday, January 29, 2014</td>
<td>Lecture</td>
<td>More CSS</td>
<td>4 CSS 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday, January 30, 2014</td>
<td>Lab</td>
<td>Assignment 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday, February 3, 2014</td>
<td>Lecture</td>
<td>Float</td>
<td>5 PHP 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday, February 4, 2014</td>
<td>Lab</td>
<td>Assignment 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday, February 5, 2014</td>
<td>Lecture</td>
<td>PHP 1</td>
<td>6 PHP 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday, February 6, 2014</td>
<td>Lab</td>
<td>Assignment 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday, February 10, 2014</td>
<td>Lecture</td>
<td>PHP 2</td>
<td>7 PHP 3 Files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday, February 11, 2014</td>
<td>Lab</td>
<td>Assignment 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday, February 12, 2014</td>
<td>Lecture</td>
<td>PHP 3</td>
<td>8 PHP 3 Forms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friday, February 14, 2014</td>
<td>Lab</td>
<td>Assignment 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday, February 17, 2014</td>
<td>Lecture</td>
<td>PHP 2</td>
<td>Week 5: Programming Interlude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday, February 18, 2014</td>
<td>Lab</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday, February 19, 2014</td>
<td>Lecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday, February 20, 2014</td>
<td>Lab</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday, February 24, 2014</td>
<td>Lecture</td>
<td></td>
<td>Week 6:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday, February 25, 2014</td>
<td>Lab</td>
<td>Demos</td>
<td>9 PHP 4 Forms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday, February 26, 2014</td>
<td>Lecture</td>
<td></td>
<td>10 PHP 5 Forms 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday, February 27, 2014</td>
<td>Lab</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The INTERNET... and a bit of history
What is the internet?

- A “series of tubes”
- How many Internets are out there?
- Is Google one of them?
What is the internet?

- A collection of computer networks that use a protocol to exchange data
- Is the World Wide Web (WWW) and the internet the same?
Brief history

- Began as a US Department of Defense network called ARPANET (1960s-70s)
- Packet switching (in the 60s)
- E-mail is born on 1971
- TCP/IP beginning on 1974 (Vinton Cerf)
- USENET (1979)
- By 1987: Internet includes nearly 30,000 hosts
Brief history (cont.)

- WWW created in 1989-91 by Tim Berners-Lee
- Popular web browsers released:
  - Netscape 1994
  - IE 1995
- Amazon.com opens in 1995
- Google January 1996
- Wikipedia launched in 2001
- MySpace opens in 2003
- Facebook February 2004
Key aspects of the internet

- Sub-networks are independent
- Computers can dynamically join and leave the network
- Built on open standards
- Lack of centralized control (mostly)
- Everyone can use it with simple, commonly available software
People and organizations

- Internet Engineering Task Force (IETF): internet protocol standards
- Internet Corporation for Assigned Names and Numbers (ICANN): decides top-level domain names
- World Wide Web Consortium (W3C): web standards
Internet Protocol (IP)

- Simple protocol for data exchange between computers
- IP Addresses:
  - 32-bit for IPv5
  - 128-bit for IPv6
Transmission Control Protocol (TCP)

- Adds multiplexing, guaranteed message delivery on top of IP
- Multiplexing: multiple programs using the same IP address
- Port: a number given to each program or service
  - port 80: web browser (port 443 for secure browsing)
  - port 25: email
  - port 22: ssh
- Some programs (games, streaming media programs) use simpler UDP protocol instead of TCP
Web Servers

- Web server: software that listens for web page requests
  - Apache
  - Microsoft Internet Information Server (IIS)
Web Browser

- Web browser: fetches/displays documents from web servers
  - Mozilla Firefox
  - Microsoft Internet Explorer (IE)
  - Apple Safari
  - Google Chrome
  - Opera
Domain Name Server (DNS)

- Set of servers that map written names to IP addresses
  - Example: ju.edu → 204.29.160.73

- Many systems maintain a local cache called a hosts file
  - Windows: C:\Windows\system32\drivers\etc\hosts
  - Mac: /private/etc/hosts
  - Linux: /etc/hosts
Uniform Resource Locator (URL)

- Identifier for the location of a document on a website
  - Example: http://dept.ju.edu/cs/index.html

- Upon entering this URL into the browser, it would:
  - ask the DNS server for the IP address of dept.ju.edu
  - connect to that IP address at port 80
  - ask the server to GET /cs/index.html
  - display the resulting page on the screen
Hypertext Transport Protocol (HTTP)

- Set of commands understood by a web server and sent from a browser

- Some HTTP commands (your browser sends these internally):
  - GET filename: download
  - POST filename: send a web form response
  - PUT filename: upload

- Exercise: simulate a browser with a terminal window
### HTTP Error Codes

- When something goes wrong, the web server returns a special "error code" number.

- **Common error codes:**

<table>
<thead>
<tr>
<th>Number</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
</tr>
<tr>
<td>301-303</td>
<td>page has moved (permanently or temporarily)</td>
</tr>
<tr>
<td>403</td>
<td>you are forbidden to access this page</td>
</tr>
<tr>
<td>404</td>
<td>page not found</td>
</tr>
<tr>
<td>500</td>
<td>internal server error</td>
</tr>
</tbody>
</table>
Internet Media ("MIME") types

<table>
<thead>
<tr>
<th>MIME type</th>
<th>file extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>text/html</td>
<td>.html</td>
</tr>
<tr>
<td>text/plain</td>
<td>.txt</td>
</tr>
<tr>
<td>image/gif</td>
<td>.gif</td>
</tr>
<tr>
<td>image/jpeg</td>
<td>.jpg</td>
</tr>
<tr>
<td>video/quicktime</td>
<td>.mov</td>
</tr>
<tr>
<td>application/octet-stream</td>
<td>.exe</td>
</tr>
</tbody>
</table>
Web Languages

- Hypertext Markup Language (HTML): used for writing web pages
- Cascading Style Sheets (CSS): stylistic info for web pages
- PHP Hypertext Processor (PHP): dynamically create pages on a web server
- JavaScript: interactive and programmable web pages
Web Languages (cont.)

- Asynchronous JavaScript and XML (Ajax): accessing data for web applications
- eXtensible Markup Language (XML): metalanguage for organizing data