Web Cite Design CSc 473

Syllabus Spring 2011

Instructor: Michael D. Grossberg
Email: grossberg@cs.engr.ccny.cuny.edu
Office: NA 7311
Office Phone: 212-650-6166
Office Hours: T 3-4pm, W 10:30-11:30am
Website: https://134.74.146.39/moodle/course/view.php?id=2

Course Meetings

Location: NA-5123
Time: T,TH 5:00-06:15PM

Course Description

From Bulletin:

The design and implementation of web sites from a Human-Computer Interaction viewpoint, with emphasis on user testing. Navigation design. Accessibility by persons with limitations in vision or motor ability is stressed and must be addressed in the final project.

This course will assume some programming ability. If you have never programmed before, you will need to allocate time for self-study. This is a CS course, not a graphic design course so there will be a significant component of server side programming. Some people assume that web programming is "easier" than application programming. It is harder. You are expected to use many technologies such as HTML, CSS, Javascript, Python, XML, YAML and many APIs. Not covered but also used technologies include Java, PHP, Ruby on Rails, Flex/Flash, Silverlight/.Net, and SQL. Lectures can only provide superficial overviews of the some of the technologies you will need. You will be expected to do significant independent study. Note: while challenging, comfort with these skills continues to be in great demand in the workplace.

Collaboration and Academic Integrity

It is acceptable, even encouraged to form study groups and collaborate in understanding homework problems, and preparing for exams. However, all the work on homework and
exams should be your own work exclusively. It is very easy to test for duplicate code. Some students do not realize that professors can use Google too! Checking that two files have similarities is easily done with unix tools like grep. Cheating will not be tolerated. For group projects, the work must be done exclusively by members of the group.

Cuny Policy on Academic Integrity

Grades

The components that will determine the grade are

<table>
<thead>
<tr>
<th>Weight</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>20%</td>
<td>Homeworks</td>
</tr>
<tr>
<td>15%</td>
<td>Midterm</td>
</tr>
<tr>
<td>40%</td>
<td>Project</td>
</tr>
<tr>
<td>25%</td>
<td>Final</td>
</tr>
</tbody>
</table>

Homework

Homework will be submitted online. Do not send by email it may be lost.

Midterm/Final

Will consist of some questions which test the material by direct questions on languages or technology, and questions which ask you to write code or web pages (at a computer).

Project

You will participate in a group project to build a functioning live web site. Your participation in this project will be measured by the number and quality of the code/documentation you write as measured by the version control system. If it is not clear from the checkins that you contributed it will be assumed your team-mates did all the work.

Textbook

There is no official textbook for the course. The following are references for the course:

Version Control:
- **Mercurial: The Definitive Guide**
- **hg tip** Learn Mercurial one bite-sized tip at a time
- **Mercurial for Beginners: The Definitive Practical Guide**
- **Tutorial on Using Mercurial**
- **Hg Init: a Mercurial tutorial**

**HTML/HTML5:**

- **W3Schools HTML**
- **W3Schools HTML5**
- **Dive Into HTML5**
- **HTML5 Rocks**
- **Google HTML Video Tutorial**

**CSS/CSS3**

- **W3Schools CSS**
- **W3Schools CSS3**
- **The 30 CSS Selectors you Must Memorize**
- **Mastering CSS, Part 1, Styling Design Elements**
- **Mastering CSS, Part 2, Advanced Techniques and Tools**
- **Google CSS Video Tutorial**

**Python**

- **Google's Python Class**
- **Understanding Python**
- **How to Think Like a Computer Scientist - 2nd edition**
- **Python for Non-Programmers**
- **Dive into Python**
- **Python for Programmers**

**Google Technologies**

- **Google App Engine**
- **Python Runtime**
- **Video: Google Spreadsheet as DB**
- **Periodic Table of Google APIs**

**Other Technologies**

- **A ReStructuredText Primer**
- **Python Sphinx**
- **Wordpress Documentation**
- **Python Twill**
- **BeautifulSoup**
- **Python Bottle**
- **Django**
- **Pinax**
- **Mako Templates for Python**
- **W3Schools Jquery**

**Magazines and Articles**

- **Learn the Basics of Design This Weekend**
- **Smashing Network**
- **Design Float**
- **DZONE**
- **A List Apart**

**Imaginary Plan**

Tentatively this is the plan of the lectures

1. Intro
2. Version Control/HTML/Bitbucket
3. Python/Test/HTML
4. HTML
5. CSS
6. Static Page Site: Sphinx
7. CMS/Wordpress/Tumblr (drupal/alfresco)
8. Testing Twill
9. Bottle Web Framework
10. Google App Engine (using bottle)
11. Web templates Mako
12. Midterm
13. Data Google Docs (spreadsheet)
14. Google App Data Model
15. Graphic Design Concepts
16. Layout
17. Type
18. Color
19. Vector Art
20. Raster Art
21. Usability I
22. Usability II
23. Accessibility
24. Javascript
25. AJAX/Jquery
26. Web APIs
27. Your Projects
28. Final