Web Programming, CSC 435, Spring 2014

Instructor: Prof. Bei Xiao, American University.

Contact: bxiao@american.edu

Time: Mon/Thurs 4-515pm

Location: SCAN (Sports Center Annex) 157

Office hours: Wed 4-6pm, Thursday 11-noon or by appointment.

Textbooks: We will **not** follow any particular textbooks, but the following are good references:
1. Learning web design, 4th Edition. Jennifer Niederst Robbins
2. Web Standards: Programmer's Reference, Steven M. Schafer, Wiley Publishing, 2005.
3. Teach Yourself Javascript in 24 Hours, Michael Moncur, Sams Publishing, 2007.
4. How to Think Like a Computer Scientist: Learning Python
5. Learning SQL, Alan Beautieu, O'Reilly Media Inc., 2005.

Prerequisite: Introduction to Computer Science 1-2 (CSC 280).

Grading: 50% Assignments, 15% Mid-term exam, 25% Final Project, 10% Class participation.

Grading Scale listed below:

93-100	A
90-92	A-
87-89	B+
83-86	B
80-82	B-
77-79	C+
73-76	C
70-72	C-
60-69	D
0-59	F

Course description:
The Web is intertwined with our lives. No mater if you decide to be a politician, a musician, a scientist, an artist, a doctor, or a journalist, or purely an interesting person, web design is an extremely useful skill. Web development is useful in advertising (personal homepage), entrepreneurship, (web API), journalism (news media and blog), multi-media (flickr), social networks (facebook), e-commerce (Amazon), user experiences and crowd-sourcing (Amazon Mechanical Turk), online games (Phatasy star online). This course presents the fundamental technologies behind the Web, as well as techniques for designing, developing, and evaluating Web-based applications.

Topics will include HTML, CSS, Dynamic HTML and JavaScript, HTML forms and CGI scripting, Python, database SQL, and dynamic web programming using Ajax (given time). The goal is to have a basic understanding of the web environment and be able to build dynamic and interactive websites that serve flexible purposes. We will use JavaScript as the client side coding tool and Python as the major server side-coding tool. But we will discuss other server-side programming options.

Learning outcomes: This is a technical and programming class. So the learning outcome is the acquired skill. First of all, I wish you all have fun making webpage and understand basic principles of making the website serve your purpose! Learning web programming is very much like learning a

foreign language in an introductory level. So you are expected to be able at least use this language in a daily conversation. Can you communicate with the users? Can you interact with other data sources and websites? The exact proficiency depends on how much work you put in!

Computers and Software:

Students should come to class with their laptops and should have installed a good browser (Chrome or Firefox+Firebug) and Python. Note Python is automatically installed on Mac OS. Download a text editor of your choice such as TextWragelar, TextMate 2 and Sublime Text 2 for text editors. My favorite is Sublime Text 2. You do not need to upload your HTML or JavaScript file to see your webpage. But at some point, we will talk about testing your webpage on the Internet by uploading your files on to the web server.

Attendance Policy: I will do lots of live programming, discussion, and quiz in class. Class attendance is strongly recommended.

Policy of collaborative work: No copy lines of code from each other! Only high-level discussion is allowed.

Late Policy: Most assignments will be submitted electronically on Blackboard, and will be due at 11:59 pm on the date given in the assignment. Assignments must be submitted by the due date to receive full credit, unless I approve prior arrangements. Projects and homework received after the deadline will incur a 50% lateness penalty. No work received after the beginning of the course's scheduled final exam will be accepted for credit.

Academic Integrity
Plagiarism and academic misconduct are defined in the University Academic Integrity Code. You should be familiar with what constitutes academic dishonesty. In particular, you should observe the following rules: only high-level discussions are allowed (i.e., not relating to a single line of code), and you have to declare whom you discussed with.

Course Schedule (This is only tentative, Please refer to course website for updated information):

Course Schedule

Date	Topic	Reading	Slides	Homework	
Mon 1.13	Introduction: Internet and the web.	The internet Chapter 1, Robbins	Lecture1		
Thu 1.16	Web servers, some basic concepts, start with HTML	Chapter 2, Robbins	Lecture 2	Homework 1 Out	
Thu 1.23	Basics of HTML	Chapter 3, Robbins Chapter 1, Schafer	Lecture 3		

		Schafer Exercise online: [HTML basic exercise track 1-2](#)			
Mon 1.27	HTML continue: text, links, images, lists.	Chapter 4, 5, Robbins	Lecture 4	Homework 1 due.	
Thur 1.30	HTML continue: table, forms, plugins	Chapter 6, Robbins	Lecture 5	Homework 2 out	
Mon 2.3	CSS basics		Lecture 6		
Thur 2.6	CSS		Lecture 7	Homework 2 due: Styling your wegpage	
Mon 2.10	Advanced CSS: Review & Questions, Styles of webpages		Lecture 8	Homework 3 Out	
Thur 2.13	JavaScript		Lecture 9		
Monday 2.17	JavaScript		Lecture 10		
Thur 2.20	JavaScript and Dynamic HTML	Chapter 22, Shafer	Lecture 10	Homework 3 due Collect user responses	
Mon 2.24	JQuery user interface		Lecture 12		
Thur 2.27	JavaScript and HTML5		Lecture 13		

Mon 3.03	Intro to CGI		Lecture 14		
Thur 3.06	*Mid-term Exam*				
Mon 3.17	Intro to Python		Lecture 15		
Thur 3.20	Python Continues				
Mon 3.24	*Python and CGI*				
Thur 3.27	MySQL				
Mon 3.31	MySQL Continues				
Thur 4.03	Python and MySQL				
Mon 4.07	Ajax				
Thur 4.10	Node.js				
Mon 4.14	Node.js		Final Project Out (Dynamic web design)		
Thur 4.17	Guest lecture: A word on User Interface Design				
Mon 4.21	Amazon Mechanical Turks				

Thur 4.24	*Possible Guest Lecture: Web App*				
Mon 4.28	Guest lecture TBA				
Mon 5.01	*Final project presentations*				
Mon 5.05	*Final Project presentations*				