CMSI 270-01

Modern Web Apps

http://myweb.lmu.edu/dondi/fall2011/cmsi270

Fall 2011 — Doolan 222
TR 9:25–10:40am, 3 semester hours
Office Hours: TR 11am–12nn, 3–4pm, or by appointment

John David N. Dionisio, PhD
e-mail: *dondi@lmu.edu,* AIM: *dondi2LMU*
Doolan 106; (310) 338-5782

Course Objectives

This course explores the technologies involved in modern web applications and examines how they are built, from a source code and language level. It seeks to provide non-majors with a glimpse of how real-world software is constructed, using technologies that they see and use everyday. Computer science majors are exposed to a specific set of languages and constructs to which they can apply the concepts and first principles seen in other computer science courses. Long after the course concludes, my hope is that you will:

1. **Write fun, cool, working code in the languages used by modern web browsers.**
2. **Know the major concepts and technologies that comprise modern web applications.**
3. **Understand the key principles and concepts involved in the effective development of such applications.**

There are no prerequisites to this course.

Materials and Texts

Assorted handouts, articles, and sample code will be distributed throughout the semester. There are no required texts.

Course Work and Grading

This course uses standards-based grading: your proficiency in each course objective is directly evaluated according to the outcomes shown on page 3 of this syllabus. Proficiency is measured according to the following key:

+	Advanced proficiency
|	Appropriate proficiency
/	Approaching appropriate proficiency
–	Needs practice and support
O	Not yet evaluated

Your submitted work is used as the basis for evaluating these outcomes (see below). Letter grades are then assigned as follows:

	+	|	/	–
A	**many**		*none*	*none*
B		**many**	*none*	*none*
C			**some**	*none*
D				**some**
F				**many**

A–, B+, B–, C+, and C– grades may be assigned based on qualitative considerations such as degree of difficulty, effort, class participation, time constraints, and overall attitude throughout the course. You may inquire at any time about the proficiency measures that I currently have on record for you.

Homework

Homework consists of questions, exercises, and programming assignments to be given throughout the semester. Homework is where you can learn from your mistakes without penalty. It is meant to *develop* proficiency, not *demonstrate* it. Thus, homework is optional.

Of course, with great flexibility comes great accountability. By doing homework (and paying attention to the feedback I provide), you will know precisely how you are doing in class. This is because feedback on submitted homework is standards-based: it is evaluated according to the same proficiency scale as your final grade. *Students who submit homework receive a clear, up-to-date picture of their current strengths and weaknesses.* By submitting homework, you will know what you need to work on, and what you have already learned sufficiently.

To receive this feedback in a timely fashion, submit your homework by the designated deadline. This deadline is always the beginning of class on the designated due date; the due date is encoded in the homework number. Quid pro quo: submissions after the deadline may eventually get feedback, but its timing is not guaranteed.

Term Portfolio

Your accumulated homework for the semester comprises the *term portfolio* — the final, definitive artifact that demonstrates the proficiencies you have reached for each course outcome. The term portfolio provides you with an opportunity to improve upon the homework submitted throughout the semester; it is how you show that you learned from your mistakes or improved upon already established knowledge.

After receiving feedback on your homework, you are encouraged to improve your work based on that feedback, and show it to me for re-evaluation. Improvements in proficiency are recorded and give you a good idea of how your term portfolio will fare long before its final version is submitted.

The final version of your term portfolio is due on December 15. Late portfolios will not be accepted.

Extra Credit

In terms of standards-based grading, "extra credit" takes on a different meaning: it indicates work that, if successfully performed, would indicate advanced proficiency (+). Extra credit tasks may be assigned for either homework or the term portfolio. Accomplish them successfully to rack up those +'s. You do not need to perform extra credit work to show advanced proficiency; it merely demonstrates such proficiency more readily.

Attendance

Attendance at all sessions is expected, but not absolutely required. If you must miss one or more class sessions, it is your responsibility to keep up with the course. The last day to add or drop a class without a grade of W is September 2. The withdrawal or credit/no-credit deadline is November 4.

University Policy on Academic Honesty

Loyola Marymount University expects high standards of honesty and integrity from all members of its community. All students are expected to follow the LMU Honor Code and Process, as stated in the *LMU Undergraduate Bulletin*.

Americans with Disabilities Act

Students with special needs as addressed by the Americans with Disabilities Act who need reasonable modifications, special assistance, or accommodations in this course should promptly direct their request to the Disability Support Services (DSS) Office. Any student who currently has a documented disability (physical, learning, or psychological) needing academic accommodations should contact DSS (Daum Hall, Room 224, x84535) as early in the semester as possible. All discussions will remain confidential. Please visit *http://www.lmu.edu/dss* for additional information.

Topics and Important Dates

Correlated outcomes are shown for each topic. Specifics may change as the course progresses. University dates (italicized) are less likely to change.

September	Introduction to modern web applications, architecture, and concepts *(2a–2f)*; introduction to HTML and CSS *(1a–1d, 3a–3c, 3e–3f)*; operational aspects of web application development *(1b–1j, 2g–2h)*
September 2	*Last day to add or drop a class without a grade of W*
October	Intermediate HTML and CSS *(1a–1d, 3a–3c, 3e–3f)*; introduction to JavaScript *(1e–1g, 3d–3f)*; introduction to Ajax *(1f)*
November	Advanced HTML and CSS *(1a–1d, 3a–3c, 3e–3f)*; intermediate JavaScript *(1e–1g, 3d–3f)*
November 4	*Withdraw/credit/no-credit deadline*
November 23–25	*Thanksgiving; no class*
December	Portfolio improvement workshops *(1a–3f)*; miscellaneous topics *(varies)*
December 15	*Term portfolios due*

You can view my class calendar in any iCalendar-savvy client such as Google Calendar or Apple iCal by subscribing to:

webcal://www.me.com/ca/sharesubscribe/1.9392690/M2CD-5-1-5B14D70A-E341-4026-A665-D391D97E01B8.ics

(I know, it's ugly; the link is also available on my web site for convenience.)

If necessary, this syllabus and its contents are subject to revision. Students are responsible for any changes or modifications announced in class.

Course Outcomes

1	**Write fun, cool, working code in the languages used by modern web browsers.**
1a	*Write valid HTML structure and markup that complies with the latest web standards.*
1b	*Use interactive HTML elements for dynamic web page functionality.*
1c	*Write valid CSS selectors and properties.*
1d	*Use elements, IDs, classes, relationships, and filters in CSS selectors.*
1e	*Write correct JavaScript behaviors and callbacks.*
1f	*Include and use JavaScript libraries such as jQuery and jQuery UI.*
1g	*Encapsulate JavaScript code in modules that do not pollute the overall JavaScript namespace.*
1h	*Use developer tools to see the structure and behavior of a web application.*
1i	*Use developer tools to try out experimental code.*
1j	*Use available resources and documentation to find the markup, properties, or functions that deliver the desired application feature or capability.*
2	**Know the major concepts and technologies that comprise modern web applications.**
2a	*Know the HTTP request and response cycle, and the technologies involved at each step of that cycle.*
2b	*Know the difference between HTTP and HTTPS.*
2c	*Be aware of how different web browsers perform in the Acid1, Acid2, and Acid3 compliance tests.*
2d	*Know the parts of a URL.*
2e	*Interpret absolute and relative URLs correctly.*
2f	*Differentiate between native browser functionality and plug-ins.*
2g	*Locate, examine, and modify (when permitted) the distinct assets that comprise a web application.*
2h	*Transfer web application files and assets between your personal computer and a web server.*
3	**Understand the key principles and concepts involved in the effective development of such applications.**
3a	*Organize the files and assets of a web application.*
3b	*Demonstrate proper separation of concerns in web apps that you author (i.e., properly separated HTML, CSS, and JavaScript files or elements).*
3c	*Identify and fix HTML and CSS validation errors as enforced by the W3C Validator.*
3d	*Follow JavaScript coding style guidelines as enforced by JSLint.*
3e	*Provide clear, appropriate inline documentation (i.e., comments).*
3f	*Write code that is properly indented and spaced for human readability.*

Sample Standards-Based Evaluation

Based on these proficiencies, the student will get a C+.

1	**Write fun, cool, working code in the languages used by modern web browsers.**	
1a	*Write valid HTML structure and markup that complies with the latest web standards.*	+
1b	*Use interactive HTML elements for dynamic web page functionality.*	+
1c	*Write valid CSS selectors and properties.*	|
1d	*Use elements, IDs, classes, relationships, and filters in CSS selectors.*	|
1e	*Write correct JavaScript behaviors and callbacks.*	|
1f	*Include and use JavaScript libraries such as jQuery and jQuery UI.*	/
1g	*Encapsulate JavaScript code in modules that do not pollute the overall JavaScript namespace.*	/
1h	*Use developer tools to see the structure and behavior of a web application.*	|
1i	*Use developer tools to try out experimental code.*	|
1j	*Use available resources and documentation to find the markup, properties, or functions that deliver the desired application feature or capability.*	|
2	**Know the major concepts and technologies that comprise modern web applications.**	
2a	*Know the HTTP request and response cycle, and the technologies involved at each step of that cycle.*	|
2b	*Know the difference between HTTP and HTTPS.*	|
2c	*Be aware of how different web browsers perform in the Acid1, Acid2, and Acid3 compliance tests.*	+
2d	*Know the parts of a URL.*	|
2e	*Interpret absolute and relative URLs correctly.*	/
2f	*Differentiate between native browser functionality and plug-ins.*	|
2g	*Locate, examine, and modify (when permitted) the distinct assets that comprise a web application.*	|
2h	*Transfer web application files and assets between your personal computer and a web server.*	|
3	**Understand the key principles and concepts involved in the effective development of such applications.**	
3a	*Organize the files and assets of a web application.*	|
3b	*Demonstrate proper separation of concerns in web apps that you author (i.e., properly separated HTML, CSS, and JavaScript files or elements).*	/
3c	*Identify and fix HTML and CSS validation errors as enforced by the W3C Validator.*	+
3d	*Follow JavaScript coding style guidelines as enforced by JSLint.*	+
3e	*Provide clear, appropriate inline documentation (i.e., comments).*	/
3f	*Write code that is properly indented and spaced for human readability.*	/