Instructor

Douglas K. Lay
Email: douglay@umd.edu
Phone: 410-302-3562 (cell)
Office Hours: Tuesday and Thursday evenings 7-8 PM EST (Conferences tab in ELMS) or by appointment.

Catalog Description

Non-programmers will learn basic programming and how to develop familiarity with web formatting and programming paradigms, including XML, REST, APIs, and authentication schemes. The class begins with an introduction to basic programming and students build on those skills by programming applications that use web-based data and services.

Prerequisites: None. All programming concepts will be introduced as needed.

Extended Course Description

This course will be divided into two parts, approximately divided by our spring break in mid-March

The first part of the course will focus almost entirely on the client-side part of the Web—that is, the technologies that are available within a Web browser. These core technologies include:

- Hypertext Markup Language (HTML): used to provide structure to Web pages.
- Cascading Style Sheets (CSS): used to provide design to Web pages.
- Javascript: used to provide interactivity to Web pages.

Our introduction to Javascript will cover several weeks and will include an introduction to fundamental concepts that appear in nearly all programming languages including variables, conditional expressions, loops and functions. For students already familiar with these concepts some optional intermediate-to-advanced Javascript concepts will be presented.

In the second part of the course we will switch focus to the server side of Web applications, and the mechanisms used to communicate between Web servers and browser clients. We will use the PHP scripting language for server-side programming, as this language is widely supported, free
to use, and has a relatively simple learning curve. Students will be provided with cloud-based visual machines with PHP installed.

Key concepts introduced in this part of the course will include:

- In-depth understanding of the Hypertext Transfer Protocol
- Connecting Web applications to central data stores
- User authentication best practices
- Using AJAX to communicate with Web servers without reloading entire Web pages
- Web API design including REST concepts
- Web security.

For final projects, students will build Web applications that incorporate the client-side and server-side technologies introduced in the course. Final projects will be done individually by students and will be presented during the final week of class.

Student Learning Outcomes

This course will not, by itself, turn students into professional-level Web developers. However, upon successful completion of the course, students should expect to:

- Have a high-level understanding of fundamental Web technologies including HTML, HTTP, CSS, Javascript, AJAX, REST and server-side scripting.
- Understand foundational programming concepts like variables, conditional logic, iteration and functions.
- Be able to build simple but useful real-world Web applications, including applications that integrate with centralized data stores.
- Have a good ability to communicate with Web programming professionals about their work, and be able to make informed decisions regarding selection of Web technologies and architectures.

Technology Requirements

Students will need a laptop or desktop computer with reliable Internet access. Software needed will include:

- The Google Chrome Web browser. Students will learn how to use the developer tools in Chrome to debug Web applications. Chrome is available free of charge for all major operating systems.
- At least one alternate Web browser, such as Mozilla Firefox, Safari (Mac only) or Edge (Windows only). Web applications have to run successfully across a range of browsers so
students should have access to multiple browsers for testing purposes. Note that all of the above browsers can be downloaded free of charge, and some are installed by default by operating system vendors.

- A text editor. Students will need to use a text editor to complete the assignments, and will be expected to have obtained a text editor by the end of the second week of class and begun to learn how to use it. Many excellent text editor options are available for all major operating systems. The instructor uses Atom (https://atom.io) which is available free of charge. Many Web developers prefer Sublime Text (https://www.sublimetext.com/) which is not free but offers an extended evaluation period. Other options are described at this link: https://www.creativebloq.com/advice/best-code-editors

- (Optional) A graphical FTP client. Students will need to transfer files between their own computers and remote Web servers. This can be accomplished using command-line tools such as `ftp` and `scp`, and it is a good idea for students to understand how to do this. However, students may prefer to use a graphical program to transfer files, and some good options are available, including FileZilla (https://filezilla-project.org/ — only the client is needed).

Students will be provided with access to cloud-hosted virtual machines for the server-side programming exercise in the second part of the course, and for use in developing their final projects. These VMs should be available by early March.

## Course Structure

This course is being conducted online and the majority of interaction between course members will be asynchronous.

The course will include the following components:

- Readings (possibly also including videos)
- Discussions
- Assignments (including final projects)
- Exams
- Conferences (including office hours)

These components are described in more detail below.

## Readings

Readings will be provided for each week’s course material. Some readings will be marked as mandatory. Students are expected to thoroughly read and understand all mandatory readings. Any material in these readings will be fair game for inclusion on the two course midterm exams. **All mandatory readings will be from sources freely available on the Internet.**
Some of the non-mandatory readings will cover intermediate-to-advanced topics that are beyond the scope of the class but would be things a professional Web developer would be expected to know about. Students can decide for themselves whether to investigate these topics.

**Readings, in general, will be posted in ELMS a full week before the topic in question is discussed.** For instance, readings for Week 3, covering Cascading Style Sheets, will be posted by the beginning of Week 2.

**Discussions**

Each week, two new discussion items will be posted in ELMS, related to that week’s subject matter. These discussion items will most often be in the form of questions. Students are expected to take part in all discussions with at least one substantive posting in each discussion thread. Discussion participation will count substantially toward students’ final grades.

**Discussion items will be posted by the beginning of each week.** For instance, discussion items related to Week 2 will be posted by 12:00 AM Monday Feb 2nd. Students must submit discussion postings during the week a discussion is active. No credit will be given for late discussion postings.

**Assignments**

There will be four graded programming assignments over the duration of the course. Assignment due dates are listed in the schedule below. Students are expected to complete assignments on time. If students are unable to complete assignments on time they may submit partial work for partial credit or may elect to submit an assignment up to one week late, incurring a 10% penalty on their grade. Assignments more than one week late will receive no credit at all.

**Graded assignments will be posted three weeks before they are due.** For instance, Assignment 1, due at the end of Week 4, will be posted by the end of Week 1 (Monday 2/2).

Students will also complete final projects, which will be substantial programming projects covering most of the concepts presented in the course. These projects will be due at the end of the final day of class (May 12). Students will also need to set up a time to present their final project to the instructor (and hopefully to other class members as well). These presentations should be about 20 minutes in length and must occur on or before May 12.

**Requirements for the final project will be presented in depth at the beginning of Week 8 (3/23).**

Some weeks will include ungraded assignments. Students are not required to complete these ungraded assignments during a given week; however it is highly advisable they do so, since most of these ungraded assignments are necessary building blocks toward completion of the graded
assignments or the final project. Ungraded assignments for a week will be provided along with readings for that week.

Students are expected to work individually on all assignments including the final project. Using the Internet or class discussion forums to ask questions about specific programming details is acceptable, but if there is evidence that a student wholesale copied an assignment from another student or an outside resource, the instructor will pursue this as a violation of academic integrity.

Exams

There are two midterm exams in the class. Each exam will be conducted through ELMS. Exams will be made available at the beginning of the week they are due and must be completed by the end of that week. Students may choose the time at which to begin the exam but once they begin, must have the exam completed within 90 minutes. Students may consult their own notes, course reading materials or general internet sources while answering exam questions. However, if there is evidence students collaborated with each other or simply copied answers from Internet sources, the instructor will pursue this as a violation of academic integrity.

Conferences

The instructor will be available for office hours at least twice per week, through the Conferences tool in ELMS. Owing to the instructor’s schedule constraints, office hours will need to be conducted in the evenings and initially will be scheduled for 7-8 PM on Tuesdays and Thursdays. Participation in office hours is not required but hopefully students will take advantage of the opportunity to ask questions and get clarification on course material, as well as to meet the instructor and other students.

Grading

Final grades for the course are computed as the sum of your scores on the individual elements below (100 possible points total), converted to a letter grade:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Score Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>97-100</td>
</tr>
<tr>
<td>A</td>
<td>93-96.99</td>
</tr>
<tr>
<td>A-</td>
<td>90-92.99</td>
</tr>
<tr>
<td>B+</td>
<td>87-89.99</td>
</tr>
<tr>
<td>B</td>
<td>83-86.99</td>
</tr>
<tr>
<td>B-</td>
<td>80-82.99</td>
</tr>
<tr>
<td>C+</td>
<td>77-79.99</td>
</tr>
<tr>
<td>C</td>
<td>73-76.99</td>
</tr>
<tr>
<td>C-</td>
<td>70-72.99</td>
</tr>
<tr>
<td>D+</td>
<td>67-69.99</td>
</tr>
<tr>
<td>D</td>
<td>63-66.99</td>
</tr>
<tr>
<td>D-</td>
<td>60-62.99</td>
</tr>
<tr>
<td>F</td>
<td>0-59.99</td>
</tr>
</tbody>
</table>

Final grades will be calculated based on the following components:

- Programming Assignments (4) 32 points (8 per assignment)
- Midterm Exams (2) 20 points (10 per exam)
- Final Project 20 points (15-project, 5-presentation)
- Participation in class discussions 28 points (2 per week)
Course Schedule (subject to Change)

While this is our planned timeline, please note that this schedule is subject to change at any time. Any revisions to this timeline will be published to ELMS; Appropriate announcements will be made in class and posted online accordingly.

<table>
<thead>
<tr>
<th>Week</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (1/27-2/2)</td>
<td>Introductions; Overview and History of the Web</td>
</tr>
<tr>
<td>2 (2/3-2/9)</td>
<td>HTML, URLs</td>
</tr>
<tr>
<td>3 (2/10-2/16)</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>4 (2/17-2/23)</td>
<td>HTML Forms</td>
</tr>
<tr>
<td></td>
<td>Assignment 1 Due 2/23</td>
</tr>
<tr>
<td>5 (2/24-3/1)</td>
<td>Javascript Part 1: event handlers, the Document Object Model</td>
</tr>
<tr>
<td>6 (3/2-3/8)</td>
<td>Javascript Part 2: variables, branding and iteration</td>
</tr>
<tr>
<td></td>
<td>Assignment 2 Due 3/8</td>
</tr>
<tr>
<td></td>
<td>Midterm 1 Due 3/15</td>
</tr>
<tr>
<td>Spring Break (3/16-3/22)</td>
<td></td>
</tr>
<tr>
<td>8 (3/23-3/29)</td>
<td>HTTP in depth, Introduction to Ajax</td>
</tr>
<tr>
<td>9 (3/30-4/5)</td>
<td>Intro to Server-side Web programming</td>
</tr>
<tr>
<td>10 (4/6-4/12)</td>
<td>Connecting to Data Stores</td>
</tr>
<tr>
<td></td>
<td>Assignment 3 Due 4/12</td>
</tr>
<tr>
<td>11 (4/13-4/19)</td>
<td>Authentication and session persistence</td>
</tr>
<tr>
<td>12 (4/20-4/26)</td>
<td>REST</td>
</tr>
<tr>
<td></td>
<td>Assignment 4 due 4/26</td>
</tr>
<tr>
<td>13 (4/27-5/3)</td>
<td>Web security</td>
</tr>
<tr>
<td></td>
<td>Midterm 2 due 5/3</td>
</tr>
<tr>
<td>14 (5/4-5/10)</td>
<td>Work on Projects; Presentations</td>
</tr>
<tr>
<td>15 (5/11-5/12)</td>
<td>Project Presentations</td>
</tr>
<tr>
<td></td>
<td>Final Project Due 5/12</td>
</tr>
</tbody>
</table>
University Course Policies

The essential purpose of the university’s course policies is to enable all of us to fully participate in an equitable, accessible and safe academic environment so that we each can be challenged to learn and contribute most effectively. They address issues such as academic integrity, codes of conduct, discrimination, accessibility, learning accommodations, etc. We are all responsible for following the policies at http://www.ugst.umd.edu/courserelatedpolicies.html. You must read them and send the instructor any questions by the end of the first week of class.

Academic Integrity and Ethical Use of Other People’s Work

In academia and in computer programming, building on the work of other people is often accepted and encouraged. In this class, there will be some situations in which it is appropriate (sometimes even necessary) to build on other people’s work. For example:

- You may get help from fellow students to understand a particular concept or technique;
- You may collaborate with other students on a project that has been designated as a group assignment;
- You may want to use a function or an algorithm published on a website;
- You may find it useful to share ideas from an article or other source in a reflection you are writing.

For the purposes of this class, the following principles govern the ethical use of other people’s work:

You have an obligation to produce original work to satisfy the learning objectives of each assignment. Other people’s work should complement, not replace, your own work. You should always give credit to individuals whose work you use. In a written document such as a reflection, this means providing a full bibliography entry as well as an in-text citation. In code, you should provide a comment indicating the following details:

- The source of the code: url (if online) or bibliographic citation;
- Whatever authorship information is available;
- The date you accessed it;
- If applicable, the version number and title of the code.

UMD students are required to abide by the student honor pledge: I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. You will be asked to complete the honor pledge as part of each assignment, quiz and test in this class.

Suspected cases of cheating, plagiarism, or other academic integrity violations will be referred to the Honor Council.
Late Work

Late work is generally not acceptable. As mentioned above in the Assignments section, if a student is unable to complete one of the four course assignments on time, they may elect to submit it up to one week late at a penalty of 10% of the grade. In all other cases, including the final project, class discussions and exams, work submitted after the due date will receive no credit.

Exceptions to this policy may be made in the case of serious illness (doctor’s not required) or truly extraordinary circumstances (such as a death in the family or military deployment). In such circumstances, please contact the instructor as soon as possible. Do not wait until a deadline has passed.

Syllabus Revision Policy

This syllabus is a guide for the course and is subject to change with advance notice. Changes will be posted in ELMS. The ELMS course site is the definitive location for all course work, and communication, including class schedules, assignments and deadlines.