CS-4350 Spring 2015: Advanced Internet Programming Syllabus

- **Course:** CS-4350, WSU BS Computer Science
- **Instructor:** Don Stringham, donstringham@weber.edu
- **Location:** Salt Lake Community College, Business Building #320
- **When:** Monday and Wednesday at 7:30 PM to 9:20 PM.
- **Need help?**
  - Look through and create an issue
  - Office/Hacker Hours will be the 4th Saturday (see Canvas Calendar for schedule)
  - Email for 1-on-1 help or to set up a time to meet

**Course Description**

This course builds on the knowledge of CS-3620 and CS-3750 to go deeper into the areas of eBusiness, multimedia, HTML, CSS, XML, Javascript, PHP and other technologies focusing on the server-side coding and database manipulation required for enterprise level web applications. It requires a high level of programming skill and knowledge of databases. Computers are provided in the lab, though you are encouraged to bring a laptop for in-class exercises.

**Prerequisites**

- **CS 3620 - Server-Side Web Development** or equivalent
- **CS 3750 - Software Engineering** or equivalent
- Understanding of variables, data types, control flow, and basic function usage in Javascript and PHP
- Strong intermediate knowledge of HTML and at least basics of CSS
- Basic jQuery knowledge (DOM interaction) is a plus

These won't be enforced by the instructor, but you will be pretty lost without understanding those concepts.

**Course Overview**

We will dive into the details of PHP and Javascript. Build interactive, data-driven websites with the potent combination of open-source technologies and web standards, even with just basic HTML knowledge. We'll tackle dynamic web programming with the help of today's core technologies: PHP, MySQL, JavaScript, CSS3, and HTML5.

Explore each technology separately, learn how to integrate them and pick up valuable web programming practices along the way. At the end of the class, you'll put everything together to build a fully functional web application site.

Topics covered include:

- Learn PHP in-depth, along with the basics of object-oriented programming
- Explore MySQL, from database structure to complex queries
- Master the JavaScript language and use it to create interactive web pages
- Use Ajax calls for background browser/server communication
- Acquire CSS skills for professionally styling your web pages
- Implement all the new HTML5 features, including geolocation, audio, video, and the canvas
- Introduction to Clean Code principles
- Test-Driven Development
Gain experience with Github, Travis CI and DigitalOcean

Topics will be demonstrated through live-code examples/slides, available at [wsu-cs4350.github.io/slidedeck](http://wsu-cs4350.github.io/slidedeck/). Additional exercises will completed in-class.

**Books**


**Homework/Projects**

All assignments are listed within the Course Outline.

**Workflow**

1. Fork the repository for the exercise/project (found under [github.com/wsu-cs4350](http://github.com/wsu-cs4350))
2. Clone the repository to your computer
3. Open the `index.html` file in a browser and open the Developer Tools
4. Modify the files to complete your solution
5. Refresh the `index.html` page to see the results, and repeat
6. Make sure all of your code is committed
7. Push-sync up to GitHub
8. Create a pull request on the original repository by the due time (generally the start of the following class)
9. You can continue to push fixes and improvements until the close date (listed in Classes) – just add a comment in the pull request to let me know it’s been updated.

When the pull request is created, you should see a message saying that “the Travis CI build is in progress” – this means that your solution is being automatically checked for syntax errors. If this “build” ends up failing (which will show a red “X”), click through the “details” link and scroll to the bottom to see what the errors were. Per the requirements below, please fix the issues and push up the changes.

Feedback will be given in the pull request, so please respond with your thoughts and questions! You are welcome to open the pull request as the work is still in-progress if you are stuck and want to ask a question – just mention @afeld with the question to make sure I know to look at it sooner.

Note that your solution will also be live at [http://USERNAME.github.io/EXERCISE](http://USERNAME.github.io/EXERCISE). For exercises with multiple levels/versions, leave a new comment in the pull request saying “Level X finished!” before pushing commits for the next level.

**Requirements**

These apply to real life, as well.

- **Travis CI build should pass**, which includes:
  - All HTML files should pass W3C Markup Validation
  - All JS files should pass JSHint
  - All PHP files should pass PHP CodeSniffer
- **Must apply “Clean Coding Principles” learned in class**:
  - 9 Qualities of Clean Code
  - 10 Principles for Keeping Your Programming Code Clean
- 12 Principles for Keeping Your Code Clean
- Sandi Metz’s Rules for Developers
- Unwritten Rules for PHP Developers
- Optimize for readability
- For projects, use Object-Oriented Programming

- Bonus points for:
  - Automated tests
  - Creativity (as long as requirements are fulfilled)

## Course Outline

### Class 1

1. Introduction
   - Install GitHub for [Mac](#) or [Windows](#)
     - You can use your own client or the command-line if you choose
   - Sign up for [GitHub](#)

2. Student checklist:
   - Access [WSU Canvas](#) page, where quizzes will be taken and grades posted.
   - Access [Piazza](#) where all questions, discussions and collaborations will be held.

3. Explain how slide deck work

4. Explain the echo slide

5. GitHub workflow
   - Walk through workflow
   - Create pull request on students repository

6. Lecture/Class Discussion
   - Introduction to 'How the Web works'

### Homework

- Work on [echo](#) exercise.
- Work on [countdown](#) exercise.

### Class 2

1. Q&A and Review
   - Workflow
   - [Piazza](#)
   - Slidedeck

2. Projects
   - PHP Application (Chapters 3-7)
   - PHP/MySQL Application (Chapters 8-11)
   - Javascript Application (Chapters 14-18)
   - Final Project (Javascript front-end and a Web Service)

3. Lecture/Class Discussion
   - Dynamic Web Content

### Homework

- Read [JavaScript Garden - Intro](#)
- Read [PHP: The Right Way - Getting Started](#)
- Read Chapter 1 - Introduction to Dynamic Web Content in Learning PHP, MySQL, JavaScript, and CSS
- Finish up and submit [echo](#) and [countdown](#) exercises

### Class 3

- Will be posted before the next class.
**Pairing Tips (Final Project)**

- Three people is possible, but two works best
- Agree on an editor and environment that you're both comfortable with
- The person who's less experienced/comfortable should have more keyboard time
- Switch who's “driving” regularly
- Make sure to save the code often and send it to both people

**Resources**

**Extra Reading**

- [Google JavaScript Style Guide](#)
- [JavaScript Garden](#)
- [Mozilla's Introduction to Object-Oriented Javascript](#)

**Beginner Materials**

This class assumes you are confident with this material, but in case you need a brush-up...

- Codecademy – JavaScript and PHP

**Suggested Extra Reading**

- [PHP QA Tools](#)
- [Functional Programming in PHP](#)
- [Functional JavaScript](#) by Michael Fogus
- [Front-end Job Interview Questions](#) by @darcyclarke (for testing yourself)
- [JavaScript Best Practices](#)
- [JavaScript Patterns](#) by Stoyan Stephanov
- [JavaScript Web Applications](#) by Alex MacCaw
- [JavaScript: The Good Parts](#) by Douglas Crockford
- [Learning Advanced JavaScript slides](#) by John Resig
- [Test-Driven JavaScript Development](#) by Christian Johansen
- [The JavaScript Interpreter, Interpreted](#) by Martha Girdler (video)

**Specific Topics**

- [Classical Inheritance in JavaScript](#) by Douglas Crockford
- [Partial Application in JavaScript](#) by Ben Alman (thanks @michaelBenin)
- [HTML5 Rocks slides](#)
- [Learning JavaScript Design Patterns](#) by Addy Osmani

**Other Lists**

- [JS: The Right Way](#)
- [PHP: The Right Way](#)

**Tools**

- code validation: [JSHint](#) / [PHPCodeSniffer](#)
- debugging: [Chrome Developer Tools](#) (tutorial) / [Firebug](#)
- sharing code snippets: [gist.github.com](#)
- asking questions: [Stack Overflow](#)

**GitHub**

- Git and GitHub
  - [Official GitHub Help](#)
  - [Recommended resources](#)
- [GitHub Pages](#)
- Official site
- Thinkful guide

**JS/PHP Sandboxes**
- CodePen
- JSFiddle
- Plunker
- Runnable
- rawgithub.com

**Frameworks**
- AngularJS
- Jasmine
- Silex
- Slim
- Phalcon
- Codeception

**Grading**
- Class/Piazza Participation – 15%
- Quizzes - 20%
- Exercises 30%
- Projects - 35%

**Statements on Plagiarism**

**SCPS**

Weber State University takes plagiarism very seriously and regards it as a form of fraud. The definition of plagiarism that has been adopted by the School of Continuing and Professional Studies is as follows: "Plagiarism is presenting someone else’s work as though it were one’s own. More specifically, plagiarism is to present as one’s own words quoted without quotation marks from another writer; a paraphrased passage from another writer’s work; or facts or ideas gathered, organized, and reported by someone else, orally and/or in writing. Since plagiarism is a matter of fact, not of the student's intention, it is crucial that acknowledgement of the sources be accurate and complete. Even where there is not a conscious intention to deceive, the failure to make appropriate acknowledgement constitutes plagiarism. Penalties for plagiarism range from failure for a paper or course to dismissal from the University.

**Instructor**

Reuse and building upon ideas or code are major parts of modern software development. As a professional programmer you will never write anything from scratch. This class is structured such that all solutions are public. You are encouraged to learn from the work of your peers. I won’t hunt down people who are simply copying-and-pasting solutions, because without challenging themselves, they are simply wasting their time and money taking this class.

Please respect the terms of use and/or license of any code you find, and if you reimplement or duplicate an algorithm or code from elsewhere, credit the original source with an inline comment.