



Course Name: Web Development

Course Number: CS 290

Credits: 4

Terms Offered: Every Term

Course Description

In this course, we will mostly talk about the frontend web development skills and a few backend techniques. We will be covering topics like HTML, CSS, JavaScript, Node.js and MySQL. We will spend most of the time talking about JavaScript.

Prerequisites: CS 162 or CS 165

Course Structure

Each week's module will be posted ahead of time (on Thursday or Friday) for you to get started early. All the contents including the learning goals and requirements, reading materials and lecture videos are put into one session and you will explore them by topics.

Assignment/Activity:

Each week you will have one or two graded assignment /activity that related to the skills covered in that week's contents and will due on Sunday night. You will learn most from writing the code on your own. The more you practice, the better you learn.

Project:

You will do an individual project which is building a simple website from week 6 to week 8 using the skills you learn in this course and you have a lot of freedom to determine the theme of the website.

Exam:

There is a proctored final exam at the end of the term. The exam will be open from Monday to Thursday of week 11 for normal term and week 8 for the summer term. Please reserve your proctor as early as possible to make sure you can get the time you want.

Instructor

Eric Ianni, Email: iannie@oregonstate.edu

Please always use your **OSU email** to contact me. The Canvas mailbox doesn't work very well. You must include the tag "**CS 290**" in your email subject to get the quickest response from me. You should expect a response to emails within 48 hours. Emails sent over the weekend sometimes take longer to respond to. Emailing should be only used for personal questions which does not include grading.



Please post all course-related questions on the Piazza discussion board **so the whole class may benefit from our conversation.** For grading questions, please make a private Piazza post to the "instructors."

Learning Resources

Required: Eloquent JavaScript <http://eloquentjavascript.net/Links> to an external site.

Optional: Jon Duckett's "HTML and CSS design and build websites", ISBN-13: 978-1118008188, ISBN-10: 1118008189

Evaluation of Student Performance

The list below indicates how the course learning outcomes will be measured:

- Activities/Exercise - 25%
- Homework Assignments - 50%
- Project - 10%
- Final Exam - 15%

Grading Policy

Grade letter	Percentage floor
A	93
A-	90
B+	87
B	83
B-	80
C+	77
C	73
C-	70
D+	67
D	63
D-	60
F	0

Accommodations

"Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students with accommodations approved by DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 541-737-4098."



Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should be aware of, or who need special arrangements in the event of an evacuation, should make an appointment with the instructor **as early as possible**, and no later than the first week of the term. Class materials will be made available in an accessible format upon request.

If you have a really tough situation that might affect your progress a lot (illness, job duties, family emergency...), you should contact the instructor immediately. **Don't wait until the due date or even past the due date to explain your personal situations and ask for extensions.** If you are not sure whether to ask for it, better do it.

Late Policy

Requests for extensions are considered on a case by case basis. Non-emergency requests must be submitted **via email at least 72 hours before the due time.** (Not having enough time to get the assignment done does not, by itself constitute an emergency, sorry!). **If you don't know if you will need an extension but might, you should ask for one.**

Time elapsed past due date	Maximum Point Percentage Possible (if no extension is granted)
T < 24 hours	90
T < 48 hours	80
T < 1 week	70

Bonus Day

You have **3 bonus days** that you can apply to any activities or assignments (except the final assignment). You can use it all at once for one assignment (if you are late for 3 days) or split it and use one day each for three assignments (no "half" day).

How to apply the bonus day: leave a comment on Canvas under that assignment submission, saying that you would like to apply x bonus days for this late submission, and you have y bonus days left after that. When TAs are applying the late penalty, they will look at your submission time and your comments. If you don't leave a comment there, TAs will directly apply the late penalty.

Course Content

This course is divided into three main sections which are largely addressed sequentially:

Layout and Styling



The first portion of the class focus on the static layout and styling of a web page (HTML/CSS). For some, this may be a review if you have done web publishing in the past. There is quite a bit of information to take in here but the problems to solve are not that intricate.

Client Side Interaction

The second portion of the class focuses on JavaScript and making interactive web pages in the browser. Things like forms that will display an error message if a password is too short or creating a drop down menus are things that will be covered in this portion of the class.

Server Side Interaction

In this portion of the class, we look at using a very simple database to store data between website visits. The technologies we will be using this term are Node.JS and MySQL. In addition, we look at how we can track a user and data from page to page which is a critical first step in designing more complex systems like shopping carts for an online shopping website.

Week	Topic(s)	
1	Web Overview and Intro to Tools HyperText Markup Language and Cascading Style Sheets	HW1 Node.js and Git Course Policy Quiz
2	Introduction to JavaScript	Activity1 JS Environment Activity2 JS Functions Activity3 JS Objects
3	JS Functions and Objects	Activity4 Fixing Closure Loop HW3 Higher-Order Functions and Objects
4	JS and the DOM JS and HTTP (forms)	HW4 DOM and Events Activity5 AJAX Interactions Project Proposal
5	Introduction to Node.js	HW5 GET and POST checker Activity6 Practice Project
6	Database Interaction	HW6 Database interactions and UI
7	Wrap-up	HW6 Database interactions and UI Sample Quiz
8	Final	Final Exam



Student Expectations

Prior Knowledge

Before starting work in this class students should know the fundamentals of object-oriented programming. The following is a list of topic that you should be very comfortable with:

Basic data types

[This list \(https://docs.microsoft.com/en-us/cpp/cpp/fundamental-types-cpp?view=vs-2017\)](https://docs.microsoft.com/en-us/cpp/cpp/fundamental-types-cpp?view=vs-2017) of data types should be familiar or at least understandable after reading the type description.

Common object types

You should know what a string is. You should know if there is a difference between a string and an array. You should know the difference between a string in C and a string object in C++

Complex data types

You should know what an array is, what a struct is and what, if any, differences there are between them.

Flow control primitives

You must be very comfortable with for, while, if/else, switch and do/while. You should know how they work, you should know what i is equal to at the completion of a loop if the condition says, for(i=0, i<5, i++) is it 5 or is it 6?

Scope

No matter where I declare int foo in your code, you should be able to figure out if any other arbitrary spot in your code can access that variable. You should know what public and private functions are and what happens (and what it means) when variables go out of scope.

Problem solving

If a language does not provide you a tool to do something, but you need that tool to make progress, you should be comfortable making it yourself. You should be comfortable making helper functions, even when the requirements don't call for it.

Organization and documentation

You should know that all of your source code probably does not belong in a single file and that every public function should have comments.

Things not on this list

This is not a comprehensive list, but it is a good start. If you don't know any of these topics, consider reviewing them before the class begins.

Code Quality

Code quality will be discussed on a section by section basis. However, the general rule is that all code should be syntactically valid, consistently formatted in a readable way and produce zero errors, warnings or notices before submission.



In this course, we follow the Google style guide for the HTML/CSS/JS code:

<https://google.github.io/styleguide/htmlcssguide.html>

<https://google.github.io/styleguide/jsguide.html>

Academic Integrity

Students are expected to comply with all regulations pertaining to academic honesty. For further information, visit [Student Conduct and Community Standards](#), or contact the office of Student Conduct and Mediation at 541-737-3656.

OAR 576-015-0020 (2) Academic or Scholarly Dishonesty:

- a) Academic or Scholarly Dishonesty is defined as an act of deception in which a Student seeks to claim credit for the work or effort of another person, or uses unauthorized materials or fabricated information in any academic work or research, either through the Student's own efforts or the efforts of another.
- b) It includes:
 - i) CHEATING - use or attempted use of unauthorized materials, information or study aids, or an act of deceit by which a Student attempts to misrepresent mastery of academic effort or information. This includes but is not limited to unauthorized copying or collaboration on a test or assignment, using prohibited materials and texts, any misuse of an electronic device, or using any deceptive means to gain academic credit.
 - ii) FABRICATION - falsification or invention of any information including but not limited to falsifying research, inventing or exaggerating data, or listing incorrect or fictitious references.
 - iii) ASSISTING - helping another commit an act of academic dishonesty. This includes but is not limited to paying or bribing someone to acquire a test or assignment, changing someone's grades or academic records, taking a test/doing an assignment for someone else by any means, including misuse of an electronic device. It is a violation of Oregon state law to create and offer to sell part or all of an educational assignment to another person (ORS 165.114).
 - iv) TAMPERING - altering or interfering with evaluation instruments or documents.
 - v) PLAGIARISM - representing the words or ideas of another person or presenting someone else's words, ideas, artistry or data as one's own, or using one's own previously submitted work. Plagiarism includes but is not limited to copying another person's work (including unpublished material) without appropriate referencing, presenting someone else's opinions and theories as one's own, or working jointly on a project and then submitting it as one's own.
- c) Academic Dishonesty cases are handled initially by the academic units, following the process outlined in the University's Academic Dishonesty Report Form, and will also be referred to SCCS for action under these rules.



Oregon State University
Ecampus

CS290 Syllabus