

# CS290 – Web Development for CS Majors

## Winter 2020, CRN: 33342, Credits: 4

Instructor: Joseph Jess

Web: <http://cf.linnbenton.edu/bcs/cs/jessj/web.cfm?pgID=5404>

email: [jessj@linnbenton.edu](mailto:jessj@linnbenton.edu)

**Initial class note:** This is **not** an introduction course to computer science (CS), though that does not mean it will be difficult. This is an intro to various web development technologies and design ideas. I assume that you have some programming experience and we will expand on the design, testing, and implementation concepts covered in an introductory CS course. Expect to do a good bit of reading, much practicing, and much discussion of the topics we cover.

### 1. LBCC catalog course description, including pre-requisites/co-requisites:

This course will cover how to design and implement a multi-tier application using Web technologies. This will include the creation of extensive custom client and server side code consistent with achieving a high-quality software architecture.

Prerequisites: CS162 Intro to Computer Science II with a grade of "C" or better or equivalent experience as determined by a Computer Systems Department instructor

### 2. Class Time-space:

2.1 Lecture + demo + lab: MW 1000 – 1150, MKH203

### 3. Measurable student learning outcomes:

At the completion of the course, students will be able to:

- 3.1 Describe the architectural elements of effective web applications.
- 3.2 Describe key threats to relevant architectural attributes in web enabled applications.
- 3.3 Demonstrate implementation of custom functionality across multiple tiers of a web enabled applications.
- 3.4 Evaluate which architectural strategies to apply to address quality requirements, with emphasis on scalability, usability and security.

### 4. My hopes are that we will also:

**Design** and create a moderately complex static web site that conforms to current standards.

**Implement** a custom user interface behavior using client-side scripting.

**Implement** asynchronous calls for sending data between the client and server.

**Implement** dynamically-generated websites using server-side scripting.

**Use** basic database commands to create, store and retrieve data in conjunction with a dynamic website

### 5. Learning resources:

5.1 **Note:** All class materials and storage will be freely available in a digital format

5.2 **The HTML, CSS, Javascript, and select server-side languages** – available through several media and available options will be provided when needed in class.

5.3 (**recommended**) Access to a computer outside of class to practice and work on assignments (there are a few labs with suitable setups around on and off campus that I can point you to, should you need it).

5.4 A text editor. We could use a smart IDE to do this work as well!

5.4.1 We will discuss some capabilities of smart code editors during the course.

5.5 (**strongly recommended**) A desire to learn, experiment, design, test, and problem solve with code (both on and off of a computer).

### 6. Grading:

6.1 Scores for coursework items will be initially available when demonstrated to the instructor. I will keep a spreadsheet in a folder that I share with you through your student email account using Google Drive. We will discuss this some in class, including how to access it and keep yourself organized (which may affect your grade).

6.1.1 My favored grading method is to have people show me where they are throughout the week so we can discuss approaches and any requirements that would affect the grade, which also allows for better understanding of requirements.

6.2 Students will be required to turn in all coursework items before 23:59 (Pacific Time Zone) on the date that they are due (generally the first day of class each week in my courses).

6.2.1 Students must be sure to give themselves plenty of time to submit coursework, as late work will not be accepted without prior consent or special circumstances.

6.3 To earn a passing grade in this course you must pass each of the following coursework categories:

6.3.1 **Demonstration:** Discussion and weekly assignments – 50%

6.3.1.1 There are a number of weekly assignments to be completed for this class, designed to challenge and solidify design, coding, and testing skills.

6.3.1.2 Project components are generally graded based on:

6.3.1.2.1 completeness (does it compile and run)

6.3.1.2.2 correctness (does it meet the listed requirements)

6.3.1.2.3 quality and explanation of the design (features in advance, organized)

6.3.1.2.4 quality and explanation of the tests (test each feature and expected successes and failures)

6.3.1.2.5 quality of the implementation (consistent and readable style, runs well, is easy to learn and use)

6.3.1.2.6 **Note:** careful design, systematic testing, consistent style, and readability of code are important software quality factors (all of which are subject to interpretation but graded by the instructor based on the spirit and letter of the requirements, so be sure to explain your decisions).

6.3.1.2.7 **Note well:** Your submission needs to be understandable and able to function from just the files in your final submission. This means that you need to include any and all files necessary for your project to function, even if the instructor provided some of them.

6.3.1.2.8 **Note very well:** Graded work must be designed and implemented by the student submitting the work and must function on one of the instructor's machines in order to get a final grade. (to create a working project quickly: get it working simply, then add to it; if at some point it stops functioning, then you will better know where an error was introduced)

6.3.2 **Final:** Final project – 50%

6.3.2.1 There will be a final project to test the overall ability to understand, design, implement, test, and reflect on the problem solving and coding knowledge and skills covered in the class.

6.3.2.2 The final project will be a mix of in-class (initial design, testing, and implementation discussion) and take-home (final design, testing, and implementation) elements.

6.3.3 Final grades will be given out based on the following based on score in the class:

90-100%: A

80-89%: B

70-79%: C

60-69%: D

00-59%: F

6.4 Reminder: A passing grade in order to count for course requirements for CS classes is generally a C or above.

## 7. Other Administrative Information:

7.1 For a list of general administration information (note that this list is not intended to be exhaustive), such as:

7.1.1 contacting me,

7.1.2 [accessibility resources](#),

7.1.3 expectations of student conduct,

7.1.4 communications,

7.1.5 student assistance,

7.1.6 miscellany,

7.1.7 nondiscrimination & nonharrasment,

(each section contains a number of sub-sections and is not meant to be exhaustive of all situations)

see my administrative information document: [administrative\\_information\\_document](#).