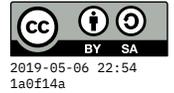


# Syllabus

23 January 2019



Welcome to CS 120, on web page and application development. Topics include HTML, Web browser and server communication using HTTP and HTTPS, browser state tracking, basic web server configuration settings, client-side JavaScript, back-end database connectivity, and application development using common tools and languages. Students are required to develop and complete several web-based applications.

**When:** Monday, Wednesday 2–3:50 PM

**Where:** M-412

**Credits:** 3

**Prerequisites:** CS102

## Contact Info

**Instructor:** Prof. Christopher League, Ph.D.

**Email:** [christopher.league@liu.edu](mailto:christopher.league@liu.edu)<sup>1</sup> — please **include “CS120”** in the subject. I have several email addresses, but all messages end up in the same place, so please use only one.

**Office hours:** Monday, Wednesday 4–4:50 PM and by appointment using [bookme.liucs.net](http://bookme.liucs.net)<sup>2</sup>

**Office phone:** +1 718 488 1137 (but email is better)

**Office location:** Pratt 122 (2nd aisle, 2nd desk on left)

## Resources

We will use several web resources:

- [liucs.net/cs120s19/](http://liucs.net/cs120s19/)<sup>3</sup> – notes, schedule, assignment handouts
- [gitlab.liu.edu](http://gitlab.liu.edu)<sup>4</sup> – discussion forum, assignment submission, feedback

We will also use a designated *virtual machine* image to complete assignments, so that we have a consistent work environment. More details about configuring and operating it will be available in Check-in 1 and Project 1.

There is no required textbook, but if you’d like a book to supplement or for reference, here are some great suggestions:

- *HTML5: The Missing Manual*<sup>5</sup> by Matthew MacDonald



<sup>1</sup>[christopher.league@liu.edu?subject=CS120](mailto:christopher.league@liu.edu?subject=CS120)



<sup>2</sup>[bookme.liucs.net/](http://bookme.liucs.net/)



<sup>3</sup>[liucs.net/cs120s19/](http://liucs.net/cs120s19/)



<sup>4</sup>[gitlab.liu.edu/](http://gitlab.liu.edu/)



<sup>5</sup>[amzn.to/1ykqPzQ](http://amzn.to/1ykqPzQ)



<sup>6</sup>amzn.to/1szc  
DDz

- *JavaScript & jQuery: The Missing Manual*<sup>6</sup> by David Sawyer McFarland

## Requirements

Your grade will be computed based on the materials you submit. There are a total of 1,000 points available, broken down as follows:

- There will be **7 programming projects** during the semester. Assignments are worth **100 points each**, for a total of **700 points**.
- There will be **12 ‘check-in’ opportunities**, roughly one per week. These vary from week to week, but may involve responding to a survey, taking a brief online quiz, participating in a discussion, or making some progress on an assignment. Check-ins are worth **30 points each** but I will **drop the lowest 2 scores** so only 10 will count, for a total of **300 points**.

On the 1,000-point scale, you can expect the following letter grades:

|                        |                        |
|------------------------|------------------------|
| $\geq 930$ : <b>A</b>  | $\geq 770$ : <b>C+</b> |
| $\geq 900$ : <b>A-</b> | $\geq 730$ : <b>C</b>  |
| $\geq 870$ : <b>B+</b> | $\geq 680$ : <b>C-</b> |
| $\geq 830$ : <b>B</b>  | $\geq 600$ : <b>D</b>  |
| $\geq 800$ : <b>B-</b> | else: <b>F</b>         |

In the end, I may choose to adjust the scale slightly to compensate for assignments or questions that turned out to be trickier than I intended. Such adjustments would never *lower* your grade from what is designated in the above table; if you achieve 930 points, you are guaranteed an A.

## Policies

It is important to **complete tasks on time**, so you don’t fall behind. Missed check-ins will receive a zero, and cannot be made up (but remember, the lowest two are dropped). **Late assignments** are accepted up until finals week, but will be penalized as follows.

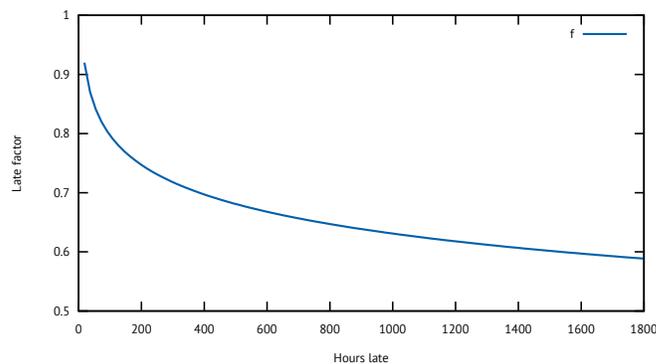
We define a *lateness factor*  $f$  as a real number in the range  $\{0 \dots 1\}$  that will be multiplied by your earned score to determine a late score. The formula is:

$$f = \min\left(1.0, \frac{18 - \log_2\left(\frac{h}{24}\right)}{20}\right)$$

where the variable  $h$  represents the number of hours the submission is late. The table below shows some sample values of the late factor for increasingly late submission times.

| weeks late | days late | hours late ( $h$ ) | late factor ( $f$ ) |
|------------|-----------|--------------------|---------------------|
| 0.01       | 0.1       | 2.4                | 1.000               |
| 0.04       | 0.3       | 7.2                | 0.987               |
| 0.07       | 0.5       | 12.0               | 0.950               |
| 0.14       | 1.0       | 24.0               | 0.900               |
| 0.29       | 2.0       | 48.0               | 0.850               |
| 0.43       | 3.0       | 72.0               | 0.821               |
| 1.00       | 7.0       | 168.0              | 0.760               |
| 2.00       | 14.0      | 336.0              | 0.710               |
| 4.00       | 28.0      | 672.0              | 0.660               |
| 8.00       | 56.0      | 1344.0             | 0.610               |

The idea is that is that the penalty is somewhat steep initially (from an **A** to a **B+** after just one day) but shallows out over time. It will still be worthwhile to submit a missing assignment, even weeks late.



**There will be no extra credit.** Students usually ask for extra credit late in the semester after they have already squandered their original opportunities. Be sure to start your work early, so that we can detect and solve any problems before they can impact your grade.

**Plagiarism** is the use or presentation of ideas, words, or work that is not one's own and that is not common knowledge, without granting credit to the originator. Plagiarism is a practice that is not only unacceptable, but which is to be condemned in the strongest terms possible on the basis of moral, educational and legal grounds. Under University policy, plagiarism may be punishable by a range of penalties from a failing grade in the assignment or course to dismissal from the School of Business, Public Administration and Information Sciences. All students are required to read the handbook on avoiding plagiarism.<sup>7</sup>

**Cheating** includes, but is not limited to the following: falsification of statements or data; listing sources that have not been used; having another individual write your paper or do your assignments; writing a paper or creating work for another student to use without proper attribution; purchase of paper or research work for one's submission as their own work; using written, verbal, or electronic or other sources



of aid during an examination (except when expressly permitted by the instructor, depending on the nature of the examination) or knowingly providing such assistance to aid other students.

In a course with programming assignments, it is usually okay to work with and learn from other students to **some** extent, but what you submit in the end needs to be your own. The most reliable way to do that would be to set aside whatever code you created together, and then recreate it from scratch on your own.

**Showing up on time** to class is extremely important. If you must be absent or more than 5 minutes late, please try to notify me in advance. I will be keeping track of whether you are in class, and when you arrive. A few missed classes will not count against you, but habitual absence will significantly hurt your grade. Additionally, there will be no make-up quizzes. I do not distinguish between “excused” and “unexcused” absence. Unless you miss an **exam** due to a severe medical emergency, I don’t need to see a doctor’s note. If you do miss an exam, the make-up exam may be somewhat different from the one given in class.

In accordance with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1990, including changes made by the Americans with Disabilities Amendments Act of 2008, the Long Island University **does not discriminate against qualified individuals with disabilities**. If you are a student with a documented disability/impairment (psychological, neurological, chronic medical, learning disability, sensory, physical) and require reasonable accommodations, please register with Student Support Services and provide me with an accommodation letter. Visit Sloan Building 1st floor, call 718 488 1044, or visit Student Support Services.<sup>8</sup>



<sup>8</sup>[www.liu.edu/  
Brooklyn/SSS](http://www.liu.edu/Brooklyn/SSS)

I participate in the **LIU Safe Zone** program. Representatives of the program serve as contacts for individuals on campus with questions or concerns related to sexual orientation and gender identity, whether of self or of a friend or family member. The goal of the program is to promote a safe and free campus for all students. Safe Zone areas can be identified by a sticker with the LIU Safe Zone logo.



The **Family Educational Rights and Privacy Act (FERPA)** gives students control over the disclosure of their educational records. During this course you may have the opportunity to create accounts or register with certain public online services. In these cases, you need not make any personally identifying information public. You may use a pseudonym or online handle, as long as you identify yourself to the instructor.

## Goals and objectives

Upon completion of this course, I expect that you will be able to...

1. describe the purpose of standard methods and headers in the HyperText Transfer Protocol (HTTP).
2. use the developer tools built in to web browsers to investigate the Document Object Model and diagnose connection problems.
3. create basic interactive web applications using JavaScript and jQuery.
4. create intermediate interactive web applications using a client-side framework such as AngularJS.
5. implement the server-side of a web API using a database and framework such as Play or NodeJS.

## Assessment of learning

This course is an elective of the B.S. program in Computer Science. This section relates programmatic objectives with objectives and assessment instruments used in this course. At the completion of the B.S. program...

**BSCS 1.1 Written Communication:** Students will prepare quality written documents that effectively communicate technical ideas and system specifications.

**BSCS 1.2 Oral Communication:** Students will develop competencies in delivering effective presentations of technical ideas and system specifications.

**BSCS 2.1 Programming Language Constructs:** Students will demonstrate mastery in the use of programming constructs, including functions. *Practiced in course objective 3.*

**BSCS 2.2 Algorithms:** Students will achieve competency in developing algorithms using linear data structures, trees, and graphs.

**BSCS 2.3 Networks:** Students will develop an understanding of internet protocols and apply the concepts to perform network configuration and troubleshooting tasks. *Mastered in course objectives 1, 2.*

**BSCS 2.4 Computer Systems:** Students will develop an understanding of the hardware and software architecture of computer systems. *Practiced in course objectives 4, 5.*

**BSCS 3.1 Programming:** Students demonstrate competency in writing and completing programs using commonly accepted programming practices. *Practiced in course objectives 3, 4, 5.*

**BSCS 4.1 Application Design:** Students develop proficiency in designing database intensive applications with demonstrated knowledge of Normalization and SQL. *Practiced in course objective 5.*

**BSCS 4.2 Application Development:** Students use appropriate development environments, tools and software engineering principles to plan, implement, and test a software application. *Practiced in course objectives 2, 4, 5.*

## Schedule

We will cover these topics:

1. HTML and CSS
2. HTTP and web developer tools
3. Static site generators, markdown
4. Client-side programming in JavaScript (including jQuery)
5. ReST API design and programming
6. Hosting options, server configuration and maintenance

The day-by-day schedule is shown below, including all deadlines. The schedule is available as `schedule.ics`<sup>9</sup> – copy that link to subscribe or import it into Google Calendar and other systems.



<sup>9</sup>[liucs.net/cs120s19/schedule.ics](https://liucs.net/cs120s19/schedule.ics)

**Wed 23 Jan: Meeting 1**

Overview, HTML background and syntax.

**Mon 28 Jan: Meeting 2**

HTML tag inventory, block vs inline content, and basic CSS rules.

**Tue 29 Jan: Check-in 1 due**

Software setup.

**Wed 30 Jan: Meeting 3**

HTML/CSS validation tool, measurement units, box model.

**Sun 3 Feb: Project 1 due**

HTML/CSS

**Mon 4 Feb: Meeting 4**

CSS fonts and color specifications.

**Tue 5 Feb: Check-in 2 due**

(Skipped)

**Wed 6 Feb: Meeting 5**

(Sick day)

**Mon 11 Feb: Meeting 6**

(Sick day)

**Tue 12 Feb: Check-in 3 due**

(Skipped)

**Wed 13 Feb: Meeting 7**

Static site generation with markdown and an HTML template.

**Mon 18 Feb: No class – Presidents' Day**

**Tue 19 Feb: Meeting 8**

Introduction to Javascript and the DOM. Work on a Hide/Show toggle.

**Wed 20 Feb: No class – I am out of town**

We will substitute some online content and exercises.

**Sun 24 Feb: Project 2 due**

Javascript and the DOM

**Mon 25 Feb: Meeting 10**

Introduction to jQuery

**Wed 27 Feb: Meeting 11**

Start on a simple jQuery app

**Thu 28 Feb: Check-in 4 due****Mon 4 Mar: Meeting 12**

Snow day

**Tue 5 Mar: Project 3 due**

jQuery app

**Tue 5 Mar: Check-in 5 due**

(Skipped)

**Wed 6 Mar: Meeting 13**

Solution to color palette project.

**Mon 18 Mar: Meeting 14**

Intro to Google Maps API

**Tue 19 Mar: Check-in 6 due**

Google Developer Console

**Wed 20 Mar: Meeting 15**

Developing an interactive map-based site

**Mon 25 Mar: Meeting 16**

Managing a navigation bar, and synchronizing with a map.

**Tue 26 Mar: Check-in 7 due**

(Skipped)

**Wed 27 Mar: Meeting 17**

Intro to web accessibility standards and tools.

**Fri 29 Mar: Project 4 due**

Google Maps

**Mon 1 Apr: Meeting 18**

Explanation of REST API technology and conventions.

**Tue 2 Apr: Check-in 8 due****Wed 3 Apr: Meeting 19**

Demonstration of REST API using Python Eve.

**Sun 7 Apr: Project 5 due**

Server-side API programming

**Mon 8 Apr: Meeting 20**

Building an end-to-end CRUD app with jQuery, Eve, MongoDB.

**Tue 9 Apr: Check-in 9 due****Wed 10 Apr: Meeting 21**

Further development of CRUD app.

**Mon 15 Apr: Meeting 22**

Technical help on your CRUD app.

***Tue 16 Apr: Check-in 10 due***

***Wed 17 Apr: Meeting 23***

Technical help on your CRUD application.

***Sun 21 Apr: Project 6 due***

Client-server CRUD application

***Mon 22 Apr: Meeting 24***

Hosting and deployment options.

***Tue 23 Apr: Check-in 11 due***

***Wed 24 Apr: Meeting 25***

***Mon 29 Apr: Meeting 26***

***Tue 30 Apr: Check-in 12 due***

***Wed 1 May: Meeting 27***

***Mon 6 May: Meeting 28***

***Tue 7 May: Project 7 due***

Server configuration and management