CS 120 Syllabus

22 January 2018

Welcome to Web Development. Here is the official course description:

> Web page and Common Gateway interface (CGI) 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 CGI 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: M411
Credits: 3
Prerequisites: CS102

Contact Info

Instructor: Prof. Christopher League, Ph.D.
Email: `christopher.league@liu.edu` — please include the course number (CS120) in the subject. I have several email addresses, but all messages end up in the same place, so **please use only one.**
Google Hangout: `cleague@gmail.com`
Office hours: Monday, Wednesday 4–4:50 PM, Thursday 3–4:50 PM, or make an appointment at `https://liucs.net/bookme`
Office phone: +1 718 488 1274
Office location: H-700, LIU Brooklyn

Resources

- We will use several web resources:
  - `https://liucs.net/cs120s18/` — notes, schedule, assignment handouts
  - `https://gitlab.liu.edu/` — assignment submission, discussion, sample code
  - `http://www.gradechamp.com/` — grade reports

If you have a question or problem that might also apply to other students, *please* ask on the discussion forum rather than by email. Then the GA and other students can help you too, and the solution is available for all to see. Try to use email only for personal matters such as your grades.

- 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* by Matthew MacDonald `http://amzn.to/1ykqPzQ`
    - *JavaScript & jQuery: The Missing Manual* by David Sawyer McFarland `http://amzn.to/1szcDDz`
    - *React Quickly: Painless web apps with React, JSX, Redux, and GraphQL* by Azat Mardan `https://www.manning.com/books/react-quickly`
- Additional office hours are available with my graduate assistant, Manasi Gore. Her hours are Thursdays 12-3 in H-700. You can also contact her at `manasi.gore@my.liu.edu`.

Requirements

Your grade will be computed based on assignments, exams, and participation. 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:

		≥ 870:	**B+**	≥ 770:	**C+**		
≥ 930:	**A**	≥ 830:	**B**	≥ 730:	**C**	≥ 600:	**D**
≥ 900:	**A−**	≥ 800:	**B−**	≥ 680:	**C−**	else:	**F**

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 scores are dropped). If you need to miss an exam, try to notify me in advance so we can make other arrangements. **Late assignments** will be graded as follows.

This formula specifies a *lateness factor* f that is multiplied by your earned score to determine a late score. The variable h represents the number of hours the submission is late.

$$f = \frac{8.5 - \log_2\left(\frac{h}{24}\right)}{10}$$

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 affect 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 by visiting `https://liucs.net/u2`

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 his/her 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 want to see a doctor's note. If you do miss an exam, the make-up exam will be different – and probably *not* easier.

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 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. Basics of HTTP and introduction to web developer tools
2. HTML and CSS
3. Static site generators (e.g. Jekyll, Hugo)
4. Client-side programming in JavaScript (including jQuery, React)
5. ReST API design and programming
6. Hosting options, server configuration and maintenance

The day-by-day schedule is shown below, including all deadlines.

Mon Jan 22 **Meeting 1** at 2 pm. HTTP basics, anatomy of URL, GET/POST.

Wed Jan 24 **Meeting 2** at 2 pm. The curl command, basic static HTML skeleton and a few CSS properties.

Sun Jan 28 **Check-in 1** due at 23:59.

Mon Jan 29 **Meeting 3** at 2 pm. Demonstration of basic multi-column Bootstrap layouts.

Wed Jan 31 **Meeting 4** at 2 pm. Mobile viewport header, custom CSS and selector syntax, Google fonts.

Sun Feb 4 **Check-in 2** due at 23:59.

Mon Feb 5 **Meeting 5** at 2 pm. Canceled due to illness.

Tue Feb 6 **Project 1** due at 23:59.

Wed Feb 7 **Meeting 6** at 2 pm. Practicing CSS selectors with flukeout (checkin 3), intro to static site generation.

Sun Feb 11 **Check-in 3** due at 23:59.

Mon Feb 12 **Meeting 7** at 2 pm. Demonstration of `jekyll` for static site generation.

Wed Feb 14 **Meeting 8** at 2 pm. Introducing hugo for static site generation.

Sun Feb 18 **Check-in 4** due at 23:59.

Wed Feb 21 **Meeting 9** at 2 pm. Using the `<script>` tag to manipulate elements on the page.

Sun Feb 25 **Project 2** due at 23:59. **Check-in 5** due at 23:59.

Mon Feb 26 **Meeting 10** at 2 pm. Introduction to the Google Maps JS API.

Wed Feb 28 **Meeting 11** at 2 pm.

Sun Mar 4 **Check-in 6** due at 23:59.

Mon Mar 5 **Meeting 12** at 2 pm.

Wed Mar 7 **Meeting 13** at 2 pm.

Thu Mar 8 **Project 3** due at 23:59.

Sun Mar 11 **Check-in 7** due at 23:59.

Mon Mar 19 **Meeting 14** at 2 pm.

Wed Mar 21 **Meeting 15** at 2 pm.

Sun Mar 25 **Check-in 8** due at 23:59.

Mon Mar 26 **Meeting 16** at 2 pm.

Wed Mar 28 **Meeting 17** at 2 pm.

Sun Apr 1 **Project 4** due at 23:59.

Mon Apr 2 **Meeting 18** at 2 pm.

Wed Apr 4 **Meeting 19** at 2 pm.

Sun Apr 8 **Check-in 9** due at 23:59.

Mon Apr 9 **Meeting 20** at 2 pm.

Tue Apr 10 **Project 5** due at 23:59.

Wed Apr 11 **Meeting 21** at 2 pm.

Sun Apr 15 **Check-in 10** due at 23:59.

Mon Apr 16 **Meeting 22** at 2 pm.

Wed Apr 18 **Meeting 23** at 2 pm.

Sun Apr 22 **Check-in 11** due at 23:59.

Mon Apr 23 **Meeting 24** at 2 pm.

Tue Apr 24 **Project 6** due at 23:59.

Sun Apr 29 **Check-in 12** due at 23:59.

Mon Apr 30 **Meeting 26** at 2 pm.

Wed May 2 **Meeting 27** at 2 pm.

Mon May 7 **Meeting 28** at 2 pm.

Thu May 10 **Project 7** due at 23:59.

Mon May 14 **Final Exam** at 12:30 pm.