Syllabus

23 January 2019
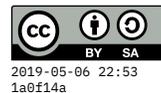
Welcome to CS 101, a broad overview of the main areas of study in Computer and Information Sciences. Topics include computer organization, information processing, algorithms, and programming. The main ideas behind the theory and design of Operating Systems, Databases, and Computer Networks, along with current views on the theory and practice of Software Engineering, and the basics of Artificial Intelligence are also explored. The course highlights the uses of computing systems in business, the sciences, and other professional fields. This course is required for all students majoring in Computer Science or Information Systems. It is also suitable for majors in other disciplines who want to go beyond being casual users of computers to gain a deeper appreciation of some of the most important computing and information technologies developed over the last fifty years.

> Computation and algorithmic thinking have become essential components for solving problems in many different fields. As such, computer scientists are intimately involved in finding solutions to some of the most pressing social, economic, and scientific problems of our day

> — Goldweber, Barr, and Patitsas, in SIGCSE 2013.

When: Monday, Wednesday 9–10:50 AM

Where: H-701

Credits: 3

Prerequisites: None

Contact Info

Instructor: Prof. Christopher League, Ph.D.

Email: `christopher.league@liu.edu`[1] — please **include "CS101"** 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`[2]

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

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

[1] `christopher.league@liu.edu?subject=CS101`

[2] `bookme.liucs.net/`

Resources

We will use several web resources:

- `liucs.net/cs101s19/`[3] – notes, schedule assignment handouts

[4]gitlab.liu.e
du/

- `gitlab.liu.edu`[4] — discussion forum, assignment submission, feedback

The textbook is *Computer Science Illuminated*[5] by Dale and Lewis (6th edition, ISBN 978-1284055917). An older edition is fine, if that saves you some money.

Requirements

Your grade will be computed based on assignments, quizzes, and exams. There are a total of 1,000 points available, broken down as follows:

[5]amzn.to/1yks
nd6

- There will be **8 assignments** during the semester. Assignments are worth **60 points each,** for a total of **480 points.**

- There are **6 quizzes** scheduled throughout the semester, to make sure you are following along with the lectures and online review resources. Quizzes are worth **30 points each,** but I will **drop the lowest two scores** so only 4 will count, for a total of **120 points.**

- There will be a midterm and final exam, worth **200 points each** for a total of **400 points.**

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

>= 930:	**A**	>= 770:	**C+**
>= 900:	**A–**	>= 730:	**C**
>= 870:	**B+**	>= 680:	**C–**
>= 830:	**B**	>= 600:	**D**
>= 800:	**B–**	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 quizzes will receive a zero, and cannot be made up (but remember, the lowest two quiz scores are dropped). If you need to miss an exam, try to notify me in advance so we can make other arrangements. **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.
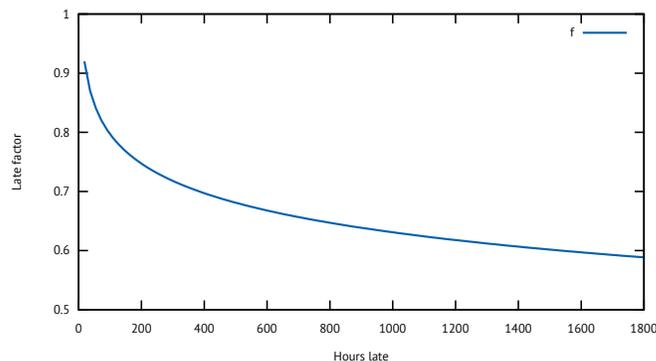
There will be no extra credit. Students usually ask for extra credit late in the semester after they have already squandered their original opportunities. Be sure to start your work early, so that we can detect and solve any problems before they can impact your grade.

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.[6]

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.[7]

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

[6]liucs.net/u2

[7]www.liu.edu/ Brooklyn/SSS

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. understand and follow basic algorithms.

2. understand the mathematical and logical foundations of computing.

3. understand the role of programming languages in software architecture.

4. explain the key concepts of operating systems and computer networks.

5. work with simple data models structured as tables and keys.

6. exhibit awareness of professional organizations and technical opportunities.

7. be inspired to attend computing seminars and workshops outside of classwork.

Assessment of learning

This course is part of a 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...

1.1: Students will demonstrate proficiency in basic algorithms and data structures. Introduced with course objective 1.

1.2: Students will understand the mathematical and logical foundations of computing. Introduced with course objective 2.

2.1: Students will master the fundamentals of programming languages and software architecture. Introduced with course objective 3.

3.1: Students will master the key concepts of operating systems and computer networks. Introduced with course objective 4.

3.2: Students will develop a data modeling design for a proposed database application. Introduced with course objective 5.

5.1: Students will exhibit awareness of professional organizations and technical opportunities. Introduced with course objective 6.

5.2: Students will attend seminars and workshops outside of classwork. Introduced with course objective 7.

Program objectives 1.2 and 2.1 are routinely assessed by scoring a selection of questions from students' midterm exams in CS101.

Schedule

Section numbers in the textbook are noted as "§1.2," and correspond to the 6th edition. If you have a different edition, see this table[8] to find corresponding section numbers. The schedule is available as `schedule.ics`[9] – copy that link to subscribe or import it into Google Calendar and other systems.

[8]`liucs.net/u5 62`

[9]`liucs.net/cs 101s19/sched ule.ics`

Wed 23 Jan: **Meeting 1**

Introduction, and positional numbering. *Read §2.1*

Mon 28 Jan: **Meeting 2**

Binary natural numbers. *Read §2.2*

Wed 30 Jan: **Meeting 3**

Signed two's complement numbers. *Read §3.1–3.2*

Mon 4 Feb: **Meeting 4 [quiz]**

Text encodings. *Read §3.3*

Wed 6 Feb: **Meeting 5**

Text compression.

Mon 11 Feb: **Meeting 6**

Graphics encoding. *Read §3.5*

Wed 13 Feb: **Meeting 7**

Audio/video encoding. *Read §3.4, 3.6*

Mon 18 Feb: **Assignment 1 due**

Text compression

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

Tue 19 Feb: **Meeting 8 [quiz]**

Boolean logic. *Read §4.1–4.2*

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

We will substitute some online content and exercises. Combinational circuits. *Read §4.3, 4.4*

Mon 25 Feb: **Meeting 10**

Memory and the Turing Machine model. *Read §4.5, 5.1*

Tue 26 Feb: **Assignment 2 due**

Image representation

Wed 27 Feb: **Meeting 11**

The von Neumann architecture and machine code. *Read §5.2, 6.1–6.3*

Mon 4 Mar: **Meeting 12**

Algorithms and complexity. *Read §6.5, 7.1–7.2*

Tue 5 Mar: **Assignment 3 due**

Combinational circuits

Wed 6 Mar: **Meeting 13 [quiz]**

Searching and sorting. *Read §7.4–7.5*

Mon 18 Mar: **Meeting 14**

Python syntax.

Wed 20 Mar: **Meeting 15 [exam]**

Midterm exam.

***Mon 25 Mar*: Meeting 16**

Python programming.

***Tue 26 Mar*: Assignment 4 due**

Sorting and searching

***Wed 27 Mar*: Meeting 17**

More programming.

***Mon 1 Apr*: Meeting 18**

Other programming languages. *Read §9.2–9.3*

***Wed 3 Apr*: Meeting 19**

What is an Operating System? *Read §10.1–10.4*

***Sun 7 Apr*: Assignment 5 due**

Python program

***Mon 8 Apr*: Meeting 20**

Introduction to Linux and the network stack. *Read §15.1–15.4*

***Wed 10 Apr*: Meeting 21 [quiz]**

Languages and protocols of the web. *Read §16.1–16.2*

***Mon 15 Apr*: Meeting 22**

The relational data model. *Read §12.3*

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

Introduction to SQL.

***Sun 21 Apr*: Assignment 6 due**

Web page

***Mon 22 Apr*: Meeting 24 [quiz]**

Overview of Artificial Intelligence. *Read §13.1–13.3*

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

Planning and searching techniques. *Read §17.1–17.2*

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

Authentication and authorization.

***Tue 30 Apr*: Assignment 7 due**

Artificial intelligence

***Wed 1 May*: Meeting 27 [quiz]**

Cryptography. *Read §17.4–17.5*

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

Wildcard day and wrap-up.

***Wed 8 May*: Final exam 8:00 am**

***Fri 10 May*: Assignment 8 due**

Cryptography