

CS 664 Syllabus

20 January 2016

Welcome to CS 664, *Compiler Theory and Design* — a study of compiler organization, including symbol tables, lexical analysis, syntactic analysis, semantic analysis, code generation and optimization techniques.

When: Wednesday evenings 6–8:30pm

Where: H-214

Credits: 3

Format: This is a *blended* course, which means that nearly half of the content is delivered online. Consult the detailed schedule to determine which weeks we meet face-to-face *vs* online.

Most software developers will never implement a full-blown native-code compiler, so why study compilers at all?

- Some of the algorithms and techniques routinely used in compilers also appear in other kinds of systems. As just a few examples: graph coloring, strongly-connected components, unification, tree walking, and pattern matching.
- Many non-compiler applications involve some amount of lexical analysis and parsing, whether for configuration files, data formats, or presentation templates.
- Languages are the tools of our trade; understanding how they are implemented can make you a better software developer.

Contact Info

Instructor: Prof. Christopher League, Ph.D.

Email: christopher.league@liu.edu — please include the course number (CS664) in the subject. I have several email addresses, but all messages end up in the same place, so use only one.

Google Hangout: cleague@gmail.com

Office hours: Monday 2:30–3:50, Wednesday 4–5:20 or make an appointment at <https://liucs.net/bookme>

Office phone: +1 718 488 1274

Office location: LLC 206, LIU Brooklyn

Resources

Web sites: We will use several web resources:

- <https://liucs.net/cs664s16/> — syllabus and schedule
- <https://piazza.com/liu/spring2016/cs664> — discussion, Q&A
- <https://git.liucs.net/> — assignment submission, 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 Piazza rather than by email. Then 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.

Text: *Language Implementation Patterns* by Terrence Parr, ISBN 978-1-934356-45-6 <https://pragprog.com/book/tpdsl/language-implementation-patterns>
The eBook is \$24 for a DRM-free PDF; a combo price is available for eBook + paper.

The above book contains many useful concepts, patterns, and techniques, but unfortunately it's based on ANTLR 3 and I prefer to work with ANTLR 4. So it may be helpful also to obtain *The Definitive ANTLR 4 Reference* by Terrence Parr, ISBN 978-1-93435-699-9 <https://pragprog.com/book/tpantlr2/the-definitive-antlr-4-reference>.

Library: Campus library resources tailored for computer science are available at <https://liucs.net/u1>

Requirements

There are a total of 1,000 points available, broken down as follows:

- There will be **8 programming assignments** scheduled throughout the semester. The exact requirements and expectations for each will be posted to the course web site. They are worth **100 points each**, for a total of **800 points**.
- There is no midterm exam, but there will be a final exam, worth **200 points**.

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

	≥ 870:	B+	≥ 770:	C+	≥ 670:	D+	
≥ 930:	A	≥ 830:	B	≥ 730:	C	≥ 600:	D
≥ 900:	A–	≥ 800:	B–	≥ 700:	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 milestone requirements **on time**, so you don't fall behind. Late work 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.

Long Island University seeks to provide **reasonable accommodations for all qualified persons with disabilities**, whether psychological, neurological, chronic medical, learning, sensory, or physical. The University will adhere to all applicable federal, state and local laws, regulations and guidelines with respect to providing reasonable accommodations as required to afford equal educational opportunity. It is the student's responsibility to register with Student Support Services as early as possible and to provide faculty members with the formal communication for suitable accommodations. Visit Pratt 410, 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.

Schedule

We will cover these topics:

1. Overview of compilers: purpose, examples, phases of compilation, and intermediate representations.
2. Formal language theory: Chomsky hierarchy, regular expressions, BNF grammars, and left factoring.
3. Lexical analysis and parsing: bottom-up, top-down (recursive descent), look-ahead, parser generators including ANTLR.
4. Tree representations and algorithms: parse tree, abstract syntax tree, homogeneous vs. heterogeneous node representation, traversals, visitor and listener patterns, pattern matching, and rewriting.
5. Semantic analysis: type checking, type inference, type promotion, symbol tables, scopes, forward references, mutual and open recursion, strongly-connected components, and topological sort.
6. Optimization of intermediate representation: three address code, SSA form, common sub-expression elimination, constant propagation, dead code elimination.
7. Code generation and transformations toward target language: tail call elimination, closure conversion, instruction selection, register allocation.

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

Wed Jan 20 Meeting (F2F) at 6 pm.
Wed Jan 27 Meeting (F2F) at 6 pm.
Thu Jan 28 Assignment 1 due at 23:59.
Wed Feb 3 Meeting (Online) at 6 pm.
Fri Feb 5 Assignment 2 due at 23:59.
Wed Feb 10 Meeting (F2F) at 6 pm.
Tue Feb 16 Assignment 3 due at 23:59.
Wed Feb 17 Meeting (Online) at 6 pm.
Wed Feb 24 Meeting (F2F) at 6 pm.
Tue Mar 1 Assignment 4 due at 23:59.
Wed Mar 16 Meeting (F2F) at 6 pm.
Tue Mar 22 Assignment 5 due at 23:59.
Wed Mar 23 Meeting (Online) at 6 pm.
Wed Mar 30 Meeting (F2F) at 6 pm.
Wed Apr 6 Meeting (F2F) at 6 pm.
Wed Apr 13 Assignment 6 due at 23:59.
Wed Apr 20 Meeting (F2F) at 6 pm.
Tue Apr 26 Assignment 7 due at 23:59.
Wed Apr 27 Meeting (Online) at 6 pm.
Wed May 4 Meeting (F2F) at 6 pm.
Fri May 6 Assignment 8 due at 23:59.