

COLLEGE OF ENGINEERING, UIC
CS 473, Compiler Design, 4/3 (grad / undergrad)

I. Instructor & Course Details

Prof. Luís Pina

Email address: luispina@uic.edu (you should use Piazza for any class-related communications)

Drop-In Office Hours:

- Remote only (via Zoom): Friday 2pm-3pm
- Hybrid (in-person + Zoom): Monday and Wednesday 3-4pm
 - Priority will be given to students attending office hours in-person
- It is possible to schedule extra office hours by appointment only

Drop-In Hours location:

- Zoom link:
<https://uic.zoom.us/j/84483006648?pwd=PyFfHT56hQPyFP9GI2aYvFekeioXv0.1>
- In-person: CDRLC 4448

Joseph Wiseman (TA)

Email address: You should use Piazza for any class-related communications

Drop-In Office Hours:

- Hybrid (in-person + Zoom): Tuesday 2-3pm
- Remote only (via Zoom): Thursday 2-3pm

Drop-In Hours location:

- Zoom link:
<https://uic.zoom.us/j/84483006648?pwd=PyFfHT56hQPyFP9GI2aYvFekeioXv0.1>
- In-person: CDRLC 2404

Course Modality and Schedule

Class meets in-person at 12h30noon to 1h45pm in **CDRLC Room 1406 SH Room 304** every Monday and Wednesday from August 25 2025 to December 3 2025.

II. Course Information

Catalog Course Description

The purpose of this course is to learn how to transform programs written in common user-level programming languages into code that can be executed by the processor. We will cover all the stages of a compiler, from processing program text into an internal representation, through optimization and program improvement, to the generation of assembly code.

Growth Mindset:

Course materials and assignments can be complex and challenging, but they are crucial to your intellectual and personal growth and development. There are times you may need extra help. Students who attend class consistently, complete all assignments, thoughtfully engage with feedback on work, develop good study strategies, visit the tutoring center, and contact faculty when they are struggling can develop a thorough understanding of the course material and ultimately succeed in the course!

Prerequisites

- Basic understanding of the software development steps: compilation, linking, debugging, etc.
- Students are expected to have completed CS 301 (Languages and Automata), CS 251 (Data Structures), and CS 261 (Machine Organization) with a grade of C or better, or its equivalent from other universities.
- You should have a good knowledge of C. This course will have a sizable portion of programming in the C language, and we will assume a decent C proficiency.

Course Goals and Learning Outcomes

By the end of this course, students should:

- **Understand** how modern compilers shape programming languages.
- **Know** the fundamental building blocks of modern compilers.
- **Be able to write** code that transforms program text (high-level code) into a data-structure (data) that can be used by a compiler.
- **Be able to write** code that transforms a suitable data-structure (data) into assembly language (low-level code).
- **Understand** existing program analysis techniques, their limitations, and write your own.

Course topics: Lexical Analysis, Regular expressions, Deterministic and non-deterministic finite automata, Parsing, Grammars: LL, LR(0), LR(1), LALR(1), Abstract Syntax Trees (ASTs), Symbol resolution, Type checking, Function frames, Stack machines, Assembly code: MIPS, Translating ASTs to assembly language, Intermediate representation (IR), Translating IR to assembly language, Liveness analysis. Register allocation. Compiler optimizations, Dataflow analysis

This is an elective course in the CS curriculum.

Required and Recommended Course Materials

Required textbook: Andrew W. Appel, Modern Compiler Implementation in C, Cambridge University Press, 1999.

You don't need to read the textbook before class. A [quick google search](#) shows many hits for the draft version of the book (which can be used for class).

Extra optional materials may be provided by the instructor during the course

Respect for Copyright

Please protect the copyright integrity of all course materials and content. Please do not upload course materials onto third-party websites or share content with anyone not enrolled in our course.

III. COURSE POLICIES & CLASSROOM EXPECTATIONS

Grading Policy and Point Breakdown

You must complete Assignment 0 and the Syllabus Quiz during the first 2 weeks of class to receive a grade higher than F in this course:

- **Assignment 0** ensures you understand how to start and submit the programming assignments. It consists of following a video tutorial and: (1) Starting an assignment on Github Classroom; (2) Checking your assignment grade through the autograder; (3) Performing a trivial change to the original code given the feedback on the autograder; (4) Submitting the updated code; and (5) Checking your final grade on the autograder.
- **Syllabus Quiz** ensures that you read and understand this document by answering 20 questions, with a focus on the logistics and academic integrity.

Grade components:

Undergraduate Students	Rubric	Graduate Students
15%	Attendance	15%
50%	Programming Assignments	50%
15%	Midterm exam	15%
20%	Final exam	20%
100%	TOTAL	100%

Attendance: Attendance will be measured using one of two alternatives:

- (1) **In-person clicker-style quizzes during class.** Attend the class in-person and answer all clicker-style questions via iClicker. For full marks on one class you have to answer 80% of all questions, regardless of correctness. This means that incorrect answers count as much as correct answers.
- (2) **Remote attendance:** Attendance via Zoom is allowed provided that the majority of students attend in-person. Students should use Zoom as an exception for days you cannot be on campus. If in-person attendance drops below 50%, iClicker availability via Zoom will be disabled.
- (3) **Asynchronous offline lecture-capture watch.** A quiz will be made available after each class on Gradescope, in which you have to answer a subset of the clicker-questions correctly as you watch the lecture recording.

- a. When multiple options are correct, you have to select them all. This should be clear when following the lecture recording.
 - b. Some clicker questions in class may be skipped in the quiz. Echo360 will have a modified slide deck in which each Gradescope question is clearly marked.
 - c. The quiz must be completed until Sunday at 5pm of the same week as the class.
- (4) For full attendance, you have to get full marks on 80% of all classes by either attending them in-person or filling out the respective offline quizzes (you don't have to do both). You can miss 4 classes (equivalent to two full weeks of lectures) and still get full attendance.
- a. Example for undergraduate students:
 - i. If you miss 4 classes, your attendance will be 15%
 - ii. If you miss 5 classes, your attendance will be $(27 - 5) / 27 \times 15\% = 81\% \times 15\% = 12\%$

Programming assignments will be made available and submitted through Github. You need to create an account with Github and fill out a form with your account name; the URL of the form will be posted on Piazza.

- To submit a programming assignment, you simply push your changes to the repository that will be created for you. There will be an autograder made available through Github. The autograder will run every time you submit new code, and it will generate a grade based on how many tests your submission passes.
- The date of submission is the date of the last commit in your repository.
- This course will use the IntelliJ CLION as the editor for your programming assignments. CLION is free for students (you have to register for an account first), available for all common platforms (e.g., OSX, Linux, and Windows) and supports editing, debugging, and submitting your assignments. You are responsible for installing CLION on your machine. We will cover in class how to import, edit, and submit your assignments.
- A portion of each assignment may be based on questions or screencast videos (submitted through Gradescope) that require you to explain a particular aspect of your submission (e.g., how do you parse complex strings with octal numbers in them).
 - Uploading videos to Gradescope may take some time, so you should not leave this to the very last minute. Please contact the instructors through Piazza if you experience technical difficulties to upload your submission.
 - You can record such screen captures for free using Zoom to start a meeting without anyone else, sharing your screen, and recording the meeting to disk.
 - Technical difficulties reported 2h or less before the deadline will result in a zero grade for the screencast component of that assignment.
- Graduate students will have more work to do than undergraduate students, to reflect the difference in ECTS
- **Assignments split in half:** Some assignments have 2 deadlines, you should complete the first half by the first deadline and the second half by the second deadline:
 - You cannot get points for the first half in the second deadline
 - The whole assignment will be released at the same time, including both halves

- Any work you complete for the second half by the first half deadline will receive bonus points
- The second half will require you to complete the first half as a starting base (see below for resubmission policies)

- Assignments will be released and due according to the following schedule:

Assignment	Release date	Due date
0	08/25/2025	09/06/2025 @ 5pm
1	09/03/2025	09/13/2025 @ 5pm
2	09/17/2025	09/27/2025 @ 5pm
3.1	10/02/2025	10/11/2025 @ 5pm
3.2		10/18/2025 @ 5pm
4.1	10/22/2025	11/01/2025 @ 5pm
4.2		11/09/2025 @ 5pm
5.1	11/12/2025	11/22/2025 @ 5pm
5.2		12/07/2025 @ 5pm

Exams will take place in-person.

- Midterm exam: During class hours (11am-12h15noon) on October 15th 2025
- Final exam: Time, date, and location to be determined

Policy for Missed or Late Work

Assignment resubmission: After a due date, the instructors will post a screen-capture video of the instructors solving the assignment (full or the current half) from the starting code to a 100% grade. You can follow this video and update your submission to fix what you got wrong. You can then resubmit your assignment and get credit for that.

- You can resubmit your assignment within one week after the due date, until the next Saturday at 5PM
- There is no minimum grade to qualify for resubmission, but your overall assignment average needs to be higher than 65%, otherwise you will receive an F grade
- Your final grade is the average of both submissions
 - Original submission 80%, resubmission 100%, results in 90%
- Resubmissions are completely optional, you can keep the original submission grade if you don't resubmit your assignment.

Exams: No makeup exams and quizzes will be permitted unless authorized by the CS director of (under)graduate advisor or the department head.

Attendance / Participation Policy

The **attendance policy** is described above.

Participation bonus: Each programming assignment will have a bonus component based on Piazza participation. Each **question/answer** that you contribute that is tagged with the assignment tag and marked as good by an instructor **is worth 5%**. Each **duplicated** question

you find by answering with the original question and answer, under the same circumstances as above, **is worth 5%**. You will be able to get up to 10% or more in bonus points for each assignment.

- Finding typos and trivial mistakes on the assignment specs will not result in bonus points, unless the mistake can impact how students understand the assignment (e.g., finding a copy-paste error showing an incorrect due date results in bonus points, but a minor typo where **ClassName** written as **Classname** does not result in bonus points)
- Bonus points count towards the original submission when resubmitting:
 - Original submission 30%, 5% bonus points, resubmission 100%, results in 67.5% (the original submission is considered to be 35%)

Early completion bonus: For assignments split in half, any work for the second half that you complete before the first deadline will receive bonus points

Other Course Policies

Use of AI assistants

You are required to be the author of all the code you submit for the assignments. Submitting code not authored by you (e.g., copy/pasted from an AI tool) violates the academic integrity policy (described below) and will result in applying all the penalties. When performing plagiarism checks, instructors will include code copied directly from common AI assistants submission to detect such cases.

Academic Integrity

As a student and member of the UIC community, you are expected to adhere to the Community Standards of [academic integrity](#), accountability, and respect. Please review the [UIC Student Disciplinary Policy](#) for additional information.

Unless stated otherwise, all work submitted for grading must be done individually. While instructors encourage you to talk to your peers and learn from them, this interaction must be superficial with regards to all work submitted for grading. This means you cannot work in teams, you cannot work side-by-side, you cannot submit someone else's work (partial or complete) as your own.

All the submissions for programming assignments will be checked for plagiarism using an automated tool that checks each submission against all other submissions. The tool is sophisticated enough to detect variable renaming, code transformation (turning a for loop into a while loop, inverting an if statement, moving functions/methods to a different place/file, etc.) and I will investigate personally submissions that are classified as much more similar than average.

Violation of academic integrity as described in this document on a programming assignment will result in an immediate zero grade for that assignment, a one letter grade drop from your final grade (e.g., from B to C, or from D to F), and a report filed with the Dean of Students for all parties involved.

Attempting to claim undeserved bonus points, (e.g., by modifying an existing and already endorsed Piazza post), will result in losing all bonus points in this course.

Academic dishonesty is unacceptable, and penalties can result in expulsion from the university. All cases are handled via the official student conduct process described at <https://dos.uic.edu/conductforstudents.shtml>.

In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. You may not solicit code from family, friends, or online forums. Other examples of academic dishonesty include emailing your program to another student, working in a group on a homework assignment, and allowing a tutor, TA, or another individual to write an answer for you. It is also considered academic dishonesty if you click someone else's iClicker with the intent of answering for that student.

If you are struggling with a particular part of your assignments, post a question on Piazza. Other students may be struggling as well and benefit from the discussion around your question. If you see a student asking a question that you know how to answer with a small code example carefully curated from your assignment (e.g., how to use an API to parse a file), it is acceptable to share that code. Use your common sense to not disclose important parts of the assignment. For instance, if you are struggling with Java but not with the concepts of an assignment, showing/sending your code to a fellow student is considered a violation of academic integrity. Posting a question on Piazza about that same issue is rewarded with bonus points.

Use of shared computers: The use of shared computers may lead to inadvertently accessing someone else's code for an assignment, which will be counted as plagiarism. A shared computer is a computer that is accessible potentially by many students. Your own personal computer can become a shared computer if you allow a fellow student to use it in the context of this course.

Examples of shared computers include (but are not limited to):

- A public access computer: Library, lab, printer, hotel, etc.
- Using your credentials on another student's computer: Downloading your code to another student's computer, logging on your account on Blackboard, Piazza, Github, or Travis on another student's computer.
- Allowing another student to use your computer

If you use your credentials on another student's computer, then fail to log-off securely, and that other student accesses your answers, both are guilty of academic dishonesty. I suggest using incognito/private mode in modern browsers when using your credentials on any computer that may be accessed by someone else.

Using shared computers can lead to complications that are considered plagiarism. When using shared computers, play close attention to each submission of each assignment. For instance, the following situations can happen:

- Alice lends Bob her computer so that Bob can check his grade. Later, Alice submits her assignment using Bob's credentials by mistake. Both Alice and Bob are guilty of plagiarism.
- Charlie uses Dana's computer to submit his assignment using IntelliJ. Later, Dana is struggling with Github and IntelliJ replaces one of Dana's files automatically with files from Charlie's assignment. Dana submits the file without realizing. Both Charlie and Dana are guilty of plagiarism.

Piazza Expectations

Students are responsible for all information instructors publish on Piazza. Faculty messages should be regularly monitored and read in a timely fashion.

Instructors may need to communicate directly with you via a private Piazza message. Instructors will assume that you have read all private Piazza messages within 24h.

IV. COURSE SCHEDULE

Weekly Schedule of Class Topics, Assignments, Assessments, Due Dates, and Deadlines

Week 1: Overview of what a compiler is and does

Week 2: Lexing

Week 3: Parsing – Context Free Grammars, yacc, and disambiguation

Week 4: Parsing – LL, LR(0), LR(1), LALR(1)

Week 5: Parsing – Conflicts, Abstract Syntax Trees (ASTs)

Week 6: Semantic analysis – Symbol resolution

Week 7: Semantic analysis – Type checking

Week 8: Intermediate Representation (IR) – AST Transformation, Translation, Basic Blocks

Week 9: Code generation and Stack Machines

Week 10: Code generation – MIPS assembly
Week 11: Code generation – Stack frames
Week 12: Register Allocation
Week 13: Analysis and Optimization – Loop optimization
Week 14: Analysis and Optimization, Compiler testing
Week 15: Conclusion and Research Topics
Week 16: Final exam week

Disclaimer

This syllabus is intended to give the student guidance on what may be covered during the semester and will be followed as closely as possible. However, as the instructor, I reserve the right to modify, supplement, and make changes as course needs arise. I will communicate such changes in advance through in-class announcements and in writing via Blackboard Announcements.

V. ACCOMMODATIONS

Disability Accommodation Procedures

UIC is committed to full inclusion and participation of people with disabilities in all aspects of university life. If you face or anticipate disability-related barriers while at UIC, please connect with the Disability Resource Center (DRC) at drc.uic.edu, via email at drc@uic.edu, or call (312) 413-2183 to create a plan for reasonable accommodations. To receive accommodations, you will need to disclose the disability to the DRC, complete an interactive registration process with the DRC, and provide me with a Letter of Accommodation (LOA). Upon receipt of an LOA, I will gladly work with you and the DRC to implement approved accommodations.

Religious Accommodations

Following [campus policy](#), if you wish to observe religious holidays, you must notify me by the tenth day of the semester. If the religious holiday is observed on or before the tenth day of the semester, you must notify me at least five days before you will be absent. Please submit [this form](#) by email with the subject heading: **“YOUR NAME: Requesting Religious Accommodation.”**

VI. CLASSROOM ENVIRONMENT

Inclusive Community

UIC values diversity and inclusion. Regardless of age, disability, ethnicity, race, gender, gender identity, sexual orientation, socioeconomic status, geographic background, religion, political ideology, language, or culture, we expect all members of this class to contribute to a respectful,

welcoming, and inclusive environment for every other member of our class. If aspects of this course result in barriers to your inclusion, engagement, accurate assessment, or achievement, please notify me as soon as possible.

Name and Pronoun Use

If your name does not match the name on my class roster, please let me know as soon as possible. My pronouns are [\[she/her; he/him; they/them\]](#). I welcome your pronouns if you would like to share them with me. For more information about pronouns, see this page:

<https://www.mypronouns.org/what-and-why>.

Community Agreement/Classroom Conduct Policy

- Be present by turning off cell phones and removing yourself from other distractions.
- Be respectful of the learning space and community. For example, no side conversations or unnecessary disruptions.
- Use preferred names and gender pronouns.
- Assume goodwill in all interactions, even in disagreement.
- Facilitate dialogue and value the free and safe exchange of ideas.
- Try not to make assumptions, have an open mind, seek to understand, and not judge.
- Approach discussion, challenges, and different perspectives as an opportunity to “think out loud,” learn something new, and understand the concepts or experiences that guide other people’s thinking.
- Debate the concepts, not the person.
- Be gracious and open to change when your ideas, arguments, or positions do not work or are proven wrong.
- Be willing to work together and share helpful study strategies.
- Be mindful of one another’s privacy, and do not invite outsiders into our classroom.

Content Notices and Trigger Warnings

Our classroom provides an open space for a critical and civil exchange of ideas, inclusive of a variety of perspectives and positions. Some readings and other content may expose you to ideas, subjects, or views that may challenge you, cause you discomfort, or recall past negative experiences or traumas. I intend to discuss all subjects with dignity and humanity, as well as with rigor and respect for scholarly inquiry. If you would like me to be aware of a specific topic of concern, please email or visit my Student Drop-In Hours.

VII. RESOURCES: Academic Success, Wellness, and Safety

We all need the help and the support of our UIC community. Please visit my **drop-in hours** for course consultation and other academic or research topics. For additional assistance, please contact your assigned college advisor and visit the support services available to all UIC students.

Academic Success

- UIC [Tutoring Resources](#)
- College of Engineering [tutoring program](#)
- [Equity and Inclusion in Engineering Program](#)
- UIC Library and UIC Library Research Guides.
- [Offices](#) supporting the UIC Undergraduate Experience and Academic Programs.
- [Student Guide for Information Technology](#)
- [First-at-LAS](#) Academic Success Program, focusing on LAS first-generation students.

Wellness

- **Counseling Services:** You may seek free and confidential services from the Counseling Center at <https://counseling.uic.edu/>.
- Access [U&I Care Program](#) for assistance with personal hardships.
- **Campus Advocacy Network:** Under Title IX, you have the right to an education that is free from any form of gender-based violence or discrimination. To make a report, email TitleIX@uic.edu. For more information or confidential victim services and advocacy, visit UIC's Campus Advocacy Network at <http://can.uic.edu/>.

Safety

- [UIC Safe App](#)—PLEASE DOWNLOAD FOR YOUR SAFETY!
- [UIC Safety Tips and Resources](#)
- [Night Ride](#)
- [Emergency Communications](#): By dialing 5-5555 from a campus phone, you can summon the Police or Fire for any on-campus emergency. You may also set up the complete number, (312) 355-5555, on speed dial on your cell phone.

VIII. Change log

- 08/20/2025: Document available
- 08/21/2025: Updated lecture room
- 08/23/2025: Added TA