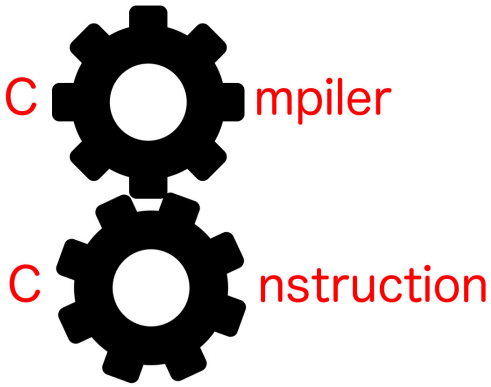

The compiler is the programmer's primary tool. Understanding the compiler is therefore critical for programmers, even if they never build one. Furthermore, many design techniques that emerged in the context of compilers are useful for a range of other application areas. This course introduces students to the essential elements of building a compiler: parsing, context-sensitive property checking, code linearization, register allocation, etc. To take this course, students are expected to already understand how programming languages behave, to a fairly detailed degree. The material in the course builds on that knowledge via a series of semantics preserving transformations that start with a fairly high-level programming language and culminate in machine code.

Course Code: COMP_SCI 322

When: Winter

Instructor:
Simone Campanoni, `simone.campanoni@northwestern.edu`

Course Objectives:
Students that complete this course should:

1. understand how to efficiently implement a programming language,
2. have an accurate performance model for the primitives in a programming language,
3. have a good sense for the kind of assembly code that a compiler produces
4. be able to maintain and extend a compiler for a programming language

Assignments:
In this class, you will learn how to design and efficiently implement a compiler able to generate Intel x86_64 machine code from a high level programming language (a modern C-based language). The work is divided in eight assignments, one per week. Subsequent assignments build on top of their previous one.

Each compiler implemented by each team will be checked in at least two ways.

1. All the tests included in the framework distributed via Canvas will be checked automatically.
2. Simone will inspect the source code.

3. If you will present your code in a panel (see below), then the manager(s) will double check its correctness.

Completing all assignments will obtain an overall compiler capable to translate a program written in a C-like language to a semantically-equivalent Intel x86_64 executable binary. Such binaries will need to run correctly on real Intel-based platforms running a Linux OS. Furthermore, the generated binaries will be able to compete with production-quality compilers like `gcc` and `clang`.

Competition:
Compilers built by successful teams will compete at the end of the class. The team that has designed and implemented the compiler that generates the most performant binary will win. The students that compose the team that wins the competition will get an A independently on how many points they have accumulated. Also, the winners will have the option to include their names and pictures in the **Hall of Fame** of this class (see the winners of prior years here).

Panels:
Every assignment is going to be evaluated in class with a code walk just after its deadline. To do so, a code walk is organized as a panel where a team describes their compiler, a manager asks questions to guarantee the correctness of the design and implementation of such compiler, manager helpers help the manager to do his/her job, and a secretary writes down all important discussions. Only teams that have submitted their correct solution on time can be selected to participate to the panels.

We will have up to five panels for every assignment. The first three panels will be composed only by teams. Simone will describe his code as the last panel and the entire class will act as managers.

Materials:
Course slides
Recommended book: Modern compiler implementation in C (or Java).
Referenced book: Compilers: Principles, Techniques, and Tools (2nd Edition)

Grading Policy:
Your grade depends on the points you will earn on the homework assignments as well as on the code walks you will perform during panels. You can earn up to eight points for each homework assigned (one per assignment). Also, you can earn up to three points for the panelist experiences. Finally, you earn an extra point if you submit the final compiler on time for the final competition and such compiler passes all tests.

The map between points and grades is the following:

A	11 (and above)
A-	10
B+	9
B	7
B-	7
C+	6
C	5
C-	4
D	3
F	0 - 2

Notes:
This course satisfies the Systems breadth of the CS Department.
This course requires significant C++ programming, but you do not need to be a C++ expert to take this class.

The quality of the C++ code you will write will not be evaluated. However, each of your assignment will be based on your previous one. So, if you write code that is hard to maintain, then you will make subsequent assignments harder for yourself.

Pre-requisites:

- CS 213: Introduction to Computer Systems (or equivalent)
- CS 214: Data Structures and Data Management

While we assume you know and remember the concepts thought in the classes listed above, peer mentors will run tutorials during their first half of their office hours to refresh your memory. While the participation to these tutorials is not mandatory, it is strongly suggested.

Recommended classes:

- CS 321: Programming Languages

Other compiler-heavy classes:

- CS 323: Code Analysis and Transformation
- CS 397/497: Advanced Topics in Compilers

Academic Integrity Students in this course are required to comply with the policies found in the booklet, "Academic Integrity at Northwestern University: A Basic Guide". All papers submitted for credit in this course must be submitted electronically unless otherwise instructed by the professor. Your written work may be tested for plagiarized content. For details regarding academic integrity at Northwestern or to download the guide, visit: https://www.northwestern.edu/provost/policies/academic-integrity/index.html

Accessibility Northwestern University is committed to providing the most accessible learning environment as possible for students with disabilities. Should you anticipate or experience disability-related barriers in the academic setting, please contact AccessibleNU to move forward with the university's established accommodation process (e: accessiblenu@northwestern.edu; p: 847-467-5530). If you already have established accommodations with AccessibleNU, please let me know as soon as possible, preferably within the first two weeks of the term, so we can work together to implement your disability accommodations. Disability information, including academic accommodations, is confidential under the Family Educational Rights and Privacy Act.

COVID-19 Testing Compliance To protect the health of our community, Northwestern University requires unvaccinated students who are in on-campus programs to be tested for COVID-19 twice per week.

Students who fail to comply with current or future COVID-19 testing protocols will be referred to the Office of Community standards to face disciplinary action, including escalation up to restriction from campus and suspension.

COVID-19 Classroom Expectations Statement Students, faculty, and staff must comply with University expectations regarding appropriate classroom behavior, including those outlined below and in the COVID-19 Code of Conduct. With respect to classroom procedures, this includes:

- Policies regarding masking and social distancing evolve as the public health situation changes. Students are responsible for understanding and complying with current masking, testing, Symptom Tracking, and social distancing requirements.
- In some classes, masking and/or social distancing may be required as a result of an Americans with Disabilities Act (ADA) accommodation for the instructor or a student in the class even when not generally required on campus. In such cases, the instructor will notify the class.
- No food is allowed inside classrooms. Drinks are permitted, but please keep your face covering on and use a straw.
- Faculty may assign seats in some classes to help facilitate contact tracing in the event that a student tests positive for COVID-19. Students must sit in their assigned seats.

If a student fails to comply with the COVID-19 Code of Conduct or other University expectations related to COVID-19, the instructor may ask the student to leave the class. The instructor is asked to report the incident to the Office of Community Standards for additional follow-up.

Exceptions to Class Modality Class sessions for this course will occur via Zoom for the first two weeks. Then, depending on how the pandmic will evolve, the class will become hybrid.

Prohibition of Recording of Class Sessions by Students Unauthorized student recording of classroom or other academic activities (including advising sessions or office hours) is prohibited. Unauthorized recording is unethical and may also be a violation of University policy and state law. Students requesting the use of assistive technology as an accommodation should contact AccessibleNU. Unauthorized use of classroom recordings – including distributing or posting them – is also prohibited. Under the University's Copyright Policy, faculty own the copyright to instructional materials – including those resources created specifically for the purposes of instruction, such as syllabi, lectures and lecture notes, and presentations. Students cannot copy, reproduce,

display, or distribute these materials. Students who engage in unauthorized recording, unauthorized use of a recording, or unauthorized distribution of instructional materials will be referred to the appropriate University office for follow-up.

Support for Wellness and Mental Health Northwestern University is committed to supporting the wellness of our students. Student Affairs has multiple resources to support student wellness and mental health. If you are feeling distressed or overwhelmed, please reach out for help. Students can access confidential resources through the Counseling and Psychological Services (CAPS), Religious and Spiritual Life (RSL) and the Center for Awareness, Response and Education (CARE). Additional information on all of the resources mentioned above can be found here:

https://www.northwestern.edu/counseling/
https://www.northwestern.edu/religious-life/
https://www.northwestern.edu/care/