

# CS 6410: Compilers

Fall 2023

Tamara Bonaci

[t.bonaci@northeastern.edu](mailto:t.bonaci@northeastern.edu)

Thank you to UW faculty Hal Perkins for all the help and inspiration in preparing these course materials and assignments.

## CS 6410 – Compilers, Fall 2023

- **Time:** Wednesdays from 10:00 – 1:00pm PT online, using NEU Zoom
- **Instructor:** Tamara Bonaci (t.bonaci@northeastern.edu)
  - **Office hours:** TBD
  - **Virtual office hours:** by appointment
- **Course material:** Canvas
- **Homework assignment submission:** Canvas
- **Project assignment submission:** Through Khoury GitHub
- **Course discussion board:** Canvas
- **Assignment grades:** Canvas

# What is CS 6410, Fall 2023?

- Graduate-level introductory compilers course
- **Focus:** Principles and practice of building efficient implementations of modern programming languages
- **Course goals:**
  - Understand a structure of a modern compiler
  - Analyze the major algorithms used to translate code from high-level to machine language
- **Learning mechanism:** building a simplified compiler

## CS 6410 - Prerequisites

- Nominal prerequisites for a compiler course:
  - Data structures and algorithms
  - Formal languages and automata
  - Machine organization
- **But we will review these topics as we need them**
- If you're missing one of the background courses, it is normally possible to work around it (but you may have to fill some of the gaps on your own)
- So, an eagerness to learn is the most important (and expected 😊)

## CS 6410 – Expected Course Progression

**Week 1 (September 6):** Course overview. Introduction to compilers

**Week 2 (September 13):** Languages, automata and regular expressions

**Week 3 (September 20):** Scanners

**Week 4 (September 27):** Grammars and LR parsing

**Week 5 (October 4):** LR construction, LL parsing, AST/visitor

**Week 6 (October 11):** Static semantics, type checkers, and symbol tables

**Week 7 (October 28):** x86-64. Runtime organization

**Week 8 (October 25):** Code shape

**Week 9 (November 1):** Optimization and transformation

**Week 10 (November 8):** Value numbering and data flow

**Week 11 (November 18):** Loops and SSA

**Week 12 (November 22):** No class – Thanksgiving week.

**Week 14 (November 29):** Code generation. Instruction selection and scheduling.

**Week 14 (December 6):** Registers allocation. Inlining and dynamic languages

**Finals week (December 13):** Memory models. Garbage collection (optional lecture)

## CS 6410 – Course Logistics

- **Course will be graded based upon:**
  - Quizzes – 10%
  - Homework – 30%
  - Project – 60%

## Course Logistics – In-Class Quizzes

- **Quizzes** – simple quizzes (activities) occasionally done after the class
- **Purpose:**
  - To review class material
  - To facilitate discussion
- **Grading scale - 0-2, where:**
  - 0 - missed or irrelevant answer
  - 1 - relevant answers submitted
  - 2 - good and interesting answers
- There may theoretically be an in-class activity every week, but we will take seven best scores when determining your grade

## Course Logistics – Homework

- **Homework** – six assignments, consisting of **written problems**, intended to reinforce the understanding of concepts and ideas presented in class
- **Expected topics for homework assignments:**
  - HW1 – Regular expressions
  - HW2 – LR grammars
  - HW3 – LL grammars
  - HW4 – Static semantic
  - HW6 – Assembly, code shape
  - HW6 – SSA, Value numbering and dataflow



## Course Logistics – Project

- **Project** – the biggest component of this course
- **Goal:** to give you a deeper understanding of concepts relevant to modern-day compilers
- The project will be due in steps, to:
  - Keep you on schedule,
  - Give you an early feedback at crucial points
- **Some project steps:**
  - Setting up the environment
  - Building your own scanner
  - Building your own parses and AST
  - Code generation
  - Compiler addition
  - Project report

## Course Logistics – Project

- **Project** – the biggest component of this course
- **Goal:** to give you a deeper understanding of concepts relevant to modern-day compilers
- **Suggestion for the project:** work in groups of two! (While it is possible to work alone, some past experiences indicate that that may just be too much work for an individual.)

## Course Material

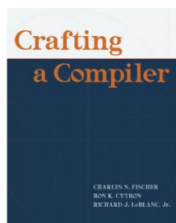
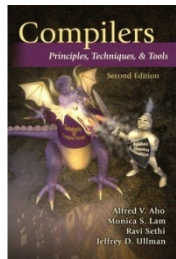
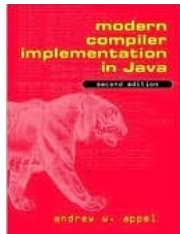
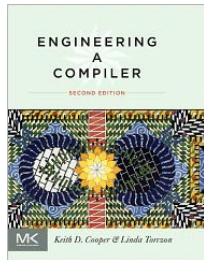
Four good books:

1. Cooper & Torczon, *Engineering a Compiler (our official text book)*

2. Appel, *Modern Compiler Implementation in Java*, 2nd ed.  
(MiniJava is from here)

3. Aho, Lam, Sethi, Ullman, “Dragon Book”

4. Fischer, Cytron, LeBlanc, *Crafting a Compiler*



## Course Material

- **If you can, try to attend lectures because:**
  - Lectures will likely cover more than provided in lecture notes, and the provided references
  - Lectures will focus on “big-picture” principles and ideas
  - Your colleagues will likely start interesting discussions during lectures
  - In-class activities and discussions they will start

## Late Turn In Policy

- All assignments are due **by 11:59pm on the assigned date**
  - Homework submission through Northeastern Canvas
  - Project steps through Khoury GitHub
- Late assignments will (generally) be dropped 10% per calendar day, and no submissions will be accepted after 7 days
- If you have a meaningful reason for delay (e.g., illness) – ***come and talk to me***
- **Exception to the late turn in policy:** final project report should to be turned in on time

# Academic Integrity

- **We want a collegial group helping each other succeed!**
- But you must never misrepresent work done by someone else as your own, without proper credit if appropriate, or assist others to do the same
- You may want to familiarize yourselves with the Northeastern University Office of Student Conduct and Conflict Resolution (OSCCR), at <http://www.northeastern.edu/osccr/academicintegrity/index.html>.

# Academic Integrity

- **In conclusion:**

We trust you to behave ethically because honest work is the most important feature of a university (or engineering or business)

# Credits For Course Material and Assignment

- **UW CSE faculty member, Hal Perkins**
- Some direct ancestors of this course:
  - UW CSE 401 (Chambers, Snyder, Notkin, Perkins, Ringenburt, Henry, ...)
  - UW CSE PMP 582/501 (Perkins)
  - Cornell CS 412-3 (Teitelbaum, Perkins)
  - Rice CS 412 (Cooper, Kennedy, Torczon)
  - Many books (Appel; Cooper/Torczon; Aho, [[Lam,] Sethi,] Ullman [Dragon Book], Fischer, [Cytron ,] LeBlanc; Muchnick, ...)



## Your Questions



[Picture credit: quickmeme.com]