CS 4240: Compilers and Interpreters
Spring 2027

Instructor: Prof. Qirun Zhang

Updated on: October 21, 2026

1 Overview

This course will cover the principles and foundations of compiler design. The course project will be structured around the design of a compiler for a small, but expressive, imperative language to be translated for execution on a simulator of the MIPS processor. After successfully completing the course, students will be able to design and develop all major components of a compiler. In particular, students will be able to design and develop:

- A compiler "front end" for input source code, which includes a scanner, a parser, and semantic analysis to generate intermediate representations (IRs) of code. Students will understand the foundations of scanner generators based on regular expressions and finite automata, as well as the foundations of parser generators based on context-free languages and LL/LR grammars.
- A compiler "middle end", which processes the IR used to bridge between the front end and the back end, while also "optimizing" the code in the process. Students will understand the foundations of systematic code optimizations on IRs using data flow analyses.
- A compiler "back end" for generating output machine code from the IR. Students will be able to design phases that start with an intermediate representation, and perform instruction selection, register allocation, code layout, code generation, and instruction scheduling.

2 Pre-requisites

Students must have taken *CS 1332* and *CS 2200* (or *ECE 3058*), and earned a grade of C for each course, at minimum.

3 Resources

References to all resources required for the course, including information for the course textbook and lecture slides, will be maintained in the Piazza and Canvas sites for the course. The course textbook is Cooper and Torczon's *Engineering a Compiler* (second edition).

4 Grading

Your final grade will be the weighted sum of grades collected from work in the following categories:

- **Written Homeworks (15%)** will consist of three exercises. They are designed to be completed using material from the textbook and the course Piazza site. However, they must be completed individually with no help from anyone else. Unlike in a traditional class setting, students will complete their homework questions and submit them in class. Each exercise will contain a combination of small-scale written assignments, with pseudocode programming in some cases.
- **Programming Projects (30%)**: each project will contribute towards a phase (front-end, middle-end, back-end) of a compiler for a simple subject language. Work can be performed in teams of two to three students.
- **Midterm (20%)**: an in-class examination that will cover all topics studied up to the last lecture before the midterm.
- **Final (30%)**: registrar-scheduled final examination that will cover all topics presented in the course.
- **Class Participation (5%)**: assessed based on your level of participation in lectures, in-class worksheets, and on Piazza. For worksheets, we will record 10 worksheets from each student (4%). There is also an "Assignment 0" mostly related to course logistics (1%).
- **Project BONUS (5%)**: an extension to project 2.

5 Policies

5.1 Class Participation

All students are expected to be respectful of other students, and to help contribute to an improved course experience for all students, both in the classroom and on Piazza. No strict guidelines will be enforced regarding whether students attend class on time or at all, or on how they use the Internet in lecture. However, all of these choices will factor into the **Class Participation** component of their grade.

5.2 Honor Code

All students are expected to follow the Georgia Tech Honor Code.

5.3 Late Submissions

- Within 1 hour after the assignment is due: You can email the TAs to request removal of the late submission penalty, provided the request is made within 24 hours of your *submission time*. If you do not send this request, your work will be treated as a late submission;
- Within 24 hours after the assignment is due: 50% point deduction;
- More than 24 hours late: 100% point deduction.

No late submission exceptions will be permitted.

5.4 Collaboration Policy

Students are expected to complete all coursework independently, except for in-class worksheets, which are encouraged to be done collaboratively with other students, and for group projects for which collaboration is only permitted within a group. Should any team member wish to join a new group for Project-X (X=1,2,3), they are required to communicate their request to both fellow team members and the teaching staff within one week of the corresponding Project-X release date. While students are, of course, welcome to discuss the course material outside of class, the material covered in homework is best understood when students

discover the key insight behind solutions independently. Sharing insights that directly lead to the solution to exercises before they are due thus is not allowed. Any such incident is cause for reporting an honor code violation.

5.5 Accommodation

Students with documented disabilities that may affect their ability to learn the course material are encouraged to discuss them with the instructor independently, and accommodations will be made.

If a student encounters a personal emergency during the semester, they must notify the instructor immediately, defined as within one day of the incident, in order to make alternate arrangements for assignments, projects, or examinations as necessary.

For non-emergency matters that may affect the submission of any deliverable, the student must notify the instructor at least five days prior to the relevant deadline. The instructor will determine whether the matter is relevant and whether an accommodation will be granted.

All requests, whether for emergency or non-emergency situations, must be supported by official documentation from an appropriate campus office (*e.g.*, Dean of Students, Registrar's Office, ODS). No accommodation will be granted without *both* timely communication and proper verification.