

CSC 431: Programming Languages II

Instructional Information

Professor: Aaron Keen

E-mail: akeen@csc.calpoly.edu

Phone: 756-2926

Office: 14-230

Office hours: M: 12–1pm, W: 12–1pm, F: 12–2pm

Course Webpage: <http://www.csc.calpoly.edu/~akeen/courses/csc431>

Lecture Time and Location

- Section 1
 - MWF 2:10pm–3:00pm, 186-C103
 - MWF 3:10pm–4:00pm, 14-303 (lab)
 - MWF 3:10pm–4:00pm, 20-127 (overflow lab)

Course Objectives

- Explore the design and implementation of a compiler.
- Focus on issues related to the “back-end” of a compiler.
- Understand and implement code transformations.

Prerequisites: CSC 430

Texts

The recommended course textbook is *Engineering a Compiler* by Cooper and Torczon. Supplemental materials will be linked from the course webpage.

Webpage

Clarifications, changes, etc. regarding the class and assignments will be posted to the course webpage (<http://www.csc.calpoly.edu/~akeen/courses/csc431>). Read it regularly, especially near when assignments are due. You are responsible for any announcements posted on the course website.

Activities

Class Participation

The lectures are for your benefit. You should ask questions when you have them. Use lecture time to discuss general approaches to the project.

Project

There will be one large project with multiple milestones. The due dates for the milestones are listed on the schedule. You are allowed, but not required, to work with a single partner on the project.

Exams

There will be one exam. The exam will focus on material relating to the project and the analyses discussed in lecture. The exam will be open book and open note.

Paper

Each group will submit a paper detailing the design and implementation of their compiler project. This paper must

- Outline the overall architecture of the solution.
- Describe the representation of key data (e.g., control flow graphs and instructions).
- Outline optimizations implemented.
- Provide a section detailing the performance of the code generated for the benchmarks. This section should contain graphs comparing the run-times of the generated code (with and without optimizations) and the C equivalent code compiled using gcc (with and without optimizations).

Grading

The percentage breakdown for the course grade is as follows.

Activity	% per	% total
Milestone #1	4	4
Milestones #2–#5	5	20
Final Submission	46	46
Exam	10	10
Paper	20	20
Total		100

Milestones

Each milestone will be demonstrated in lab on the day that it is due. Each group must complete a status sheet (linked from the course webpage) and prepare (ahead of time) a simple demonstration of functionality.

Final Project Submission

You must submit your final project by the date specified on the schedule. This submission must include all of your source code, instructions on how to build your project, and instructions on using your compiler.

Minimal Proficiency

Your grade in the course will be primarily determined based on the weights for each activity but with a ceiling based on how well your compiler performs. Specifically, **your grade will be no better than the letter grade on a straight scale corresponding to the percentage of the benchmarks for which your compiler generates correct code.**

For example, if your compiler generates correct code for only 85% of the benchmarks, then your grade in the course will be no better than a B.

Missed Exams

Make-up or early exams will not be given except in the most extreme situations. If you must miss an exam due to extreme illness, etc., contact the instructor (by phone or by e-mail) or leave a message with the Department of Computer Science office (805-756-2824) *before* the exam. Be sure to leave both the reason for missing the exam and how to reach you.

Collaboration and Cheating

Policy on Collaboration

Each student is to do his or her own work. Students may work in pairs on the project, but not on the exam. It is fine to talk with others about general approaches used to solve the assignments, *but* each student/pair is to

develop his/her/their own solution; collaborative efforts beyond a recognized pair are **not** allowed. Students/pairs are not to view any other student's code or exchange code in any form (hardcopy or electronically). Sharing pseudo-code is not allowed.

In addition, using solutions from any other source is forbidden; in particular, using solutions (either instructors' or other students') from previous offerings of this course is not allowed. Using solutions found on the Internet is not allowed. Referring to previous solutions while developing your solution is not allowed.

Collaboration that goes beyond a high-level discussion of general approaches will be considered cheating. If you are unsure about what constitutes proper or improper collaboration, consult the instructor for guidance.

To summarize: all assignments and exams are to be *individual* (or paired) and *original* efforts.

Policy on Cheating

Don't. Any instance of cheating or plagiarism will be referred to the campus Office of Student Rights and Responsibilities. The Cal Poly rules and policies are available on the OSRR web site, <http://www.calpoly.edu/~osrr/index.html>. Ask the instructor for clarification *beforehand* if the above rules are not clear.

The Last Page

This page is so that I can gather a little information about you at the beginning of the class. Please fill it out, tear it off and leave it with me on the way out.

Who are you?

Name: _____
Section: _____
Major: _____
Email: _____
Enrollment: _____ Enrolled
 _____ Enrolled, thinking about dropping
 _____ Thinking about signing up

Class Expectations?

Please take a minute to write out what your goals and expectations are for CSC 431. What do you want to learn? What do you expect to learn? Are these the same thing?