CSC 431: Programming Languages II

Instructional Information

   Professor: Aaron Keen
   E-mail: akeen@calpoly.edu
   Office: 14-228
   Office hours: M: 10–11am, W: 10–11am, R: 1–3pm, F: 2–3pm
   Course Webpage: http://www.csc.calpoly.edu/~akeen/courses/csc431

Lecture Time and Location

   Lecture: MWF 11:10am – 12:00pm, 14-247                           Lab: MWF 12:10pm – 1:00pm, 14-301
   Lecture: MWF 3:10pm – 4:00pm, 52-E27                           Lab: MWF 4:10pm – 5:00pm, 14-301

Learning Environment

   I believe in a supportive learning environment wherein every person deserves respect, every question deserves an answer, and we all work together toward a goal of improved understanding and life-long learning. All members of this class are expected to contribute to a respectful, welcoming, and inclusive environment for every other member of the class.

   There is an old adage that encourages those with questions to ask because others may have the same question; you do not need some imagined quorum in order to validate your question or your request for assistance. If you have a question or need assistance, that alone is sufficient. You belong here and your participation in the course is not only welcomed but encouraged.

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.
   • Gain appreciation for what an optimizing compiler can do and the implications on how you write code.

Prerequisites: CSC 430

Texts

   The recommended course textbook is Engineering a Compiler by Cooper and Torczon or Compilers: Principles, Techniques, and Tools by Aho, Lam, Sethi, and Ullman. 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.

Each milestone must be demonstrated in lecture or lab on the day that it is due to earn full credit. Milestones are graded based on a somewhat subjective measure of how “complete” the required functionality is. Milestone completion may be demonstrated by the “final demonstration” date for partial credit.

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.

Exams
There will be no exams. You may rejoice now or later.

Paper
Each group will submit a paper detailing the design and implementation of their compiler project. At least half of the grade for the paper will depend on the presentation of some performance analysis of the code generated by the group’s compiler. Further details are provided on the course website.

Grading
The percentage breakdown for the course grade is as follows.

<table>
<thead>
<tr>
<th>Activity</th>
<th>% per</th>
<th>% total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milestones</td>
<td>5</td>
<td>35</td>
</tr>
<tr>
<td>Final Submission</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Paper</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

Choose Your Own Adventure – Project Paths
Each final project submission will be evaluated based on the ability to generate valid (ideally, assembly) code for the provided acceptance tests. There are multiple legitimate (primary) paths to completing the project, with varying time commitments and, as such, varying potential final course grades (as dictated by the grading breakdown). These are summarized as follows.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Maximum Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLVM-only</td>
<td>Register-based LLVM - without optimizations (1–3)</td>
<td>D</td>
</tr>
<tr>
<td>LLVM-opt</td>
<td>Register-based LLVM - with optimizations (1–3, 6–7)</td>
<td>C</td>
</tr>
<tr>
<td>Barebones</td>
<td>Assembly from stack-based LLVM - without optimizations (1–2, 4–5)</td>
<td>C</td>
</tr>
<tr>
<td>Registered</td>
<td>Assembly from register-based LLVM - without optimizations (1–5)</td>
<td>B</td>
</tr>
<tr>
<td>Straight Line</td>
<td>Assembly from stack-based LLVM - with optimizations (1–2, 4–7)</td>
<td>B</td>
</tr>
<tr>
<td>All In</td>
<td>Assembly from register-based LLVM - with optimizations (1–7)</td>
<td>A</td>
</tr>
</tbody>
</table>

Collaboration and Cheating
Students may work in pairs on the project. Each student/pair is expected to complete their own project without collaboration with others.

It is fine to talk with others about general approaches to the project, but each student/pair is to develop 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.
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?