# CS 252 Project Requirements

Thomas H. Austin
San José State University
thomas.austin@sjsu.edu

February 22, 2016

## 1 Introduction

The class project is your chance to explore an aspect of programming language design that you find interesting. The choice of a topic is open-ended, so you are encouraged to explore options beyond what I suggest here.

**By March 10th** you must turn in a proposal, including your topic and a schedule broken down by the week. If you are planning to do a survey or a research project, include a couple of references of papers that you plan to read. Make sure your citations are well-formatted. Lists of urls are never sufficient. All of your citations must have a date; if it is a living document, specify "date accessed".

Broadly, there will be three types of projects:

- A survey project – read several papers on an aspect of programming languages and report what you have learned. You will be expected to provide extensive code snippets to demonstrate the different concepts that you have learned.
- An implementation project – implement an interpreter for an interesting language, or a library exploring the concepts that we have discussed in class.
- A research project – a mix of survey and implementation. These projects allow you to explore the edge of programming language design. Some of these could (potentially) be the basis for a 297 project.

## 2 Survey projects

Many students ask if it is possible to get an 'A' with a survey project. The answer is yes. Survey projects can be fascinating and insightful if you dig in to some interesting areas.

Survey projects can be terrible and get a poor grade if you do not go into depth on your subject.

A good survey project should have strong references. At a minimum, you should read 6-8 **peer-reviewed** publications, meaning journal papers, conference papers, etc. You are welcome to cite blog posts, wikipedia, bug reports, etc, but none of those count as your 6-8 citations.

You should know at least 2-3 of these papers **in depth**. If I ask you about subtle details of these papers, you should be able to answer my questions.

Some possible survey projects:

- Do a comparison of the Go, Rust, and D programming languages. What is each language trying to achieve? What are there comparative advantages?
- Explore the macro system in Scheme/Racket. How is the design of the Sweet.js JavaScript library influenced by Scheme/Racket?

- Explore object capabilities. How have the concepts in the E programming language influenced the design of Google's Caja?
- Contrast different mechanisms for handling concurrency? What are the advantages of the different approaches?
- Compare and contrast object capabilities and information flow analysis. What security guarantees do each of these languages make?

# 3 Implementation projects

If you enjoy hacking on code more than reading papers, the implementation projects might be more fun. You should implement a significant project related to programming language design. You are encouraged to come up with your own ideas, but here are some possibilities to consider:

- Implement a domain specific language for games, graphics, music generation, or whatever area interests you.
- Implement a Firefox addon that inspects page content and transforms it in interesting ways.
- Write an interpreter for a small language of your design, preferably implementing some feature that we have not covered in class.
- Modify the Zaphod Firefox addon to support alternate languages in the web browser. (`https://github.com/taustin/Zaphod`).

# 4 Research projects

These projects will require extensive coding **and** lots of paper reading. The idea is to find some unusual concept in PL research, and to implement these ideas. These projects will be very challenging, but if done well, may be the basis for a CS 297 project.

Some possibilities:

- Integrate security controls into JavaScript using the Sweet.js library.
- Integrate policy-agnostic programming for JavaScript. See `http://projects.csail.mit.edu/jeeves/papers/popl088-yang.pdf` for more details on policy-agnostic programming.
- Integrate object capabilities into the Lua programming language.
- Explore the Jeeves programming language, and extend it with some additional security controls. (`http://projects.csail.mit.edu/jeeves/`).

# 5 Citation format

Note that your paper should look professional. As part of that, I expect you to cite your references with some care and with proper formatting. **A list of urls are not valid citations, and will earn you a F for the project.** If you are unsure, see me.