CS 6390: Programming Languages
Fall 2019

Instructor: Prof. Vivek Sarkar
TA: Caleb Voss

1 Overview

The foundational principles of programming languages run a broad spectrum from formal, precise mathematical reasoning through the definition of proof systems and academic languages, to the realization of these principles in real-world languages and tools. This course will cover topics across this spectrum, including:

- defining semantics of typed languages
- progress and preservation theorems
- functional languages with recursion over inductively defined data types
- formal proof systems on "pencil-and-paper" and in Coq
- the Curry-Howard correspondence
- concurrency and parallelism
- substructural types for permission and ownership

Toward the end of the semester, we will take a closer look at real-world examples of languages and tools which make use of these concepts, such functional programming in OCaml, move semantics in Rust, concurrency in Go, and features in other mainstream languages like Java and Scala.

2 Pre-requisites

There are no formal graduate pre-requisites for this course.

Undergraduate students wishing to take this course must have taken *CS 2050/2051: Intro to Discrete Mathematics for CS*. Some assignments in this course will require programming in a functional language and the use of a formal proof system. Students should ideally be familiar with basic functional programming techniques such as recursion on lists and trees, as well as be able to articulate a proof by induction. Please contact the instructor for any questions.

3 Resources

There is no required textbook for this course; however, we will rely on selections from a number of reading materials freely available online:

- *Software Foundations*, vols. 1–2, Benjamin Pierce et al., 2019.

  https://softwarefoundations.cis.upenn.edu

- *Practical Foundations for Programming Languages*, Robert Harper, 2016.

  http://www.cs.cmu.edu/~rwh/pfpl/index.html

  Much of the second edition is publicly available. The first edition is available online through the library.

- *Types and Programming Languages*, Benjamin Pierce, 2002.

  https://www.cis.upenn.edu/~bcpierce/tapl/

  Available online through the library.

- Various academic papers, TBD.

Course announcements will be posted on Piazza. Students are encouraged to discuss concepts and help answer one another's questions on Piazza. Assignments will be released and turned in through Canvas.

4 Office Hours

- Vivek Sarkar (vsarkar@gatech.edu): TBD
- Caleb Voss (cvoss@gatech.edu): TBD

5 Grading

- **Homework (20%)** Periodic homework assignments will consist of written work, small programming exercises, or small proof exercises. These will be designed to deepen your understanding of topics covered in the lectures and readings.
- **Project (30%)** One large project will give you the opportunity to combine knowledge and skills acquired throughout the semester and apply them to a problem selected by you and the instructor. This problem may involve applying or adapting an academic solution to real programs or languages, formally proving known results, demonstrating a programming language concept by formalizing it for a small language or subset of a real language, extending the definition of a language or otherwise implementing a concept within an existing language, etc. Expect this project to involve some combination of actual programming, formal proof writing, and a written report. Presentations will be scheduled toward the end of the semester.

- **Midterm (20%)** An in-class written exam, date TBD (75 minutes), will cover topics from the first half of the semester.
- **Final (25%)** The registrar-scheduled exam will take place on Friday, 6 Dec 2019, 2:40–5:30 pm (170 minutes), and will cover all topics in the course.
- **Participation (5%)** Participation credit is awarded based on students' demonstrated commitment to the course, including engagement in discussions during lecture and on Piazza, as well as any in-class worksheets.

6 Policies

6.1 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 during lecture. However, all of these choices will factor into the participation component of the course grade.

6.2 Honor Code

All students are expected to follow the Georgia Tech Honor Code with respect to all graded materials, including homework solutions, exams, and project reports and project code.

6.3 Collaboration Policy

Students are expected to complete all coursework independently, except for in-class worksheets, for which collaboration is always encouraged. While students are welcome to discuss the course material outside of class, the material covered in homeworks 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 is therefore not allowed. Any such incident is cause for reporting of an honor code violation.

6.4 Disability Accommodation

Students with disabilities that present a challenge for learning 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 should bring it to the attention of the instructor immediately. Assignments missed as a result of the emergency will not be graded.