Syllabus, CSCI312, Programming Languages

Winter 2023

Instructor: Dr. Geoffrey Matthews, Parmly 407A, `gmatthews <at> wlu <dot> edu`

Web page: `https://github.com/geofmatthews/csci312`

Office hours: MWF 1:30-2:30

Lectures: MWF 12:15 - 1:15, Parmly 405

Goals. This class is an introduction to the semantics of programming languages. Although we will do some simple programming in a variety of programming languages, most of our programming will be in Scheme, `http://www.schemers.org/`, an advanced functional programming language. Functional programming is a methodology, style, and suite of programming language features that facilitates rapid programming and algorithmic design. Some required reading on the usefulness of functional programming: *Beating the Averages*, `http://www.paulgraham.com/paulgraham/avg.html`. Although we will program in a single language, we will build interpreters for a wide variety of languages. Each language you encounter in your career will partake of some of these features; understanding how they are implemented, and what their runtime support entails, will give you a deep understanding of any programming language you come across.

Texts:

- *Programming Languages, Application and Interpretation*, Shriram Krishnamurthi,
  `https://cs.brown.edu/~sk/Publications/Books/ProgLangs/2007-04-26/plai-2007-04-26.pdf`
  This will be our textbook. We will use the **first** edition.
  Note that the text is pedagogical; frequently he will start with the obvious (but wrong) way to solve a problem, and then work his way to the right solution 5 or 10 pages later. If you don't spend the time reading, and rereading, you may get confused. However, if you put in the time, you will understand at a much deeper level.
- *Teach Yourself Scheme in Fixnum Days*,
  `https://ds26gte.github.io/tyscheme/`
  Best introductory Scheme tutorial I've found. There are many more online.
  Note: skip Chapter 1, it deals with a different Scheme system. Instead, run through the **Quick** tutorial found here: `https://docs.racket-lang.org/` or in the Racket help desk available from DrRacket.
- *The Scheme Programming Language*, (4/e),
  `http://www.scheme.com/tspl4/`
  Good reference manual for Scheme.

Software:

- *The Racket programming language*, `https://racket-lang.org/`

Grading: Homework: 50%; Midterm: 20%; Final: 30%; Extra Credit: 10%.

Letter grades: A $\geq$ 90% > B $\geq$ 80% > C $\geq$ 70% > D $\geq$ 60% > F

Extra credit: A maximum of 10 percentage points of extra credit may be given for a special project. Special projects must: (a) be proposed to the instructor, in writing, three weeks before the last lecture, (b) be approved by the instructor at that time, (c) must include a substantive software component, written in Scheme, (d) must include a writeup with an introduction outlining the purpose of the project, a section discussing the results, and a conclusion, (e) must include well annotated source code, a user's guide, and a programmer's guide to the software, and (f) must be complete and turned in before the last lecture of class.

Schedule:

		January 2023					
Su	Mo	Tu	We	Th	Fr	Sa	
8	9	10	11	12	13	14	Scheme intro
15	16	17	18	19	20	21	Scheme intro, PLAI 1
22	23	24	25	26	27	28	PLAI 2,3; Python AST
29	30	31					PLAI 4
		February 2023					
Su	Mo	Tu	We	Th	Fr	Sa	
			1	2	3	4	PLAI 5,6
5	6	7	8	9	10	11	PLAI 7,8,9
12	13	14	15	16	17	18	PLAI 10,11; review
19	20	21	22	23	24	25	holiday
26	27	28					PLAI 12,13
		March 2023					
Su	Mo	Tu	We	Th	Fr	Sa	
			1	2	3	4	PLAI 13,14
5	6	7	8	9	10	11	PLAI 15,16,17
12	13	14	15	16	17	18	PLAI 18,19,20
19	20	21	22	23	24	25	review
26	27	28	29	30	31		PLAI 24,25,26
		April 2023					
Su	Mo	Tu	We	Th	Fr	Sa	
						1	
2	3	4	5	6	7	8	review
9	10	11	12	13	14	15	final exam