Preliminaries

CS 440: Programming Languages
Michael Lee <lee@iit.edu>
Michael Lee

- lee@iit.edu
- http://moss.cs.iit.edu
- Office: SB 226C
- Hours: Tue/Thu 10AM-12PM
  (make appointments on homepage)
TA: Xincheng Yang
- xyang76@hawk.iit.edu
- Hours: TBA
Agenda

- Course overview
- Administrivia
- Grading
- Assessments
- Resources
§ Programming Languages
Programming Languages …

- Theoretically all the same, yet practically very different!
- “Same” in a deep sense: Turing completeness
- Learning different languages and language features can vastly expand your repertoire of programming techniques
- PLs are among our most important and fundamental tools!
PL features

- Must learn to precisely dissect and discuss PLs

- Terminology: imperative, functional, compilers, interpreters, types, type checking, etc.

- Many terms are used imprecisely in conversation!
Reasoning about PLs

- What does a program (or PL construct) mean?
- Can we prove a program’s correctness?
- Many different ways of modeling and reasoning about program semantics
- Goal: inject mathematical rigor into programming
Not just a consumer!

- You will modify and create your own PLs
- Understand how PLs tick
- Where is the overhead? Is it useful/necessary/worthwhile?
- Fun and useful skill!
We will …

1. Use a new language, *Racket*, to learn about different programming language constructs and ideas.

2. Learn different methods of language specification, focusing on *semantics* and *verification*.

3. Understand how programs are *interpreted, compiled, represented, evaluated, and optimized*.

4. Implement our own programming language interpreters
Topics

- Racket
- Syntax
- Higher order functions
- Recursion
- Closures
- Metaprogramming
- Continuations

- Grammars and Languages
- Semantics
- Operational / Axiomatic
- Evaluation strategies
- Interpreters and Compilers
- Type inference and Unification
- Memory management
§ Administrivia
Prior knowledge

- Programming experience (CS 115/116/201)
- First-order / Predicate logic (CS 330)
- Rules of inference and logical proofs (CS 330)
- Formal languages and Grammars (CS 330)
- Analysis of algorithms (CS 331 / 430)
CS 440: Programming Languages

Announcements

- We will be using Discord to facilitate student discussion, and to run open TA office hours. If you aren’t already a member, please join at https://discord.gg/Co8alekN
- Welcome to the Fall 2022 edition of CS 440: Programming Languages!

Calendar

The lecture calendar is tentative and may be updated. All lecture source files can be found in the class GitHub repository.

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Notes</th>
<th>Readings(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug 24</td>
<td>Syllabus and Course overview</td>
<td>slides intro.pdf</td>
<td>Syllabus</td>
</tr>
<tr>
<td>Aug 26</td>
<td>Compilers, Interpreters, and Rocket</td>
<td>slides rocket.pdf</td>
<td>Ch. 2</td>
</tr>
<tr>
<td>Aug 31 - Sep 2</td>
<td>Intro to Rocket and Basic types</td>
<td>01.intro.rkt</td>
<td>R0, ch. 2-3</td>
</tr>
<tr>
<td>Sep 7</td>
<td>Functions and Special forms</td>
<td>02.functions.rkt</td>
<td>R0, ch. 4</td>
</tr>
<tr>
<td>Sep 9</td>
<td>Lists and Recursion</td>
<td>03.recursion.rkt</td>
<td>BPL, Lists</td>
</tr>
<tr>
<td>Sep 14 - Sep 16</td>
<td>Higher-order functions</td>
<td>04.hol.pdf</td>
<td></td>
</tr>
<tr>
<td>Sep 21</td>
<td>Macros and Syntax transformers</td>
<td>05.macros.rkt</td>
<td>R0, ch. 10</td>
</tr>
<tr>
<td>Sep 23 - Sep 28</td>
<td>Building an Interpreter</td>
<td>06.interpreters.rkt</td>
<td>PLAI ch. 3-4, R0, ch. 5-12</td>
</tr>
<tr>
<td>Sep 30</td>
<td>Closures</td>
<td>07.closures.rkt</td>
<td>PLAI ch. 5-6</td>
</tr>
<tr>
<td>Oct 5</td>
<td>Debugging</td>
<td>08.debugging.rkt</td>
<td></td>
</tr>
<tr>
<td>Oct 7 - Oct 12</td>
<td>Big Step Semantics</td>
<td>big-step.pdf</td>
<td>Hasper, ch. 2</td>
</tr>
<tr>
<td>Oct 14</td>
<td>Midterm Exam</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oct 19 - Oct 26</td>
<td>Lambda Calculus</td>
<td>lambda-calculus.pdf</td>
<td>A-calculus Tutorial</td>
</tr>
<tr>
<td>Oct 28</td>
<td>Continuation Passing Style</td>
<td>09-continuations.rkt</td>
<td></td>
</tr>
<tr>
<td>Nov 2</td>
<td>Pair-wise Continuations</td>
<td>10-pairs.rkt</td>
<td>BPL, Continuations</td>
</tr>
<tr>
<td>Nov 4 - Nov 9</td>
<td>Small-Step Semantics</td>
<td>small-step.rkt</td>
<td></td>
</tr>
<tr>
<td>Nov 11 - Nov 16</td>
<td>Automaton Semantics and Hoare Logic</td>
<td>automatic.rkt</td>
<td></td>
</tr>
</tbody>
</table>

Course website: http://moss.cs.iit.edu/cs440
Blackboard: http://blackboard.iit.edu
Discord: TA class discussion and Q/A  
(invite on course website)
References (in addition to notes)

- Programming Languages: Application and Interpretation, by Shriram Krishnamurthi

- Crafting Interpreters, by Robert Nystrom

Grading

- 50% Assignments
- 25% Midterm Exam
- 25% Final Exam (Cumulative)
Assignments

- 5-7 total
  - Some written, some machine problems (coding problems)
  - Written submitted via Blackboard, MPs via GitHub
Late Policy

- 7-day late pool, distributed however you like across labs (a day at a time)
- If you’re out of late days, late submissions will not be accepted for a grade!
Exams

- Scores may be linearly scaled so that median/mean (whichever lower) is 75%

- Midterm tentatively scheduled for March 8
<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\geq 90%$</td>
</tr>
<tr>
<td>B</td>
<td>80-89%</td>
</tr>
<tr>
<td>C</td>
<td>70-79%</td>
</tr>
<tr>
<td>D</td>
<td>60-69%</td>
</tr>
<tr>
<td>E</td>
<td>$&lt; 60%$</td>
</tr>
</tbody>
</table>
For Friday

- Read chapter 2 of *Crafting Interpreters*: “A Map of the Territory”
- Download and install DrRacket ([https://racket-lang.org](https://racket-lang.org))
- Clone the class lecture repository from GitHub ([https://github.com/cs440lang/lectures/](https://github.com/cs440lang/lectures/))