CS3100 Paradigms of Programming

Fall 2019

Who am I

- Your Instructor: KC Sivaramakrishnan
- I go by "KC"

Single instruction programming language

```
subleq a, b, c   ; Mem[b] = Mem[b] - Mem[a]
                 ; if (Mem[b] ≤ 0) goto c
```

If the branch target is the next instruction, then drop the third argument.

```
subleq a, b
```

is equivalent to

```
    subleq a, b, L1
L1: ...
```

What does this program do?

```
subleq a, Z
subleq Z, b
subleq Z, Z
```

Answer: Mem[b] = Mem[a] + Mem[b]

What does this program do?

```
subleq b, b
subleq a, Z
subleq Z, b
subleq Z, Z
```

Answer: Mem[b] = Mem[a]

In fact, this one instruction PL is as powerful as *every* PL.

- But good luck writing quicksort in this PL
- ..or Swiggy.
- ..or Grand Theft Auto V.

The `subleq` instruction is from One Instruction Set Computer (https://en.wikipedia.org/wiki/One_instruction_set_computer). If you thought such a machine is hypothetical, think again. It has been shown that the x86 `mov` instruction is turing complete (https://esolangs.org/wiki/Mov) and is as powerful as every programming language.

So why study programming languages?

- Analogy -- studying a foreign language
- Learn about another culture; incorporate aspects into your own life
- Shed preconceptions and prejudices about others
- Understand your native language better

The Goal of CS3100

Become a better programer

through the study of

programming languages

Java is to Programming Languages

as

Japanese is to Linguistics

Programming Languages: Language design, implementation, semantics, compilers, interpreters, runtime systems, programming methodology, testing, verification, security, reliability ...

Adjacent to **Software Engineering** in the CS family tree.

Linguistic Relativity

```
The principle of linguistic relativity holds that the
structure of a language affects its speakers world view
or cognition.
```

Or more simply:

```
Programming Language shapes Programming Thought.
```

Language affects how ideas and computation are expressed.

Alan J. Perlis

**"A language that doesn't affect the way you
think about programming is not worth knowing"**

First recipient of the Turing Award for his “influence in the area of advanced programming techniques and compiler construction”

New languages come (and go ..)

There was no

- Java 25 years ago
- C# 20 years ago
- Rust 10 years ago
- WebAssembly 2 years ago

What is CS3100 about?

- Concepts in programming languages
- Programming paradigms
- Language design and implementation

Goal: Learn the Anatomy of PL

- What makes a programming language?
- Which features are *fundamental* and which are *syntactic sugar*?

Goal: Learn New Languages / Constructs

New ways to *describe* and *organize* computation, to create programs that are:

- Correct
- Readable
- Extendable
- Reusable

Goal: How to Design new Languages

New hot lanuages being designed in industry as we speak:

- Flow, React @ Facebook
- Rust @ Mozilla
- TypeScript @ Microsoft
- Swift @ Apple
- WebAssembly @ Google + Mozilla + Microsoft

Goal: How to Design new Languages

Buried in every large system is a (domain-specific) language

- DB: SQL
- Word, Excel: Formulas, Macros, VBScript
- Emacs: LISP
- Latex, shell scripts, makefiles, …
- All the smart contract languages on *Blockchains*.

If you work on a large system, you **will** design a new PL!

Goal: Enable You To Choose Right PL

But isn't that decided by

- Libraries
- Standards
- Hiring
- Your Boss?!

My goal: Educate tomorrow's leaders so you'll make **informed** choices.

Course Syllabus

- **Functional Programming:** OCaml & Lambda Calculus
- **Logic Programming:** Prolog
- **Concurrent Programming:** (in) OCaml

Course Logistics

Course website

- http://kcsrk.info/cs3100_f19 (http://kcsrk.info/cs3100_f19)
- Schedule, Lecture Notes, Assignments, etc.
- Look at the schedule (http://kcsrk.info/cs3100_f19/schedule/) to know if the class is on.

Lectures

- Delivered through interactive Jupyter notebooks.
- Instruction for setting up available on course website (http://kcsrk.info/cs3100_f19/resources/).
- **Highly recommend that you practice in the notebooks**

Grading

- 6 Assignments = 6 * 5% = 30%
- 2 Quizzes = 2 * 15% = 30%
- End Sem = 40%

Software

- We will use OCaml and Prolog in this course. The installation information is available in the course webpage (http://kcsrk.info/cs3100_f19/resources/).
- Docker image is available for lectures in Jupyter notebooks.
  - Get familiar with basic Docker commands. It is likely that you will use them in the future.
- Get a local installation of OCaml on your machine (instructions (http://kcsrk.info/cs3100_f19/resources/)).
  - Get familiar with `utop` (a great top-level for OCaml),
  - `merlin` (IDE server for OCaml that works with vim, emacs, vscode, sublime)
  - `dune` (a build tool for OCaml).

Tutorial on Git, Docker, Jupyter

Anmol Sahoo (Research Associate in my group) will give a tutorial on the tools this

Friday 2nd August 13:00 to 13:50 (C slot).

Fin.