CSE130: Programming Languages

Summer 2019

MTuWTh 11:00 AM-12:20 PM

Kaiser Pister

Your instructor

- UCSD CSE B.S. & M.S.
- Lecturer in CSE
- Research: Natural Language Processing
  - Lots of PL ideas appear in daily work
- Academic Focus: Systems + PL + Security

Your TAs

- Yinglong Miao
- Yanzhi Ding
- Kiana Lucin
- Dongyao Zhu

What is CSE 130 about?

What this course is **not** about?

- Learning how to write…
  - JavaScript in January
  - Haskell in February
  - C++ in March
  - etc.
- Learning C++, JavaScript, etc. to spec

What this course **is** about

- Concepts in programming languages
  - Fundamentals and core features and building blocks
  - Different programming paradigms and their use
- Design and implementation of languages
  - Goals and trade-offs (with historical context)
  - The cost of a language feature

Why?

- Concepts in programming languages
  - Language shapes your thinking! Language features dictate how we express ideas and computation
  - E.g., t

This program prints “Hello World!”:

```
++++++++[>++++[>++>+++>+++>+<<<<-]>+>+>->>+[<]<-]>>.>---.
+++++++..+++.>>.<-.<.+++.------.--------.>>+.>++.
```

https://en.wikipedia.org/wiki/Brainfuck

- Design and implementation of languages
  - Nothing is free: understand what you’re giving up and what you’re gaining when choosing a language
  - E.g., exception handling, garbage collection, etc.

Why else?

- You can learn any of those languages… once you have a grasp of the fundamentals and understand features

- You'll usually want to use the right lang for the job… this ultimately comes down to what features you need

- You will be able to think about programs differently… since you will understand what's going on underneath

- You will be in better shape to design and implement new languages… great features ➠ great language!

I'll be working on languages?

- Lots of systems have their own languages or have a language runtime system at their core:
  - Editors (Lisp for Emacs, JavaScript for Atom)
  - DBs (SQL, MongoDB's JavaScript, …)
  - PDF viewers (JavaScript!?)
- PL is hot! Likely to work on something new in industry

Flow, React @ Facebook Rust, Emscripten @ Mozilla,
TypeScript @ Microsoft Swift @ Apple CUDA @ NVIDIA

If nothing else...

You can put Haskell and (maybe) Rust on your resume!

Syllabus: The great ideas [Ramsey]

Expressive power (say more with less)

First-class functions

Pattern matching

Type inference

Exception handling

Monads

Continuations

Reliability and reuse

Type polymorphism

Type classes

Modules

Objects & inheritance

Cross-cutting concerns

Memory management

Concurrency

The great ideas [JavaScript]

Expressive power (say more with less)

First-class functions

Pattern matching

Type inference

Exception handling

Monads

Continuations

Reliability and reuse

Type polymorphism

Type classes

Modules

Objects & inheritance

Cross-cutting concerns

Memory management

Concurrency

The great ideas [Haskell]

Expressive power (say more with less)

First-class functions

Pattern matching

Type inference

Exception handling

Monads

Continuations

Reliability and reuse

Type polymorphism

Type classes

Modules

Objects & inheritance

Cross-cutting concerns

Memory management

Concurrency

The great ideas [C++]

Expressive power (say more with less)

First-class functions

Pattern matching

Type inference

Exception handling

Monads

Continuations

Reliability and reuse

Type polymorphism

Type classes

Modules

Objects & inheritance

Cross-cutting concerns

Memory management

Concurrency

The great ideas [Rust]

Expressive power (say more with less)

First-class functions

Pattern matching

Type inference

Exception handling

Monads

Continuations

Reliability and reuse

Type polymorphism

Type classes

Modules

Objects & inheritance

Cross-cutting concerns

Memory management

Concurrency

Week of	Monday & Tuesday	Wednesday & Thursday	Monday & Friday (section)
Jul 01- Jul 04	JavaScript and high-order functions	Lambda Calculus	Lambda Calc Review
Jul 08- Jul 11	Closures & Haskell Crash Course	Type polymorphism and type inference	Haskell Crash Course (cont.)
Jul 15- Jul 18	Catch-up / Review & **Midterm**	Type classes	Midterm Review (Monday section)
Jul 22- Jul 25	Objects	vtables, subtyping, inheritance	C++ Review
Jul 29- Aug 02	Control Flow, Continuations, Monads	PL Topics & Final Review	Additional Final Review
Aug 03	**Final**		

Logistics & course mechanics

Contact information

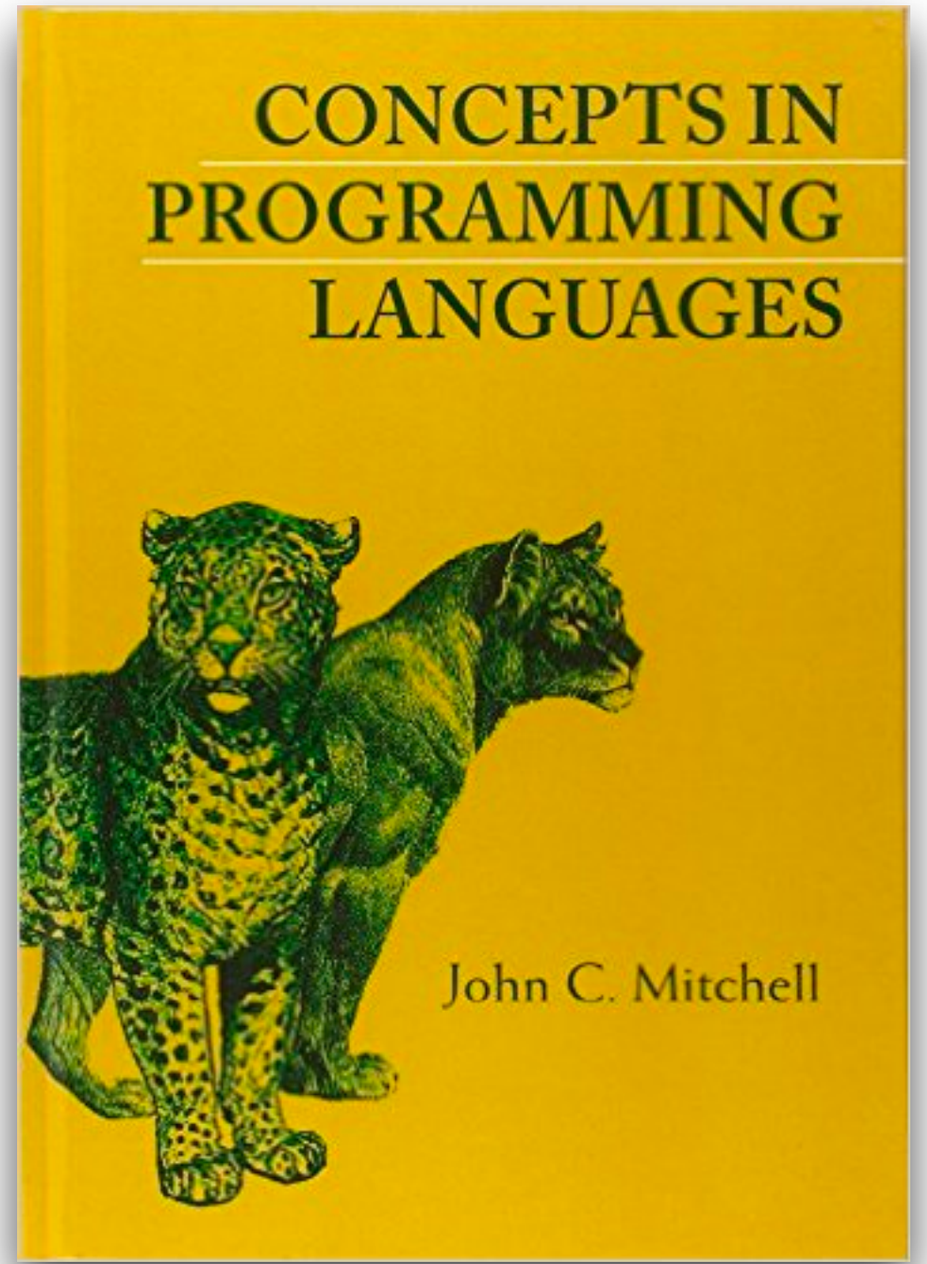
- Course website:
  - cseweb.ucsd.edu/~kpister/cse130-summer19
  - Goto place for links and resources
- Piazza: piazza.com/ucsd/summer2019/cse130
  - Use this for general discussions and questions
- My email: kpister@eng.ucsd.edu
  - Use this if you need to get in touch with us directly

Logistics: Lectures & Section

- Lectures: Mondays, Tuesdays, Wednesdays, Thursdays
  - We will assign reading before every class
  - Come prepared: we will ask questions during lecture
- Section: Mondays Fridays
  - Come to section with questions!
  - Goal: go over course material and problems similar to those assigned for homework

Reading from:

- Optional course textbook
  - Concepts in Programming Languages by John Mitchell
  - Renting: cheaper option
  - We'll be distributing new Chapters
- Papers & online resources



Logistics: Participation [5%]

- In class: answer questions, ask questions
- OH: ask questions, answer questions
- Online: ask questions, answer questions
  - I've give an additional 5% to the students who really put in the effort to help on Piazza

Logistics: Assignments [50%]

- Homework: one assignment per week
  - Work in groups of 3 (but try to do it on your own first!), submit using gradescope
  - Will be released Sundays
  - Early deadline: following Thursday night
    - You get 10% of your grade if you turn it in early!
  - Hard deadline: following Saturday night

Logistics: Assignments [50%]

- Programming labs: roughly one per week
  - Submit solution by yourself using gradescope
  - Will be released Sundays
    - Early deadline: following Thursday night
      - You get 10% of your grade if you turn it in early!
    - Hard deadline: 1 week from the release date on Saturday night

Exams [50%]

- Midterm exam: July 16, in class [20%]
  - Can screw up; we'll compute your score as:

```
midterm > 0 ?  max(final, midterm) : 0
```

  - Will reflect assignments, pretty straight forward
- Final exam: August 3, location and time TBA [30%]
  - Will test you in new setting, expect to learn!

Summary: grading breakdown

- Participation: 5%
- Assignments: 50%
- Exams: 50%

Collaboration policy

- Talk with each other, talk on Piazza, use resources
  - Collaboration is a good thing! Just credit the person or resource in you submission
- That said: I expect you to turn in your own work
  - Don’t discuss particularities of a solution with others
  - Don’t ask for a solution on StackOverflow and the like
  - See academic integrity statement

Academic integrity, conduct, etc.

- Goal: welcoming class where all can learn and feel included, safe, healthy
  - I don’t want to run the class like a police state, but these two rules will be enforced: these matter even once you graduate!
  - Eat, sleep, take care of your health
  - Talk to me if you’re concerned

Feedback wanted!

- How’s the pace?
- Are there particular topics you want to spend more time on?
- What can I do to make your learning experience better?

Participation

Questions?