Hello, world!

A Programming Language

Two Variables

- `x`, `y`

Three Operations

- `x++`
- `x--`
- `x = 0 ? L1 : L2`

Example Program

(What *does* it do?)

```
L1: x++
    y--
    y = 0 ? L2 : L1
L2: ...
```

The above language is “equivalent to” every PL!

- But good luck writing QuickSort ...
- ... or Plants v. Zombies
- ... or Spotify!

So Why Study Programming Languages?

Federico Fellini

> *A different language is a different vision of life.*

So Why Study Programming Languages?

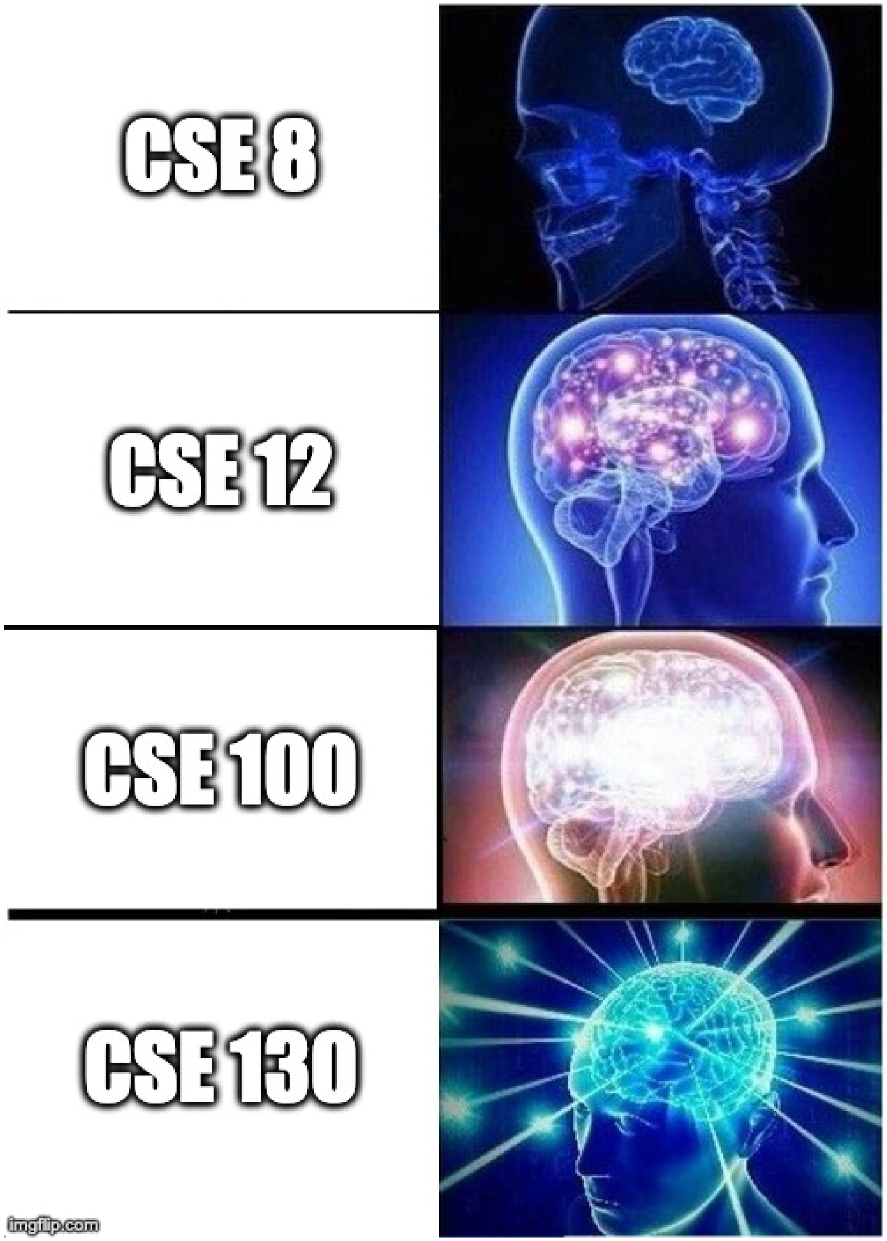
> *The principle of **linguistic relativity** holds that the structure of a language affects its speakers world view or cognition.*

Or more simply:

> *Programming Language shapes Programming Thought.*

Language affects how ideas and computation are expressed

Course Goals



130 Brain

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 CSE 130 about?

- Concepts in programming languages
- Programming paradigms

- Language design and implementation

Goal: Learn the Anatomy of PL

Zygote Media Group, Inc. ©2006
Image provided courtesy of 3DSCIENCE.COM

Anatomy

- What makes a programming language?

- Which features are **fundamental** and which are **syntactic sugar**?

Goal: Learn New Languages / Constructs

Musical Score

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

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

- DB: SQL
- Word, Excel: Formulas, Macros, VBScript
- Emacs: LISP
- Latex, shell scripts, makefiles, …

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?!

Yes.

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

*What is CSE 130 **not** about?*

Learning…

- JavaScript in April
- Haskell in May
- C++ in June

etc.

Who am I?

Ranjit Jhala (https://ranjitjhala.github.io/)

- Professor at CSE since 2005
- Research: Tools and Techniques to make programs better

The Crew

Teaching Assistants

- Matt Kolosick (https://kolosick.com/)
- Rose Kunkel (https://github.com/rosekunkel)
- George Sakkas (https://www.linkedin.com/in/gsakkas)

Tutors

- Darya Verzhbinsky (https://darya-ver.github.io/)

Course Syllabus

Functional Programming

- Lambda calculus (2 weeks)
- Haskell (6 weeks)

Logic Programming

- Prolog (2 weeks)

QuickSort *in C*

```
void sort(int arr[], int beg, int end){
  if (end > beg + 1){
    int piv = arr[beg];
    int l = beg + 1;
    int r = end;
    while (l != r-1)
       if(arr[l] <= piv) l++;
       else swap(&arr[l], &arr[r--]);
    if(arr[l]<=piv && arr[r]<=piv)
       l=r+1;
    else if(arr[l]<=piv && arr[r]>piv)
       {l++; r--;}
    else if (arr[l]>piv && arr[r]<=piv)
       swap(&arr[l++], &arr[r--]);
    else r=l-1;
    swap(&arr[r--], &arr[beg]);
    sort(arr, beg, r);
    sort(arr, l, end);
  }
}
```

QuickSort in Haskell

```
sort []     = []
sort (x:xs) = sort ls ++ [x] ++ sort rs
  where
    ls      = [ l | l <- xs, l <= x ]
    rs      = [ r | r <- xs, x <  r ]
```

(not a wholly fair comparison… (http://stackoverflow.com/questions/7717691/why-is-the-minimalist-example-haskell-quicksort-not-a-true-quicksort))

Course Logistics

webpage (https://ucsd-cse130.github.io/wi20)

- Calendar
- Lecture notes
- Programming assignments

piazza (https://piazza.com/ucsd/winter2019/cse130/home)

- Go-to place if you have a question or need help
- **Email is futile**

Grading

- 30% Assignments
- 30% Midterm

- 35% Final
- 05% Class participation (clickers)
- 05% Piazza Extra Credit
    - To **top 20** best participants

Assignments

6 programming assignments

- Released online (assignments.html)
- At least a week before due date
- On github

Four late days

- used as **whole unit**
- 5 mins late = 1 late day

Groups of two can do hw together

- On `ieng6` or your own machine
- Register your group here (https://forms.gle/jP4fGinSii5vPmNW8)

- **Submit individually** (via github)

Exams

Midterm

- in-class on *Monday, Feb 10*
- 2-sided “cheat sheet”

Final

- March 18, 8-11a
- 2-sided “cheat sheet”
- The final is cumulative
- Midterm grade is calculated as `midterm > 0 ? max(final, midterm) : 0`
    - you get a second chance if you don’t do well on the midterm
    - you **must take** the midterm and the final

Clickers

Assigned Seating

- From next lecture
- See your group (../static/raw/seating.txt) and seat (https://ucsd-cse130.github.io/wi20/static/raw/Center__113__groups.pdf)

Make class interactive

- Help **you** and **me** understand what’s tricky

Clickers Are Not Optional

- **Cheap** ones are fine
- **Respond** to 75% questions
- Register your clicker here (https://forms.gle/8gzTuqmVxRZ7anGX6)

TEST QUIZ

Do you have a clicker?

A. Yes

B. No

C. What’s a clicker?

Clicker Protocol

1. **Solo Vote**
    - Think for yourself, select answer
2. **Discuss**
    - Analyze Problem in Groups

    - Reach consensus
    - Have questions, raise your hand!

3. **Group Vote**
    - Everyone in group votes
    - Hopefully the same way but not enforced
    - You don’t have to answer correctly to get points!

4. **Class Discuss**
    - What was easy or tricky?

Your Resources

Discussion section

- Wednesday 05:00-06:00pm in CENTER 113

Office hours

- Every day, check **calendar**

Piazza

- We answer during work hours and office hours

No text

- Online lecture notes and links

Academic Integrity

Programming assignments: do not copy from classmates or from previous years

Exams done **alone**

- Zero Tolerance
- Offenders punished ruthlessly
- Please see academic integrity statement

Students with Disabilites

Students requesting accommodations for this course due to a disability or current functional limitation must provide a current **Authorization for Accommodation (AFA)** letter issued by the Office for Students with Disabilities (OSD) which is located in University Center 202 behind Center Hall.

Students are required to present their AFA letters to Faculty (please make arrangements to contact me privately) and to the CSE OSD Liaison Christina Rontell (https://cse.ucsd.edu/people/administrative-staff/christina-rontell) in advance so that accommodations may be arranged.

Diversity and Inclusion

Our goal is to create a diverse and inclusive learning environment where all students feel comfortable and can thrive. If there is a way we can make you feel more included please let one of the course staff know, either in person, via email/discussion board, or even in a note under the door.

Our learning about diverse perspectives and identities is an ongoing process, and we welcome your perspectives and input.

We also expect that you, as a student in this course, will honor and respect your classmates, abiding by the UCSD Principles of Community (https://ucsd.edu/about/principles.html). Please understand that others' backgrounds, perspectives and experiences may be different than your own, and help us to build an environment where everyone is respected and feels comfortable.

*If you experience any **harassment or discrimination***

Contact the Office of Prevention of Harassment and Discrimination (https://ophd.ucsd.edu/)

Students may receive confidential assistance

- Sexual Assault Resource Center (http://care.ucsd.edu) at (858) 534-5793
- Counseling and Psychological Services (http://caps.ucsd.edu.) (CAPS) at (858) 534-3755.

Basic Needs

- Are you eating properly?
- Do you have adequate access to nutritious food?
- Do you have stable housing?
- Are you homeless or couch surfing?

If you or someone you know has food and/or housing insecurity, be aware of Basic Needs (http://basicneeds.ucsd.edu)

- The Triton Food Pantry (https://www.facebook.com/tritonfoodpantry/) (in the old Student Center), is free and anonymous, and includes produce.
- Financial aid resources, the possibility of emergency grant funding, and off-campus housing referral resources are available.
- CAPS and college deans can connect students to the above resources, as well as other community resources and support.

(https://ucsd-cse130.github.io/wi20/feed.xml) (https://twitter.com/ranjitjhala)
(https://plus.google.com/u/0/104385825850161331469)
(https://github.com/ranjitjhala)

Generated by Hakyll (http://jaspervdj.be/hakyll), template by Armin Ronacher (http://lucumr.pocoo.org), suggest improvements here (https://github.com/ucsd-progsys/liquidhaskell-blog/).