

CMPS 116: Fall 2019

Introduction to Functional Programming

Lecture 1: Course Overview.

Owen Arden
UC Santa Cruz

A Programming Language

- Two variables

- x, y

- Three operations

- $x++$

- $x--$

- $(x=0) ? L1 : L2 ;$

```
L1 :  x++ ;
```

```
      y-- ;
```

```
      (y=0) ? L2 : L1
```

```
L2 :  ...
```

Fact: This is “equivalent to” to **every** PL!

Good luck writing quicksort

... or Windows, Google, Spotify!

So why study PL ?

Programming language
shapes
Programming thought

So why study PL ?

Language affects how:

- Ideas are expressed
- Computation is expressed

Course Goals



“Free your mind”
-Morpheus

Learn New Languages/Constructs

Lorenzo da Ponte
English version by
Ruth and Thomas Martin

Wolfgang Amadeus Mozart

Overture

Andante



New ways to:

- describe
- organize
- think about computation

Goal: Enable you to Program

Lorenzo da Ponte
English version by
Ruth and Thomas Martin

Wolfgang Amadeus Mozart

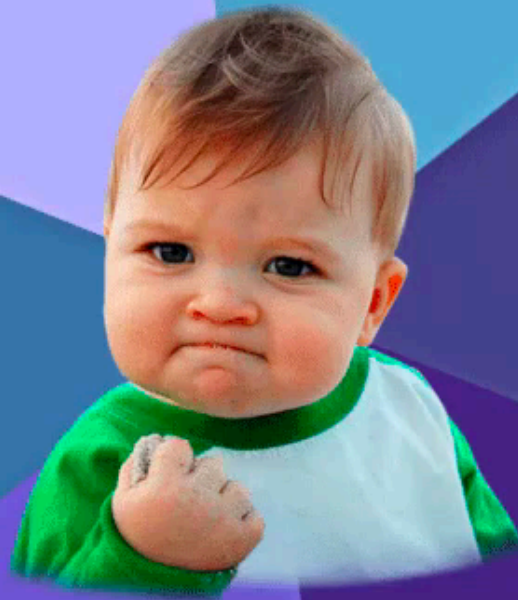
Overture

Andante



- Readable
- Correct
- Extendable
- Modifiable
- Reusable

#goals



Learn How To Learn

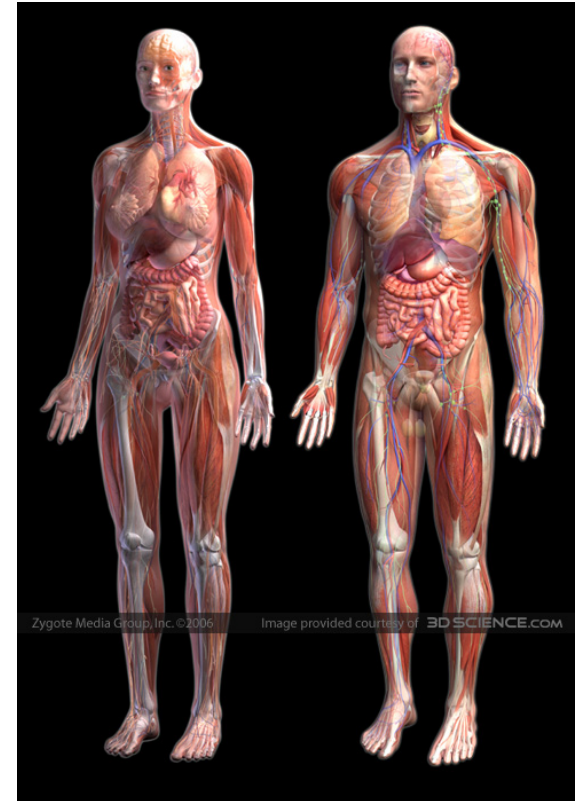
Goal: How to learn new PLs

No Java (C#) 15 (10) years ago
AJAX? Python? Ruby? Erlang? F#?...

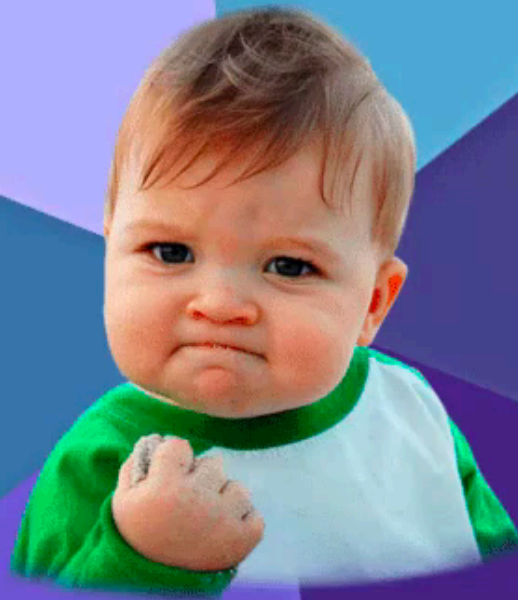
Learn the **anatomy** of a PL

- Fundamental **building blocks**
- Different guises in different PLs

Re-learn the PLs you already know



#goals



Design new languages

livememe.com

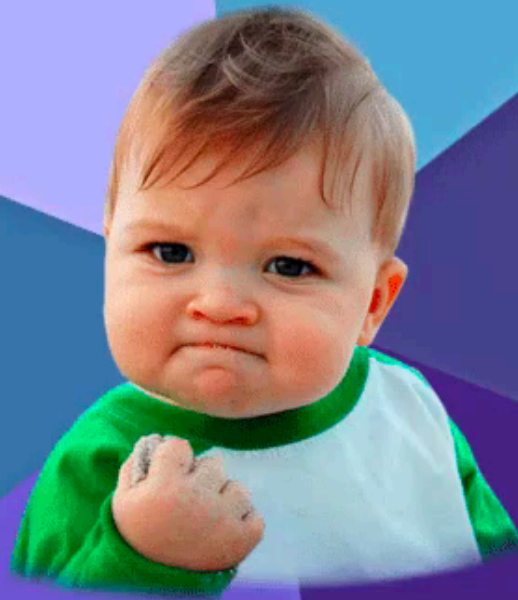
Goal: How to design new PLs

...“who, me ?”

Buried in **every extensible** system is a PL

- Emacs, Android: Lisp
- Word, Powerpoint: Macros, VBScript
- Unreal: UnrealScript (Game Scripting)
- Facebook: FBML, FBJS
- SQL, Renderman, LaTeX, XML ...

#goals



Choose right language

livememe.com

Enables you to choose right PL

“...but isn't that decided by

- libraries,
- standards,
- and my boss ?”

Yes.



My goal: educate tomorrow's tech leaders & bosses, so you'll make informed choices

Speaking of Right and Wrong...

Imperative Programming

$$x = x + 1$$

WTF?

$x = x + 1$

Imperative = Mutation

Imperative = Mutation

Bad!

Don't take my word for it

John Carmack Creator of FPS: Doom, Quake,...



John Carmack
@ID_AA_Carmack



I am starting to remove op= operator overloads to discourage variable mutation.

39
RETWEETS

16
FAVORITES



2:55 PM - 28 Feb 12 via web · Embed this Tweet

 Reply  Retweeted  Favorite

Don't take my word for it

Tim Sweeney (Epic, Creator of UNREAL)

*“In a concurrent world,
imperative is the wrong default”*



Functional Programming

Functional Programming ?

No Assignment.

No Mutation.

No Loops.

OMG! Who uses FP?!

So, Who Uses FP ?

The Google logo, featuring the word "Google" in its characteristic multi-colored font (blue, red, yellow, blue, green, red).

MapReduce

So, Who Uses FP ?



Microsoft®

Linq, F#

So, Who Uses FP ?

The Facebook logo, consisting of the word "facebook" in white lowercase letters on a blue rectangular background.

facebook

Erlang

So, Who Uses FP ?



twitter

Scala

So, Who Uses FP ?

Wall Street

(all of the above)

So, Who Uses FP ?

...CSE 116

Course Mechanics

Mechanics

Course website:

<https://owenarden.github.io/cse116-fall19/>

Course texts (optional):

- An Introduction to Functional Programming Through Lambda Calculus by Greg Michaelson. Free pre-print.
- Thinking Functionally with Haskell by Richard Bird. Available online (free via library).
- Programming in Haskell (2nd ed.) by Graham Hutton.
- Real World Haskell by Bryan O'Sullivan. Available online (free via library).
- Learn You a Haskell for Great Good by Miran Lipovača. Available free online
- Write You a Haskell by Stephen Diehl. (incomplete, but useful) Available free online

Peer Instruction (ish)

Peer Instruction

- Make class interactive
 - Help YOU and ME understand whats tricky
- Respond to in-class quizzes
 - 5% of your grade
 - Respond to 75% questions
- Bring laptop/phone if you have one

In Class Exercises

1. Solo Vote: Think for yourself, select answer
2. Discuss: Analyze Problem in Groups of 3
 - Practice analyzing, talking about tricky notions
 - Reach consensus
 - Have questions, raise your hand!
3. Group Vote: Everyone in group votes
4. Class-wide Discussion:
 - What did you find easy/hard?
 - Questions from here show up in exams

Requirements and Grading

- In-Class Exercises: 5%
- Midterm: 30%
- Programming Assignments (6): 30%
- Final: 35%

Two hints/rumors:

1. Lot of work
2. Don't worry (too much) about grade

Note: Regrades must be requested *in person within two weeks of receiving grade*

Resources

- Online lecture notes
- Readings and exercises
- Webcasts:
 - User: cse-116-1
 - Pass: lambda
- Pay attention to lecture and section!
- Do assignments yourself (+partner)!

Ask for help!

- Lots of help available, will be adding more soon. (watch website)
- Lab sessions 4 days/wk with tutors to help with assignments
- Discussion sections with TAs to help with lecture concepts

Programming Assignments

Schedule up on webpage. May be done in groups of two, if desired. See link on website.

Code in **GitLab** (sign up!). Submit on **Canvas**

- **You must *push* your submitted code.**

Deadline Extension:

- Four “late days”, used as “whole unit”
- 5 mins late = 1 late day
- Plan ahead, **no other extensions**

See course webpage for HW deadlines

Programming Assignments

Unfamiliar languages
+ Unfamiliar environments

Start Early!

Weekly Programming Assignments

Scoring = Style + Test suite

No Compile, No Score

Weekly Programming Assignments



Forget Java, C, C++ ...
... other 20th century PLs

Don't complain
... that Haskell is hard
... that Haskell is @!%@#

Immerse yourself in new language

It is not.

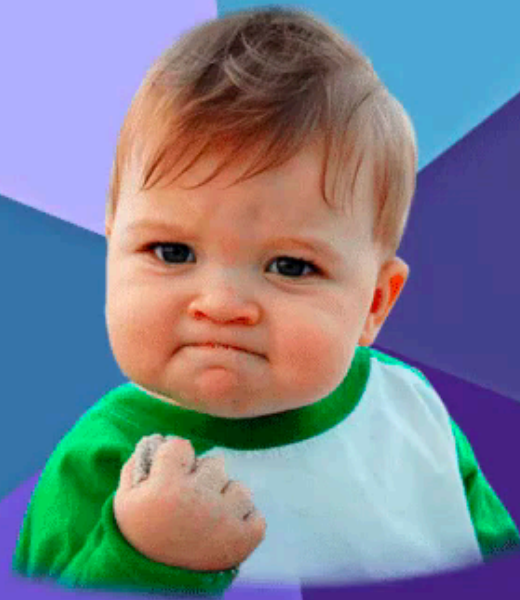
Immerse yourself in new language



Word from our sponsor ...

- Programming Assignments done **ALONE** or in (official) **groups of two**
- We use plagiarism detection software
 - MOSS is fantastic, plagiarize at your own risk
- **Zero Tolerance**
 - offenders punished ruthlessly
- Please see academic integrity statement:
 - <https://ue.ucsc.edu/academic-misconduct.html>

#goals



Ask me questions