Meeting 01: Welcome

Principles of Programming Languages - CSCI 3155 - Fall 2024

Professor Bor-Yuh Evan Chang

IF I HAD A COMPUTER, I'M SURE I'D GET BETTER GRADES ON MY BOOK REPORTS.
YOU'D STILL HAVE TO READ THE BOOK AND TELL THE COMPUTER WHAT YOU WANT TO SAY, YOU KNOW.
MAN, WHAT'S ALL THE FUSS ABOUT COMPUTERS ?!

Today

- Bootstrapping
- Introductions
- Course Approach
- Discuss Your Goals and Worries
- Getting Your Money's Worth
- Syllabus Policies

Bootstrapping

A puzzle …

Suppose you want to run lots of JavaScript programs — all the cool kids are doing it, but unfortunately you have a computer that can only execute x86 assembly. So you decide to **write a compiler from JavaScript to x86 assembly**.

Furthermore, suppose that you think Scala is the best language for writing a compiler; you find the language has particularly effective constructs for implementing compilers. Unfortunately, **your computer only has a C to x86 compiler**. Having taken 3155, you know that **you can write an awesome compiler in Scala** that produces efficient x86 code, but you're not so sure you can write an efficient compiler in C. What do you do?

Back up. What is program?

set of instructions

```
function plusOne(n) {
    return n+1;
}
```

How do you use a program?

: Data → Data

plusOne : Int → Int

- Device that runs program (Joseph)
- Call it
- Brandon: compile it

How do you run a program?

What is compilation?

(JavaScript)

- Translate from human-read code to executable instructions (x86)

Declan

compiler : Program[JS] $\rightarrow$ Program[x86]

compiler(plusOne) $\Downarrow$ plusOne$_{x86}$

$\uparrow$

"produces"

"evaluates to"

What is an interpreter?

a program (device) (computer) that can run programs

$$\text{interpreter} : \text{Program} \times \underset{\text{(input)}}{\text{Data}} \to \text{Data}$$

$$\text{interpreter}_{\text{JS}}(\text{plusOne}, 3) \Downarrow 4$$

Uh oh ...

$\text{interpreter}(\text{interpreter}, (\text{plusOne}, 3)) \Downarrow 4$

~~x86~~ your machine

Let’s Restart with Some PL Terminology

Idea: Bootstrapping

Key Terms: A *meta language* versus an *object language*.

Some Terminology

A *programming language* is

instructions for interpreter

representation of code

notation for describing computation

A *program* is

something (an instance) written

in the PL notation

An Analogy: Cooking

recipe = program

cook = interpreter

Syntax and Semantics

Syntax is

how it looks (Greg)

↳ program

what things you can write that are programs (in the PL of interest)

Semantics is

meaning = how the program executes

$3+1 \Downarrow 31$?

Meta Language versus Object Language

object language — the language under study

meta language — the language used to study

Interpreter

An interpreter is a program that executes other programs.

An interpreter is written in a meta language (also called an *implementation language*) that interprets programs in an object language (also called a *source language*).

interpreter : $\text{Program} \times \text{Data} \rightarrow \text{Data}$

JavaScript

|

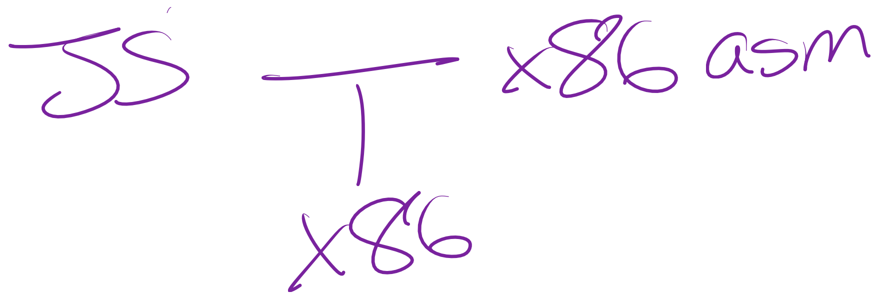
x86

"I diagram"

Compiler

A compiler is a program from one object language (the *source language*) translating a program in another object language (the *target language*).

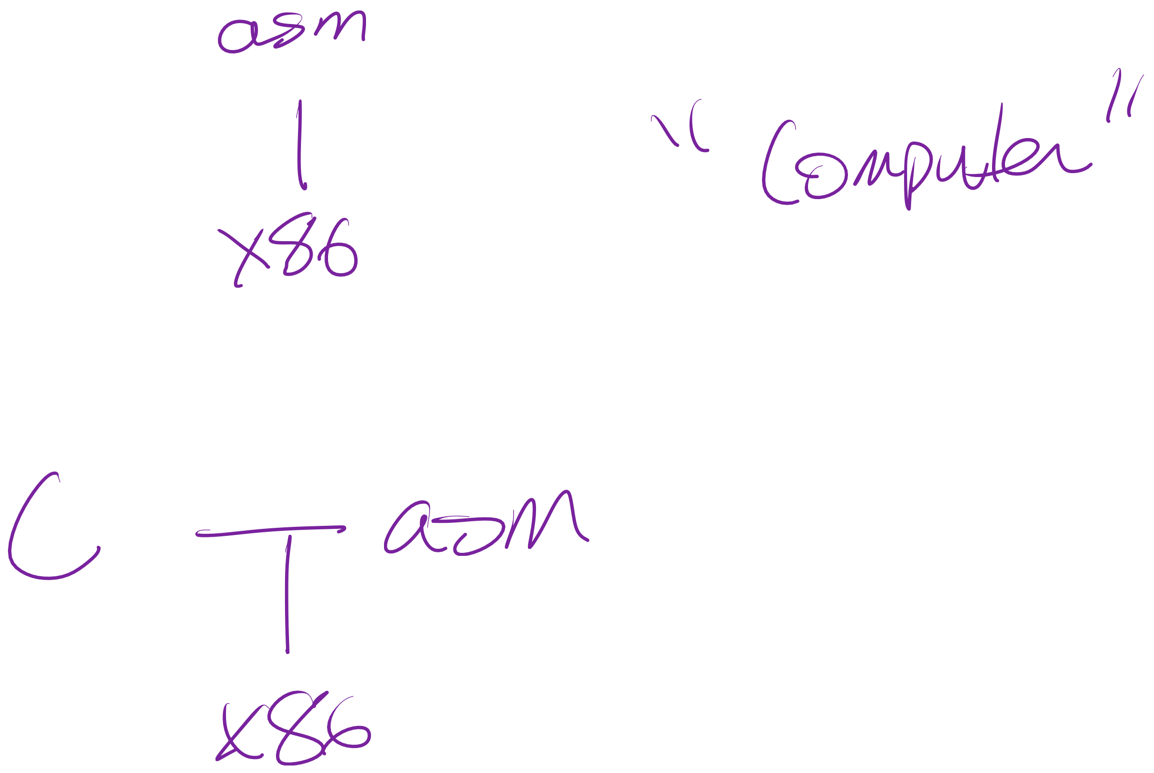
(obj)



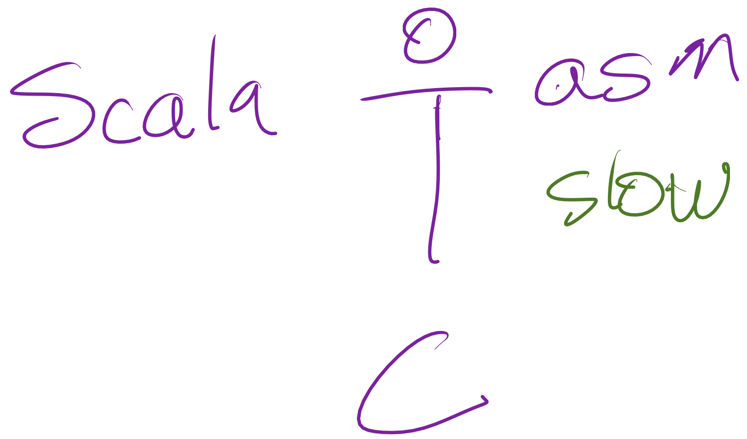
T diagram

↳ "translator"

Back to the puzzle …

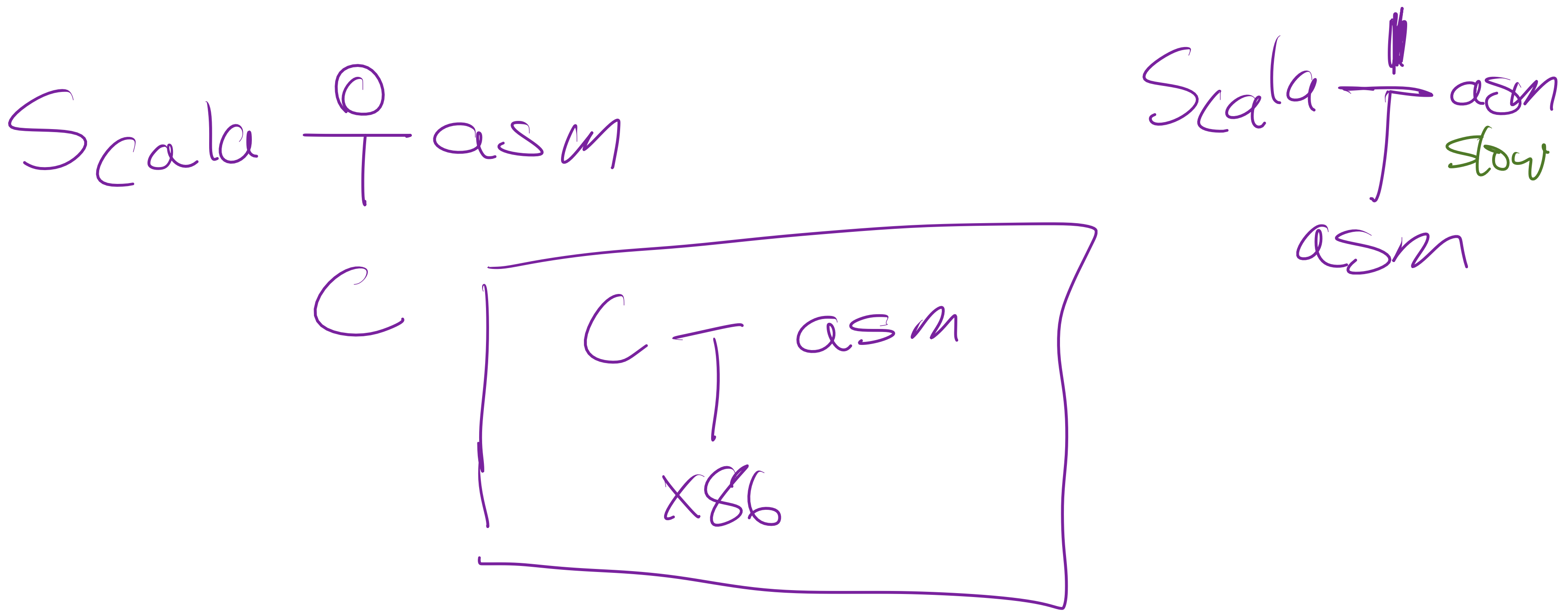


Step 0

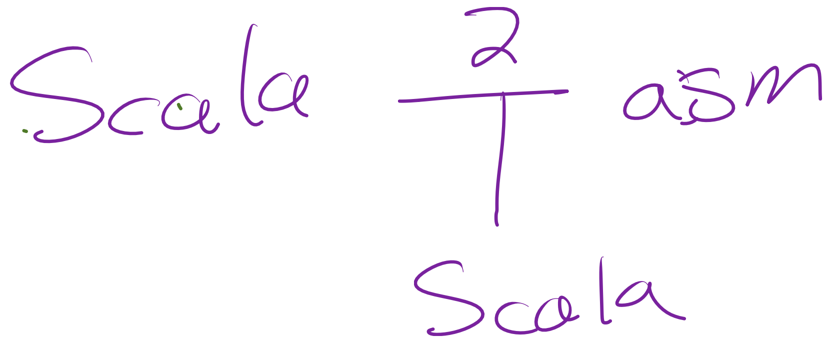


"quick and dirty"

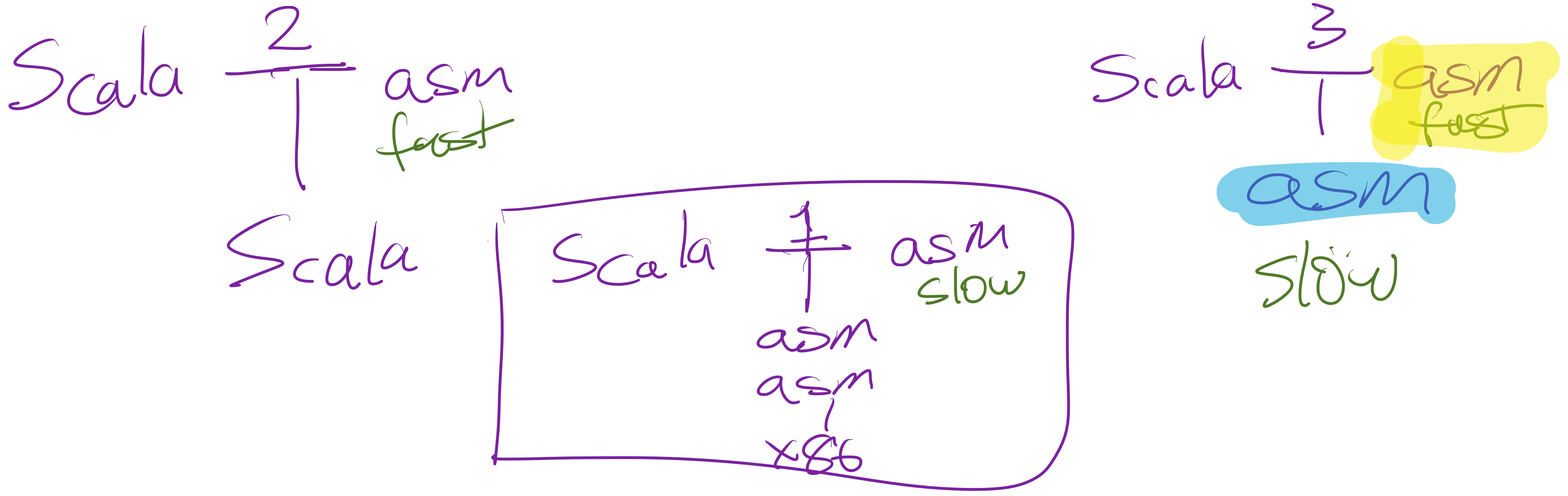
Step 1



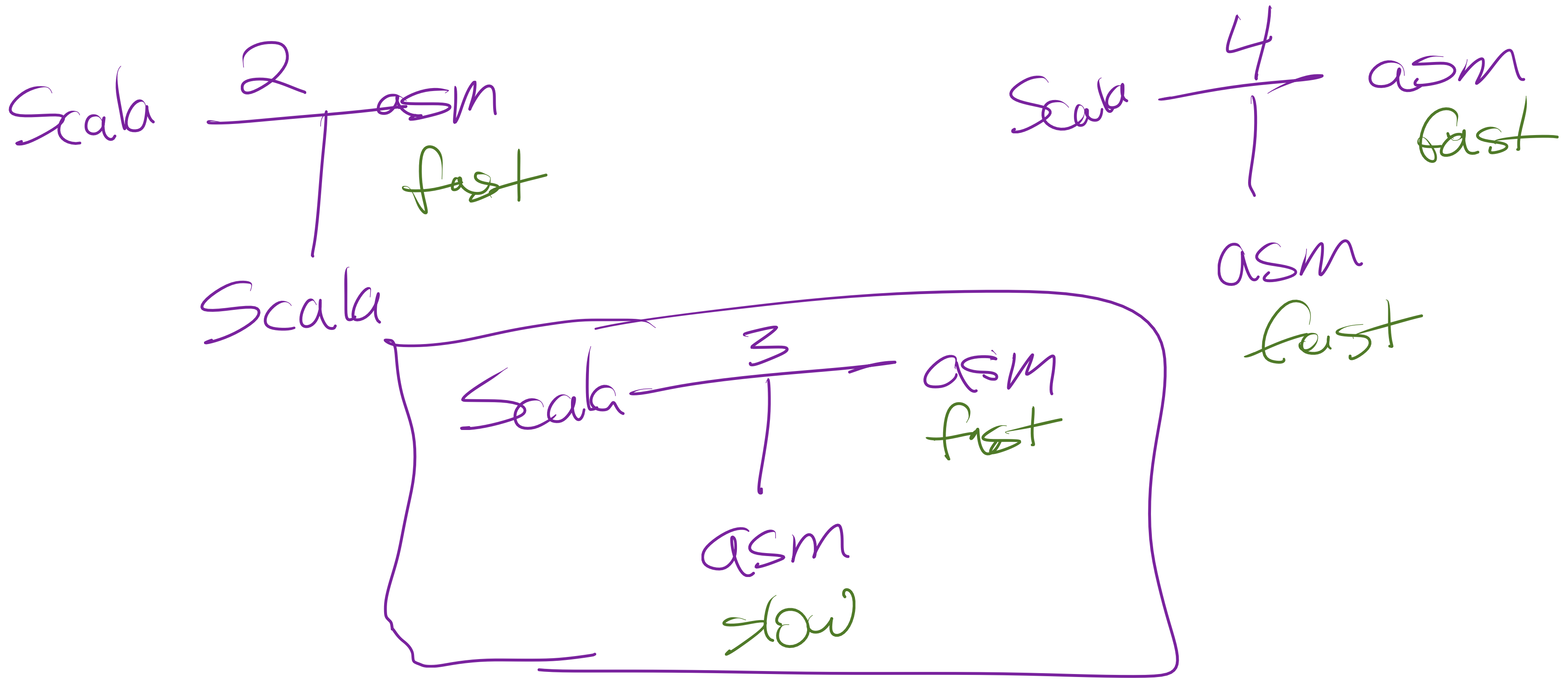
Step 2



Step 3


Scala
2
asm
fast
Scala
Scala
1
asm
slow
asm
asm
x86
Scala
3
asm
fast
asm
slow

Step 4


Scala
2
asm
fast
Scala
Scala
3
asm
fast
asm
slow
Scala
4
asm
fast
asm
fast

Step 5

Meeting 01: Welcome

Principles of Programming Languages - CSCI 3155 - Fall 2024

Professor Bor-Yuh Evan Chang

IF I HAD A COMPUTER, I'M SURE I'D GET BETTER GRADES ON MY BOOK REPORTS.
YOU'D STILL HAVE TO READ THE BOOK AND TELL THE COMPUTER WHAT YOU WANT TO SAY, YOU KNOW.
MAN, WHAT'S ALL THE FUSS ABOUT COMPUTERS ?!
WATTERSON

Announcements

- Today’s Homework: Post an introduction of yourself on Piazza
- Today’s Reading from Schedule
  - Syllabus: Coming next time means you have read and agreed
  - PPPL Introduction: Course Approach, Getting Your Money’s Worth

Getting to Know You

“I, …, wonder …”

Distraction-Free Classroom

Let’s turn off our cell phones and laptops.

Just imagine that we have class in Rocky Mountain National Park.

Your Guide this semester

CUPLV

Programming Languages and Verification
University of Colorado Boulder
plv.colorado.edu

scholars

More Guides

Section Guides (= Teaching Assistants or TAs)

Dakota Bryan

Anish Thilagar

Md Rezwanur Rahman

Juan Vasquez

Your Section Guide is responsible for being your guide through the learning materials.

Tutors (= Course Assistants or CAs)

Tome Dudanov

Yash Singh

Your Tutors are responsible for providing additional one-to-one or small group tutoring and review sessions.

Feedback Staff

Reviewers (= Graders)

Your Reviewer is responsible for providing personal and targeted feedback on the artifacts you produce.

Administrative Staff

Course Manager (CM)

Lawrence Khadka

Your Course Manager is responsible for the logistics and administration of the course.

Even More Guides

Introduce yourself to someone you haven’t met before.

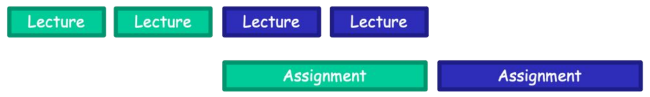
Two minutes!

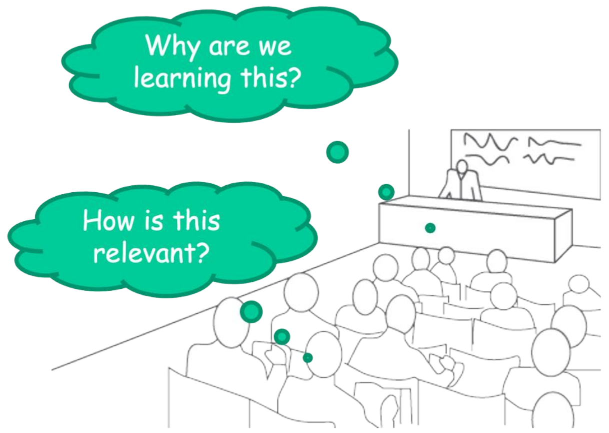
Course Approach: Active Classroom

- **Build** interpreters for mini-languages.
- 2-3 weeks of active discussion — in class and online (Piazza) — towards incrementally completing a lab! Do intermediate assignments along the way.
- Recitation sections are lab sessions. Bring a laptop.
- Assignments due Friday 6pm. Everyone gets an automatic “stuff happens” 24-hour grace period.
- No late assignments but generous “redo” policy and lowest score dropped.

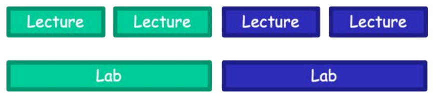
Course Approach: Active Classroom

Traditional Format





Our Class



Course Approach: Active Classroom

Please interrupt at any time!

It’s completely ok to say, “I don’t understand. Please say it another way. Slow down!”

Attempting assignments and reading before class prompts the discussion. There is an expectation on you to be active.

Learning Oath

About You: Your Goals and Worries

1. What do you want to get out of this class?
2. What do you think this course is about?
3. What worries you about this class?

What worries you about this class?

Getting Your Money's Worth

Why study PL?

Why study PL when we have ChatGPT?

Testimonials

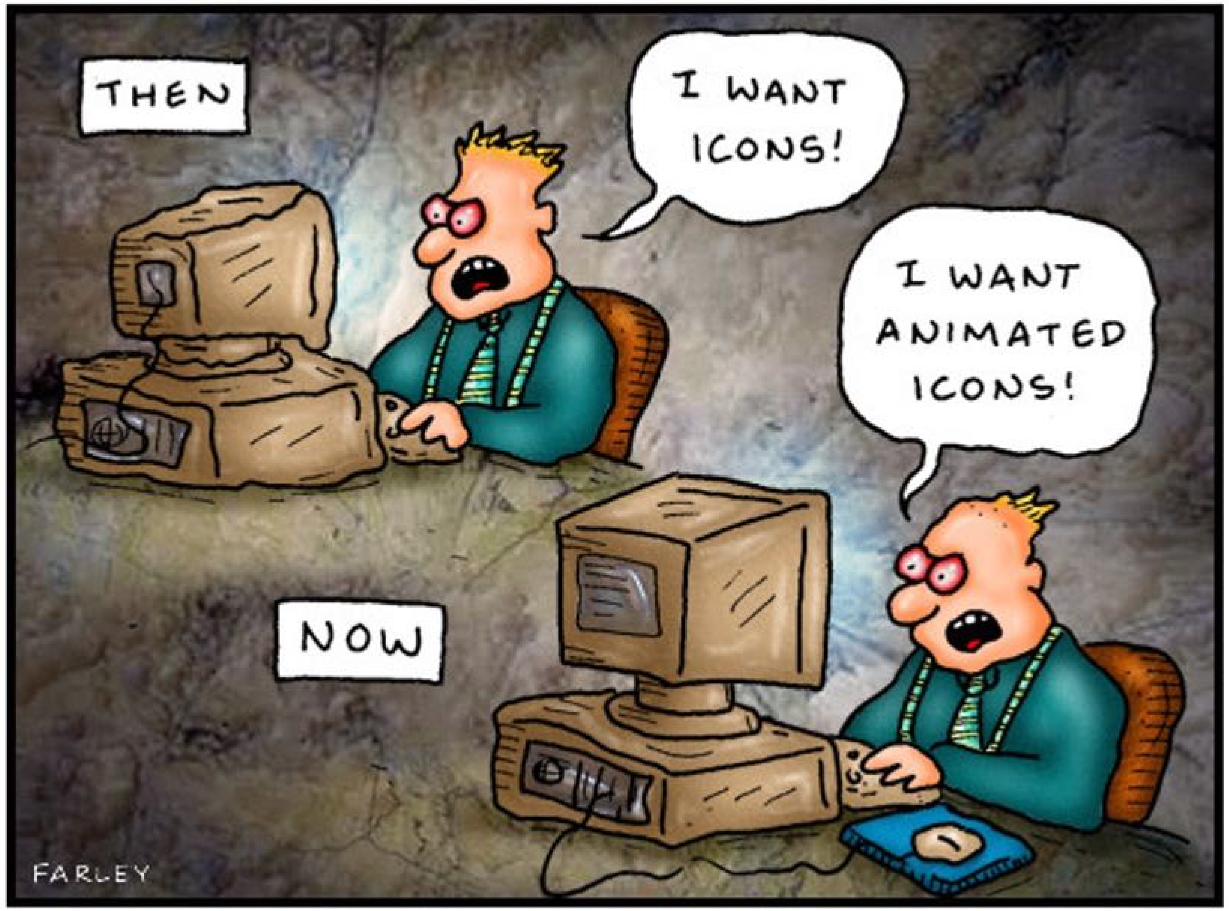
Come on Thursday to hear from 3155 alumni!

- Luis Olivas (Workday)
- Andrew Arnopoulous (Fitbod)

Don’t we have enough PLs?

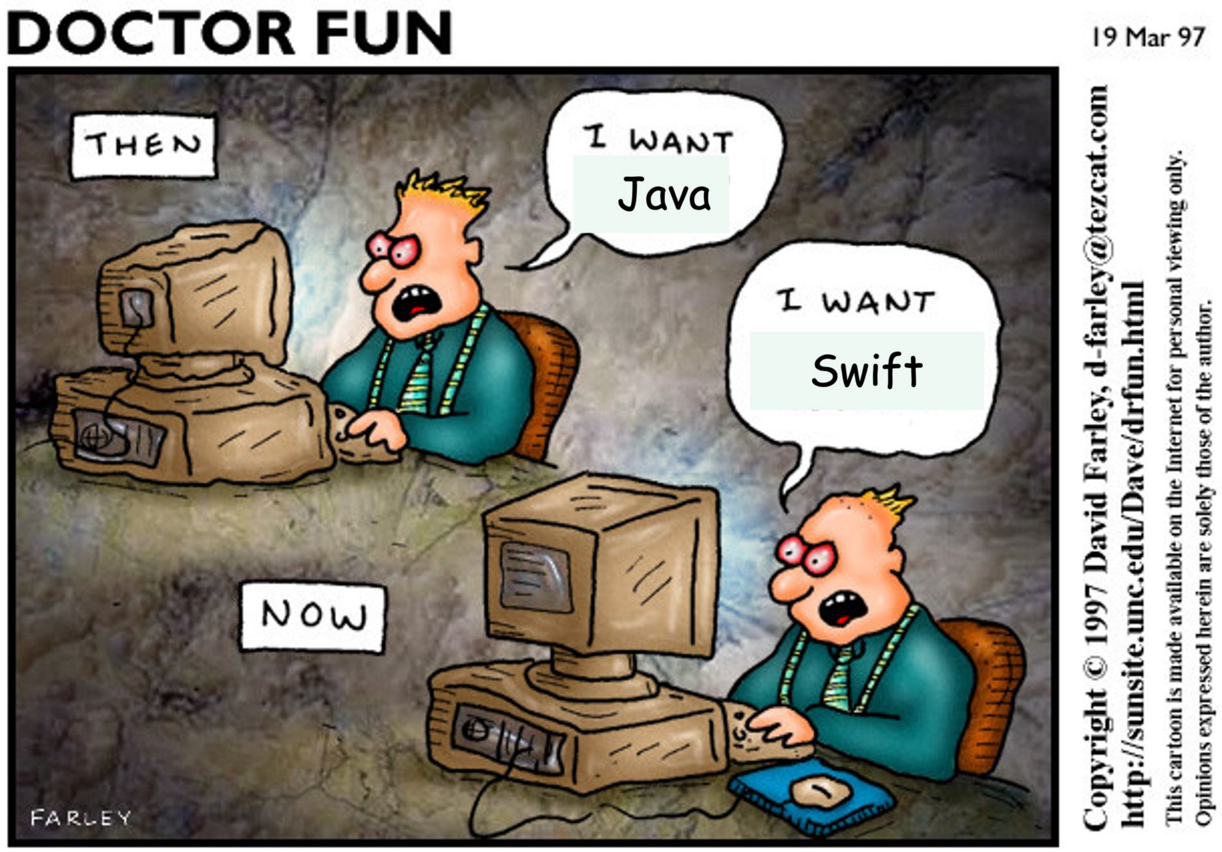
DOCTOR FUN

19 Mar 97



Progress

A Dismal View



Progress

This is not a history course.

This is not a trip to the zoo

This is more like a study of anatomy

Learn new languages

You will need to learn many languages during your careers.

You will learn concepts that make it easier for you to learn new languages in this class.

Improve background for selecting a suitable one for your task

Have you ever had to pick a language?

What criteria did you use?

How do you understand “about pages”?

Gain new ways of viewing computation and approaching algorithmic problems

Have you heard of Google MapReduce?

How about Meta React?

… inspired by ideas from PL theory

Gain new ways of viewing programs

What does it mean for two programs to “behave the same”?

You will be able to *reason* about computation.

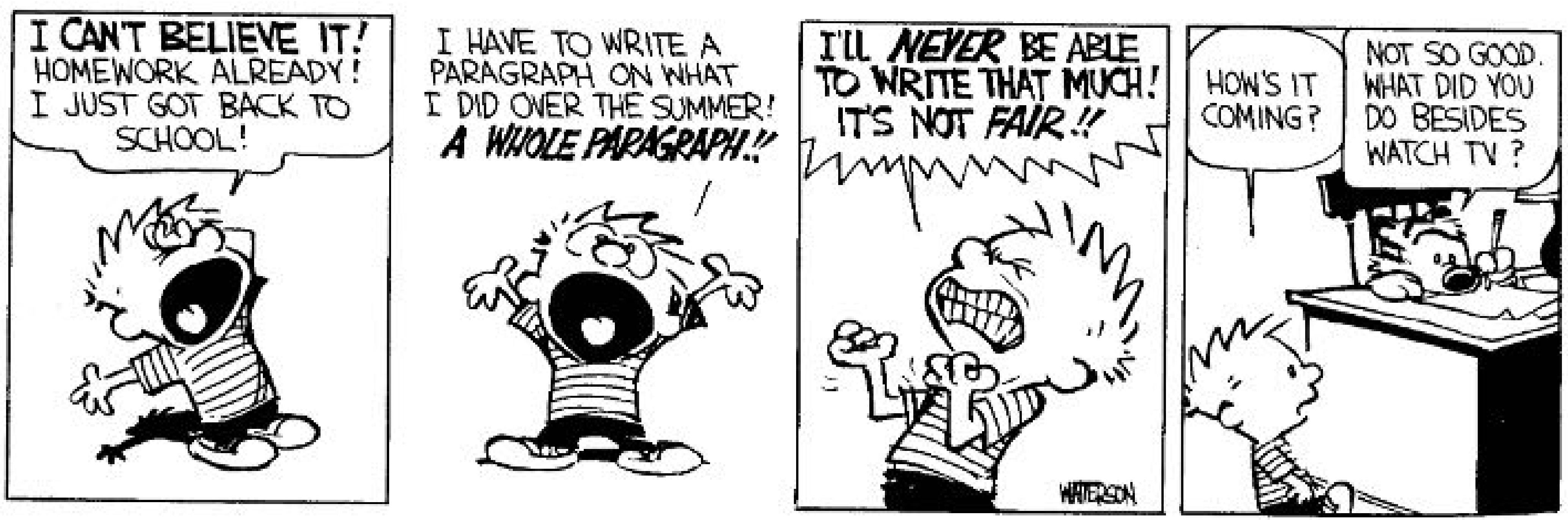
Being able to compute versus being about to reason about computation is very different (analogy: arithmetic versus algebra).

Gain insight into avoiding mistakes for when (not if!) you design languages

Use AI assistants to accelerate your creative design

Other reasons?

Syllabus Policies


I CAN'T BELIEVE IT! HOMEWORK ALREADY! I JUST GOT BACK TO SCHOOL!
I HAVE TO WRITE A PARAGRAPH ON WHAT I DID OVER THE SUMMER! A WHOLE PARAGRAPH!!
I'LL NEVER BE ABLE TO WRITE THAT MUCH! IT'S NOT FAIR!!
HOW'S IT COMING?
NOT SO GOOD. WHAT DID YOU DO BESIDES WATCH TV?
WATERSON

Requirements

Distinguishes between learning activities:

- Reading
- Quizzes and Exercises (25%)
- Lab Assignments (30%)
- Participation (5%)

And summative assessments:

- In-Class Exams (20%)
- Final Exam (20%)

Highlights

- Integrity of the Course Materials
- Collaboration Policy
- Redo Policy
- No late assignments but one drop
- No make-up exams (unless emergency or special accommodation)
- Special accommodation requests (religious observances) within first four weeks
- Regrades/redo requests within one week