CIS 552
Advanced Programming

Fall 2022

Welcome!

- Sit anywhere
- Make a name tag
- Introduce yourself to your table
- Sign in and pick a team name

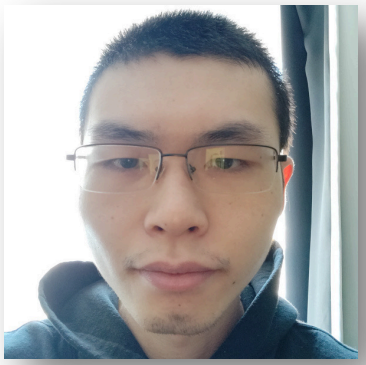
Course Staff

Instructor: Dr. Stephanie Weirich

sweirich@seas.upenn.edu

OH: *Wednesdays, 3:30-5pm Levine 510*

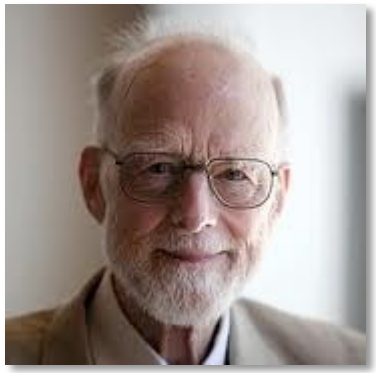
TAs: Yiyun Liu, Joe Cutler, Emmanuel Suárez



What is Advanced Programming?

- Good programmers get the job done

- Excellent programmers
  - write code that other people can understand, maintain and modify
  - rewrite/refactor code to make it clear and simple
  - use and create *abstractions* to capture fundamental designs



Tony Hoare
Turing Award Lecture 1980

"There are two ways of constructing a software design:

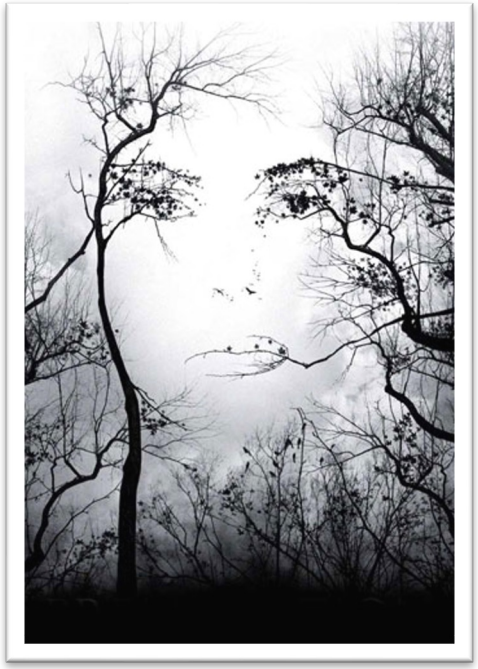
One way is to make it so simple that there are obviously no deficiencies,

and the other way is to make it so complicated that there are no obvious deficiencies.

The first method is far more difficult."

Simplicity through Abstraction

- Readable
- Reusable
- Modifiable
- Predictable
- Checkable



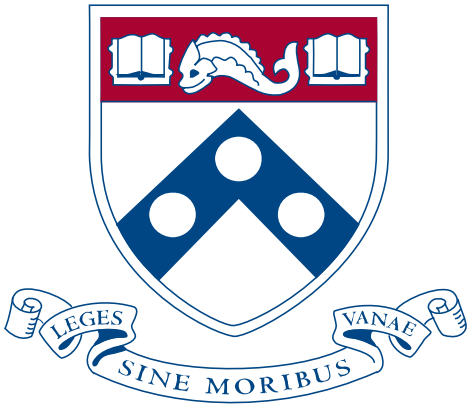
- Advanced type systems: Multiple levels of abstraction available

Simplicity through Purity

- Readable
- Reusable
- Modifiable
- Predictable
- Checkable

- Functional Programming: Focus on what code **means** instead of what it does
- Programming inspired by pure mathematics

CIS 552



Haskell

Why Haskell?

- Leading edge language research.

Why Haskell?

- Beautiful mathematics.

Why Haskell?

- Stretches Your Mind.

Why Haskell?

- Fun.

Course content

Functional Programming

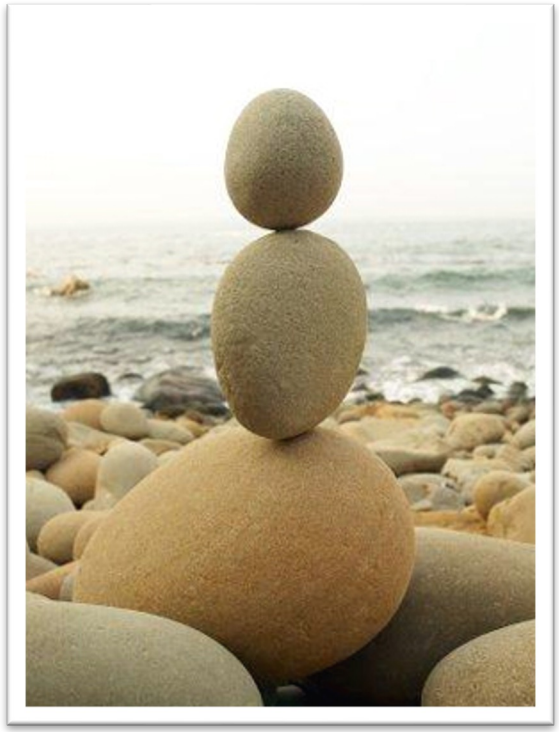
- Black-belt Haskell
- Mathematical approach to programming
- Many small-scale case studies

Advanced Programming Techniques

- Modular design and abstraction
- Black-belt types
- Test driven development
- Collaboration (pair programming)

Lots of programming!

- Small in-class exercises
- Bi-weekly homework assignments
- End of semester project



What this course is not

- CIS 3500/5730, Software Engineering
  - Focuses on "Software in the large”
  - How to deal with code you didn’t write
  - Problems that arise in projects that are too large for one person
    - lifecycle models
    - project management
    - design modeling notations (UML)
    - formal specification
- The two courses complement each other

What are you most excited about for CIS 5520?

- Haskell
- Functional Programming
- Monads
- Learning a different way of programming
- Learning to be a better programmer
- Interactive class format
- Fun projects
- Random partners

What concerns do you have about CIS 5520?

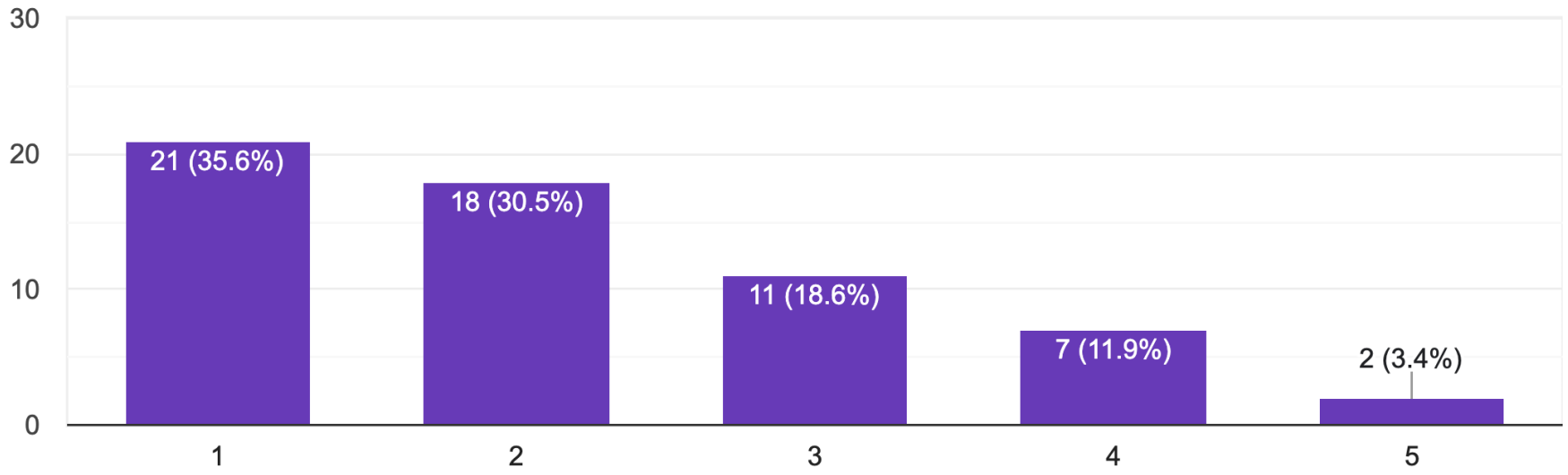
- What does active learning mean for CIS?
- Will the workload be too much? Can I balance it with everything else? (other classes, recruiting, etc)
- Will the math/functional programming be hard?
- Will I be able to keep up?
- Will random partners and collaborative assignments work?
- Will this course be useful to me in my future career?
- Will I get off the waitlist?

Audience

- People with strong background in programming and mathematics
- No experience with FP expected, but helps
  - We'll split the class by background for the first few weeks, but will converge quickly
- Undergraduates, Masters, and PhD students together

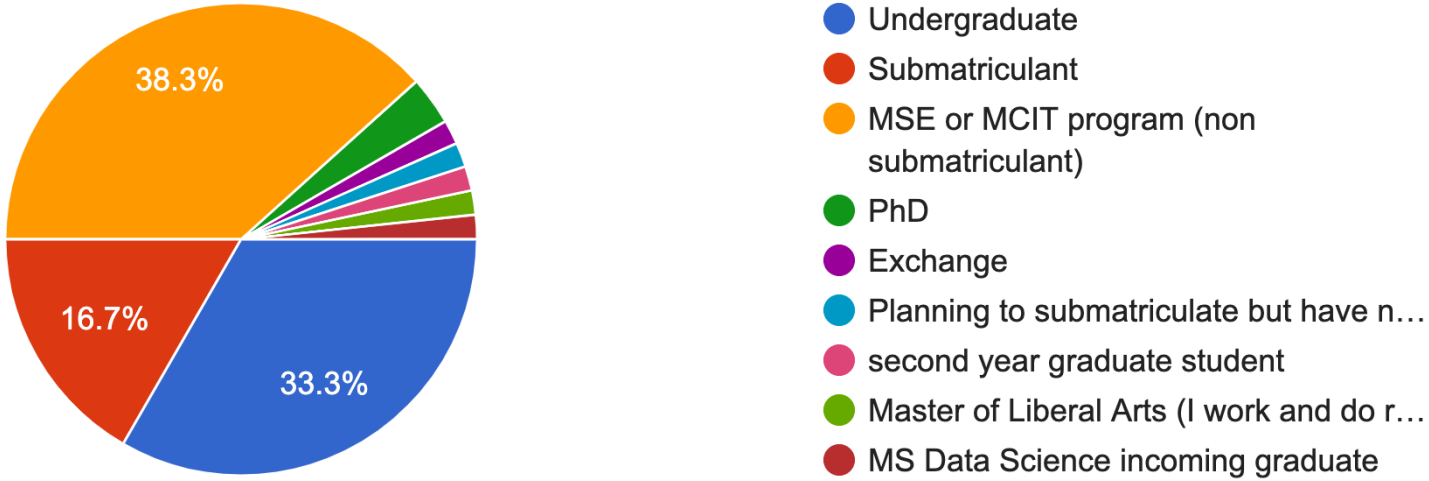
How much experience do you have with functional programming ?

59 responses



What is your status in Fall 2022

60 responses



How will this all work?

General Course Structure

- Every week has a topic
  - Read module and complete quiz by Sunday night
  - Active in-class discussion Monday based on module
  - Active in-class exercise Wednesday
  - Homework due alternate Thursdays (midnight), covers two topics
- Some weeks will be different
- End-of-semester: final project

Grading Structure

- 15 % Quizzes
  - pre-class quizzes (study "lecture" content asynchronously)
  - **first module/quiz available now**
- 15 % Active learning / Attendance
  - in class exercises and discussion
  - office hours – let's chat!
- 50 % Programming assignments
  - in pairs, most *randomly* assigned
  - graded on correctness, style and (asymptotic) efficiency
  - **first assignment available now**
- 20 % Final Projects (your choice)

Because of the active learning component, in person participation is essential!

Asynchronous "Lecture" Content

- Course content available in two forms
  - Formatted reading: on the public course website (under "Schedule")
  - IDE experimentation (*recommended*): public repo in github
- **Read module "Basics" before next class**
  - Part of the "01-intro" project on github
  - Fill in the "undefined" parts in your IDE
- Canvas quiz on material due before next class (midnight prior)
  - Make sure that you ask a question to guide our next in-class discussion

Active Learning Goals

- Goal for the semester: create a CIS 5520 *community*
  - You should get to know me and the TAs (they're great!)
  - You should get to know each other (you are all great!)
- Forced, random interactions during class time and outside
  - Small and large group discussions
  - In-class exercises with a partner or table
  - Random homework partners
  - TODAY: PL-themed icebreaker game

Fall 2022 and COVID-19

- Masks are **required** for everyone, out of consideration
- Attendance required, but don't come to class if you are sick
- No class recordings / hybrid option (doesn't work for 5520)

Homework #1

- Based on "Basics" (available now) and "HigherOrder" modules (tba)
- You will be provided with a *private* repo to complete the assignment
- Work alone or with a partner (your choice), only one person should submit
- Must compile to get any credit
- **Due Thursday, Sept 15th at midnight**
- Late policy (all homework assignments)
  - 10 point penalty for up to 24 hours late
  - 20 point penalty for up to 48 hours late
  - no credit for assignments submitted after 48 hours
  - if you have an emergency, please ask for an extension

Where to go for what

- Public site (http://www.seas.upenn.edu/~cis5520)
  - Haskell related asynchronous lecture material, HW instructions
- Github organization (https://github.com/upenn-cis5520)
  - Code repos for lecture content, in-class exercises (public) & HW (private)
- Canvas site (https://canvas.upenn.edu/courses/1675188)
  - Syllabus
  - Quizzes
  - Link to Ed (Announcements and questions)
  - Link to Gradescope (Homework submission)

Things to do right now

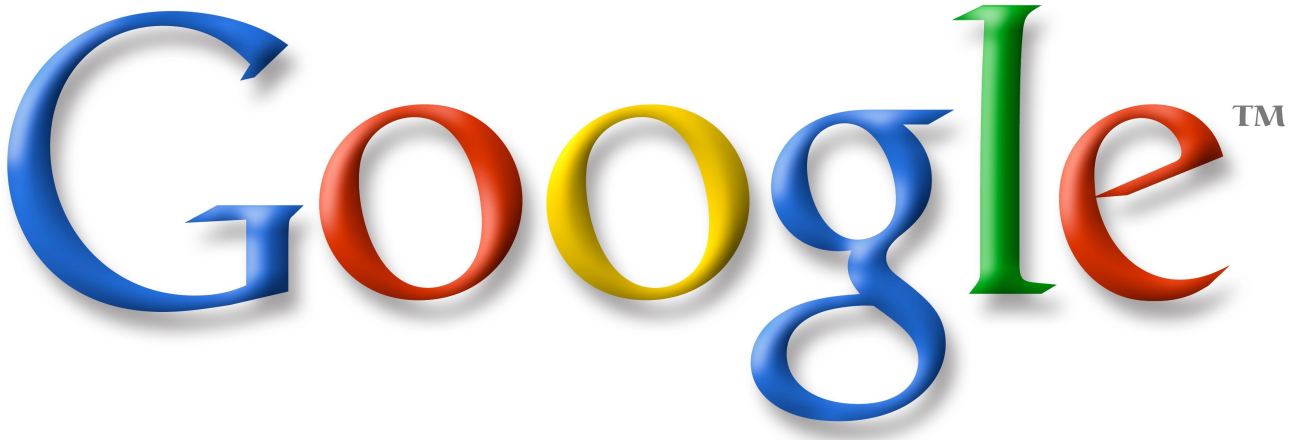
- Create a github account (if you do not have one)
- Respond to Fall 2022 survey (if you haven't already)
- Introduce yourself to the others at your table
- Start reading "Basics" module, install software, watch for access to your hw01 repo (after class)
- Office hours this week:
  Stephanie: Today, 3:30-5 PM, Levine 510
  Yiyun: Friday, 10-11AM, Levine/GRW 054

https://pl-quiz.herokuapp.com/

- Each table is a team and should choose a team name
- **One** person should enter the answers for the whole team
- Don't refresh or leave the page or you will lose your score
- Winner is the team with the most points by 1:20PM
- Rules
  - No google / web searching allowed
  - No "extra" teams allowed

fin

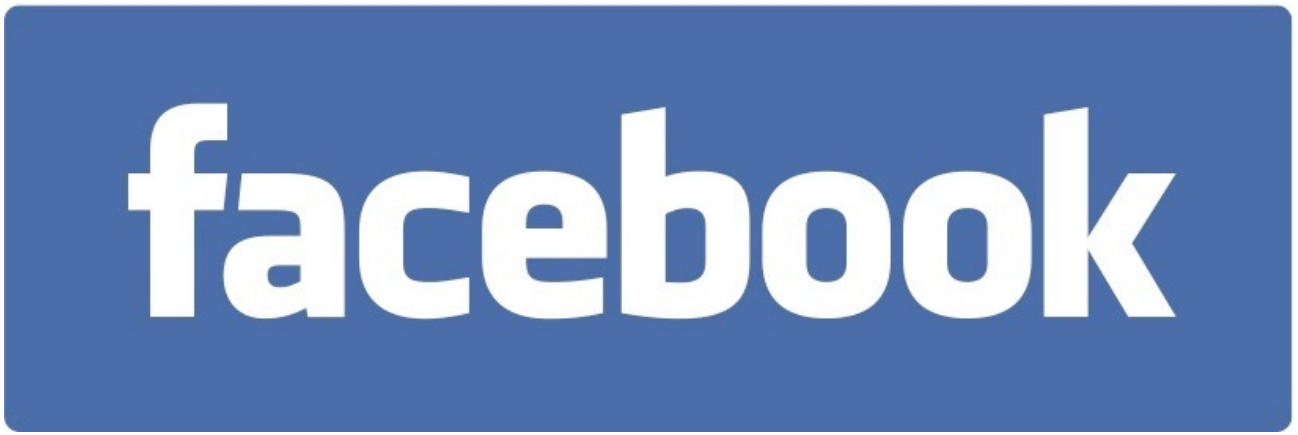
So, Who Uses FP?



So, Who Uses FP?

Microsoft®

So, who uses FP?

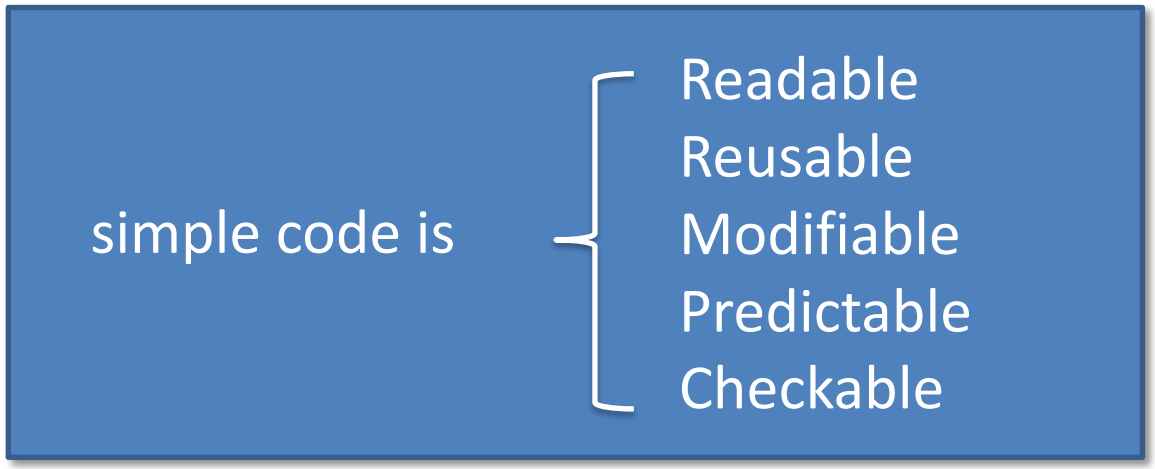


So, Who Uses FP?

So, Who uses FP?

Goal: Obviously no deficiencies

- Want code that is so simple, it obviously works



- OK… so what makes code simple?