Course Staff

Instructor: Dr. Stephanie Weirich  
sweirich@cis.upenn.edu  
OH: Today after class

TAs: Nicholas Rioux  
      Harrison Goldstein  
      Calvin Beck
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
  – use and create *abstractions* to capture fundamental designs
"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 structure
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
Why Haskell?

• Leading edge language technology.
Why Haskell?

• Beautiful mathematics.
Why Haskell?

• Blows 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
- (Remote) Collaboration (pair programming)

Lots of programming!
- Small in-class exercises
- Weekly-ish homework assignments
- Larger project at the end
What this course is not

• CIS 350/573, 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
Audience

• People with strong background in programming and mathematics
• No experience with FP expected, but helps
• Priority to CIS students
• If not registered, add your name to the waitlist on course website
How will this all work?
Course Structure

• 30% Active learning / participation
  – pre-class quizzes (study "lecture" content asynchronously!)
  – in class exercises, in class discussion … on Discord
  – office hours – come at least once during the semester

• 50% Programming assignments
  – in pairs, mostly *randomly* assigned
  – graded on correctness, style and (asymptotic) efficiency
  – First assignment available soon

• 20% Final Projects (your choice)

Because of the active learning component of this course you must "attend" every class!
Asynchronous Content

• Course content will be posted on the public course website (under "Schedule") for asynchronous reading
  – Available now: "Basics" module
• The code extracted from the module is available.
  – You should follow along in your IDE
  – Try to fill in the "undefined" parts as you go
• There will be quizzes in Canvas.
  – First one (on "Basics") is due before class next Wednesday!
Active Learning Goals

• Goal for the semester: create a CIS 552 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 synchronous class time
  – (Quizzes on asynchronous content due before start of class)
  – Small and large group discussions
  – In-class exercises with a partner or table
  – TODAY: PL-themed icebreaker game
Homework #1

• Available soon, based on "Basics" module
• 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 Tuesday, 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
Where to go for what

• Public site (http://www.seas.upenn.edu/~cis552)
  – Haskell related asynchronous content, HW instructions

• Github organization (https://github.com/cis552)
  – Code repos for lecture content, in-class exercises & HW

• Canvas site (https://canvas.upenn.edu/courses/1528990)
  – Syllabus, Zoom links, lecture recordings, quizzes, grades
  – Piazza (Announcements and questions)
  – Gradescope (Homework submission)

• Discord server
  – Social interaction & community, in-class exercises
Things to do right now

• Respond to Fall 2020 survey (if you haven't already)
• Create a github account (if you do not have one) and tell us your github id (link to form on Canvas)
• Sign up for class Discord server (link on Canvas)
  – Don't use your personal account
  – Your nickname should be your full name
  – Introduce yourself on #introductions, pick a Table
• (After class) Start reading "Basics" module on public website, install software (stack, VSCode)