Preliminaries

CS 340: Programming Paradigms and Patterns
Michael Saelee <lee@iit.edu>
Michael (Sae) Lee

- lee@iit.edu
- http://moss.cs.iit.edu
- Office: SB 226C
- Hours: Wed & Fri 3:15PM-5:15PM
TA: Samuel Golden

- sgolden2@hawk.iit.edu
- Office: SB 108
- Hours: Tue 3:30PM-6:00PM, Thu 4:30PM-6:30PM
Agenda

- What is “PPP”?
- Why Haskell?
- Why take CS 340?
- Administrivia
  - Websites, References, Grading, etc.
“Programming Paradigms and Patterns”
Paradigm

- Model for how a program in a given language is organized, expressed, and/or executed
  - e.g., procedural, imperative, object-oriented, functional, declarative
- We will be focusing on the *functional* paradigm
Why Functional?

- Very different set of operating assumptions from your (likely) first model, imperative programming
- No state mutation $\rightarrow$ referential transparency
- Arguably easier to reason about (rigorously) and use for concurrency
- You’ll read a paper on this for Friday!
Pattern

- A reusable template for solving a common class of problem(s)
- May be paradigm/language specific, and typically as abstract as possible to encourage reuse
E.g., Imperative & OOP patterns

- Loops/Iterators for array, list, or collection traversal
- Encapsulation with setter/getter methods
- Singleton & Factory patterns
- Observer pattern, aka Publish/Subscribe
Our focus: Functional patterns

- Structural and Generative recursion
- Functors and Monads
- Monoids and Foldables
- Monadic parsing
- Etc.
Haskell

- Our functional language of choice: Haskell
  - **Pure**: purely functional; side-effects are contained/earmarked
  - **Statically typed**: types are checked at compile time
  - **Lazy**: expressions aren’t evaluated until absolutely necessary
- Likely very different from another language you’ve used!
Why Haskell?

- It’s fun, surprising, and powerful!
- Learning a (different) new language gives you an entirely new way to think about and tackle problems
- Valuable, even if you don’t actually code the solution up in said language
A Taste of Haskell

```haskell
fibs = 0 : 1 : zipWith (+) fibs (tail fibs)

primes = filterPrime [2..]
  where filterPrime (p:xs) =
    p : filterPrime [x | x <- xs, x `mod` p /= 0]

quicksort :: Ord a => [a] -> [a]
quicksort [] = []
quicksort (p:xs) = (quicksort lesser) ++ [p] ++ (quicksort greater)
  where lesser = filter (< p) xs
        greater = filter (>= p) xs
```
Why take CS 340?

- You love to program
- You love programming languages
- You are frustrated with languages you currently know
- You want to learn new ways to reason about programming
- This will help in later classes and your career
Topics (not exhaustive)

- Functional programming
- Haskell Types and Typeclasses (like OOP on steroids)
- Higher Order Functions
- Functors and Monads
- Automated Property-Based Testing
- Concurrency and Software Transactional Memory
§ Administrivia
Prerequisites

- I assume you are …
  - fluent in some programming language
  - familiar with procedural & OO paradigms
- comfortable with development processes:
  - compilation, debugging, testing
Online resources

1. Course website: moss.cs.iit.edu/cs340
   - static information
     - lecture calendar, readings, assignment write-ups, links, etc.
Online resources

2. Blackboard
   - only for grade reporting!
Online resources

3. Piazza: discussion forum
   - all class-related questions
   - monitored by TA
   - scales way better than e-mail
Online resources

4. Bitbucket: Git repository hosting
   - you will share a private repository with the TA and me (invitations coming soon)
   - lecture notes will be distributed via Git
   - machine problems will be submitted via Git
References

- Miran Lipovača, *Learn You a Haskell for Great Good!*
- Graham Hutton, *Programming in Haskell*
- O’Sullivan, Stewart, Goerzen, *Real World Haskell*
Grading

- 50% Machine Problems
- 5-7 Haskell programming assignments
- 25% Midterm Exam
- 25% Final Exam (Cumulative)
For Friday

- Read Hughes’s “Why Functional Programming Matters” (at least sections 1 & 2, if you can get further, great!)
- Start reading “Learn You a Haskell”
- Clone the repo: https://bitbucket.org/michaelee/cs340.git
- Install the Haskell Platform (ideally, using Haskell Stack, haskellstack.org) & bring laptop to class