Detailed Course Schedule

Week 1

Monday

- What is Functional Programming (1.2)
- Haskell Language features (1.3)
  - Conciseness
  - Powerful types
  - List comprehensions
  - Recursive functions and types
  - Higher-order functions
  - Pure functions separated from effectful functions
  - Lazy evaluation
- Examples: sum (two versions), qsort (1.5)
- In-class activity: Write product

Wednesday

- Introduction to GHC, GHCi (2.1, 2.2)
  - Math operations. Compute the 10th power of 2.
- Standard Prelude (2.3, 2.4)
  - Basic list methods
  - Function application needs no parentheses
- Loading script files (2.5)
  - Write in the custom qsort function
    - load it
    - ask for its type
    - test it with strings and numbers
  - Comments, layout rule
- In-class activities:
  - Write function addInterest
  - Write function weirdOp x y that does "sum minus product"
  - Write function haveEnough n lst that checks if list has that many elements
  - Write a function last that given a list returns the last element (use other existing functions)

Friday

- Basic types (3.1, 3.2)
  - Bool
  - String vs Char
  - Int vs Integer
- List types (3.3)

    - Strings are really `[Char]`
    - List of lists
- Tuple types (3.4)
    - Difference between tuples and lists?
    - Activity: Think of things we might represent with tuples.
- Function types (3.5)
    - Look at types of standard functions
    - In-class activity:
        * TODO

Week 2

Monday

- Curried functions (3.6) and their type
- Polymorphic types (3.7)
- Overloaded types (3.8)
- Type classes (3.9)
- In-class activity:
    - Determine type for certain functions, with type classes in mind
    - Build up to the Map and Set ADTs

Wednesday

- Conditionals (4.2)
- Guarded equations (4.3)
- In-class activity:
    - Define Collatz function, explore its behavior
    - Define function that given pair (string1, string2) returns whichever string is longer
    - Do the same problem by first writing a function "min" (return to this later as a higher-order-function — argmax)

Friday

- Pattern matching (4.4)
    - Booleans
    - Tuples
    - Lists
- In-class activity:
    - Define function that does `||`
    - Define function that finds the maximum element on a list by dropping the smallest of the top two elements.
    - Write a function that tests if a list of numbers is ordered in increasing order.

Week 3

Monday

- Anonymous functions (4.5)
- In-class activity: Write anonymous functions that:
  - given two lists return the longest of the two
  - given a list take out the first 3 elements
  - given a pair of numbers (x, y) return the y
  - given a number and a list, return the list if the number is even, and the reverse list if the number is odd
- Sections (4.6)
- In-class activity: Write sections that:
  - cut a number in half.
  - turn a number into its negative.
  - turn a number into a list with one element.
  - append to any list the number 1.
  - compute the division-by-2 remainder of a number.
  - take out the first 3 elements from a list.

Wednesday

- Basics of list comprehensions (5.1-5.3)
- In-class activity: Write list comprehensions that:
  - given a list of pairs return a list of the first entries in those pairs
  - given a list of pairs of numbers return a list of the largest number on each pair
  - given a list of numbers return the squares, but only for the numbers that are odd
  - given a list of elements, return a list of pairs where the first coordinate is the same element from the original list, and the second coordinate is the number 1
  - given two lists of numbers produce pairs of the sum and product for each combination, but only for the combinations where the number from the first list is bigger than the number from the second list
  - given a list of lists of numbers, flatten out into one list, but ignoring any lists that have less than 3 elements
  - given a list of numbers, return the 1st, 3rd, 5th etc (use zip with [1. . .])

Friday

- String comprehensions (5.4)
- Long example: Caesar cipher (5.5)

Week 4

Monday

- Numerical recursion (6.1)

- In-class activity:
  - Implement fast exponentiation
  - TODO
- Recursion on lists (6.2)
- In-class activity:
  - TODO

Wednesday
- Recursion with multiple arguments (6.3)
- Multiple/Mutual recursion (6.4, 6.5)

Friday
- Recursion practice (6.6)
  - Insert on a sorted list
  - Search on a sorted list (boolean result)
  - Check if list sorted
  - Group up consecutive same elements in list
- Lab: Implementation of map via sorted lists of pairs

Week 5

Monday
- Higher-order functions (7.1)
- Processing lists (7.2)

Wednesday
- foldr (7.3), foldl (7.4)

Friday
- composition, point-free style (7.5)
- Example: voting algorithms (7.7)

Week 6

Monday
- More practice with higher-order functions?

Wednesday
- type declarations, type aliases (8.1)
- data declarations (8.2)
- the Maybe type
- newtype declarations (8.3)

Friday
- Recursive types (8.4)
- Tree structures

Week 7

Monday
- Example use of types (Heaps?)
- Type-directed programming
- Creating a custom module (Heaps?)

Wednesday
- Information hiding and abstraction with modules (TODO)

Friday
- Class and instance declarations (8.5)

Week 8

Monday
- Example: Tautology checker (8.6)

Wednesday
- Example: TODO (abstract machine? 8.7 or something "better")

Friday
- More on modules and classes

Week 9

Monday
- Countdown problem setup (9.1, 9.2, 9.3, 9.9)
- Plan the solution

Wednesday
- Countdown problem solution (9.4, 9.5, 9.6)
- Combining generation and evaluation (9.7, 9.8)

Friday
- Interactive programming intro (10.1, 10.2, 10.3)
- Interactive programming sequencing, derived (10.4, 10.5)

Week 10

Monday
- Interactive programming example: Nym (10.7)
- Idea of a main loop?

Wednesday
- Graphics? (or tic-tac-toe?)

Friday
- Functors (12.1)
- Applicatives (12.2)

Week 11

Monday
- Monads (12.3)

Wednesday
- State in a pure world: The state monad (12.4)

Friday
- Lazy Evaluation intro (15.1, 15.2)
- Call-by-name may terminate (15.3)

Week 12

Monday
- Number of reductions and lazy evaluation (15.4)
- Infinite structures (15.5)

Wednesday
- Separation of control and data generation (15.6)

Friday

:

Week 13

Monday

:

Wednesday

:

Friday

:

Week 14

Monday

:

Wednesday

:

Friday

: