Schedule

A week-by-week breakdown of the material.

Overview

- Introduction to Haskell and Fuctional Programming (1.1-1.14)
- Compiler and interpreter (2.1-2.7)
- Basic types (3.1-3.7)
- Designing and writing programs (4.1-4.8)
- Tuples and lists (5.1-5.7)
- More programming with lists (6.1-6.8)
- Defining functions over lists (7.1-7.6)
- Input and output in Haskell (8.1-8.6)
- Patterns of computation (10.1-10.5)
- Higher-order functions (11.1-11.5)
- Developing higher-order programs (12.1-12.7)
- Overloading and type-classes (13.1-13.8)
- Algebraic types (14.1-14.6)
- Case study: Huffman codes (15.1-15.7)
- Abstract data types (16.1-16.9)
- Lazy evaluation (17.1-17.8)
- I/O programming and Monads (18.1-18.6)

Week 1

Mon
- Reading: 1.1-1.11. Optional: 1.12-1.14
- Setting up[1]
- Introduction to Haskell and Fuctional Programming[2]

Wed
- Reading: 2.1-2.7

[1] notes/setup.html
[2] notes/intro.html

- Commands for GHCi interactive mode[3]
- Practice with GHCi[4]

Fri
- Reading: 3.1-3.7
- Standard Haskell values and types.[5]
- Conditionals. Guarded Expressions.[6]
- Assignment 0. Due Wed 01/15[7]

Week 2

Mon
- Reading: 4.1-4.3, 4.8, 5.1-5.3
- Compound Types[8]
- Type Aliases and Custom Types.[9]

Wed
- Reading: 5.4-5.7
- Working with the GHC compiler and interpreter. Lists.[10]
- List Comprehensions.[11]
- Assignment 1. Due Mon 01/20[12]

Fri
- Catchup/Practice

Week 3

Mon
- List comprehension practice: Book Library[13]

Wed
- Reading: 6.1-6.3, 6.7
- Parametric polymorphism[14]
- The supermarket billing example[15]
- Assignment 2. Due Wed 1/29[16]

Fri
- Reading: 7.1-7.4
- Pattern Matching.[17]

[3] notes/ghci_commands.html
[4] notes/ghci_practice.html
[5] notes/standard.html
[6] notes/functions_conditionals.html
[7] assignments/assignment0.html
[8] notes/compoundTypes.html
[9] notes/types_custom.html
[10] notes/lists.html
[11] notes/list_comprehensions.html
[12] assignments/assignment1.html
[13] notes/list_comp_practice.html
[14] notes/parametric_polymorphism.html
[15] notes/supermarket_billing.html
[16] assignments/assignment2.html
[17] notes/pattern_matching.html

Week 4

Mon
- Reading: 7.5-7.6
- More practice with Pattern Matching.[18]

Wed
- Reading: 8.2-8.5
- Interactive Programming[19]

Fri
- Practice with Interactive Programming[20]
- Assignment 3. Due Fri 2/7[21]

Week 5

Mon
- Reading: 10.1-10.2
- Higher-order functions[22]

Wed
- Reading: 10.3-10.5
- Folding[23]

Fri
- Reading: 11.2-11.3
- Types of recursion[24]
- Anonymous Functions. Sections.[25]

Week 6

Mon
- Reading: 11.1, 11.4
- Currying and partial application[26]
- Function Composition[27]

Wed TBD

Fri TBD

[18] notes/more_pattern_matching.html
[19] notes/interactive.html
[20] notes/interactive_hangman.html
[21] assignments/assignment3.html
[22] notes/more_pattern_matching_functions.html
[23] notes/folding.html
[24] notes/recursion.html
[25] notes/anonymous_functions.html
[26] notes/currying.html
[27] notes/function_composition.html

Week 7

Mon TBD

Wed TBD

Fri TBD

Week 8

Mon TBD

Wed TBD

Fri TBD

Week 9

Mon TBD

Wed TBD

Fri TBD

Week 10

Mon TBD

Wed TBD

Fri TBD

Week 11

Mon TBD

Wed TBD

Fri TBD

Week 12

Mon TBD

Wed TBD

Fri TBD

Week 13

Mon TBD

Wed TBD

Fri TBD

- Folding trees[28]
- Reading: 14.1-14.3 Recursive Types[29]

Old links

- More advanced typing: Curried Functions. Polymorphism, Type classes.[30] (3.6-3.9)
- Version Control[31]
- The Maybe (Option) Type.[32]
- Functions as Values: Difference Lists, Composition[33] (7.5)
- Functions as Values: Difference Lists, Composition (cont)[34] (7.5)
- MIDTERM (study guide[35])
- BREAK
- Assignment 4. Due 11/03[36]
- Overview of Software Development Practices[37]
- Information hiding and abstraction with modules[38]
- Testing[39]
- The State Monad[40]

[28] notes/foldingTrees.html
[29] notes/recursive_types.html
[30] notes/types_advanced.html
[31] notes/version_control.html
[32] notes/maybe.html
[33] notes/difference_lists.html
[34] notes/difference_lists.html
[35] notes/midterm_study_guide.html
[36] assignments/assignment4.html
[37] notes/dev_overview.html
[38] notes/modules.html
[39] notes/testing.html
[40] notes/functors_monads.html

- Functors, Applicatives, Monads[41]
- Final Study Guide[42]

[41] notes/functors_monads.html
[42] notes/final_study_guide.html