

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 up¹
 - Introduction to Haskell and Fuctional Programming²

- Wed**
- Reading: 2.1-2.7

¹[notes/setup.html](https://haskell-lang.org/docs/setup)

²[notes/intro.html](https://haskell-lang.org/docs/intro)

- Commands for GHCi interactive mode³
- Practice with GHCi⁴

- Fri**
- Reading: 3.1-3.7
 - Standard Haskell values and types.⁵
 - Conditionals. Guarded Expressions.⁶
 - Assignment 0. Due Wed 01/15⁷

Week 2

- Mon**
- Reading: 4.1-4.3, 4.8, 5.1-5.3
 - Compound Types⁸
 - 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 01/20¹²

- Fri**
- Catchup/Practice

Week 3

- Mon**
- List comprehension practice: Book Library¹³

- Wed**
- Reading: 6.1-6.3, 6.7
 - Parametric polymorphism¹⁴
 - The supermarket billing example¹⁵
 - Assignment 2. Due Wed 1/29¹⁶

- 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 Programming¹⁹
- Fri**
- Practice with Interactive Programming²⁰
 - Assignment 3. Due Fri 2/7²¹

Week 5

- Mon**
- Reading: 10.1-10.2
 - Higher-order functions²²
- Wed**
- Reading: 10.3-10.5
 - Folding²³
- Fri**
- Reading: 11.2-11.3
 - Types of recursion²⁴
 - Anonymous Functions. Sections.²⁵

Week 6

- Mon**
- Reading: 11.1, 11.4
 - Currying and partial application²⁶
 - Function Composition²⁷
- Wed**
- Higher order functions practice: 12.5²⁸
- Fri** Random number generation in Haskell²⁹

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](#)

²⁸[notes/higher_order_functions_practice.html](#)

²⁹[notes/random_numbers.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 trees³⁰
- Reading: 14.1-14.3 Recursive Types³¹

Old links

- More advanced typing: Curried Functions. Polymorphism, Type classes.³² (3.6-3.9)
- Version Control³³
- The Maybe (Option) Type.³⁴
- Functions as Values: Difference Lists, Composition³⁵ (7.5)
- Functions as Values: Difference Lists, Composition (cont)³⁶ (7.5)
- MIDTERM (study guide³⁷)
- BREAK
- Assignment 4. Due 11/03³⁸
- Overview of Software Development Practices³⁹
- Information hiding and abstraction with modules⁴⁰
- Testing⁴¹
- The State Monad⁴²

³⁰[notes/foldingTrees.html](#)

³¹[notes/recursive_types.html](#)

³²[notes/types_advanced.html](#)

³³[notes/version_control.html](#)

³⁴[notes/maybe.html](#)

³⁵[notes/difference_lists.html](#)

³⁶[notes/difference_lists.html](#)

³⁷[notes/midterm_study_guide.html](#)

³⁸[assignments/assignment4.html](#)

³⁹[notes/dev_overview.html](#)

⁴⁰[notes/modules.html](#)

⁴¹[notes/testing.html](#)

⁴²[notes/functors_monads.html](#)

- Functors, Applicatives, Monads⁴³
- Final Study Guide⁴⁴

⁴³[notes/functors_monads.html](#)

⁴⁴[notes/final_study_guide.html](#)