

# 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<sup>1</sup>
  - Introduction to Haskell and Fuctional Programming<sup>2</sup>

- Wed**
- Reading: 2.1-2.7

---

<sup>1</sup>[notes/setup.html](#)

<sup>2</sup>[notes/intro.html](#)

- Commands for GHCi interactive mode<sup>3</sup>
- Practice with GHCi<sup>4</sup>

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

## Week 2

- Mon**
- Reading: 4.1-4.3, 4.8, 5.1-5.3
  - Compound Types<sup>8</sup>
  - Type Aliases and Custom Types.<sup>9</sup>

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

- Fri**
- Catchup/Practice

## Week 3

- Mon**
- List comprehension practice: Book Library<sup>13</sup>

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

- Fri**
- Reading: 7.1-7.4
  - Pattern Matching.<sup>17</sup>

---

<sup>3</sup>[notes/ghci\\_commands.html](#)

<sup>4</sup>[notes/ghci\\_practice.html](#)

<sup>5</sup>[notes/standard.html](#)

<sup>6</sup>[notes/functions\\_conditionals.html](#)

<sup>7</sup>[assignments/assignment0.html](#)

<sup>8</sup>[notes/compoundTypes.html](#)

<sup>9</sup>[notes/types\\_custom.html](#)

<sup>10</sup>[notes/lists.html](#)

<sup>11</sup>[notes/list\\_comprehensions.html](#)

<sup>12</sup>[assignments/assignment1.html](#)

<sup>13</sup>[notes/list\\_comp\\_practice.html](#)

<sup>14</sup>[notes/parametric\\_polymorphism.html](#)

<sup>15</sup>[notes/supermarket\\_billing.html](#)

<sup>16</sup>[assignments/assignment2.html](#)

<sup>17</sup>[notes/pattern\\_matching.html](#)

## Week 4

- Mon**
- Reading: 7.5-7.6
  - More practice with Pattern Matching.<sup>18</sup>
- Wed**
- Reading: 8.2-8.5
  - Interactive Programming<sup>19</sup>
- Fri**
- Practice with Interactive Programming<sup>20</sup>
  - Assignment 3. Due Fri 2/7<sup>21</sup>

## Week 5

- Mon**
- Reading: 10.1-10.2
  - Higher-order functions<sup>22</sup>
- Wed**
- Reading: 10.3-10.5
  - Folding<sup>23</sup>
- Fri**
- Reading: 11.2-11.3
  - Types of recursion<sup>24</sup>
  - Anonymous Functions. Sections.<sup>25</sup>

## Week 6

- Mon**
- Reading: 11.1, 11.4
  - Currying and partial application<sup>26</sup>
  - Function Composition<sup>27</sup>
- Wed**
- Higher order functions practice: 12.5<sup>28</sup>
  - Assignment 4. Due Fri 3/6<sup>29</sup>
- Fri**
- Random number generation in Haskell<sup>30</sup>

---

<sup>18</sup>[notes/more\\_pattern\\_matching.html](#)

<sup>19</sup>[notes/interactive.html](#)

<sup>20</sup>[notes/interactive\\_hangman.html](#)

<sup>21</sup>[assignments/assignment3.html](#)

<sup>22</sup>[notes/more\\_pattern\\_matching\\_functions.html](#)

<sup>23</sup>[notes/folding.html](#)

<sup>24</sup>[notes/recursion.html](#)

<sup>25</sup>[notes/anonymous\\_functions.html](#)

<sup>26</sup>[notes/currying.html](#)

<sup>27</sup>[notes/function\\_composition.html](#)

<sup>28</sup>[notes/higher\\_order\\_functions\\_practice.html](#)

<sup>29</sup>[assignments/assignment4.html](#)

<sup>30</sup>[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 trees<sup>31</sup>
- Reading: 14.1-14.3 Recursive Types<sup>32</sup>

## Old links

- More advanced typing: Curried Functions. Polymorphism, Type classes.<sup>33</sup> (3.6-3.9)
- Version Control<sup>34</sup>
- The Maybe (Option) Type.<sup>35</sup>
- Functions as Values: Difference Lists, Composition<sup>36</sup> (7.5)
- Functions as Values: Difference Lists, Composition (cont)<sup>37</sup> (7.5)
- MIDTERM (study guide<sup>38</sup>)
- BREAK
- Assignment 4. Due 11/03<sup>39</sup>
- Overview of Software Development Practices<sup>40</sup>
- Information hiding and abstraction with modules<sup>41</sup>
- Testing<sup>42</sup>
- The State Monad<sup>43</sup>

---

<sup>31</sup>[notes/foldingTrees.html](#)

<sup>32</sup>[notes/recursive\\_types.html](#)

<sup>33</sup>[notes/types\\_advanced.html](#)

<sup>34</sup>[notes/version\\_control.html](#)

<sup>35</sup>[notes/maybe.html](#)

<sup>36</sup>[notes/difference\\_lists.html](#)

<sup>37</sup>[notes/difference\\_lists.html](#)

<sup>38</sup>[notes/midterm\\_study\\_guide.html](#)

<sup>39</sup>[assignments/assignment4.html](#)

<sup>40</sup>[notes/dev\\_overview.html](#)

<sup>41</sup>[notes/modules.html](#)

<sup>42</sup>[notes/testing.html](#)

<sup>43</sup>[notes/functors\\_monads.html](#)

- Functors, Applicatives, Monads<sup>44</sup>
- Final Study Guide<sup>45</sup>

---

<sup>44</sup>[notes/functors\\_monads.html](#)

<sup>45</sup>[notes/final\\_study\\_guide.html](#)