

DUBLIN HASKELL BOOK CLUB

October 2021

BUILDING UP OUR AXIOMS (SORRY SUPER TEXT HEAVY)

- A lot of this will be from my personal experience.
- A lot of this especially early will be stuff you already know and might be boring
- The group will have different experiences so we have to find the right balance of making it interesting to old hands and possible to new members
- Might make sense to have two session a month, one general session and an additional tutorial for people struggling.
- I'll start making supporting material like summary videos for people that start late or miss weeks.

REASONS FOR STARTING BOOK CLUB

- Excuse to use and talk about haskell.
- Fun way to build community.
- “Peer pressure” effect to help keep motivation and work through large text books.
- It’s easier and more fun to study topics in a group
- Opportunity to share experiences and learn from others.

SOME EXPECTATIONS(FROM MY PERSPECTIVE)

- Improve my Haskell knowledge.
- Introduction to new ideas.
- Meet new people who share a love of programming.
- Maybe become better at explaining concepts from haskell to others
- Build up the confidence to use haskell more

HOPES FOR THE GROUP

- Build a community of dedicated haskellers in Dublin.
- Get some networking effects where we all improve our haskell skills.
- Id absolutely love to start some open source projects in haskell from this group.(Open to any and all ideas)
- A support group where we can share and get excited about ideas and concepts in programming.

LOGISTICS

- One meeting a month, generally late in the month and late in the week. These will be virtual for now, but later we can also do a meetup somewhere in Dublin with the option of signing in virtually.
- Github repo with the solutions to exercises, these slide decks and any other supporting media, feel free to make pull requests, also I'll probably move it to a dedicated Dublin haskell org account.
- Generally around three chapters a month(depending on chapters) trying to average between 50 to 100 pages of content. Which means it will take at least 6 months to get through Real World Haskell.
- I'll try to stay a few chapters ahead and post material ahead of time, also will post summary videos after each session.
- Would be great if people could volunteer to handle the meetups if i cant for any reason(this should be super rare).
- Also would be really cool if others thought of ideas and hosted separate meetups, eg code alongs, open source projects, speaker series etc.
- Will be using Discord.

WHY REAL WORLD HASKELL

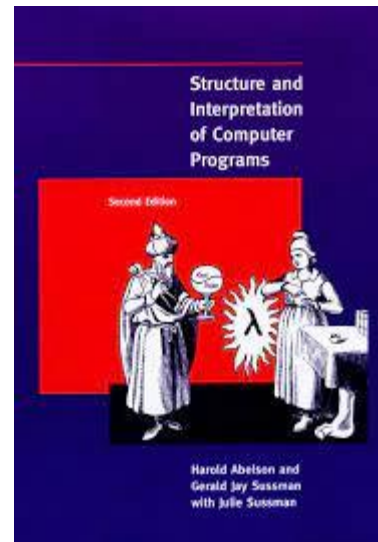
- It available free.
- It's a classic.
- Personally I've never finished it.
- Comprehensive coverage, and nice balance between being beginner friendly and not shying away from complexity.
- Some really cool example projects (e.g JSON parser)
- Been around for a while so errata have been addressed.
- One of the Authors is from Dublin.

ONE OF IREALND'S GREAT HASKELLERS



FUNCTIONAL BACKGROUND(FROM A C DEVELOPER)

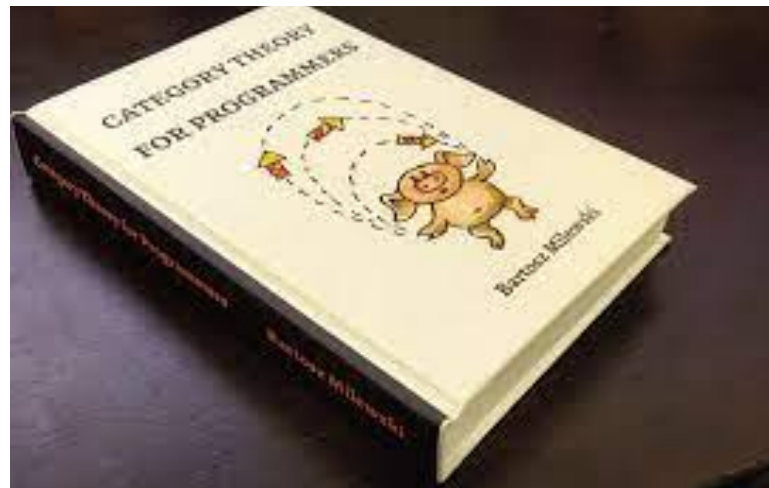
- Church and Turning
- The lambda Calculus
- Lisp's and C
- Stalmans lisp machines and Bell labs systems
- Scheme's and SICP
- Category Theory
- OOP obsession
- Functionals return to the mainstream
- Java, C++, Pythons “borrowing”
- The hybrids Scala, Closure and Rust.



MY INTRODUCTION CAME FROM A C++ CONFERENCE



PURE MATHEMATICS VS APPLIED II FUNCTIONAL VS OBJECT



SOME REASONS FOR MY PERSONAL INTEREST IN HASKELL

- Reduced lines of code, means less bugs and more readability and easier to maintain.
- Different way of thinking make my generally coding ability better
- Power of abstraction
- Use for writing parsers and compilers.
- Ambition to create my own language and rewrite Minix.

CHAPTER ONE - GETTING STARTED

- Getting Haskell setup
- GHCi
- Calculator
- Lists
- Strings and Chars
- Types
- A Simple Program

CHAPTER ONE PROBLEMS + SOLUTION

- Question 4 and 5 page 16
- WC.hs

```
1 main = interact wordCount
2   where wordCount input = show (length input) ++ "\n"
3   -- where wordCount input = show(length (lines input)) ++ "\n"
4
5
6   -- wrote this with vim, using tab causes compile error, I think this can be fixed in the vim setting, but need to investigate
7   --runghc WC < text.txt
```

CHAPTER TWO TYPES AND FUNCTIONS

- Why types
- Haskell Type System
- Basic Types
- Functional Application
- Composite Data Types
- Functions Lists and Tuples
- Types and Purity
- Simple Functions
- Evaluation by example
- Polymorphism
- Purity

```
1
2  myDrop :: Int -> [a] -> [a]
3  myDrop n xs = if n <= 0 || null xs
4                then xs
5                else myDrop (n-1)(tail xs)
```

LASTBUTONE.HS