Notes on
Functional Programming in Scala
Chapter Index

H. Conrad Cunningham

20 April 2022

Contents

Professor of Computer and Information Science
University of Mississippi
214 Weir Hall
P.O. Box 1848
University, MS 38677
(662) 915-7396 (dept. office)

Browser Advisory: The HTML version of this textbook requires a browser that supports the display of MathML. A good choice as of April 2022 is a recent version of Firefox from Mozilla.

Instability Warning: This version of the *Notes on Functional Programming in Scala* is in work beginning in January 2022. The author may change its structure and content without warning. No changes are planned to the 2018 version upon which the this version is based. The stable version is on the Fall 2018 CSci 450 course website.

Feedback Request: The author plans to publish this textbook eventually. He invites anyone using this book to give him feedback on its current structure and content: to point out typos and other errors or suggest improvements and extensions. He can be contacted at hcc AT cs DOT olemiss DOT edu.

Notes on Functional Programming in Scala

- **Integrated Chapters 3-5 of *Functional Programming in Scala* [2]:**
  - HTML
  - PDF

Programming Paradigms (ELIFP Ch. 2)

- Chapter:
  - HTML
  - PDF
- HTML Slides:
  - Programming Paradigms Scala Version
  - Programming Paradigms (regular ELIFP slides)

Object-Based Paradigms (ELIFP Ch. 3)

- Chapter:
  - HTML
  - PDF
- HTML Slides:
  - Object-Based Paradigms (regular ELIFP slides)

Scala for Java Programmers

- Chapter:
  - HTML
  - PDF
- HTML Slides: None yet

Type System Concepts

- Chapter:
  - HTML
  - PDF
- HTML Slides:

  - Type Concepts

Recursion Styles, Correctness, and Efficiency (Scala)

- Chapter:
  - HTML
  - PDF
- HTML Slides:
  - Recursion Styles, Correctness, and Efficiency (Scala)

Examples from SICP [1]

- Chapter:
  - HTML
  - PDF
- HTML Slides: None yet

This document also examines implementations in Elixir, Lua, Racket Scheme, Elm, and Python.

Chapter 3: Functional Data Structures

- Chapter:
  - HTML
  - PDF
- HTML Slides: None yet

Chapter 4: Handling Errors without Exceptions

- Chapter:
  - HTML
  - PDF
- HTML Slides: None yet

Chapter 5: Strictness and Laziness

- Chapter:
  - HTML
  - PDF
- HTML Slides: None yet

Sandwich DSL in Scala Project

- Chapter:
  - HTML
  - PDF
- HTML Slides: None yet

TODO: Mention and link to Haskell and Lua discssions.

Natural Number Arithmetic Examples

- Chapter:
  - HTML
  - PDF
- HTML Slides: None yet

This document also examines implementations in Elixir, Lua, Ruby, and Java.

CookieJar Abstract Data Type

TODO: Better order, integrate, and title the discussions of ADTs in Scala.

We discuss Scala immutable (i.e., functional) implementations of a CookieJar abstract data type (ADT) in this possible future chapter.

- Chapter:
  - as HTML
  - as PDF
- HTML Slides: None yet

This document also examines mutable Scala, Python, and Ruby implementations.

Abstract Data Types in Scala

TODO: Better order, integrate, and title the discussions of ADTs in Scala.

- Chapter:
  - HTML
  - PDF
- HTML Slides: None yet

TODO: Mention and link to Haskell and Elixir implementations.

Acknowledgements

In Spring 2016, I drafted this set of notes to accompany my lectures on functional programming in Scala using, in part, the book *Functional Programming in Scala* [2] (i.e., the Red Book) for my Scala-based, Spring 2016 version of CSci 555 (Functional Programming) at the University of Mississippi. I constructed the notes around the ideas, general structure, and Scala examples from the Red Book and its associated materials [3,4]. I also adapted some discussion and examples from my Haskell-based *Notes on Functional Programming with Haskell* [5], Odersky's *Scala by Example* [6], and my previous notes on teaching multiparadigm programming using Scala.

The Red Book [2] is an excellent book for self-study, but it needed expanded explanations to be useful to me as a textbook for students who were novices in both Scala and functional programming.

In Spring 2019, I again offered a Scala-based version of CSci 555 and continued to revise and extend these lecture notes.

I retired from the full-time faculty in May 2019. As one of my post-retirement projects, I am continuing work on the ELIFP textbook and other instructional materials. In January 2022, I began refining the ELIFP content and related documents such as this one. I am integrating separately developed materials better, reformatting the documents (e.g., using CSS), constructing a unified bibliography (e.g., using citeproc), and improving the build workflow and use of Pandoc. I adapted this index page from aportions of my Fall 2018 CSci 556 and Spring 2019 CSci 555 course notes pages.

I maintain these notes as text in Pandoc's dialect of Markdown using embedded LaTeX markup for the mathematical formulas and then translate the notes to HTML, PDF, and other forms as needed.

References

[1] Harold Abelson and Gerald Jockay Sussman. 1996. *Structure and interpretation of computer programs (SICP)* (Second ed.). MIT Press, Cambridge, Massachusetts, USA. Retrieved from https://mitpress.mit.edu/sicp/

[2] Paul Chiusano and Runar Bjarnason. 2015. *Functional programming in Scala* (First ed.). Manning, Shelter Island, New York, USA.

[3] Paul Chiusano and Runar Bjarnason. 2022. FP in Scala exercises, hints, and answers. Retrieved from https://github.com/fpinscala/fpinscala

[4] Paul Chiusano and Runar Bjarnason. 2022. FP in Scala community guide and chapter notes. Retrieved from https://github.com/fpinscala/fpinscala/wiki

[5] H. Conrad Cunningham. 2014. *Notes on functional programming with Haskell.* University of Mississippi, Department of Computer and Information Science, University, Mississippi, USA. Retrieved from https://john.cs.olemiss.edu/~hcc/docs/Notes_FP_Haskell/Notes_on_Functional_Programming_with_Haskell.pdf

[6] Martin Odersky. 2014. *Scala by example.* École Polytechnique Fédérale Lausanne (EPFL), Programming Methods Laboratory, Lausanne, Switzerland.