

Notes on
Abstraction and Modularity:
Chapter Index

H. Conrad Cunningham

13 July 2022

Contents

| | |
|--|----------|
| Notes on Abstraction and Modularity | 1 |
| Notes on Data Abstraction | 1 |
| Notes on Modular Design | 1 |
| ELIFP Chapter 2: Programming Paradigms | 2 |
| ELIFP Chapter 3: Object-Based Paradigms | 2 |
| ELIFP Chapter 6: Procedural Abstraction (in Haskell) | 2 |
| ELIFP Chapter 7: Data Abstraction (in Haskell) | 2 |
| ELIFP Chapter 22: Data Abstraction Revisited | 2 |
| Abstract Data Types in Scala | 3 |
| CookieJar Abstract Data Type | 3 |
| Acknowledgements | 3 |
| References | 3 |

Copyright (C) 1996-2022, H. Conrad Cunningham
Professor of Computer and Information Science
University of Mississippi
214 Weir
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 July 2022 is a recent version of Firefox from Mozilla.

Notes on Abstraction and Modularity

Notes on Data Abstraction

1. Notes on Data Abstraction Index
 - as HTML
 - as PDF

Notes on Modular Design

2. Notes on Modular Design
 - as HTML
 - as PDF

ELIFP Chapter 2: Programming Paradigms

This chapter discusses the primary imperative and declarative programming paradigms and various subparadigms. It presents the imperative examples in Java and Scala, the declarative/functional examples in Haskell and Scala, the declarative/relational examples in Prolog, the imperative/procedural examples in Python and Scala, and the imperative/modular examples in Python and Scala,

- **Chapter:** as HTML
as PDF
- **Slides:** (HTML) Programming Paradigm
(HTML) Programming Paradigms Scala Version

ELIFP Chapter 3: Object-Based Paradigms

This chapter discusses the object model and the object-based, object-oriented, and prototype-based programming paradigms. It presents the object-based and object-oriented examples in Python and Scala and the prototype-based example in Lua.

- **Chapter:** as HTML
as PDF
- **Slides:** (HTML) Object-Based Paradigms

ELIFP Chapter 6: Procedural Abstraction (in Haskell)

This chapter discusses procedural abstraction and modular design in Haskell.

- **Chapter:** as HTML
as PDF
- **Slides:** (HTML) Top-Down Stepwise Refinement
(HTML) Modular Design and Programming

- Other: (Haskell) Quick Overview of Basic Haskell

ELIFF Chapter 7: Data Abstraction (in Haskell)

This chapter discusses data abstraction and modular design in Haskell.

- **Chapter:** as HTML
as PDF
- Slides: (HTML) Using Data Abstraction

ELIFF Chapter 22: Data Abstraction Revisited

This chapter presents the **Digraph ADT case study in Haskell**.

- **Chapter:** as HTML
as PDF
- Slides: NONE YET

Abstract Data Types in Scala

This chapter presents the **Digraph ADT case study in Scala**. It is similar to ELLIFF chapter 22, but also includes some general concepts on data abstraction and modular design.

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

CookieJar Abstract Data Type

This possible future chapter presents Scala, Python, and Ruby implementations of a CookieJar ADT.

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

Acknowledgements

The oldest notes in this pile are the Notes on Data Abstraction that I wrote originally when I was learning and teaching object-oriented programming and the relatively new Java language in 1996. I wrote the other notes as my understanding evolved and I learned and taught other languages over the subsequent 23 years. See the acknowledgements sections for the various documents for more specific information.

I retired from the full-time faculty in May 2019. As one of my post-retirement projects, I am continuing work on possible textbooks based on the course materials I had developed during my three decades as a faculty member. In January 2022, I began refining the existing content, integrating separately developed materials together, reformatting the documents, constructing a unified bibliography (e.g., using citeproc), and improving my build workflow and use of Pandoc.

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

References