The book was found

The Little Schemer - 4th Edition
The notion that "thinking about computing is one of the most exciting things the human mind can do" sets both The Little Schemer (formerly known as The Little LISPer) and its new companion volume, The Seasoned Schemer, apart from other books on LISP. The authors’ enthusiasm for their subject is compelling as they present abstract concepts in a humorous and easy-to-grasp fashion. Together, these books will open new doors of thought to anyone who wants to find out what computing is really about. The Little Schemer introduces computing as an extension of arithmetic and algebra; things that everyone studies in grade school and high school. It introduces programs as recursive functions and briefly discusses the limits of what computers can do. The authors use the programming language Scheme, and interesting foods to illustrate these abstract ideas. The Seasoned Schemer informs the reader about additional dimensions of computing: functions as values, change of state, and exceptional cases. The Little LISPer has been a popular introduction to LISP for many years. It had appeared in French and Japanese. The Little Schemer and The Seasoned Schemer are worthy successors and will prove equally popular as textbooks for Scheme courses as well as companion texts for any complete introductory course in Computer Science.

**Book Information**

Series: MIT Press
Paperback: 216 pages
Publisher: The MIT Press; fourth edition edition (December 21, 1995)
Language: English
ISBN-10: 0262560992
Product Dimensions: 6.9 x 0.6 x 8.9 inches
Shipping Weight: 12 ounces (View shipping rates and policies)
Average Customer Review: 4.2 out of 5 starsÂ Â See all reviewsÂ (65 customer reviews)
Best Sellers Rank: #54,895 in Books (See Top 100 in Books)  
#2 inÂ Books > Computers & Technology > Programming > Languages & Tools > Lisp  
#246 inÂ Books > Textbooks > Computer Science > Programming Languages  
#367 inÂ Books > Computers & Technology > Software

**Customer Reviews**

In 1974, Daniel P. Friedman published a remarkable little book called The Little LISPer. It was only 68 pages, but it did a remarkable thing: It could teach you to think recursively. It used some pretend dialect of LISP (which was written in all caps in those days). The dialect didn’t fully conform to any
real LISP. But that was ok because it wasn't really about LISP, it was about recursive functions. You
didn't need a computer in order to work through the exercises. After reading the book, I was
changed. Or perhaps transformed. Or altered. In a good way. There are very few books that deeply
change the way that you think. This is one of those books. The format is a programmed text with
questions on the left side and answers on the right. The way you use it is to read a question, think
about the question, come up with an answer, and then compare your answer to Friedman's
answer. He used the names of foods as the symbols that are manipulated by your functions, and
little jokes were scattered around to pull you back when things get so deep that your head is going
to pop off. It even has a place reserved for JELLY STAINS! The book has been through several
revisions. The latest, The Little Schemer (Fourth Edition), updated by Matthias Felleisen, now
conforms more closely to a real programming language, Scheme, and has new chapters which
delve much deeper into recursive function theory and language processors. Felleisen is not as
comfortable with the programmed text format, so instead of questions and answers, he has a
deranged dialog going on which reads a little like SmĂ©agol and Gollum discussing fishes. The Little
Schemer is not a complete book on programming. It is weak in practical concerns like
documentation, defensive programming, and computational efficiency. The development of a
system of arithmetic from three primitives is delightful from a mathematical perspective and
shockingly horrible from an engineering perspective. It also will not teach you very much about
Scheme. It touches on only a very small part of the language: a very good part. Despite its flaws, the
book has a very loyal following and that is because it works. It teaches one thing, a thing that is very
difficult to teach, a thing that every profession programmer should know, and it does it really well.
These are lessons that stick with you. You need to grab a sandwich and study this book.

This is a wonderful book for people who enjoy having their minds stretched. It starts from the most
elementary concepts (this is a number, this is a symbol) and then proceeds to teach you how to
program in scheme (a lisp dialect) using a question-and-answer approach, with the questions on the
left hand side of the page and the answers on the right. Most of the teaching is by example; the
authors show you something several times in several different guises in order to get you to
understand the pattern underlying the programming examples. This form of
teaching-by-pattern-recognition is especially useful for scheme, because lisp-based languages
represent such a different paradigm from more conventional computer languages that it really helps
to have the pattern in mind when you want to write a new function. The authors show how the basic
elements of lisp (atoms, numbers and lists) can be used to solve an amazing variety of problems,
many of which would be much harder (or impossible) in more conventional computer languages. Most of the book is so easy that a complete novice who had never programmed before could understand it, but the authors sneakily keep increasing the complexity until in the last three chapters they cover continuation-passing style, the applicative-order Y combinator (!) and writing a scheme interpreter in scheme (!!). Some of these topics would go over the head of most computer science Ph.D.'s (go ahead, ask one what the Y combinator is -- I dare you!). This is not the book to read if you’re looking for a "teach yourself visual basic in 20 minutes" kind of book, but if you like programming and you enjoy having your mind stretched, you could not do better than this book (or its companion book, the Seasoned Schemer).

I am reading this book now, after wanting to get back into LISP programming after a ten year hiatus. My last memory of LISP was when I learned it in high school with an excellent teacher. I wish we had had this book back then! As it stands, I devoured the book in a few sittings -- its amusing, fast paced, rigorous and low-BS structure make it an excellent read. I'm of the last generation of students who were able to switch on a computer and get a BASIC prompt. The huge heft of "introductory" programming books today leaves me cold and uninspired -- I would hate to have seen these when I was first exploring the excitement of programming. The Little Schemer, by some of the old gurus of the (I believe) MIT Artificial Intelligence Lab (if not that particular lab, then at least those early, heady days in the '70s when AI wasn't a joke), reminded me of what it used to be like -- slowly building up a repertoire of commands and associated concepts that made programming seem a lot more like playing a Bach fugue and a lot less like debugging window objects. Things like recursion -- the essential part of this book -- are inherently wonderful. Were I teaching an advanced class for high school students, this book would be at the top of my list. Were I wanting to introduce a liberal arts student into the joys of mathematics, this book would be at the top. Were I wanting to deprogram a bad-habited CS student, this book. Indeed, with so many Universities wanting to stuff some kind of logical, syntactical reasoning requirements into their required courses, this book should be a best seller. It is a book that recaptures the joys and frustrations of programming and goes a long way to explaining why so many of the brightest people of the 20th century, at some point or another, sat down and cons’ed up a list.

Download to continue reading...
