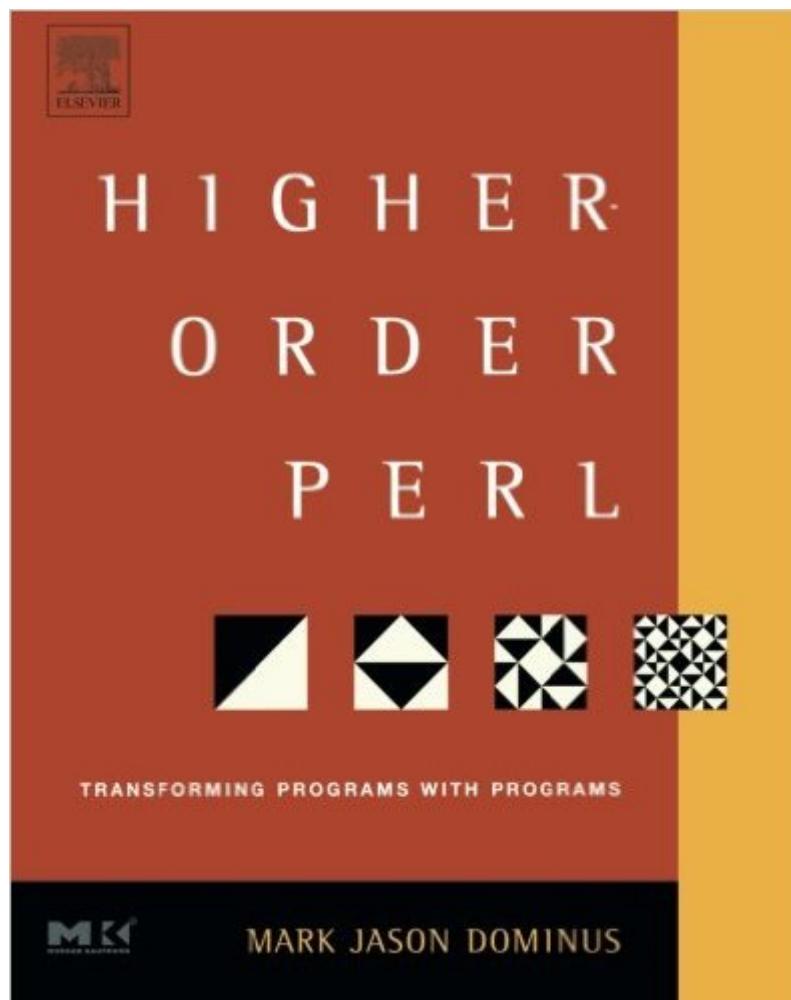


The book was found

# Higher-Order Perl: Transforming Programs With Programs



## Synopsis

Most Perl programmers were originally trained as C and Unix programmers, so the Perl programs that they write bear a strong resemblance to C programs. However, Perl incorporates many features that have their roots in other languages such as Lisp. These advanced features are not well understood and are rarely used by most Perl programmers, but they are very powerful. They can automate tasks in everyday programming that are difficult to solve in any other way. One of the most powerful of these techniques is writing functions that manufacture or modify other functions. For example, instead of writing ten similar functions, a programmer can write a general pattern or framework that can then create the functions as needed according to the pattern. For several years Mark Jason Dominus has worked to apply functional programming techniques to Perl. Now Mark brings these flexible programming methods that he has successfully taught in numerous tutorials and training sessions to a wider audience. \* Introduces powerful programming methods new to most Perl programmers that were previously the domain of computer scientists\* Gradually builds up confidence by describing techniques of progressive sophistication\* Shows how to improve everyday programs and includes numerous engaging code examples to illustrate the methods

## Book Information

Paperback: 600 pages

Publisher: Morgan Kaufmann; 1 edition (March 28, 2005)

Language: English

ISBN-10: 1558607013

ISBN-13: 978-1558607019

Product Dimensions: 7.4 x 1.4 x 9.4 inches

Shipping Weight: 2.8 pounds (View shipping rates and policies)

Average Customer Review: 4.7 out of 5 starsÂ See all reviewsÂ (18 customer reviews)

Best Sellers Rank: #87,476 in Books (See Top 100 in Books) #6 in Books > Computers & Technology > Programming > Languages & Tools > Perl #106 in Books > Textbooks > Computer Science > Software Design & Engineering #144 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design

## Customer Reviews

In "Higher Order Perl" (or HOP as it's affectionately called in the Perl community), the renown Perl wizard Mark Jason Dominus (MJD) shows how to take Perl coding to the next level by applying advanced programming techniques from the domain of functional programming. The book covers

recursion (including methods to convert recursive code to iterative code), iterators, streams, memoization, currying, parsing, constraint programming and higher order functions (functions that take functions as arguments and/or return other functions). It is packed with great, sophisticated code which is explained very well and is a model for correct programming. The author takes an approach similar to Peter Norvig's PAIP - advanced coding techniques are presented, and then non-trivial programs are written to demonstrate these concepts. The comparison with Lisp here is unavoidable, and MJD talks about Lisp in his preface. He claims that Perl shares 6 of the "7 features unique to Lisp" quoted from Norvig's PAIP, and that this basically means that most of what can be written in Lisp can be written in Perl in roughly the same manner. But as he himself admits in a later interview, the 7th "missing feature" of Lisp, namely its uniform syntax, is what \*really\* differentiates Lisp from the rest. Lisp's syntax allows a very clean handling of higher-order functions, list-processing, and most importantly macros. The contrast between MJD's own code in HOP and Norvig's PAIP code is the best example for this fundamental difference. Be MJD's code as clean and nice as it is (for Perl, anyway), it is nowhere near the sheer aesthetic appeal of Norvig's Lisp. Still, Lisp is Lisp and Perl is Perl, and each has its respectable place in the world of programming.

I had originally planned to read Higher Order Perl and write my review immediately. When it became clear that I wasn't going to read HOP straight through, I figured I'd write a review when I was done. More and more, it looks like my reading of the last two chapters is being indefinitely delayed. I'm going to write what I think so far, while it's still more or less fresh in my mind. Higher Order Perl was originally going to be given some fairly bland name, like "Handbook of Advanced Perl Techniques." This would have been a spot-on (but uninteresting) title. HOP provides the reader with explanation and demonstration of techniques for problem-solving that are often overlooked. The examples are complex and detailed, but not byzantine, and they're built up slowly, piece by piece, so that each line of code's meaning and significance are made clear. The title "Higher Order Perl" refers to the book's most central technique, functional programming. While many programmers understand how to abstract a specific solution into a more general one, Dominus helps the reader learn to push the envelope, abstracting generic solutions into extremely generalized solutions that can be applied to seemingly-unrelated problems. This is frequently done by the construction of functions that build functions that build functions -- and so on, functions all the way down. Instead of solving the problem in base, earthly Perl, the programmer produces Perl elements of a higher order which, operating in harmony, become all things to all people. Well, I'm hyperbolizing, but I think it would be

hard for me to over-emphasize the value of techniques like closures, iterators, and currying. They are, in part, what make Lisp so powerful, and the marriage of Lisp's power and Perl's expressivity is a happy one.

[Download to continue reading...](#)

Higher-Order Perl: Transforming Programs with Programs Higher! Higher! Higher! (Leslie Patricelli board books) Order,Order,Order - Kids Coding book (Coding Palz - Computer programming for kids) Christmas Mail Order Angels #1: A collection of 6 historical romantic novellas of mail order brides (Mail Order Angels Collection) Perl: Crash Course - The Ultimate Beginner's Course to Learning Perl Programming in Under 12 Hours Effective Perl Programming: Ways to Write Better, More Idiomatic Perl (2nd Edition) (Effective Software Development Series) Mastering Perl/Tk: Graphical User Interfaces in Perl Perl Programming Success in a Day: Beginners Guide to Fast, Easy, and Efficient Learning of Perl Programming Programming the Perl DBI: Database programming with Perl Overcoming student retention issues in higher education online programs: A Delphi study. Perl One-Liners: 130 Programs That Get Things Done The Process of Creating Life: Nature of Order, Book 2: An Essay on the Art of Building and the Nature of the Universe (The Nature of Order)(Flexible) Robert's Rules of Order Newly Revised In Brief, 2nd edition (Roberts Rules of Order in Brief) Mail Order Bride: 16 Book Mega Mail Order Bride Box Set: (With A Never Before Released Surprise Book) Mail Order Bride: The Accidental FiancÃƒÂ©: A Clean & Wholesome Western Historical Romance (Mail Order Brides of Fort Condor Book 2) Mail Order Bride Box Set - Sun River Brides - 9 Mail Order Bride Stories Collection: Clean and Wholesome Historical Western Romance Box Set Bundle Mail Order Bride: More Than Anything: Clean and Wholesome Western Historical Romance (Mail Order Brides On The Run Book 5) Mail Order Bride: Emily and the Overbearing Mother-in-Law (Mail Order Brides and Mother-in-Laws Book 1) Mail Order Bride: Rain in Kansas: Sweet and Clean Western Historical Romance (A Kansas Mail Order Bride Story Book 11) MARKUS ZUSAK - SERIES READING ORDER (SERIES LIST) - IN ORDER: THE BOOK THIEF, I AM THE MESSENGER, WHEN DOGS CRY, THE UNDERDOG, GETTING THE GIRL & FIGHTING RUBEN WOLFE!

[Dmca](#)