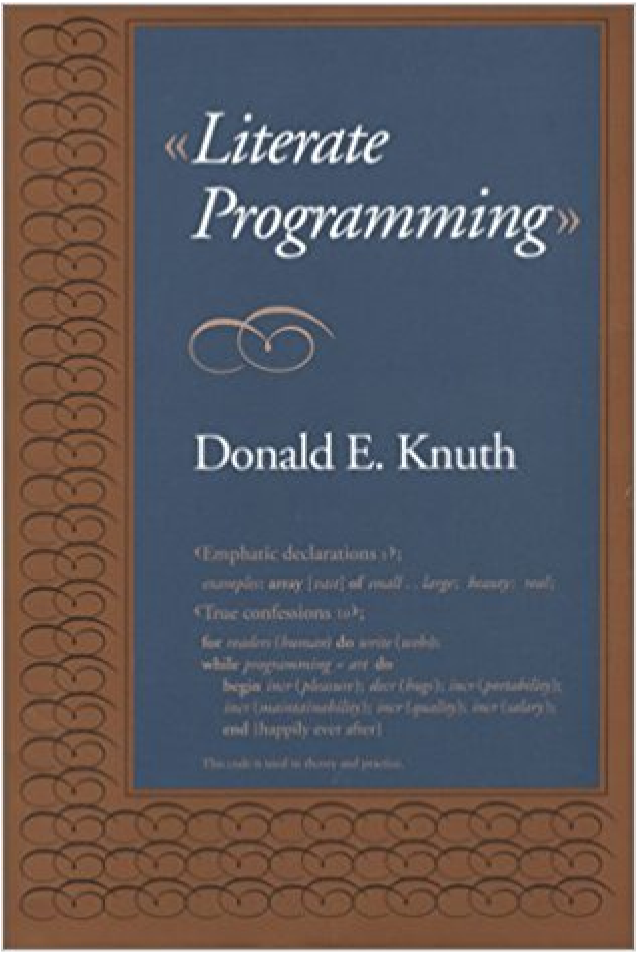
Literate Programming (Lecture Notes)


«Literate Programming»
Donald E. Knuth
⟨Emphatic declarations 1⟩;
examples: array [vast] of small . . large; beauty: real;
⟨True confessions 10⟩;
for readers (human) do write (webs);
while programming = art do
begin incr(pleasure); decr(bugs); incr(portability);
incr(maintainability); incr(quality); incr(salary);
end {happily ever after}
This code is used in theory and practice.

Synopsis

This anthology of essays from Donald Knuth, "the father of computer science," andÃ Â the inventor of literate programming includes early essays on related topics such as structured programming, as well as The Computer Journal article that launched literate programming itself. Many examples are given, including excerpts from the programs for TeX and METAFONT. The final essay is an example of CWEB, a system for literate programming in C and related languages.This volume is first in a series of Knuth's collected works.

Book Information

Series: Lecture Notes (Book 27)

Paperback: 384 pages

Publisher: Center for the Study of Language and Inf; 1 edition (June 1, 1992)

Language: English

ISBN-10: 0937073806

ISBN-13: 978-0937073803

Product Dimensions: 6 x 1.2 x 9 inches

Shipping Weight: 6.4 ounces (View shipping rates and policies)

Average Customer Review: 4.0 out of 5 stars 8 customer reviews

Best Sellers Rank: #457,879 in Books (See Top 100 in Books) #174 inÃ Â Books > Textbooks > Computer Science > Object-Oriented Software Design #608 inÃ Â Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design #1797 inÃ Â Books > Textbooks > Computer Science > Programming Languages

Customer Reviews

Even though technology has advanced considerably, the principles are still valuable. I would strongly recommend this book. I believe anybody from a beginner programmer to advanced software engineer could benefit from this book.

This is, indeed, a great book. I had read several reviews and decided to pick this one as my into to Literate Programming. I was not dissatisfied. This is a very good book and I love reading Knuth's works. I have all the volumes of his The Art of Programming (the original 3 volumes I got just before I graduated from Graduate School and have, in the intervening 35 years, found them to be a steady, reliable and wonderful reference...a great source of information. Knuth is a very accessible, readable author. This book on Literate Programming (a series of monographs by the author presenting) follows in the tradition and do not disappoint the reader who enjoys Knuth.One of the things I most like about ordering through is their teamwork with a wonderful group of 3rd party suppliers. I have not been dissatisfied with any that I have worked through and this one was no excpetion. They supplier was quick and thorough in processing the order and, in my experience, live up to the fine standards that I have always had with and their partners. I am most pleased with the service and ease of ordring.

This book is excellent. It was written by one of the pioneers of the computing field. It is the definitive work on Literate Programming. Programmers should really document their code more, but having the code and documentation as one document is pretty extreme - to some. But that is exactly what D. Knuth proposed in this classic book.

This book is the only one that I can say has truly changed my view of software development.The premise of this book matches my experience: technical communication with people is critical, and harder than communicating with the machines. Knuth carries that idea forward by one bold, logical step: in Literate Programming (LP), the main goal is to get technical ideas across to people. Programs are a co-product of the description process. This inverts the premise of JavaDoc and the like, in which human communication is incidental to the code.A literate program, by the way, reads like a standard human document, whether an essay or an IEEE standard specification. JavaDoc output reads like an HTML dump of a cross-linked tree data structure - which it is. JavaDoc serves a valuable purpose, but does not permit system description in the order required by human reasoning.My own experience with LP (a custom system) was very happy - I actually reached the "impossible" goal of true requirements traceability. I unified the system requirements, design, multi-language implementation, configuration control, and even tests under one document set. With HTML output, traceability was made real using interactive links. Anywhere else, traceability is mostly wishful thinking shared by the many owners of physically disconnected documents. (Process gurus - I hope you’re paying attention.)LP practice, however, has not caught on. LP, in today’s form, does not support programming in the large. What LP does to the compilable form of a program brings C++ name-mangling to mind. I don’t know of any WYSIWYG LP systems, so today’s window-icon-mouse-pointer (WIMP) programmers will have nothing to do with it. And, ironically, the people who need the most support in communicating with their peers are the ones most resistant to tools for effective communication.It’s a grand vision and an exciting experiment. LP deserves more attention.

This book is a collection of articles Prof. Knuth wrote about programming. He promoted a particular programming methodology called "literate programming", which weaves comments into codes and make them more readable and easier to maintain. This book was published in 1992, but Chapter 4, "Literate Programming", was originally published in 1984, which was an idea way ahead of his time (JavaDoc was first released in 1998, 12 years after the Knuth’s article). Chapter one is Knuth’s Turing Award lecture and still worth reading for his view on why programming is an art. I was wrongly impressed that Knuth is a very theoretical people and doesn’t do much programming. As you would discover from these lecture and other articles in the book, he indeed did a lot of programming and arguably in a very clever and beautiful way, "the program of which I personally am most pleases and proud is a compiler I once wrote for a primitive minicomputer that had only 4096 words of memory, 16 bites per word (pg. 10)." The discussion about the "goto" statement in Chapter 3 is not relevant in today’s programming and computer environment. The last few chapters are more like manuals of the WEB and CWEB programs (C version of WEB), which are the programs generating documents and source codes. These manuals may not interest readers unless they are well motivated to write program "literally." One gem should not be missed is is Chapter 10, "The Errors of TeX" (and the accompanying Chapter 11, "The Error Log of TeX). Seeing how Prof. Knuth meticulously documented all of his bugs in TeX is just amazing. Overall this book is more of historical value and for people who love Knuth and his work on literate programming.

Download to continue reading...

Literate Programming (Lecture Notes) Python Programming: Python Programming for Beginners, Python Programming for Intermediates, Python Programming for Advanced C++: The Ultimate Crash Course to Learning the Basics of C++ (C programming, C++ in easy steps, C++ programming, Start coding today) (CSS,C Programming, ... Programming,PHP, Coding, Java Book 1) C++ and Python Programming: 2 Manuscript Bundle: Introductory Beginners Guide to Learn C++ Programming and Python Programming C++ and Python Programming 2 Bundle Manuscript. Introductory Beginners Guide to Learn C++ Programming and Python Programming Python Programming: The Complete Step By Step Guide to Master Python Programming and Start Coding Today! (Computer Programming Book 4) Lecture Ready Student Book 2, Second Edition (Lecture Ready Second Edition 2) English Majors: A Comedy Collection for the Highly Literate (Prairie Home Companion (Audio)) The Information-Literate Historian The Highly Selective Dictionary of Golden Adjectives: For the Extraordinarily Literate (Highly Selective Reference) Teaching Information Literacy Reframed: 50+ Framework-Based Exercises for Creating Information-Literate Learners

USMLE Step 1 Lecture Notes 2017: 7-Book Set (Kaplan Test Prep) USMLE Step 1 Lecture Notes 2017: Pharmacology (USMLE Prep) USMLE Step 3 Lecture Notes 2017-2018: 2-Book Set (USMLE Prep) Ultracold Gases and Quantum Information: Lecture Notes of the Les Houches Summer School in Singapore: Volume 91, July 2009 Dynamic Response of Infrastructure to Environmentally Induced Loads: Analysis, Measurements, Testing, and Design (Lecture Notes in Civil Engineering) Statistical Methods for Data Analysis in Particle Physics (Lecture Notes in Physics) Biological Wastewater Treatment, Second Edition, Revised and Expanded (Lecture Notes in Pure and Applied Mathematics) Telescopes and Techniques (Undergraduate Lecture Notes in Physics) The Measurement of Biological Shape and Shape Change (Lecture Notes in Biomathematics, Volume 24)