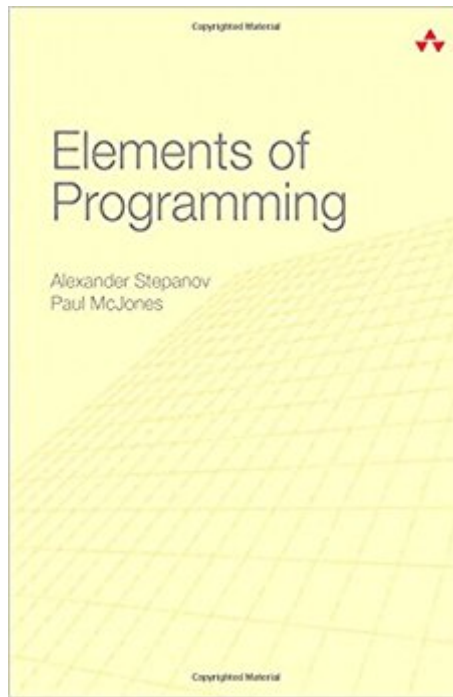The book was found

Elements Of Programming

Synopsis

â œAsk a mechanical, structural, or electrical engineer how far they would get without a heavy reliance on a firm mathematical foundation, and they will tell you, â ˆnot far.â ™ Yet so-called software engineers often practice their art with little or no idea of the mathematical underpinnings of what they are doing. And then we wonder why software is notorious for being delivered late and full of bugs, while other engineers routinely deliver finished bridges, automobiles, electrical appliances, etc., on time and with only minor defects. This book sets out to redress this imbalance. Members of my advanced development team at Adobe who took the course based on the same material all benefited greatly from the time invested. It may appear as a highly technical text intended only for computer scientists, but it should be required reading for all practicing software engineers.â •Â Â Â â ”Martin Newell, Adobe Fellow Â â œThe book contains some of the most beautiful code I have ever seen.â •Â Â Â â ”Bjarne Stroustrup, Designer of C++ â œI am happy to see the content of Alexâ ™s course, the development and teaching of which I strongly supported as the CTO of Silicon Graphics, now available to all programmers in this elegant little book.â •Â Â Â â ”Forest Baskett, General Partner, New Enterprise Associates â œPaulâ ™s patience and architectural experience helped to organize Alexâ ™sÂ mathematical approach into a tightly-structured edificeâ ”an impressive feat!â •Â Â Â â ”Robert W. Taylor, Founder of Xerox PARC CSL and DEC Systems Research Center Elements of Programming provides a different understanding of programming than is presented elsewhere. Its major premise is that practical programming, like other areas of science and engineering,must be based on a solid mathematical foundation. The book shows that algorithms implemented in a real programming language, such as C++, can operate in the most general mathematical setting. For example, the fast exponentiationÂ algorithm is defined to work with any associative operation. Using abstract algorithms leads to efficient, reliable, secure, and economical software. This is not an easy book. Nor is it a compilation of tips and tricks for incremental improvements in your programming skills. The bookâ ™s value is more fundamental and, ultimately, more critical for insight into programming. To benefit fully, you will need to work through it from beginning to end, reading the code, proving the lemmas, and doing the exercises. When finished, you will see how the application of the deductive method to your programs assures that your systemâ ™s software components will work together and behave as they must. The book presents a number of algorithms and requirements for types on which they are defined. The code for these descriptionsâ ”also available on the Webâ ”is written in a small subset of C++ meant to be accessible to any experienced programmer. This subset is defined in a special language appendix coauthored by Sean Parent and Bjarne Stroustrup. Whether you are a software developer, or any

other professional for whom programming is an important activity, or a committed student, you will come to understand what the bookâ ™s experienced authors have been teaching and demonstrating for yearsâ ”that mathematics is good for programming, and that theory is good for practice.

Book Information

Hardcover: 288 pages

Publisher: Addison-Wesley Professional; 1 edition (June 19, 2009)

Language: English

ISBN-10: 032163537X

ISBN-13: 978-0321635372

Product Dimensions: 6.5 x 1.1 x 9.2 inches

Shipping Weight: 1.3 pounds (View shipping rates and policies)

Average Customer Review: 4.0 out of 5 starsÂ Â See all reviewsÂ (25 customer reviews)

Best Sellers Rank: #186,534 in Books (See Top 100 in Books)   #130 inÂ Books > Computers & Technology > Programming > Algorithms   #135 inÂ Books > Computers & Technology > Programming > Languages & Tools > C & C++ > C++   #741 inÂ Books > Textbooks > Computer Science > Programming Languages

Customer Reviews

"I believe that iterator theories are as central to Computer Science as theories of rings or Banach spaces are central to Mathematics. Every time I would look at an algorithm I would try to find a structure on which it is defined. So what I wanted to do was to describe algorithms generically. That's what I like to do. I can spend a month working on a well known algorithm trying to find its generic representation. So far, I have been singularly unsuccessful in explaining to people that this is an important activity. But, somehow, the result of the activity - STL - became quite successful." -StepanovI had been waiting for this book for a while, as I greatly enjoy Stepanov's unorthodox views on programming. His flat rejection of the object-oriented paradigm was what caught my attention, but he differed from the unwashed newsgroup naysayers in an important respspect -- he offered an alternative. The fact that his alternative seemed to involve applying concepts from the realm of abstract algebra to computer programming made me realize I would be spending a lot of time and thought catching up.This is a short, but dense book. There is little trace of Knuth's sympathetic humor or Dijkstra's aesthetic passion. The material is presented as a series of definitions and sample programs, written in a programming language based on C++. Importantly,

there are also exercises and projects throughout each chapter. At first attempt, these puzzlers seem to contain as much insight as the prose itself.

What could be one of the most important books for developers of low-risksystems has come to my attention, and deserves your consideration if you areserious about understanding the mathematical foundations of programming andapplying them sensibly to your practice. It is not an easy read, but it isa very compelling approach. To support its mathematically orientedcrispness, the book includes the definition of a small but elegant C++subset that has been crafted by Sean Parent and Bjarne Stroustrup forillustrative use in the book. I believe this material should be taughtwithin all computer science curricula.A long quote and a short one on the back jacket give an idea of what isinvolved: Ask a mechanical, structural, or electrical engineer how far they would get without a heavy reliance on a firm mathematical foundation, and they will tell you, `not far.' Yet so-called software engineers often practice their art with little or no idea of the mathematical underpinnings of what they are doing. And then we wonder why software is notorious for being delivered late and full of bugs, while other engineers routinely deliver finished bridges, automobiles, electrical appliances, etc., on time and with only minor defects. This book sets out to redress this imbalance. Members of my advanced development team at Adobe who took the course based on the same material all benefited greatly from the time invested. It may appear as a highly technical text intended only for computer scientists, but it should be required reading for all practicing software engineers.

I was really looking forward to reading this book. Now that I'm done proving nearly all the lemmas the reader is asked work out, I have to admit to reading it with gritted teeth and what is best described as nearly constant annoyance. I will say that the subjects covered here are important and carefully chosen. The authors are undeniably passionate about the content, and they definitely know more about writing good programs than I will ever learn (particularly when said programs are written in C++). So what was it that rubbed me the wrong way?Basically, I could not help but feel like I was re-reading old and very good ideas from the 1970's about writing provably correct and reusable programs. The book often combines these with algebraic structures to produce efficient and reusable C++ code that is as general as possible. It does all this quite well, though it's a bit like being hit on your head with a sledgehammer at times. The code here is hardly beautiful, but on the whole it was a reminder of why the C++ Standard Template Library was such a brilliant effort. However, at the end of every chapter I felt like the focus on C++ implementations limited the discussion about what the universal "elements of programming" are supposed to be. In particular, I

would fault the book on not including any material on mathematically rigorous type inference systems that have been developed since the early 1980's, particularly because types are so central to what the book discusses, and since the book's errata online suggests the authors are dabbling with re-implementing all the code in Haskell.The bibliographic references also suggested a certain arrogance (I cannot imagine it is ignorance): there are no references to the extensive literature on the formal verification of software.

Programming #8:C Programming Success in a Day & Android Programming In a Day! (C Programming, C++programming, C++ programming language, Android , Android Programming, Android Games) Programming #57: C++ Programming Professional Made Easy & Android Programming in a Day (C++ Programming, C++ Language, C++for beginners, C++, Programming ... Programming, Android, C, C Programming) Programming #45: Python Programming Professional Made Easy & Android Programming In a Day! (Python Programming, Python Language, Python for beginners, ... Programming Languages, Android Programming) Programming: Computer Programming for Beginners: Learn the Basics of Java, SQL & C++ - 3. Edition (Coding, C Programming, Java Programming, SQL Programming, JavaScript, Python, PHP) Raspberry Pi 2: Raspberry Pi 2 Programming Made Easy (Raspberry Pi, Android Programming, Programming, Linux, Unix, C Programming, C+ Programming) Android: Programming in a Day! The Power Guide for Beginners In Android App Programming (Android, Android Programming, App Development, Android App Development, ... App Programming, Rails, Ruby Programming) DOS: Programming Success in a Day: Beginners guide to fast, easy and efficient learning of DOS programming (DOS, ADA, Programming, DOS Programming, ADA ... LINUX, RPG, ADA Programming, Android, JAVA) ASP.NET: Programming success in a day: Beginners guide to fast, easy and efficient learning of ASP.NET programming (ASP.NET, ASP.NET Programming, ASP.NET ... ADA, Web Programming, Programming) C#: Programming Success in a Day: Beginners guide to fast, easy and efficient learning of C# programming (C#, C# Programming, C++ Programming, C++, C, C Programming, C# Language, C# Guide, C# Coding) FORTRAN Programming success in a day:Beginners guide to fast, easy and efficient learning of FORTRAN programming (Fortran, Css, C++, C, C programming, ... Programming, MYSQL, SQL Programming) Prolog Programming; Success in a Day: Beginners Guide to Fast, Easy and Efficient Learning of Prolog Programming (Prolog, Prolog Programming, Prolog Logic, ... Programming, Programming Code, Java) R Programming: Learn R Programming In A DAY! - The Ultimate Crash Course to Learning the Basics of R Programming Language In No Time (R, R Programming, ... Course, R Programming Development Book 1) Parallel Programming:

Success in a Day: Beginners' Guide to Fast, Easy, and Efficient Learning of Parallel Programming (Parallel Programming, Programming, ... C++ Programming, Multiprocessor, MPI) MYSQL Programming Professional Made Easy 2nd Edition: Expert MYSQL Programming Language Success in a Day for any Computer User! (MYSQL, Android programming, ... JavaScript, Programming, Computer Software) Programming Raspberry Pi 3: Getting Started With Python (Programming Raspberry Pi 3, Raspberry Pi 3 User Guide, Python Programming, Raspberry Pi 3 with Python Programming) VBScript: Programming Success in a Day: Beginner's Guide to Fast, Easy and Efficient Learning of VBScript Programming (VBScript, ADA, ASP.NET, C#, ADA ... ASP.NET Programming, Programming, C++, C) Apps: Mobile App Trends in 2015 (iOS, Xcode Programming, App Development, iOS App Development, App Programming, Swift, Without Coding) ((Android, Android ... App Programming, Rails, Ruby Programming)) XML Programming Success in a Day: Beginner's Guide to Fast, Easy, and Efficient Learning of XML Programming (XML, XML Programming, Programming, XML Guide, ... XSL, DTD's, Schemas, HTML5, JavaScript) The Encyclopedia of Crystals, Herbs, and New Age Elements: An A to Z Guide to New Age Elements and How to Use Them SQL: Beginner's Guide for Coding SQL (database programming, computer programming, how to program, sql for dummies, java, mysql, The Oracle, python, PHP, ... (HTML, Programming, Coding, CSS Book 7)