

The book was found

C++ Footprint And Performance Optimization (Sams Professional)



Synopsis

This text aims to supply programmers with the knowledge they need to write code for the increasing number of hand-held devices, wearable computers and intelligent appliances. It contains case studies and sample code and is written with experienced C and C++ programmers in mind.

Book Information

Series: Sams Professional

Paperback: 400 pages

Publisher: Pearson Education; 1st edition (September 20, 2000)

Language: English

ISBN-10: 0672319047

ISBN-13: 978-0672319044

Product Dimensions: 7.2 x 0.9 x 9.2 inches

Shipping Weight: 2.1 pounds

Average Customer Review: 3.4 out of 5 stars [See all reviews](#) (11 customer reviews)

Best Sellers Rank: #1,595,583 in Books (See Top 100 in Books) #50 in [Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Performance Optimization](#) #763 in [Books > Computers & Technology > Programming > Languages & Tools > C & C++ > C++](#) #1139 in [Books > Computers & Technology > Programming > Microsoft Programming > C & C++ Windows Programming](#)

Customer Reviews

The book is well written but there is not enough facts. Not enough meat for your money. The chapter on compilers and profilers is very weak. It talks about the Gnu C++ compiler and mentions MS Visual C++ in short. The Intel compiler is not mentioned even though it is probably the most optimizing compiler for Intel and AMD processors. VTune code analyzer and profiler is the best tool available for measuring code performance and reveal processor stalls. It is not mentioned. Performance of different datatypes is handled in 1/3 page and it concludes that integer computations are faster than floating point computations and should be used where ever possible ! "So, to judge whether to replace floats and doubles with integers, an implementer must determine how often the different arithmetic functions are likely to be used" !!. In my world integer and floating point numbers are two entirely different things. The chapter about sorting algorithms contains nothing else than any other book on the subject. The O notation is the only thing regarding algorithm performance. The entire chapter looks like something that is thrown in to give the book a little

thickness. There is no information on modern processors. Data caching and optimizations on it is explained in one page. No such thing as level 1, level 2 caches or cache line splits are mentioned. Who cares about printf and puts performance these days. The book contains little that could not be found out doing a little timing on a few basic code snippets by yourself. A lot of easy going language is used - walking carefully around substance without touching it. The book can serve as a light introduction on code performance for beginning SW developers - nothing else.

Concentrates on a broad range of performance optimization techniques. The author strikes a nice balance between programming in-the-large and programming in-the-small techniques. The programming in-the-large chapters discuss language selection (assembly, C, C++, Pascal, Visual Basic, or Java), O notation, memory managers, string searching, sorting, and data structures (arrays, linked lists, hash tables, binary trees, and red/black trees). Includes a lucid analysis of memory managers. Programming in-the-small issues are variable types, variable scope, variable initialization, structures, bit fields, unions, selectors, jump tables, loops, invoking functions, macros, inline functions, iteration vs. recursion, pass by reference vs. pass by value, global data, virtual functions, and templates. Excellent set of charts containing timing comparisons of the various techniques. The source code is available from the publishers web site. There are several lapses in this text. All of the timing charts for comparing the various techniques should have been printed in the book. The third section of the book on avoiding C traps and pitfalls is off topic. Programming in-the-small optimization techniques are not covered comprehensively. One glaring omission is the technique of using a sentinel element as a loop termination signal. The author does not provide a bibliography or references to standard industry texts. At a minimum the sorting and string searching sections should have referenced Donald Knuth's 'The Art of Computer Programming: Sorting and Searching, Vol. 3' and Dan Gusfield's 'Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology'.

Apart from a solid introduction into writing more efficient code, this book covers mostly advanced topics. However, I found these were explained very clearly, with plenty of programming examples given in the text making it surprisingly easy to follow. I would certainly suggest picking up this book if you want to know more about optimized programming, or if you just need a quick reference to, say; speedy sorting techniques, efficient storage structures, pattern matching or intelligent IO. Beware though, this is not a C++ tutorial. For that you will have to go elsewhere. But if you are able to write programs a little more complicated than 'hello world' on your own, then you are already ready for the

next step in your education. This book will advance your c++ programming skills a good many more levels.

A couple of young IT guys trying to write a book for C++ experts. Very uneven coverage. Not much of interest. Short on decent examples. Somewhat dated. Try "Efficient C++" for a better (but still somewhat dated) book on optimization.

Basically this book does a good job of showing you what's going on behind the scenes when your compiler turns your source-code into a program. And what can go wrong, read: end up being wildly inefficient. The cool thing is, the book not only shows you how to optimize your programmes but also explains exactly when and why certain solutions can enhance your software's performance. I feel I have really gained much needed insight, which makes it possible for me to work out my own solutions to problems concerning stack frames, multitasking, data processing, parsing and so on. And helps me in finding those bugs that seem invisible because they have crept deep into the architecture.

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

C++ Footprint and Performance Optimization (Sams Professional) Hybrid Particle Swarm Algorithm for Multiobjective Optimization: Integrating Particle Swarm Optimization with Genetic Algorithms for Multiobjective Optimization Sams Teach Yourself UNIX System Administration in 24 Hours (Sams Teach Yourself in 24 Hours) Swift in 24 Hours, Sams Teach Yourself (Sams Teach Yourself -- Hours) Sams Teach Yourself ADO 2.5 in 21 Days (Sams Teach Yourself...in 21 Days) Network Performance and Optimization Guide: The Essential Network Performance Guide For CCNA, CCNP and CCIE Engineers (Design Series) WordPress: A Beginner to Intermediate Guide on Successful Blogging and Search Engine Optimization. (Blogging, SEO, Search Engine Optimization, Free Website, WordPress, WordPress for Dummies) Seo 2017: Search Engine Optimization for 2017. On Page SEO, Off Page SEO, Keywords (SEO Books, Search Engine Optimization 2016) SEO 2017: Search Engine Optimization for 2017. On Page SEO, Off Page SEO, Keywords (SEO Books, Search Engine Optimization 2017) SEO+Clickbank (Search Engine Optimization 2016): Use The Power of Search Engine Optimization 2016+ Clickbank Sams Teach Yourself Java 2 in 21 Days, Professional Reference Edition (3rd Edition) Teach Yourself Java in 21 Days: Professional Reference Edition (Sams Teach Yourself) Sams Teach Yourself Java 2 in 21 Days, Professional Reference Edition (2nd Edition) Sams Teach Yourself Java 2 Platform in 21 Days, Professional Reference Edition Developing Java Servlets (Sams Professional) Power and Performance:

Software Analysis and Optimization Managing Your Digital Footprint (Digital and Information Literacy) Enhancing Your Academic Digital Footprint (Digital and Information Literacy) Zero Footprint: The True Story of a Private Military Contractor's Covert Assignments in Syria, Libya, and the World's Most Dangerous Places 2013 ICD-9-CM for Hospitals, Volumes 1, 2, and 3 Professional Edition (Spiral bound), 2013 HCPCS Level II Professional Edition and 2013 CPT Professional Edition Package, 1e

[Dmca](#)