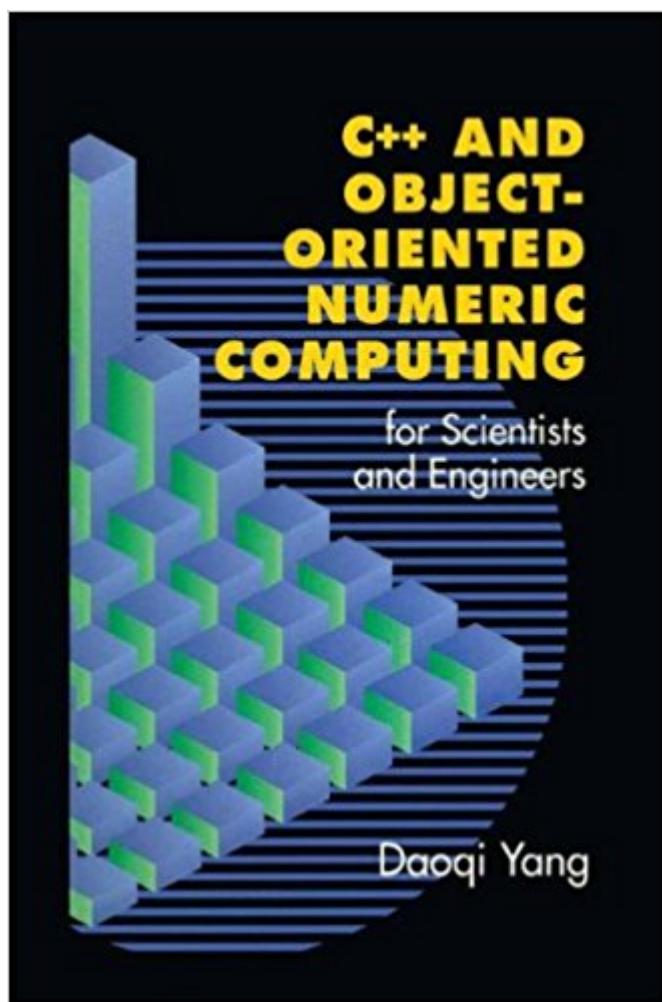


The book was found

C++ And Object-Oriented Numeric Computing For Scientists And Engineers



Synopsis

This book is an easy, concise but fairly complete introduction to ISO/ANSI C++ with special emphasis on object-oriented numeric computation. A user-defined numeric linear algebra library accompanies the book and can be downloaded from the web.

Book Information

Hardcover: 440 pages

Publisher: Springer; 2001 edition (October 23, 2000)

Language: English

ISBN-10: 0387989900

ISBN-13: 978-0387989907

Product Dimensions: 6.1 x 1 x 9.2 inches

Shipping Weight: 1.7 pounds (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 stars 20 customer reviews

Best Sellers Rank: #1,181,954 in Books (See Top 100 in Books) #92 in Books > Science & Math > Experiments, Instruments & Measurement > Microscopes & Microscopy #178 in Books > Science & Math > Mathematics > Number Systems #271 in Books > Computers & Technology > Programming > Languages & Tools > Compilers

Customer Reviews

Â¢This is one of the few good application-oriented C++ books that I have come across for students and professionals in mathematics, science, and engineering....Yang makes it attractive to all professionals in these fields and promotes the use of good objects-oriented programming practices....This book is complete and well written....it can promote good design practices among students in science and engineering. Its precision can make it a valuable reference for professionals in these areas.Â¢Â¢ACM COMPUTER REVIEWS

The "introduction" to C++ is done in stages as each basic feature of the language becomes relevant for numerically intensive computation. But the explanations are very brief -- accurate but with limited explanation. These areas of the book might be best for someone who knows C++ basics or is using a companion text for learning C++. The application of C++ to useful computational issues and algorithms is excellent - starting with simple examples and building logically to more complex methods and issues. The exercises are useful. Even simple problems are not that simple and very illustrative when the user gets the exercise "solved" with a working program. The C++ programming

style is acceptable and clear but sparse rather than elegant. I think that instructors who teach C++ would want some style changes in the code and more annotation. However I don't think those improvements are necessary for the issues and methods being explained by the author. The book seems to be what it says it is -- a very good introduction to numeric programming using C++. The author says that using C++ may be clumsy at first, but pays off handsomely in the long term. You will eventually gain efficiency in coding and speed of execution that surpasses other programming languages. There are some approaches to complex problems that you can use with an OOP language that would be much more tedious in older languages like FORTRAN, Algol, C or PL-1 (which I used years ago). A prime example is that a method for computing a solution using integers can be "inherited" by a method for real numbers, extra precision computation, complex numbers, etc. Each "new" method requires only modest additioins/changes from the "parent" method instead of a whole new "subroutine" with all the equations done again from scratch. So far (about the first 25% of the book), it looks like he is right..

If studying computational engineering, this book is a nice addition to one of the standard Deitel texts or similar. It maybe isn't quite as comprehensive on all C++ features as a 2000 page text, but it offers nice insights/ examples from a more engineering-oriented perspective. I read both and get a lot out of the combination of the two. One issue is that the accompanying code could be more well put-together. Some of the code in the book is not provided online and overall it would be nice to be able to quickly run the code shown in the book without having to type it all up and write your own dependent header files and such. This book might also be out of date with the 2011 C++ standard, which may or may not be an issue to some people.

There is widespread agreement that Kernighan & Richie's introduction to C is one of the best books on any programming language ever written. Sadly, no book on C++ has achieved this distinction. The complexity of the language is mostly to blame. Personally, I am averse to any 1000-page "primer" that aims to explain C++ from scratch up to intermediate level, with plenty of diversions, illustrations of secondary language features and a menagerie of typographic symbols, all the same getting to discuss classes at page 500. Those are better thought of as bloated references. Instead, C++ a third language, and C a second if only because a) C is not for absolute beginners and b) C programming style is often found in C++ legacy code. But if this is the case, the reader can read and understand succinct explanations of the new features of C++, and be quickly reminded of its similarities with C, or warned of the differences. If you agree with my own preferences, this book is

for you. It goes to the point and follows a linear, focused thread. Knowledge of C is not strictly required, but I can't see how can you accept double pointers as your personal saviours if you haven't seen them before. It's rather complete, but not all-encompassing. Topics are discussed clearly and the organizing principles are emphasized first, with the inevitable exceptions coming later. Examples are many. They are sometime thought in C but written in C++, which is no mortal sin. Because of the plain, assured style, and the approach to its subjects, the book reminds me of K&R. Not a small compliment to pay. The book also offers advice on programming style, mostly with an eye on numerical computing. It can be used a very high-level reference, to be complemented by a comprehensive one (my favorite is C++ in a nutshell). And each advanced subject of C++ deserves its own book, especially templates and metaprogramming. Avoid the "primers". Summing up, this is not a book for complete beginners, but more a textbook for a course or self-instruction. It's by no means a first and last book on C++. Such book has not been written yet. Know your needs and what you get.

As a C++ programmer, familiar to both numeric programming and RAD programming, I must say that this book is the best choice for anyone searching for a clear introduction to C++, OOP focused on numeric programming. Many books lack of numeric examples or OOP concepts, while Yang's book manage to balance them with uncommon sucess. A great book!

It's a huge topic and the book does offer a promising title. But it doesn't deliver, at least it confused about what is object-oriented. Writing stuff neat and reusable is not quite OO; when you are talking about a field, ex. numeric computing, framework or the concept of organizing that much of stuff depend on for the rest of your life is more important than some specific topic like vector, matrix, etc and their implementation. They are dead objects, at most you can call them numeric computing, but no OO please. By far I havn't seen any decent OO numeric computing book, 'cause most of these numeric stuff was first used by fortran ppl for long long time without any OO used. IMHO, fortran should be dead 5-10 years ago, but it's not, such a shame. I think one feasible way to start OO computing is to learn using the latest c++ tools (ex. boost, blitz, loki) to develope your own framework for your computing evironment.

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

C++ and Object-Oriented Numeric Computing for Scientists and Engineers A First Course in Scientific Computing: Symbolic, Graphic, and Numeric Modeling Using Maple, Java, Mathematica, and Fortran90 by Rubin H. Landau (2005-05-01) Physics for Scientists and Engineers: Vol. 2:

Electricity and Magnetism, Light (Physics, for Scientists & Engineers, Chapters 22-35) Physics for Scientists and Engineers with Modern Physics: Volume II (3rd Edition) (Physics for Scientists & Engineers) Scientific Computing: For Scientists and Engineers (De Gruyter Textbook) Introduction to High Performance Computing for Scientists and Engineers (Chapman & Hall/CRC Computational Science) Advice to Rocket Scientists: A Career Survival Guide for Scientists and Engineers (Library of Flight) Object Lessons for a Year: 52 Talks for the Children's Sermon Time (Object Lesson Series) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Introduction to Programming with Greenfoot: Object-Oriented Programming in Java with Games and Simulations (2nd Edition) Systems Analysis and Design: An Object-Oriented Approach with UML An Object-Oriented Approach to Programming Logic and Design Object-Oriented and Classical Software Engineering Object-Oriented Modeling and Design with UML (2nd Edition) Head First Object-Oriented Analysis and Design Tools For Structured and Object-Oriented Design (7th Edition) Object-Oriented Analysis and Design with Applications (3rd Edition) Object Oriented Systems Analysis and Design Object-Oriented Design and Patterns Object-Oriented Analysis and Design with the Unified Process (Available Titles CengageNOW)

[Contact Us](#)

[DMCA](#)

[Privacy](#)

[FAQ & Help](#)