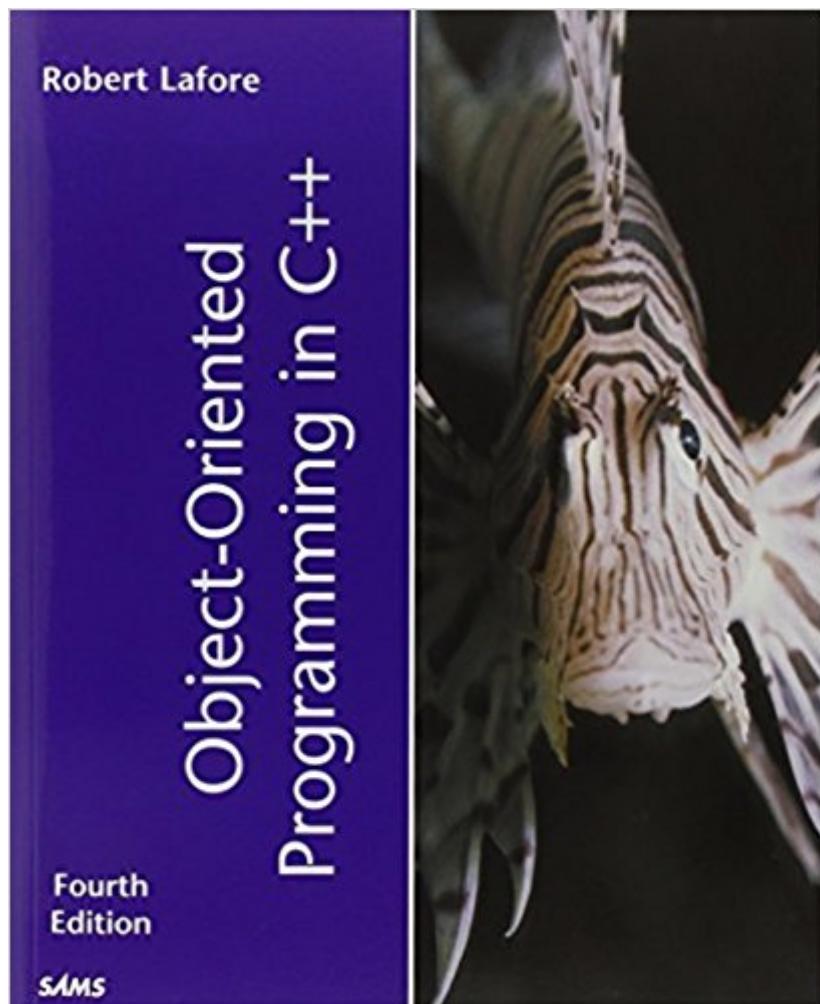The book was found

Object-Oriented Programming In C++ (4th Edition)

Synopsis

Object-Oriented Programming in C++ begins with the basic principles of the C++ programming language and systematically introduces increasingly advanced topics while illustrating the OOP methodology. While the structure of this book is similar to that of the previous edition, each chapter reflects the latest ANSI C++ standard and the examples have been thoroughly revised to reflect current practices and standards. Ã Â Educational Supplement Suggested solutions to the programming projects found at the end of each chapter are made available to instructors at recognized educational institutions. This educational supplement can be found at www.prenhall.com, in the Instructor Resource Center.

Book Information

Paperback: 1040 pages

Publisher: Sams Publishing; 4 edition (December 29, 2001)

Language: English

ISBN-10: 0672323087

ISBN-13: 978-0672323089

Product Dimensions: 7.4 x 2.2 x 9.1 inches

Shipping Weight: 3.8 pounds (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 stars   61 customer reviews

Best Sellers Rank: #43,786 in Books (See Top 100 in Books)   #15 inÃ Â Books > Computers & Technology > Programming > Microsoft Programming > C & C++ Windows Programming   #18 inÃ Â Books > Computers & Technology > Programming > Languages & Tools > C & C++ > C #23 inÃ Â Books > Textbooks > Computer Science > Object-Oriented Software Design

Customer Reviews

"Object-Oriented Programming in C++" begins with the basic principles of the C++ programming language and systematically introduces increasingly advanced topics while illustrating the OOP methodology. While the structure of this book is similar to that of the previous edition, each chapter reflects the latest ANSI C++ standard and the examples have been thoroughly revised to reflect current practices and standards. Educational SupplementSuggested solutions to the programming projects found at the end of each chapter are made available to instructors at recognized educational institutions. This educational supplement can be found at www.prenhall.com, in the Instructor Resource Center.

Robert Lafore has been writing books about computer programming since 1982. His best- selling titles include Assembly Language Programming for the IBM PC, C Programming Using Turbo C++, C++ Interactive Course, and Data Structures and Algorithms in Java. Mr. Lafore holds degrees in mathematics and electrical engineering, and has been active in programming since the days of the PDP-5, when 4K of main memory was considered luxurious. His interests include hiking, windsurfing, and recreational mathematics.

Let me start off by saying that I have a degree in fine art and NOT computer science or anything else that's related to computers.With that out of the way, I can honestly say that the author of this book is methodical, concise, and complete in his descriptions. If you're looking for a picture book with drawings and diagrams you may want to look somewhere else.I will say that I approached this book with an absolute basic understanding of the concept of a variable, a string and that was about it. I had tried to read/surf the net about functions, classes, and loops but never really found a definitive source that could explain the concepts for my "art" brain to digest. This book has answered the call. I'm sure you've read similar statements but I'll say it again, don't expect to learn a programming language after reading a couple of pages and copying some code. I'm starting chapter 3 after having the book for approx. 1 week and I have a solid understanding of variable types, cout, cin, how to perform basic arithmetic in C++, calling functions, and parameters just to name a few. For a person with a creative and not-so-computer tech mindset, this is a major accomplishment.The pacing is nice if you're willing to read the material, review, and actually do the exercises at the end of the chapters. Overall, I'm happy with this purchase and confidently learning C++ thanks to this book.If I have just one criticism I would say that I wish there were answers for every exercise question. I understand the reasoning for not having answers to all of the exercise problems(the amount of pages it would take to create the answers would be insane). Still, there is an answer key for the review questions and some of the exercises.

The writing is thorough and easy to read. The book has helped iron out the wrinkles in my patchwork C++ knowledge. However, the book is written with frequent examples of code to illustrate the points being made. These code sections are very difficult to read because the lines wrap around due to the book page being wider than the Kindle screen. Computer code, of all things, has to be very precise, and the unfortunate formatting harms the readability and prevents the reader from being able to develop a feel for what good code should look like. Still, the book has been helpful in spite of the examples, and I would recommend it.PS - Setting the Kindle to display horizontally and

shrinking the font will pretty much eliminate the wrapping problems. This is a good solution if you can tolerate reading the rest of the text in this mode as well. Changing the settings for every example would be tedious at best because the examples are so frequent.

Good, but has typos and mistakes in code samples, sometimes you'll see things like char * str = new char[length + 1]; but then author does delete str; instead of delete [] str; so watch out for those things and read everything carefully! Other than that it's a really great book, if you read this book to the end, I strongly suggest you read Scott Meyers' Effective C++ series (Effective C++, More Effective C++, Effective STL), I personally ordered those 3, and also The Boost C++ Libraries: Generic Programming by Sergei Nakariakov and C++ Templates: The Complete Guide by David Vandevoorde after I read this book.

I took several C++ Program In Computing courses while in college, and I must say, this book did a fantastic job of turning the what was then "clear as mud" concepts to something I could understand.Since those courses, I've learned C and a few other languages, but this book does a fantastic job of introducing OO concepts to a programmer, or a beginner who was otherwise doing procedural programs, or was just starting fresh. Lafore goes over the benefits of OO, and slowly builds on top of concepts throughout the book until you get a good overall understanding of the theory and the practice.Two things I didn't like about the book:1. There was very little discussion about doing things the "right way". By this, I mean, Lafore didn't really mention how certain methods of writing programs were less efficient than others. It looks like he was using the best practices in the book, so maybe that would have been overkill, but going a bit more in depth about why to do certain things certain ways would have been nice. While this is a beginner book, it certainly doesn't hurt to have sidenotes of this nature for those interested.2. While the majority of the code in this book was portable to most platforms, it's still assumed that you're going to be on a Windows plaform, and Lafore uses several Windows-only graphics libraries to illustrate some points. Being a UNIX guy, I would have preferred that he stick to a platform neutral explanation.

If I had read this book years ago, I'd have dropped everything to go for a career in programming/software engineering. Many thanks to the author for sharing his gift.

I bought this book long time ago when I was a student, but had not read it seriously. However, I started to read this book a month ago because of my job duty. Then I found that this book is

excellent in helping readears understand essencial conepts and principles which are required for C++ programming. The reasons why I am thinking like that are as below. This book explains 'why' as well as 'how', describing what's happening inside of system or C++ compiler. Each chapter starts off with some possible real-programming issues that we may face, then moves to fundamental and essential concepts, which make me think about some applications of the contents. Example codes which are shown in this book are very easy and clear, but very helpful for understanding concepts and principles which are very essential.I strongly recommend you to read this book to be an expert in C++ programming.

Download to continue reading...

Introduction to Programming with Greenfoot: Object-Oriented Programming in Java with Games and Simulations (2nd Edition) Programming Python: Powerful Object-Oriented Programming Java Programming: Intermediate Concepts for the Fundamentals of Object Oriented Programming Object-Oriented Programming in C++ (4th Edition) Object-Oriented Programming in Java: A Graphical Approach, Preliminary Edition An Introduction to Object-Oriented Programming (3rd Edition) Java Methods: An Introduction to Object Oriented Programming An Object-Oriented Approach to Programming Logic and Design Microsoft Visual C#: An Introduction to Object-Oriented Programming An Introduction to Object-Oriented Programming with Java Beginning Java Programming: The Object-Oriented Approach 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) The Object-Oriented Thought Process (4th Edition) (Developer's Library) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Object-Oriented Modeling and Design with UML (2nd Edition) Tools For Structured and Object-Oriented Design (7th Edition) Object-Oriented Analysis and Design with Applications (3rd Edition) Object Oriented Software Development Using Java (2nd Edition) Object Lessons for a Year: 52 Talks for the Children's Sermon Time (Object Lesson Series)