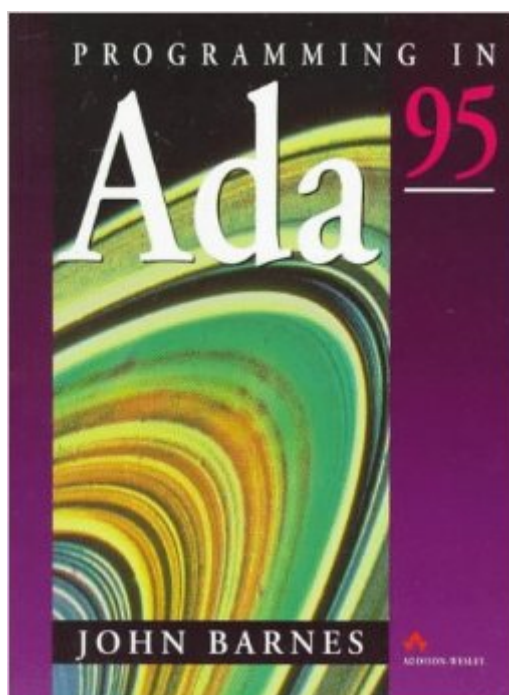


The book was found

Programming In Ada 95 (International Computer Science Series)



Synopsis

This text introduces and highlights the important changes from ANSI 83 for the ADA 92 standard, including increased support for object-oriented programming, the introduction of a hierarchical library structure, and the inclusion of protected objects.

Book Information

Series: International Computer Science Series

Paperback: 720 pages

Publisher: Pearson Education; 1st edition (September 28, 1995)

Language: English

ISBN-10: 0201877007

ISBN-13: 978-0201877007

Product Dimensions: 6.6 x 1.2 x 9.2 inches

Shipping Weight: 2.5 pounds

Average Customer Review: 3.5 out of 5 stars [See all reviews](#) (32 customer reviews)

Best Sellers Rank: #2,137,048 in Books (See Top 100 in Books) #32 in [Books > Computers & Technology > Programming > Languages & Tools > Ada](#) #15117 in [Books > Computers & Technology > Software](#) #36628 in [Books > Science & Math > Mathematics](#)

Customer Reviews

John Barnes' book is a very easy to read, complete coverage of a powerful language. The book is intended for those with some knowledge of programming, providing both details and, especially, rationale for the language features. Explaining the rationale and possible use of language features is, in fact, the best aspect of the book, and has long been a characteristic of Barnes' writing. So many books in this category simply reiterate syntax and semantics; read this book and you will *understand* Ada as well as recognize it. If you pay attention there is even subtle humor strewn throughout, making an otherwise potentially dry subject a pleasure to read about. As a long-time Ada developer and a provider of industrial Ada-related courses, I strongly recommend this book to those who have no prior experience with the Ada language. As the owner of over forty books on the subject (including both editions of this one!) I rate this second edition in the top handful of all books concerning the Ada language.

Man, these Ada books are expensive! But this was 1/2 of most others. The paperback cover is kinda flimsy, but it'll just make me look like a more experienced programmer as it gets dogeard

'-)I'm a long time C++'er but knew nothing about Ada, which 95% of my new job involves. So I needed something to bring me up to speed quickly. I find this book very readable and pretty well organized. The first few chapters teach you the 10% you'll use 90% of the time and the other 90%, if you need it, is covered in depth later in the book. I think those that don't like this book probably don't like Ada (but who does?) because I find this book great, Ada is just a tough language that you just can hack with like C++.I'm very pleased with this "bargan" book and think it'll be the only Ada book I'll need to buy. Strongly recommend

If you believe all the bad reviews you've read about this book...don't! Anyone who thinks this book is hard to read and should only be used by experienced programmers probably shouldn't even be trusted to tie their own shoes. I didn't know a thing about Ada before buying this book and after reading it I feel I know as much as Mr. Barnes himself. The history of Ada83 and Ada95 - as well as OO programming - is well-organized and top-notch. The sections clearly mark differences between Ada83 and Ada95. It has numerous real-world examples which are helpful for my every day applications. It's also a great progressive learning tool with numerous exercises at the end of each section.

This should be one of your books if you are really serious about Ada. The author was one of the leaders of the small team that actually designed the Ada language and does a great job of presenting the essentials of the language in a easy to read format. While more complete examples would have helped, this is still the book you want to have on your desk when creating an Ada program. It should not, however, be the only book. Ada is a complex and beautiful language - it helps to see it from more than one view. Many reviewers of this book who gave it a poor score appear to be students or people who don't like the language in any case. If you don't see the value of learning Ada, no book will help that much. Some reviewers could not even spell the language (i.e. it is Ada, named after Ada Lovelace Byron, not ADA as in the American Dental Association). Ada is not as highly used as some other languages but is still a very popular language in certain segments and can provide a rewarding career for the serious student. I have been steadily employed for over 10 years using only Ada and have no trouble making a six figure income. I certainly don't see Ada as a 'dead language'!

I learned Ada 83 years ago with the help of an earlier version of this book. I have recently been continuing this pattern with Ada 95. Mr. Barnes does a great job explaining the subject matter,

without getting bogged down in minutiae. The code examples and exercises are also quite helpful. I highly recommend this book.

The question that needs to be asked and that no other book on Ada directly answers is: Why Ada? More specifically what will Ada do for me that C++ cant. I started learning C++ 10 years ago and once I got a handle on the language I thought there had to be a better way. A language that is used for almost all major large scale programming that puts all the responsibility of type checking, bounds checking and allocation/deallocation of resources on the shoulders of the programmer cant be right. Sure a lot of C++ books will list some general advantages and disadvantages of the language in the beginning of the book, but not one single book will answer the question why....why is the C++ way of doing things the right way or at least the most efficient way. I looked for an alternative to C++ and found Ada. All the Ada books Ive owned list general advantages of the language but none of the books could explicitly explain what Ada is doing for you and why that is the better way of doing it, none until I found this book. John Barnes is the only author I know who can directly explain what Ada does. Everything about Ada is polar opposite of C++, it has very strong typing, strong typing is the foundation on which makes Ada object orientation, pointers, resource allocation and even threading work together seamlessly. People mistake the static typing of C++ for strong typing, but that couldnt be further from the truth. Static typing in C++ just allows you to tell the compiler how you want the bits of certain data to be interpreted, thats all, nothing more. When I read the reviews for this book here on I thought I should check it out and bought it used. I wish I found this book a long time ago and after I finish it I will buy Barnes up to date book on Ada 2005 to keep as a reference.

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

Programming in Ada: Plus an Overview of Ada 9X (International Computer Science Series) Python: Python Programming For Beginners - The Comprehensive Guide To Python Programming: Computer Programming, Computer Language, Computer Science Python: Python Programming For Beginners - The Comprehensive Guide To Python Programming: Computer Programming, Computer Language, Computer Science (Machine Language) Programming in Ada 95 (International Computer Science Series) Programming in Ada Plus Language Reference Manual (International computer science series) Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX (4th Edition) (International Computer Science Series) ADA Programming Success In A Day: Beginner's guide to fast, easy and efficient learning of ADA programming Ada's Ideas: The Story of Ada Lovelace, the World's First Computer Programmer Ada 95 Reference Manual. Language and Standard Libraries: International Standard ISO/IEC 8652:1995 (E) (Lecture

Notes in Computer Science) PowerShell: For Beginners! Master The PowerShell Command Line In 24 Hours (Python Programming, Javascript, Computer Programming, C++, SQL, Computer Hacking, Programming) Software Components With Ada: Structures, Tools, and Subsystems (The Benjamin/Cummings Series in Ada and Software Engineering) Reusable Ada Components Sourcebook (The Ada Companion Series) El violÃn de Ada (Ada's Violin): La historia de la Orquesta de Instrumentos Recicladados del Paraguay (Spanish Edition) Alex + Ada Volume 2 (Alex + ADA Tp) Ada's Algorithm: How Lord Byron's Daughter Ada Lovelace Launched the Digital Age Haskell: The Craft of Functional Programming (3rd Edition) (International Computer Science Series) Fortran 77 Programming: With an Introduction to the Fortran 90 Standard (International Computer Science Series) An Introduction to Logic Programming Through Prolog (Prentice Hall International Series in Computer Science) Productive PROLOG Programming (Prentice-Hall International series in computer science) Ada Lovelace, Poet of Science: The First Computer Programmer

[Dmca](#)