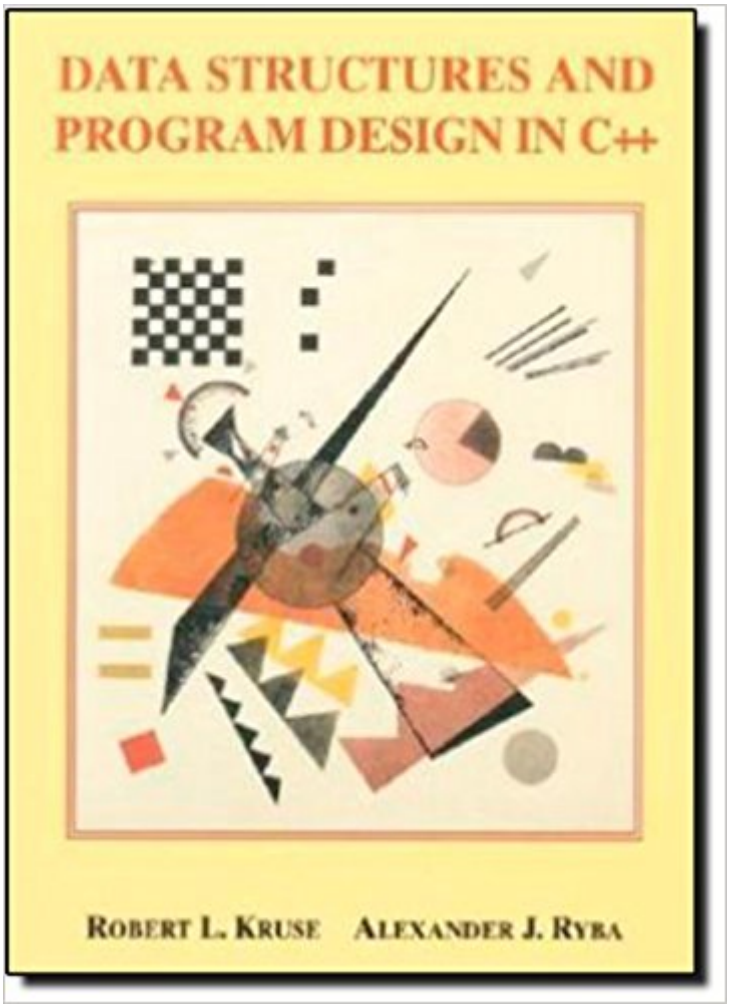
Data Structures And Program Design In C++



Synopsis

Progressing from the concrete to the abstract â ” and using numerous, substantial case studies and sample programs â ” this book explores structured problem solving, data abstraction, software engineering principles, and the comparative analysis of algorithms as fundamental tools of program design. The book and all programs have been completely written from the Object-Oriented perspective. Uses the C++ programming language throughout. Briefly reviews the syntax of C++ and provides a brief introduction to the language. The book is native C++ making full use of C++ features and object-oriented programming. Discusses major principles of software engineering and applies them to large programming projects. Covers several more advanced, modern topics, e.g.: Splay trees, Red-black trees, Amortized algorithm analysis.

Book Information

Hardcover: 717 pages

Publisher: Prentice Hall; 1st edition (October 3, 1998)

Language: English

ISBN-10: 0137689950

ISBN-13: 978-0137689958

Product Dimensions: 7.5 x 1.6 x 9.1 inches

Shipping Weight: 2.7 pounds (View shipping rates and policies)

Average Customer Review: 2.5 out of 5 stars 39 customer reviews

Best Sellers Rank: #87,670 in Books (See Top 100 in Books) #16 inÂ Books > Computers & Technology > Programming > Algorithms > Data Structures #34 inÂ Books > Computers & Technology > Programming > Microsoft Programming > C & C++ Windows Programming #42 inÂ Books > Computers & Technology > Programming > Languages & Tools > C & C++ > C

Customer Reviews

In a surprisingly lively textbook-style treatment, Data Structures and Program Design in C++ delivers expertise and plenty of sample programs for the working C++ programmer or computer science student. While some books on data structures stress theory and mathematical concepts over real-world sample code, this guide illustrates its tour of data structures--such as stacks, lists, queues, trees, and graphs--with clear, engaging samples. Throughout, the authors make use of built-in C++ features (such as the Standard Template Library [STL] and templates) where appropriate. Early chapters use such interesting examples as Conway’s Game of Life, chess and game programming, a simple calculator, and an airport simulation. Along the way, the reader will

learn about lists, stacks, and queues. In later chapters, covering thornier topics such as sorting algorithms, trees, and graphs, the authors do not skimp on the mathematical underpinnings for measuring efficiency. Instead, they take extreme care to introduce everything required to understand such conventions as the "Big O" notation and principles of logarithms. The book closes with a case study that combines several data structures and strategies. (The example, a Polish notation expression parser, is a difficult and common real-world sample.) -- Richard Dragan

Progressing from the concrete to the abstract -- and using numerous, substantial case studies and sample programs -- this text explores structured problem solving, data abstraction, software engineering principles, and the comparative analysis of algorithms as fundamental tools of program design. --This text refers to an alternate Hardcover edition.

I just finished an online course on data structures at UMass, Lowell (secured an A grade without any problem) and this is the textbook used for this course. I agree with some of the comments made by the reviewers here - that typedef’s are used beyond reason, code is presented in fragments. I must strongly disagree with the reviewers who said - the book is not worth the money (it’s cheap if you buy a used one), it is a bad book, useful only as a paper weight. The strong plus point for this book is that it presents theory well - there are many figures using which it is easy to understand complicated stuff. The other plus point is that it also analyzes the performance of algorithms and I felt the book does a great job of explaining this part in clear terms. The main drawback is that the code is fragmented and spread out, which is certainly frustating. Also, in order to get the code to work on a compiler (I used Microsoft VC++ version 6), you will need to "fill in the gaps" as most code examples (I tried 90% of the examples in the text on MVC++ compiler) are by no means complete but this was not difficult. You can download the code from the publisher’s website and the code is actually not organized into different files (sorted by chapter and example) but it comes in a few files where examples from different chapters are mixed. You will have to search and find the part you need. I do agree usage of dummy functions creates confusion. This book has a rocky start with the life game example, which was not very easy to follow with the explanation provided. While I was frustated with the first two chapters, the rest of the chapters are presented well. To the reviewer who said that the code has bugs (which certainly isn’t true), my guess that the gaps are not properly filled. I patiently tested most of the code on the computer (after filling in the gaps of course!) and find absolutely no problems with it. This however took lot of time since significant effort is needed from the student to fill the missing code to make it work. I have supplied working code to my fellow

students who were facing difficulties in getting the code to run. In a couple of places in the text, the author surprised me with C syntax I did not know was legal (I consider myself intermediate programmer).I suspect the reason why many readers have problems with this book is two-fold: 1.The code is not available in a format that can be tested on the compiler. Having read a lot of CS books which supply with readily usable code, this book gets annoying. I was wondering why the authors did not give downloadable working code for all examples in the text (which is a definite minus point) 2. The reader will have to go back and forth between the chapters as some functions developed in former chapters are used in later chapters. This does get irritating. I have read books that do this to a ridiculous level but this book stays with in tolerable bounds.I will not rate this book as the best one on the subject out there but it is definitely good enough to learn data structures. I have used another book as a supplement: Data Structures & Algorithms in Java (Mitchell Waite Signature Series) (Hardcover) By Mitchell Waite, Robert Lafore ISBN: 1571690956. This book has lot of applets that show step by step how algorithms work. I am a visual learner and this helped tremendously. However, I felt that Waite Series book did not present the theory as thoroughly as Robert Kruse's text.In summary, I consider this book as "decent". I was very much concerned when I first purchased this text because of so many bad reviews. As it turns out, those that can understand C and are willing to sit in front of the computer to make the code samples in this book work need not have any fear. As I said earlier, the theory is presented well, and all it needs is patience and diligence from the student to go through the code examples on a compiler. I believe that I now have a good understanding of the subject and I can move on books that deal with it at more depth. There are a plethora books on this topic out there and there may be better books than the ones mentioned in this review.

The author starts out by giving the reader one of the most confusing examples (The Game of Life) I have ever seen, then proceeds to base everything in the book on that one example.It will give you a headache, I promise.Try Data Structures Using C by Yedidyah Langsam. I had a copy that I got cheap as a backup and it saved my life.

I took two half year C language programming classes. The book for my first semester C class was much better and included all the information we needed for the 2nd course, as well as being written much better. The code in the book is confusing and convoluted.I would not suggest this for an experienced or new programmer.

I expected so much more from a book that rode the high praises of my univeristy's computer science department. The author presents the data structures themselves in a great manner but when it comes to code, you would be better suited facedesking your keyboard. We had 3 programming assignments based on code in certain sections. 1. They were all wrapped in this bad templating that just didnt work out at all. 2. Every single program he gave us to use segfaults. His logic is so broken he outta be ashamed of writing this book. so when it came to the programming assignments to modify the author's code to work for something else, the class was screwed and we all failed cuz we couldnt even get the normal code to run, which as an author of a textbook is his sole responsibility: To make sure everything is accurate and works; he failed this hard. Therefore i strongly urge anyone who has to use this book to not use this book for any kind of coding and to beg your professor to use a different book, its only use is for when it gets cold and u and ur classmates wanna have a good old fashion book burning.

The item that I bought was EXCELLENT. It's like brand new and the shipping was super duper fast! Thank you so much for your business.

Do you have basic knowledge of C++ syntax? Have you taken a first semester class, or worked through an intro C++ book? Are you the kind of programmer who learns by tinkering with code, rather than complaining that they aren't given 100% correct code from the start?The best way to learn code (as anybody from Google, Microsoft, or will tell you) is to actually get your hands dirty with it, and this book does this, both implicitly and explicitly.What does that mean? It explicitly does it by ending each chapter with a LARGE program. They're still toy programs, but they're actually quite instructive. The very first chapter puts together a functioning Game of Life program while teaching basic C++ syntax. The Queue chapter builds an airport runway with airplanes coming and going at random times. Chapter 6 builds a very simple text editor! These projects are some of the most successful projects I've ever seen in taking basic data structures and actually showing how they can be used in real world applications. In other words, they're excellent at turning you from a student of C++ into someone who is a SOFTWARE ENGINEER.However, the book has some implicit difficulties to be honest. The code is old-fashioned in the sense that it feels like the type of code a C programmer writes while learning C++. Modern C++ just feels much more object-oriented than this code does. There are some errors in it, but I feel that anyone with basic coding knowledge should be able to figure out what's not working right. Syntax errors are some of the easiest and most basic errors to fix.All in all, this is the BEST data structures book in C++ I've ever seen. There

are some difficulties, and I wish that the author would update the book to meet C++11 standards, but if you do all the projects in this book, you will come out as more than just a student who understands data structures. You will be a novice software engineer.