The Definitive C Book Guide and List

This question attempts to collect a community-maintained list of ***quality*** books on the C programming language, targeted at various skill levels.

C is a complex programming language that is difficult to pick up on-the-go by reading online tutorials. A comprehensive book is often the best way to learn the language, and finding a good book is the first step. It is important to avoid badly-written books, and even more importantly, books that contain serious technical errors.

Please suggest edits to the accepted answer to add quality books, with an approximate skill level and a short blurb/description about each book. (Note that the question is locked, so no new answers will be accepted. A *single answer* is being maintained with the list.)

Feel free to debate book choices, quality, headings, summaries, skill levels, and anything else you see that is wrong. Books that are deemed satisfactory by the C community here will stick around on the list; the rest will be regularly removed.

For books that have reviews by the Association of C and C++ Users (ACCU), a link to those reviews should be added along with the book.

See also:

- Other C-related resources in the c tag wiki
- A similar list for C++: The Definitive C++ Book Guide and List

c

edited Aug 12 '17 at 12:16

community wiki
29 revs, 19 users 21%
lillq

locked by George Stocker ♦ Aug 1 '13 at 17:25

This question's answers are a collaborative effort: if you see something that can be improved, just edit the answer to improve it! *No additional answers can be added here*

31 Not to belittle the question, which is good, but...most people take years to master K&R. There is more in there than you think. The thinness of the book is deceptive. – dmckee Jul 14 '09 at 21:32

4 If you are new to C, keep in mind that modern C is ANSI C and anything predating that standard (1989) may be wildly out of date. Shoot for the mid-90s or later. – Dana Robinson Jun 7 '11 at 1:28

7 @Dhaivat I think not, be careful to jump on the K&R bandwagon. K&R does not address good program design nor good programming practice, mainly because it was originally written before anyone knew what good programming practice was. It does not mention which parts of the C language that are superfluous or even dangerous. The book is correctly listed as a reference manual, it should not be used for teaching/learning modern programming. – Lundin Aug 12 '11 at 7:45

I agree with Dana that nothing written before the ANSI/ISO standardizations in 89/90 should appear on a list like this. You may also consider stating which books that address C99 and which that doesn't. – Lundin Aug 12 '11 at 7:48

1 Answer

Reference (All Levels)

- The C Programming Language (2nd Edition) - Brian W. Kernighan and Dennis M. Ritchie (1988). Still a good, short but complete introduction to C, written by the the inventor of C. However, the language has changed and good C style has developed in the last 25 years, and there are parts of the book that show its age.
- C: A Reference Manual (5th Edition) - Samuel P. Harbison and Guy R. Steele (2002). An excellent reference book on C, up to and including C99. It is not a tutorial, and probably unfit for beginners. It's great if you need to write a compiler for C, as the authors had to do when they started.
- C Pocket Reference (O'Reilly) - Peter Prinz and Ulla Kirch-Prinz (2002).
- The comp.lang.c FAQ - Steve Summit. Web site with answers to many questions about C.
- Various versions of the C language standards can be found here.
- The new C standard - an annotated reference (Free PDF) - Derek M. Jones (2009). The "new standard" referred to is the old C99 standard rather than C11.

- Rationale for C99 Standard.

Beginner

- Programming in C (4th Edition) - Stephen Kochan (2014). A good general introduction and tutorial.
- C Primer Plus (5th Edition) - Stephen Prata (2004)
- C Programming: A Modern Approach (2nd Edition) - K. N. King (2008). A good book for learning C.
- A Book on C - Al Kelley/Ira Pohl (1998).
- The C Book (Free Online) - Mike Banahan, Declan Brady, and Mark Doran (1991).
- C: How to Program (8th Edition) - Paul Deitel and Harvey M. Deitel (2015). Lots of good tips and best practices for beginners. The index is very good and serves as a decent reference (just not fully comprehensive, and very shallow).
- Head First C - David Griffiths and Dawn Griffiths (2012).
- Beginning C (5th Edition) - Ivor Horton (2013). Very good explanation of pointers, using lots of small but complete programs.
- Sams Teach Yourself C in 21 Days - Bradley L. Jones and Peter Aitken (2002). Very good introductory stuff.
- Applications Programming in ANSI C - Richard Johnsonbaugh and Martin Kalin (1996).

Intermediate

- Object-oriented Programming with ANSI-C (Free PDF) - Axel-Tobias Schreiner (1993). The code gets a bit convoluted. If you want C++, use C++.
- C Interfaces and Implementations - David R. Hanson (1997). Provides information on how to define a boundary between an interface and implementation in C in a generic and reusable fashion. It also demonstrates this principle by applying it to the implementation of common mechanisms and data structures in C, such as lists, sets, exceptions, string manipulation, memory allocators, and more. Basically, Hanson took all the code he'd written as part of building Icon and lcc and pulled out the best bits in a form that other people could reuse for their own projects. It's a model of good C programming using modern design techniques (including Liskov's data abstraction), showing how to organize a big C project as a bunch of useful libraries.
- The C Puzzle Book - Alan R. Feuer (1998)
- The Standard C Library - P.J. Plauger (1992). It contains the complete source code to an implementation of the C89 standard library, along with extensive discussion about the design and why the code is designed as shown.
- 21st Century C: C Tips from the New School - Ben Klemens (2012). In addition to the C language, the book explains gdb, valgrind, autotools, and git. The comments on style are found in the last part (Chapter 6 and beyond).
- Algorithms in C - Robert Sedgewick (1997). Gives you a real grasp of implementing algorithms in C. Very lucid and clear; will probably make you want to throw away all of your other algorithms books and keep this one.
- Pointers on C - Kenneth Reek (1997).
- Pointers in C - Naveen Toppo and Hrishikesh Dewan (2013).
- Problem Solving and Program Design in C (6th Edition) - Jeri R. Hanly and Elliot B. Koffman (2009).
- Data Structures - An Advanced Approach Using C - Jeffrey Esakov and Tom Weiss (1989).
- C Unleashed - Richard Heathfield, Lawrence Kirby, et al. (2000). Not ideal, but it is worth intermediate programmers practicing problems written in this book. This is a good cookbook-like approach suggested by comp.lang.c contributors.

- Modern C — Jens Gustedt (2017). Covers C in 5 levels (encounter, acquaintance, cognition, experience, ambition) from beginning C to advanced C. It covers C11 threads and atomic access, which few other books do and not all compilers recognize in all environments.

Expert

- Expert C Programming: Deep C Secrets - Peter van der Linden (1994). Lots of interesting information and war stories from the Sun compiler team, but a little dated in places.
- Advanced C Programming by Example - John W. Perry (1998).

- Advanced Programming in the UNIX Environment - Richard W. Stevens and Stephen A. Rago (2013). Comprehensive description of how to use the Unix APIs from C code, but not so much about the mechanics of C coding.
- Advanced C: Food for the Educated Palate - Narain Gehani (1985). Great on pointers, pointers to functions, and a variety of advanced topics, such as how stuff is stored in memory, dynamic memory, stack usage, function calling, parameter passing, etc. Assumes you have a good grasp of C to start with. Warning: pre-dates the ANSI standard and a lot of modern programming design.
- Computer Programming: An Introduction for the Scientifically Inclined - Sander Stoks (2008). Great book about scientific use of programming languages.
- Reversing: Secrets of Reverse Engineering - Eldad Eilam (2005). For those who want to test the limits of their ethics.

Uncategorized

- Essential C (Free PDF) - Nick Parlante (2003). Note that this describes the C90 language at several points (*e.g.*, in discussing `//` comments and placement of variable declarations at arbitrary points in the code), so it should be treated with some caution.
- C Programming FAQs: Frequently Asked Questions - Steve Summit (1995).
- C in a Nutshell - Peter Prinz and Tony Crawford (2005). Excellent book if you need a reference for C99.
- Functional C - Pieter Hartel and Henk Muller (1997). Teaches modern practices that are invaluable for low-level programming, with concurrency and modularity in mind.
- The Practice of Programming - Brian W. Kernighan and Rob Pike (1999). A very good book to accompany K&R.
- C Traps and Pitfalls by A. Koenig (1989). Very good, but the C style pre-dates standard C, which makes it less recommendable these days.

  Some have argued for the removal of 'Traps and Pitfalls' from this list because it has trapped some people into making mistakes; others continue to argue for its inclusion. Perhaps it should be regarded as an 'expert' book because it requires a moderately extensive knowledge of C to understand what's changed since it was published.
- Computer Systems: A Programmer's Perspective (3rd Edition) - Randal E. Bryant and David R. O'Hallaron (2015). Explains the C language in a disjointed narrative style, like *Pulp Fiction*.
- Abstraction and Specification in Program Development - Barbara Liskov and John V. Guttag (1986) (*not* the newer Java-based version by Liskov alone). This is an undergraduate text, with some ideas worth thinking about.
- Composite/Structured Design - Glenford J. Myers (1978). This and other books from the late 1970s and early 1980s by Yourdon and Myers provide excellent insights on structured design.
- Build Your Own Lisp — Daniel Holden (2014). An enjoyable way to learn C.
- MISRA-C - industry standard published and maintained by the Motor Industry Software Reliability Association. Covers C89 and C99.

  Although this isn't a book as such, every experienced C programmer should read and implement as much of it as possible. MISRA-C was originally intended as guidelines for safety-critical applications in particular, but it applies to any area of application where stable, bug-free C code is desired (who doesn't want fewer bugs?). MISRA-C is becoming the de facto standard in the whole embedded industry and is getting increasingly popular even in other programming branches. There are (at least) three publications of the standard, one from 1998, one from 2004, and one from 2012, where the last is the currently active, relevant one. There is also a MISRA Compliance Guidelines document from 2016, and MISRA C:2012 Amendment 1 — Additional Security Guidelines for MISRA C:2012 (published in April 2016).
  Note that some of the strictures in the MISRA rules are not appropriate to every context. For example, directive 4.12 states "Dynamic memory allocation shall not be used". This may well be appropriate in the embedded systems for which the MISRA rules are designed; it is not appropriate everywhere. (Compilers, for instance, generally use dynamic memory allocation for things like symbol tables, and to do without dynamic memory allocation would be difficult, if not preposterous.)
- Archived lists of ACCU-reviewed books on Beginner's C (116 titles) from 2007 and Advanced C (76 titles) from 2008. Most of these don't look to be on the main site anymore, and you can't browse that by subject anyway.

Warnings

Be wary of books written by Herbert Schildt. In particular, you should stay away from C: The Complete Reference, known in some circles as C: The Complete Nonsense.

Also be wary of the book "Let Us C" by Yashwant Kanetkar. It is a horribly outdated book that teaches Turbo C and has lot of obsolete, misleading and downright incorrect material.

Learn C The Hard Way - Zed Shaw. A critique of this book by Tim Hentenaar:

> To summarize my views, which are laid out below, the author presents the material in a greatly oversimplified and misleading way, the whole corpus is a bundled mess, and some of the opinions and analyses he offers are just plain wrong. I've tried to view this book through the eyes of a novice, but unfortunately I am biased by years of experience writing code in C. It's obvious to me that either the author has a flawed understanding of C, or he's deliberately oversimplifying to the point where he's actually misleading the reader (intentionally or otherwise.)

"Learn C The Hard Way" is not a book that I could recommend to someone who is both learning to program and learning C. If you're already a competent programmer in some other related language, then it represents an interesting and unusual exposition on C, though I have reservations about parts of the book. *Jonathan Leffler*

Outdated

- Practical C Programming (3rd Edition) - Steve Oualline (1997)(Beginner)

Other contributors, not credited in the revision history:
Alex Lockwood, Ben Jackson, Bubbles, claws, coledot, Dana Robinson, Daniel Holden, Dervin Thunk, dwc, Erci Hou, Garen, haziz, Johan Bezem, Jonathan Leffler, Joshua Partogi, Lucas, Lundin, Matt K., mossplix, Matthieu M., midor, Nietzche-jou, Norman Ramsey, r3st0r3, ridthyself, Robert S. Barnes, Tim Ring, Tony Bai, VMAtm

edited Jan 2 at 22:21

community wiki
33 revs, 17 users 25%
Cody Gray

8 Discussion on meta about this post – Lundin Aug 24 '17 at 13:22