The Definitive C++ Book Guide and List

This question attempts to collect the few pearls among the dozens of bad C++ books that are published every year.

Unlike many other programming languages, which are often picked up on the go from tutorials found on the Internet, few are able to quickly pick up C++ without studying a well-written C++ book. It is way too big and complex for doing this. In fact, it is so big and complex, that ***there are very many very bad C++ books*** out there. And we are not talking about bad style, but things like sporting *glaringly obvious factual errors* and *promoting abysmally bad programming styles*.

Please edit the accepted answer to provide **quality books** and an approximate skill level — *preferably* **after** *discussing your addition in the C++ chat room*. (The regulars might mercilessly undo your work if they disagree with a recommendation.) Add a short blurb/description about each book that you have personally read/benefited from. Feel free to debate quality, headings, etc. Books that meet the criteria will be added to the list. Books that have reviews by the Association of C and C++ Users (ACCU) have links to the review.

Note: FAQs and other resources can be found in the C++ tag info and under c++-faq . There is also a similar post for C: The Definitive C Book Guide and List

c++ c++-faq

edited Mar 20 '14 at 11:46

community wiki
94 revs, 51 users 23%
sbi

locked by George Stocker ♦ Mar 20 '14 at 11:46

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*

1 Answer

Beginner

Introductory, no previous programming experience

- *Programming: Principles and Practice Using C++* (Bjarne Stroustrup) (**updated for C++11/C++14**) An introduction to programming using C++ by the creator of the language. A good read, that assumes no previous programming experience, but is not only for beginners.

Introductory, with previous programming experience

- *C++ Primer* * (Stanley Lippman, Josée Lajoie, and Barbara E. Moo) (**updated for C++11**) Coming at 1k pages, this is a very thorough introduction into C++ that covers just about everything in the language in a very accessible format and in great detail. The fifth edition (released August 16, 2012) covers C++11. [Review]

- *A Tour of C++* (Bjarne Stroustrup) The "tour" is a quick (about 180 pages and 14 chapters) tutorial overview of all of standard C++ (language and standard library, **and using C++11**) at a moderately high level for people who already know C++ or at least are experienced programmers. This book is an extended version of the material that constitutes Chapters 2-5 of The C++ Programming Language, 4th edition.

- *Accelerated C++* (Andrew Koenig and Barbara Moo) This basically covers the same ground as the *C++ Primer*, but does so on a fourth of its space. This is largely because it does not attempt to be an introduction to *programming*, but an introduction to *C++* for people who've previously programmed in some other language. It has a steeper learning curve, but, for those who can cope with this, it is a very compact introduction into the language. (Historically, it broke new ground by being the first beginner's book to use a modern approach at teaching the language.) [Review]

- *Thinking in C++* (Bruce Eckel) Two volumes; is a tutorial style *free* set of intro level books. Downloads: vol 1, vol 2. Unfortunately they’re marred by a number of trivial errors (e.g. maintaining that temporaries are automatically `const`), with no official errata list. A partial 3rd party errata list is available at (http://www.computersciencelab.com/Eckel.htm), but it’s apparently not maintained.

* Not to be confused with *C++ Primer Plus* (Stephen Prata), with a significantly less favorable review.

Best practices

- *Effective C++* (Scott Meyers) This was written with the aim of being the best second book C++ programmers should read, and it succeeded. Earlier editions were aimed at programmers coming from C, the third edition changes this and targets programmers coming from languages like Java. It presents ~50 easy-to-remember rules of thumb along with their rationale in a very accessible (and enjoyable) style. For C++11 and C++14 the examples and a few issues are outdated and Effective Modern C++ should be prefered. [Review]

- Effective Modern C++ (Scott Meyers) This is basically the new version of *Effective C++*, aimed at C++ programmers making the transition from C++03 to C++11 and C++14.

- *Effective STL* (Scott Meyers) This aims to do the same to the part of the standard library coming from the STL what *Effective C++* did to the language as a whole: It presents rules of thumb along with their rationale. [Review]

Intermediate

- *More Effective C++* (Scott Meyers) Even more rules of thumb than *Effective C++*. Not as important as the ones in the first book, but still good to know.

- *Exceptional C++* (Herb Sutter) Presented as a set of puzzles, this has one of the best and thorough discussions of the proper resource management and exception safety in C++ through Resource Acquisition is Initialization (RAII) in addition to in-depth coverage of a variety of other topics including the pimpl idiom, name lookup, good class design, and the C++ memory model. [Review]

- *More Exceptional C++* (Herb Sutter) Covers additional exception safety topics not covered in *Exceptional C++*, in addition to discussion of effective object oriented programming in C++ and correct use of the STL. [Review]

- *Exceptional C++ Style* (Herb Sutter) Discusses generic programming, optimization, and resource management; this book also has an excellent exposition of how to write modular code in C++ by using nonmember functions and the single responsibility principle. [Review]

- *C++ Coding Standards* (Herb Sutter and Andrei Alexandrescu) "Coding standards" here doesn't mean "how many spaces should I indent my code?" This book contains 101 best practices, idioms, and common pitfalls that can help you to write correct, understandable, and efficient C++ code. [Review]

- *C++ Templates: The Complete Guide* (David Vandevoorde and Nicolai M. Josuttis) This is *the* book about templates as they existed before C++11. It covers everything from the very basics to some of the most advanced template metaprogramming and explains every detail of how templates work (both conceptually and at how they are implemented) and discusses many common pitfalls. Has excellent summaries of the One Definition Rule (ODR) and overload resolution in the appendices. A second edition is scheduled for 2016. [Review]

Advanced

- *Modern C++ Design* (Andrei Alexandrescu) A groundbreaking book on advanced generic programming techniques. Introduces policy-based design, type lists, and fundamental generic programming idioms then explains how many useful design patterns (including small object allocators, functors, factories, visitors, and multimethods) can be implemented efficiently, modularly, and cleanly using generic programming. [Review]

- *C++ Template Metaprogramming* (David Abrahams and Aleksey Gurtovoy)

- *C++ Concurrency In Action* (Anthony Williams) A book covering C++11 concurrency support including the thread library, the atomics library, the C++ memory model, locks and mutexes, as well as issues of designing and debugging multithreaded applications.

- *Advanced C++ Metaprogramming* (Davide Di Gennaro) A pre-C++11 manual of TMP techniques, focused more on practice than theory. There are a ton of snippets in this book, some of which are made obsolete by typetraits, but the techniques, are nonetheless useful to know. If you can put up with the quirky formatting/editing, it is easier to read than Alexandrescu, and arguably, more rewarding. For more experienced developers, there is a good chance that you may pick up something about a dark corner of C++ (a quirk) that usually only comes about through extensive experience.

Reference Style - All Levels

- *The C++ Programming Language* (Bjarne Stroustrup) (**updated for C++11**) The classic introduction to C++ by its creator. Written to parallel the classic K&R, this indeed reads very much alike it and covers just about everything from the core language to the standard library, to programming paradigms to the language's philosophy. (Thereby making the latest editions break the 1k page barrier.) [Review] The fourth edition (released on May 19, 2013) covers C++11.

- *C++ Standard Library Tutorial and Reference* (Nicolai Josuttis) (**updated for C++11**) *The* introduction and reference for the C++ Standard Library. The second edition (released on April 9, 2012) covers C++11. [Review]

- *The C++ IO Streams and Locales* (Angelika Langer and Klaus Kreft) There's very little to say about this book except that, if you want to know anything about streams and locales, then this is the one place to find definitive answers. [Review]

C++11 References:

- *The C++ Standard (INCITS/ISO/IEC 14882-2011)* This, of course, is the final arbiter of all that is or isn't C++. Be aware, however, that it is intended purely as a reference for *experienced* users willing to devote considerable time and effort to its understanding. As usual, the first release was *quite* expensive ($300+ US), but it has now been released in electronic form for $60US

- *Overview of the New C++ (C++11/14) (PDF only)* (Scott Meyers) (**updated for C++1y/C++14**) These are the presentation materials (slides and some lecture notes) of a three-day training course offered by Scott Meyers, who's a highly respected author on C++. Even though the list of items is short, the quality is high.

Classics / Older

Note: Some information contained within these books may not be up-to-date or no longer considered best practice.

- *The Design and Evolution of C++* (Bjarne Stroustrup) If you want to know *why* the language is the way it is, this book is where you find answers. This covers everything *before the standardization* of C++.

- *Ruminations on C++* - (Andrew Koenig and Barbara Moo) [Review]

- *Advanced C++ Programming Styles and Idioms* (James Coplien) A predecessor of the pattern movement, it describes many C++-specific "idioms". It's certainly a very good book and might still be worth a read if you can spare the time, but quite old and not up-to-date with current C++.

- *Large Scale C++ Software Design* (John Lakos) Lakos explains techniques to manage very big C++ software projects. Certainly a good read, if it only was up to date. It was written long before C++98, and misses on many features (e.g. namespaces) important for large scale projects. If you need to work in a big C++ software project, you might want to read it, although you need to take more than a grain of salt with it. The first volume of a new edition is expected in 2015.

- *Inside the C++ Object Model* (Stanley Lippman) If you want to know how virtual member functions are commonly implemented and how base objects are commonly laid out in memory in a multi-inheritance scenario, and how all this affects performance, this is where you will find thorough discussions of such topics.

edited Nov 24 at 5:56

community wiki
26 revs, 22 users 38%
George Stocker

164 Is there a list of, perhaps, c++ books that SHOULD NOT be read? – G.Rassovsky Oct 21 '14 at 10:08

78 @G Rassovsky: All books which promise to teach X in Y hours. For example Learn C++ in 24 hours. I believe all such books are better avoided. – akhil_mittal Dec 29 '14 at 4:47

18 I hate to step on anybody's shoes, but I do not recommend Bruce Eckel's "Thinking in C++" even though I respect the author for publishing his materials online for free. The book's perspective suggests relatively poor or ineffective use of C++ and "object oriented" programming, akin to poor application of the GoF Design Patterns. I found it an interesting introductory book to programming in general, but as someone becomes more familiarized with programming and (especially) computer science as a whole, I find books which think purely in "classic" OOP terms detrimental to education. – Alexander Pritchard Jan 16 at 5:55

7 Which ones would be best for introducing C++14 to a C++03 programmer? – Michael Jun 4 at 23:00

6 @Michael: There's Scott Meyers' *Effective Modern C++* plus his *Overview of the New C++* (a PDF with, I believe, slides from his seminars). – sbi Jun 26 at 10:24

protected by Robert Harvey ♦ Apr 16 '12 at 17:10

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site.

Would you like to answer one of these unanswered questions instead?