



More Effective C++

Scott Meyers

[Download now](#)

[Read Online](#) ➔

More Effective C++

Scott Meyers

More Effective C++ Scott Meyers

More than 150,000 copies in print! Praise for Scott Meyers' first book, *Effective C++* "I heartily recommend *Effective C++* to anyone who aspires to mastery of C++ at the intermediate level or above."

- *The C/C++ User's Journal* From the author of the indispensable *Effective C++*, here are 35 new ways to improve your programs and designs. Drawing on years of experience, Meyers explains how to write software that is *more effective* more efficient, more robust, more consistent, more portable, and more reusable. In short, how to write C++ software that's just plain better. *More Effective C++* includes:

Proven methods for improving program efficiency, including incisive examinations of the time/space costs of C++ language features

Comprehensive descriptions of advanced techniques used by C++ experts, including placement new, virtual constructors, smart pointers, reference counting, proxy classes, and double-dispatching

Examples of the profound impact of exception handling on the structure and behavior of C++ classes and functions

Practical treatments of new language features, including bool, mutable, explicit, namespaces, member templates, the Standard Template Library, and more. If your compilers don't yet support these features, Meyers shows you how to get the job done without them. *More Effective C++* is filled with pragmatic, down-to-earth advice you'll use every day. Like *Effective C++* before it, *More Effective C++* is essential reading for anyone working with C++.

More Effective C++ Details

Date : Published January 8th 1996 by Addison-Wesley Professional (first published January 8th 1995)

ISBN : 0785342633719

Author : Scott Meyers

Format : Paperback 336 pages

Genre : Computer Science, Programming, Technical, Nonfiction, Software, Science

 [Download More Effective C++ ...pdf](#)

 [Read Online More Effective C++ ...pdf](#)

Download and Read Free Online More Effective C++ Scott Meyers

From Reader Review More Effective C++ for online ebook

Ken says

Skip this book. There's some interesting and useful content relating to exceptions and casting, in case you haven't encountered that advice elsewhere, but otherwise the material is largely dated. In addition, some of the recommendations in the Techniques section are IMO too clever by half, and not really good advice for maintainable software. Meyers is a great author with a great understanding of C++, but this book should be left behind in favor of other ones.

Benjamin says

Very insightful, but a little dated, hence only 4/5 stars.

Kirill says

Never aging classic, although some items from 1995 are deprecated - like `auto_ptr`. Most of all liked the "Techniques"- Items. It was a pleasure to read about reference counting, double dispatch and prohibiting heap-based objects.

Fond of the Scott's sparkling sense of humor, just like here:

"Programmers looking for a C++ compiler can ask vendors whether the return value optimization is implemented. If one vendor says yes and another says "The what?", the first vendor has a notable competitive advantage. Ah, capitalism. Sometimes you just gotta love it."

gargamelscat says

Just great - still after all these years. A must read for any C++ developer.

Craig says

Good read, A bit out-dated and most of it has now been covered in Effective C++ (Third Edition) so should just read that instead :)

Patrick says

Great way to get up to speed with post-1998 C++

Timothy Culp says

The inspiration behind most designs in the Origin baseline. Skip Effective C++ and jump right to this gem.

Alexander says

I found most of the book to be a little too wordy and some of the examples are not as concise as they could've been. But some tips were useful. The author talks mostly about general concepts that apply to c++ without too much referencing to STL. It's a very good book for people without too much experience with c++.

Xavrin says

Many things in this book didn't seem very interesting to me. Some of them I knew before, some have doubtful use in a modern programming.

Some of the presented techniques were trying to solve problems that you should almost never try to solve: e. g. find out if your object is constructed on the heap or force that there are at most N objects of your class.

Benoit Blanchon says

Unfortunately, this one is not as good as the others Effective C++.
It's less practical and too wordy.

Elliott Bignell says

I have owned this book for several years and typically re-read it about every year or so. To say that a book is necessary reading is sometimes hyperbolic, but if you want to understand C++ then it is probably the truth in this case, if less so than for Meyers' first book. Meyers has written three books on C++ - "Effective C++", this volume, and "Effective STL". You, fellow programmer, need them all. Plus "Design Patterns", of course. Stroustrup and the ACR probably belong beside them, but Meyers is uniquely readable. He's even funny. Some C++ experts will point to Stroustrup as the basis of your library. I have a copy of that, but I would not describe it as learning material; it is reference material. You can actually read Meyers for pleasure and for learning, and that's an unusual combination to find in software literature.

Meyers will help you get the best out of C++. If you can write classes in Java, C# or, heaven forfend, Visual Basic, then you can certainly write them in C++ and come up with a working design. That is not the same as using the language effectively. C++ is a work of beauty, but it has many pitfalls and it requires some depth of knowledge to get the best out of it. It's also portable, and Meyers knows how to write portable code, which must surely be a consideration with Ubuntu Linux and Apple advancing into the market. All genuine C++

compilers comply with an ISO/ANSI standard, which in theory means that properly-designed C++ code will compile on any conformant compiler and can be ported to any chipset for which a compiler exists. In practice, of course, this is not quite so easy. Meyers is very familiar with the standard and has been involved in its evolution, which along with his experience of developing in C++ positions him perfectly to point out where portability can and cannot be taken for granted.

The book is divided into 35 items, any of which will do you nicely for the train-journey to work. It is divided into "Basics", "Operators", "Exceptions", "Efficiency", "Techniques" and "Miscellany". It also throws in an `auto_ptr` implementation, which I would advise any C++ developer to study in detail, as smart pointers are one of the jewels of C++ design. (I would also then advise you to use the `shared_ptr` supplied with boost, as you will not find a better implementation.) Smart pointers are, in fact, a perfect synecdoche of the broader epiphany that you will go through as you master C++, and illustrate why Meyers is so worth reading. They can take the place of the garbage-collector in the more highly-managed languages, as they provide a way to guarantee that clean-up (or other) code is performed when a code section is exited - no matter how it is exited, if you also follow Meyers' advice on exception-handling. What this means is no more "delete" in your life. You hand the resource over to your smart pointer and the smart pointer cleans it up when its lifetime is over, no matter how many times "return" appears in a function, and even if the code section exits due to an exception. The standard also provides guarantees that this happens in a particular sequence and at a particular time during stack unwinding, in contrast to the vagaries and runtime cost of commercial garbage-collectors. I won't micturate on Meyers' geraniums by describing the mechanism in detail, but this should demonstrate the worth of mastering the C++ idiom.

See-also list:

Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition)

Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library

Design Patterns: Elements of Reusable Object-Oriented Software

Flavio Perri says

It was really great: I learn a lot of things about C++ designs and language traps. I give it 4/5 because it's a little dated (edited in 1997): few things were already explained in Effective C++ Third Edition (edited recently), and compilers now support most of the "future" features mentioned.

Beside this, the book it's really good, I hope it will come out with a new edition.

I'll start soon the third book of Meyers (Effective STL)

Alex says

This particular book is rather old among the other books in the series but very helpful nonetheless. All books by Scott Meyers are just must-read for any C++ dev (arguably even more so than the books by the language author himself). What I like about the book (and the whole series) is that they do not meticulously list all the language features where the API reference can suffice. Rather, they provide a practical advice on how to use C++ effectively (and efficiently) while avoiding the numerous traps which sadly C++ excels at setting.

Mickey Kawick says

Gave me a great background years ago and deals a bit with design patterns.

Prakash says

I get why Scott Meyers is such a celebrated author in the C++ world.

This is a sequel to the Effective C++ but is older because it was last edited in 1996-97.

There is a lot of wisdom and effective design issues discussed in the book. There is also the miscellany section which will help you become a better C++ programmer if not immediately providing you something practical.

Meyer's quite funny in this book and the itemized style works great as it did in Effective C++. (And it will in Effective Java & JCIP in future)

Not bad for a 20 year old book. I only take off a star as it is bit too wordy at some places.
