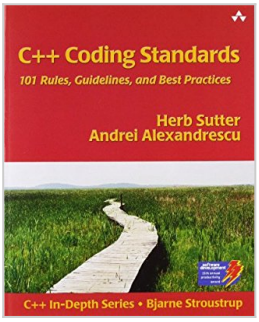

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices

By Herb Sutter, Andrei Alexandrescu

Download now

Read Online

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu

Consistent, high-quality coding standards improve software quality, reduce time-to-market, promote teamwork, eliminate time wasted on inconsequential matters, and simplify maintenance. Now, two of the world's most respected C++ experts distill the rich collective experience of the global C++ community into a set of coding standards that every developer and development team can understand and use as a basis for their own coding standards.

The authors cover virtually every facet of C++ programming: design and coding style, functions, operators, class design, inheritance, construction/destruction, copying, assignment, namespaces, modules, templates, genericity, exceptions, STL containers and algorithms, and more. Each standard is described concisely, with practical examples. From type definition to error handling, this book presents C++ best practices, including some that have only recently been identified and standardized-techniques you may not know even if you've used C++ for years. Along the way, you'll find answers to questions like

- What's worth standardizing--and what isn't?
- What are the best ways to code for scalability?
- What are the elements of a rational error handling policy?
- How (and why) do you avoid unnecessary initialization, cyclic, and definitional dependencies?
- When (and how) should you use static and dynamic polymorphism together?
- How do you practice "safe" overriding?
- When should you provide a no-fail swap?
- Why and how should you prevent exceptions from propagating across module boundaries?
- Why shouldn't you write namespace declarations or directives in a header file?

- Why should you use STL vector and string instead of arrays?
- How do you choose the right STL search or sort algorithm?
- What rules should you follow to ensure type-safe code?

Whether you're working alone or with others, ***C++ Coding Standards*** will help you write cleaner code--and write it faster, with fewer hassles and less frustration.

Download C++ Coding Standards: 101 Rules, Guidelines, and B ...pdf

Read Online C++ Coding Standards: 101 Rules, Guidelines, and ...pdf

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices

By Herb Sutter, Andrei Alexandrescu

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu

Consistent, high-quality coding standards improve software quality, reduce time-to-market, promote teamwork, eliminate time wasted on inconsequential matters, and simplify maintenance. Now, two of the world's most respected C++ experts distill the rich collective experience of the global C++ community into a set of coding standards that every developer and development team can understand and use as a basis for their own coding standards.

The authors cover virtually every facet of C++ programming: design and coding style, functions, operators, class design, inheritance, construction/destruction, copying, assignment, namespaces, modules, templates, genericity, exceptions, STL containers and algorithms, and more. Each standard is described concisely, with practical examples. From type definition to error handling, this book presents C++ best practices, including some that have only recently been identified and standardized-techniques you may not know even if you've used C++ for years. Along the way, you'll find answers to questions like

- What's worth standardizing--and what isn't?
- What are the best ways to code for scalability?
- What are the elements of a rational error handling policy?
- How (and why) do you avoid unnecessary initialization, cyclic, and definitional dependencies?
- When (and how) should you use static and dynamic polymorphism together?
- How do you practice "safe" overriding?
- When should you provide a no-fail swap?
- Why and how should you prevent exceptions from propagating across module boundaries?
- Why shouldn't you write namespace declarations or directives in a header file?
- Why should you use STL vector and string instead of arrays?
- How do you choose the right STL search or sort algorithm?
- What rules should you follow to ensure type-safe code?

Whether you're working alone or with others, ***C++ Coding Standards*** will help you write cleaner code--and write it faster, with fewer hassles and less frustration.

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu Bibliography

- Sales Rank: #115755 in Books
- Published on: 2004-11-04
- Original language: English
- Number of items: 1
- Dimensions: 9.10" h x .60" w x 7.30" l, 1.06 pounds
- Binding: Paperback
- 240 pages

Download C++ Coding Standards: 101 Rules, Guidelines, and B ...pdf

Read Online C++ Coding Standards: 101 Rules, Guidelines, and ...pdf

Download and Read Free Online C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu

Editorial Review

From the Back Cover

Consistent, high-quality coding standards improve software quality, reduce time-to-market, promote teamwork, eliminate time wasted on inconsequential matters, and simplify maintenance. Now, two of the world's most respected C++ experts distill the rich collective experience of the global C++ community into a set of coding standards that every developer and development team can understand and use as a basis for their own coding standards.

The authors cover virtually every facet of C++ programming: design and coding style, functions, operators, class design, inheritance, construction/destruction, copying, assignment, namespaces, modules, templates, genericity, exceptions, STL containers and algorithms, and more. Each standard is described concisely, with practical examples. From type definition to error handling, this book presents C++ best practices, including some that have only recently been identified and standardized-techniques you may not know even if you've used C++ for years. Along the way, you'll find answers to questions like What's worth standardizing--and what isn't? What are the best ways to code for scalability?What are the elements of a rational error handling policy? How (and why) do you avoid unnecessary initialization, cyclic, and definitional dependencies?When (and how) should you use static and dynamic polymorphism together?How do you practice "safe" overriding?When should you provide a no-fail swap? Why and how should you prevent exceptions from propagating across module boundaries?Why shouldn't you write namespace declarations or directives in a header file?Why should you use STL vector and string instead of arrays?How do you choose the right STL search or sort algorithm?What rules should you follow to ensure type-safe code?

Whether you're working alone or with others, "C++ Coding Standards" will help you write cleaner code--and write it faster, with fewer hassles and less frustration.

About the Author

Herb Sutter is the author of three highly acclaimed books, *Exceptional C++ Style*, *Exceptional C++*, and *More Exceptional C++* (Addison-Wesley). He chairs the ISO C++ standards committee, and is contributing editor and columnist for C/C++ Users Journal. As a software architect for Microsoft, Sutter leads the design of C++ language extensions for .NET programming.

Andrei Alexandrescu is the author of the award-winning book *Modern C++ Design* (Addison-Wesley, 2001) and is a columnist for *C/C++ Users Journal*.

Excerpt. © Reprinted by permission. All rights reserved.

Get into a rut early: Do the same process the same way. Accumulate idioms. Standardize. The only difference(!) between Shakespeare and you was the size of his idiom list-not the size of his vocabulary.

--Alan Perlis emphasis ours

The best thing about standards is that there are so many to choose from.

--Variously attributed

We want to provide this book as a basis for your team's coding standards for two principal reasons:

A coding standard should reflect the community's best tried-and-true experience: It should contain proven idioms based on experience and solid understanding of the language. In particular, a coding standard should be based firmly on the extensive and rich software development literature, bringing together rules, guidelines, and best practices that would otherwise be left scattered throughout many sources.

Nature abhors a vacuum: If you don't consciously set out reasonable rules, usually someone else will try to push their own set of pet rules instead. A coding standard made that way usually has all of the least desirable properties of a coding standard; for example, many such standards try to enforce a minimalistic C style use of C++.

Many bad coding standards have been set by people who don't understand the language well, don't understand software development well, or try to legislate too much. A bad coding standard quickly loses credibility and at best even its valid guidelines are liable to be ignored by disenchanted programmers who dislike or disagree with its poorer guidelines. That's "at best"--at worst, a bad standard might actually be enforced.

How to Use This Book

Think. Do follow good guidelines conscientiously; but don't follow them blindly. In this book's Items, note the Exceptions clarifying the less common situations where the guidance may not apply. No set of guidelines, however good (and we think these ones are), should try to be a substitute for thinking.

Each development team is responsible for setting its own standards, and for setting them responsibly. That includes your team. If you are a team lead, involve your team members in setting the team's standards; people are more likely to follow standards they view as their own than they are to follow a bunch of rules they feel are being thrust upon them.

This book is designed to be used as a basis for, and to be included by reference in, your team's coding standards. It is not intended to be the Last Word in coding standards, because your team will have additional guidelines appropriate to your particular group or task, and you should feel free to add those to these Items. But we hope that this book will save you some of the work of (re)developing your own, by documenting and referencing widely-accepted and authoritative practices that apply nearly universally (with Exceptions as noted), and so help increase the quality and consistency of the coding standards you use.

Have your team read these guidelines with their rationales (i.e., the whole book, and selected Items' References to other books and papers as needed), and decide if there are any that your team simply can't live with (e.g., because of some situation unique to your project). Then commit to the rest. Once adopted, the team's coding standards should not be violated except after consulting with the whole team.

Finally, periodically review your guidelines as a team to include practical experience and feedback from real use.

Coding Standards and You

Good coding standards can offer many interrelated advantages:

- *Improved code quality:* Encouraging developers to do the right things in a consistent way directly works to improve software quality and maintainability.
- *Improved development speed:* Developers don't need to always make decisions starting from first principles.
- *Better teamwork:* They help reduce needless debates on inconsequential issues and make it easier for teammates to read and maintain each other's code.
- *Uniformity in the right dimension:* This frees developers to be creative in directions that matter.

Under stress and time pressure, people do what they've been trained to do. They fall back on habit. That's why ER units in hospitals employ experienced, trained personnel; even knowledgeable beginners would panic.

As software developers, we routinely face enormous pressure to deliver tomorrow's software yesterday. Under schedule pressure, we do what we are trained to do and are used to doing. Sloppy programmers who in normal times don't know good practices of software engineering (or aren't used to applying them) will write even sloppier and buggier code when pressure is on. Conversely, programmers who form good habits and practice them regularly will keep themselves organized and deliver quality code, fast.

The coding standards introduced by this book are a collection of guidelines for writing high-quality C++ code. They are the distilled conclusions of a rich collective experience of the C++ community. Much of this body of knowledge has only been available in bits and pieces spread throughout books, or as word-of-mouth wisdom. This book's intent is to collect that knowledge into a collection of rules that is terse, justified, and easy to understand and follow.

Of course, one can write bad code even with the best coding standards. The same is true of any language, process, or methodology. A good set of coding standards fosters good habits and discipline that transcend mere rules. That foundation, once acquired, opens the door to higher levels. There's no shortcut; you have to develop vocabulary and grammar before writing poetry. We just hope to make that easier.

We address this book to C++ programmers of all levels:

If you are an apprentice programmer, we hope you will find the rules and their rationale helpful in understanding what styles and idioms C++ supports most naturally. We provide a concise rationale and discussion for each rule and guideline to encourage you to rely on understanding, not just rote memorization.

For the intermediate or advanced programmer, we have worked hard to provide a detailed list of precise references for each rule. This way, you can do further research into the rule's roots in C++'s type system, grammar, and object model. At any rate, it is very likely that you work in a team on a complex project. Here is where coding standards really pay off-you can use them to bring the team to a common level and provide a basis for code reviews.

About This Book

We have set out the following design goals for this book:

Short is better than long: Huge coding standards tend to be ignored; short ones get read and used. Long Items tend to be skimmed; short ones get read and used.

Each Item must be noncontroversial: This book exists to document widely agreed upon standards, not to

invent them. If a guideline is not appropriate in all cases, it will be presented that way (e.g., "Consider X..." instead of "Do X...") and we will note commonly accepted exceptions.

Each Item must be authoritative: The guidelines in this book are backed up by references to existing published works. This book is intended to also provide an index into the C++ literature.

Each Item must need saying: We chose not to define new guidelines for things that you'll do anyway, that are already enforced or detected by the compiler, or that are already covered under other Items.

Example: "Don't return a pointer/reference to an automatic variable" is a good guideline, but we chose not to include it in this book because all of the compilers we tried already emit a warning for this, and so the issue is already covered under the broader Item 1, "Compile cleanly at high warning levels."

Example: "Use an editor (or compiler, or debugger)" is a good guideline, but of course you'll use those tools anyway without being told; instead, we spend two of our first four Items on "Use an automated build system" and "Use a version control system."

Example: "Don't abuse goto" is a great Item, but in our experience programmers universally know this, and it doesn't need saying any more.

Each Item is laid out as follows:

- *Item title:* The simplest meaningful sound bite we could come up with as a mnemonic for the rule.
- *Summary:* The most essential points, briefly stated.
- *Discussion:* An extended explanation of the guideline. This often includes brief rationale, but remember that the bulk of the rationale is intentionally left in the References.
- *Examples (if applicable):* Examples that demonstrate a rule or make it memorable.
- *Exceptions (if applicable):* Any (and usually rare) cases when a rule doesn't apply. But beware the trap of being too quick to think: "Oh, I'm special; this doesn't apply in my situation"-that rationalization is common, and commonly wrong.
- *References:* See these parts of the C++ literature for the full details and analysis.

In each section, we chose to nominate a "most valuable Item." Often, it's the first Item in a section, because we tried to put important Items up front in each part; but other times an important Item couldn't be put up front, for flow or readability reasons, and we felt the need to call it out for special attention in this way.

Users Review

From reader reviews:

Dedra Clark:

Inside other case, little folks like to read book C++ Coding Standards: 101 Rules, Guidelines, and Best Practices. You can choose the best book if you like reading a book. Provided that we know about how is important a new book C++ Coding Standards: 101 Rules, Guidelines, and Best Practices. You can add expertise and of course you can around the world by way of a book. Absolutely right, because from book you can recognize everything! From your country right up until foreign or abroad you may be known. About

simple matter until wonderful thing you are able to know that. In this era, we can easily open a book or searching by internet unit. It is called e-book. You may use it when you feel uninterested to go to the library. Let's study.

Brian Crowe:

Hey guys, do you would like to finds a new book to learn? May be the book with the title C++ Coding Standards: 101 Rules, Guidelines, and Best Practices suitable to you? The particular book was written by well-known writer in this era. The actual book untitled C++ Coding Standards: 101 Rules, Guidelines, and Best Practicesis one of several books this everyone read now. This particular book was inspired many men and women in the world. When you read this e-book you will enter the new shape that you ever know prior to. The author explained their idea in the simple way, therefore all of people can easily to be aware of the core of this book. This book will give you a great deal of information about this world now. So that you can see the represented of the world with this book.

Anna Bailey:

Guide is one of source of information. We can add our expertise from it. Not only for students but also native or citizen have to have book to know the revise information of year to year. As we know those books have many advantages. Beside all of us add our knowledge, can also bring us to around the world. Through the book C++ Coding Standards: 101 Rules, Guidelines, and Best Practices we can take more advantage. Don't you to definitely be creative people? To become creative person must like to read a book. Simply choose the best book that ideal with your aim. Don't end up being doubt to change your life at this time book C++ Coding Standards: 101 Rules, Guidelines, and Best Practices. You can more desirable than now.

Lola Behrendt:

A lot of people said that they feel weary when they reading a guide. They are directly felt it when they get a half regions of the book. You can choose the actual book C++ Coding Standards: 101 Rules, Guidelines, and Best Practices to make your own reading is interesting. Your own personal skill of reading expertise is developing when you such as reading. Try to choose easy book to make you enjoy to read it and mingle the opinion about book and looking at especially. It is to be initial opinion for you to like to available a book and learn it. Beside that the e-book C++ Coding Standards: 101 Rules, Guidelines, and Best Practices can to be your friend when you're experience alone and confuse with the information must you're doing of that time.

Download and Read Online C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu #U1QAZIY6SOK

Read C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu for online ebook

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu books to read online.

Online C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu ebook PDF download

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu Doc

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu Mobipocket

C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu EPub

U1QAZIY6SOK: C++ Coding Standards: 101 Rules, Guidelines, and Best Practices By Herb Sutter, Andrei Alexandrescu