

Checks based on the book "Effective C++ Third Edition: 55 Specific Ways to Improve Your Programs and Designs" by Scott Meyers

Amazon.com Purchase link: <http://amzn.com/0321334876>

"Every C++ professional needs a copy of *Effective C++*. It is an absolute must-read for anyone thinking of doing serious C++ development. If you've never read *Effective C++* and you think you know everything about C++, think again."

— **Steve Schirripa**, Software Engineer, Google

"C++ and the C++ community have grown up in the last fifteen years, and the third edition of *Effective C++* reflects this. The clear and precise style of the book is evidence of Scott's deep insight and distinctive ability to impart knowledge."

— **Gerhard Kreuzer**, Research and Development Engineer, Siemens AG

The first two editions of *Effective C++* were embraced by hundreds of thousands of programmers worldwide. The reason is clear: Scott Meyers' practical approach to C++ describes the rules of thumb used by the experts — the things they almost always do or almost always avoid doing — to produce clear, correct, efficient code.

The book is organized around 55 specific guidelines, each of which describes a way to write better C++. Each is backed by concrete examples. For this third edition, more than half the content is new, including added chapters on managing resources and using templates. Topics from the second edition have been extensively revised to reflect modern design considerations, including exceptions, design patterns, and multithreading.

Important features of *Effective C++* include:

- Expert guidance on the design of effective classes, functions, templates, and inheritance hierarchies.
- Applications of new "TR1" standard library functionality, along with comparisons to existing standard library components.
- Insights into differences between C++ and other languages (e.g., Java, C#, C) that help developers from those languages assimilate "the C++ way" of doing things.

From the Back Cover

"Every C++ professional needs a copy of *Effective C++*. It is an absolute must-read for anyone thinking of doing serious C++ development. If you've never read *Effective C++* and you think you know everything about C++, think again."

— **Steve Schirripa**, Software Engineer, Google "C++ and the C++ community have grown up in the last fifteen years, and the third edition of *Effective C++* reflects this. The clear and precise style of the book is evidence of Scott's deep insight and distinctive ability to impart knowledge."

— **Gerhard Kreuzer**, Research and Development Engineer, Siemens AG

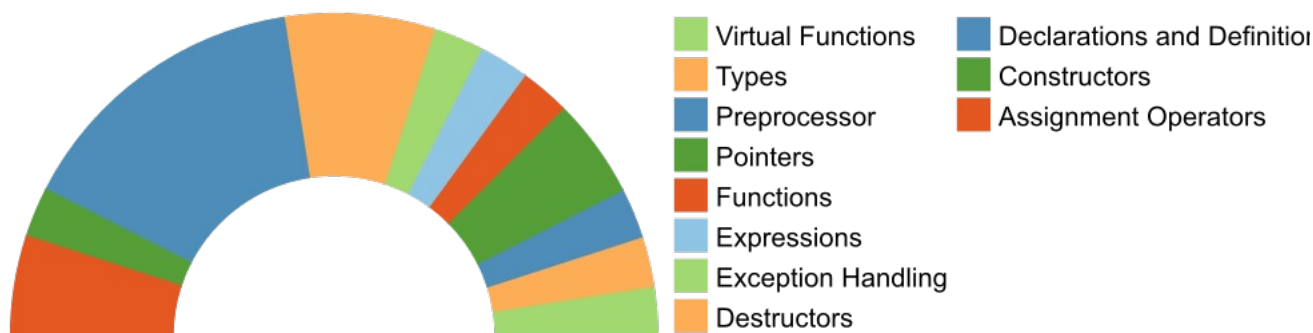
The first two editions of *Effective C++* were embraced by hundreds of thousands of programmers worldwide. The reason is clear: Scott Meyers' practical approach to C++ describes the rules of thumb used by the experts — the things they almost always do or almost always avoid doing — to produce clear, correct, efficient code.

The book is organized around 55 specific guidelines, each of which describes a way to write better C++. Each is backed by concrete examples. For this third edition, more than half the content is new, including added chapters on managing resources and using templates. Topics from the second edition have been extensively revised to reflect modern design considerations, including exceptions, design patterns, and multithreading.

Important features of *Effective C++* include:

- Expert guidance on the design of effective classes, functions, templates, and inheritance hierarchies.
- Applications of new "TR1" standard library functionality, along with comparisons to existing standard library components.
- Insights into differences between C++ and other languages (e.g., Java, C#, C) that help developers from those languages assimilate "the C++ way" of doing things.

Checks by Tags



Checks

Check ID	Check Name	Supported
EFFECTIVECPP_02	2. Do Not Use #define	Yes
EFFECTIVECPP_03	3. Use Const whenever possible	Yes
EFFECTIVECPP_04	4. Make sure that objects are initialized before they are used	Yes
EFFECTIVECPP_07	7. Non-Virtual Destructors in Base Classes	Yes
EFFECTIVECPP_08	8. Exceptions in Destructors	Yes
EFFECTIVECPP_09	9. Virtual Call in Constructor/Destructor	Yes
EFFECTIVECPP_10	10. Assignment Operator Return This	Yes
EFFECTIVECPP_11	11. Assignment Operator Self Assignment	Yes
EFFECTIVECPP_16	16. Use the same form in corresponding uses of new and delete	Yes
EFFECTIVECPP_17	17. Store newed objects in smart pointers in standalone statements	Yes
EFFECTIVECPP_20	20. Prefer pass-by-reference-to-const to pass by value	Yes
EFFECTIVECPP_22	22. Datamembers should be declared private	Yes
EFFECTIVECPP_26	26. Postpone variable definitions as long as possible	Yes
EFFECTIVECPP_27	27. Minimize casting	Yes
EFFECTIVECPP_33	33. Avoid hiding inherited names	Yes
EFFECTIVECPP_35	35. Consider alternatives to virtual functions	Yes
EFFECTIVECPP_36	36. Never redefine an inherited non-virtual function	Yes