华中科技大学出版社
http://press.hust.edu.cn
Pearson Education (培生教育出版集团)

Effective C++ 中文版

2nd Edition

50 Specific Ways to Improve Your Programs and Designs
改善程序技术与设计思维的50个有效做法

Scott Meyers 著
侯 捷 译

Conforms to the new ISO/ANSI C++ standard!

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Effective C++中文版

2nd Edition

改善程序技术与设计思维的50个有效做法

Scott Meyers 著

侯 捷 译

华中科技大学出版社

中国·武汉

Effective C++中文版

Effective C++

Scott Meyers

图书在版编目(CIP)数据

Effective C++中文版/(美)Scott Meyers 著；侯捷 译

武汉：华中科技大学出版社，2001年9月

ISBN 7-5609-2525-1

Ⅰ.E…

Ⅱ.①S… ②侯…

Ⅲ.面向对象-语言，C++

Ⅳ.TP312

责任编辑：周 筠 孟 岩

出版发行：华中科技大学出版社 （武昌喻家山 邮编：430074）

http://press.hust.edu.cn

经 销：新华书店湖北发行所

录 排：华中科技大学惠友科技文印中心

印 刷：湖北新华印务有限公司

开 本：787mm×1092mm 1/16 印 张：17.75 插 页：4 字 数：300 000

版 次：2001年9月第1版 印 次：2002年3月第3次印刷

印 数：16 101—24 000 定 价：49.80元

ISBN 7-5609-2525-1/TP·436

For Nancy,
without whom nothing
would be much worth doing.

Wisdom and beauty form a very rare combination.

—— Petronius Arbiter
Satyricon, XCIV

献给每一位对 C++/OOP 有所渴望的人

正确的观念　重于一切

– 侯 捷 –

引介：一本绝妙好书

（孟 岩）

您手上这本书，是世界顶级C++大师Scott Meyers成名之作的第二版。其第一版诞生于1991年。在国际上，本书所引起的反响之大，波及整个计算机技术出版领域，余音至今未绝。几乎在所有C++书籍的推荐名单上，本书都会位于前三名。作者高超的技术把握力、独特的视角、诙谐轻松的写作风格、 独具匠心的内容组织，都受到极大的推崇和仿效。甚至连本书简洁明快的命名风格，也有着一种特殊的号召力。我可以轻易列举出一大堆类似的名字，比如Meyers本人的More Effective C++和Effective STL，Don Box的Essential COM，Stan Lippman主编的Efficient C++系列，Herb Sutter的Exceptional C++等等。要知道，这可不是出版社的有意安排，而且上面这些作者，同样是各自领域里的绝顶大师，决非人云亦云、欺世盗名之辈。这种奇特的现象，只能解释为人们对这本书衷心的赞美和推崇。

然而这样一本掷地有声的C++世界名著，不仅迟迟未能出版简体中文版，而且在国内其声誉似乎也并不显赫。可以说在一年之前，甚至很少有C++的学习者听说过这本书，这实在是一种遗憾。今天，在很多人的辛勤努力之下，这本书终于能够展现在我们的面前，对于真正的C++程序员来说，这确实是一件值得弹冠相庆的事。

我是一名普通的C++爱好者，因为机缘巧合，有幸参与了这本书的繁简转译工作，这使我能够比较早地看到本书的原版和繁体中文版。在这里我必须表达对本书中文译者、台湾著名技术作家侯捷先生的敬意和感谢。因为在我看来，这本书的中文版在质量上较其英文版兄长分毫不差，任何人都知道，达到这一点是多么地不容易。侯先生以其深厚的技术功底、卓越的语言能力和严谨细致的治学态度，为我们跨越了语言隔阂所带来的理解障碍，完整而生动地将原书的内容与精神表达无遗。更令人钦佩的是，中文版的行文风格与原文也达到了高度的统一，可谓神形兼备，实在令人赞叹。因此，我非常乐意向大家推荐这本书，相信它会在带给您技术享受的同时，也带给您阅读的享受。

在转译的过程中，对于大陆和台湾两地技术术语的差异，侯先生与我做了许多考虑。您在书中可能会发现一些术语并不符合自己的习惯，这些都是我们经过反复思考之后保留下来的译法，大部分都有充分的理由。比如把 type 翻译成“型别”而不是“类型”，是因为“类”在 C++中实在是一个太过敏感的字眼；再比如 instance 没有按常规译为“实例”，是因为这种译法并不符合面向对象思想的本意。技术翻译毕竟还要讲求技术上的严谨和准确，所以我做出了一些抉择。究竟效果如何，还得要由广大的读者评价。我欢迎这方面的讨论。下面是一份关于术语的整理表格：

英文术语	大陆惯用译法	本书译法
adapter	适配器	配接器
argument	实参（实质参数）	引数
by reference	传参考,传地址	传址
by value	传值	传值
dereference	反引用,解参考	提领
evaluate	评估,计算	评估，核定
instance	案例,实例	实体
instantiated	实例化	实体化、具现化
library	库,函数库	程序库
range	范围	区间（使用于 STL 时）
resolve	解析	决议
parameter	形参（形式参数）	参数
type	类型	型别

曾经在网络讨论组中看到这样的说法：C++程序员可以分成两类，读过 Effective C++的和没读过的。或许有点夸张了，但无论如何，当您拥有这本书之后，就获得了迅速提升自己 C++功力的一个契机。这本书不是读完一遍就可以束之高阁的快餐读物，也不是能够立刻解决手边问题的参考手册，而是需要您去反复阅读体会，极力融入自己的思想之中，融入自己每一次敲击键盘的动作之中。C++是真正程序员的语言，它的背后有着精深的思想与无以伦比的表达能力，这使得它具有类似宗教般的魅力。希望这本书能够帮助您跨越 C++的重重险阻，领略高处才有的壮美，做一个成功而快乐的 C++程序员！

孟岩 2001/07/20 于北京

译序（侯捷）

C++ 是一个难学易用的语言！

C++ 的难学，不仅在其广博的语法，以及语法背后的语意，以及语意背后的深层思维，以及深层思维背后的对象模型；C++ 的难学，还在于它提供了四种不同（但相辅相成）的程序设计思维模式：procedural-based, object-based, object-oriented, generic paradigm。

世上没有白吃的午餐。又要有效率，又要有弹性；又要前瞻望远，又要回溯兼容；又要能治大国，又要能烹小鲜，学习起来当然就不可能太简单。

在如此庞大复杂的机制下，万千使用者前仆后继的动力是：一旦学成，妙用无穷。

C++ 相关书籍之多，车载斗量；如天上繁星，如过江之鲫。广博如四库全书者有之（*The C++ Programming Language*、*C++ Primer*）；深奥如重山复水者有之（The *Annotated C++ Reference Manual*, *Inside the C++ Object Model*）；细说历史者有之（*The Design and Evolution of C++*, *Ruminations on C++*）；独沽一味者有之（*Polymorphism in C++*, *Genericity in C++*）；独树一帜者有之（*Design Patterns*, *Large Scale C++ Software Design*, *C++ FAQs*）；程序库大全者有之（*The C++ Standard Library*）；另辟蹊径者有之（*Generic Programming and the STL*）；工程经验之累积者亦有之（*Effective C++*, *More Effective C++*, *Exceptional C++*）。

这其中，“工程经验之累积”对已具 C++ 相当基础的程序员而言，有着非凡的吸引力与立竿见影的效果。Scott Meyers 的 *Effective C++* 和 *More Effective C++* 是此类的佼佼者，Herb Sutter 的 *Exceptional C++* 则是后起之秀。

这类书籍的一个共同特色是轻薄短小，并且高密度地纳入作者浸淫于C++/OOP 领域多年而广泛的丰富经验。它们不但开拓读者的视野，也为读者提供各种 C++/OOP 常见问题或易犯错误的解决模型。某些小范围主题诸如“在 base classes 中使用 virtual destructor”、“令 operator= 传回 *this 的 reference”，可能在百科型 C++ 语言书籍中亦曾概略提过，但此类书籍以深度探索的方式，让我们了解问题背后的成因、最佳的解法，以及其它可能的牵扯。至于大范围主题，例如 smart pointers, reference counting, proxy classes, double dispatching, 基本上已属 design patterns 的层级！

这些都是经验的累积和心血的结晶！

我很高兴将以下三本极佳书籍，规划为一个系列，以郑重的态度推荐给您：

1. *Effective* C++ 2/e, by Scott Meyers, AW 1998
2. *More Effective* C++, by Scott Meyers, AW 1996
3. *Exceptional* C++, by Herb Sutter, AW 1999

不论外装或内容，中文版比其英文版兄弟毫不逊色。本书不但与原文本页页对译，保留索引，并加上译注、书籍交叉参考[1]、完整范例代码[2]、读者服务[3]。

这套书对于您的程序设计生涯，可带来重大帮助。翻译这套书籍（编注：后两本为繁体中文版）使我感觉非常快乐。我祈盼（并相信）您在阅读此书时拥有同样的心情。

侯捷 2000/05/15 于新竹.台湾
jjhou@ccca.nctu.edu.tw
http://www.jjhou.com

[1] *Effective* C++ 2/e 和 *More Effective* C++ 之中译，事实上是以 Scott Meyers 的另一个产品 *Effective* C++ *CD* 为本，不仅资料更新，同时亦将 CD 版中两书之交叉参考保留下来。这可为读者带来旁征博引时的莫大帮助。

[2] 书中程序多为片段。我将陆续完成完整的范例程序，并在 Visual C++,C++Builder, GNU C++ 上测试。请至侯捷网站（http://www.jjhou.com）下载。

[3] 欢迎读者对本书范围所及的主题提出讨论，并感谢读者对本书的任何误失提出指正。来信请寄侯捷电子信箱（jjhou@jjhou.com）。

前言

Preface

这本书是多年来我对专业程序员所进行的 C++ 课程教学的一个附产物。我发现，大部分学生在一个星期的密集训练之后，即可适应这个语言的基本架构，但要他们“将这些基础架构以有效的方式组合运用”，我实在不感乐观。于是我开始尝试组织出一些简短、明确、容易记忆的准则，作为 C++ 高效性程序开发过程之用。那都是经验丰富的 C++ 程序员几乎总是会奉行或几乎肯定要避免的一些事情。

我最初的兴趣在于整理出一些可被某种“lint-like 程序”施行的规则，最后我甚至领导一个计划，研究某种可将 C++ 原始代码中违反用户指定条件之处检验出来的工具[1]。不幸的是在我尚未完成其完整原型之前，这个研究计划便结束了。幸运的是，目前市面上已有这类 C++ 检验工具（商品），而且不只一个。

虽然我最初的兴趣是在研究可被（某种工具）自动实施的程序设计准则，但我很快了解到那个研究方向的局限性。优秀的 C++程序员所奉行的准则，多数都难以“公式化”；要不就是虽然它们有许多重要的例外情况，却被程序员盲目地奉行不渝。这使我念头一转：某些东西虽然不比计算机程序精准，但仍能比一本泛泛的 C++教科书更集中火力，更能抓住重点。这个念头的结果就是你手上的这本书：一本内含 50 个有效建议（如何改善你的 C++程序技术和你的设计思维）的书。

在这本书中，你会发现一些忠告，告诉你应该做些什么，为什么如此；告诉

[1] 读者可以在 Effective C++ 网站上找到此研究的一份概要报告。

你不应该做些什么，又为什么如此。基本而言，当然 whys 比 whats 更重要，但检阅一个个准则，也确实比强记一本或两本教科书更轻松、更方便得多。

和大部分的 C++ 书籍不同，我的组织方式并非以语言特性作为依据。也就是说，我并不在某处集中讨论 constructors（构造函数），在另一处集中讨论 virtual functions（虚拟函数），又在第三个地方集中讨论 inheritance（继承关系）。不，不是这样，本书的每一个讨论主题都剪裁合度地以一个个准则陈列出来。至于我对某个特定语言性质的探讨，散布面积可能涵盖整本书。

这种做法的优点就是比较容易反映出“特意挑选 C++ 作为开发工具”的那些软件系统的复杂度。在那些系统之中，仅只了解个别语言特性是不够的。例如，有经验的 C++ 程序员知道，了解 inline 函数和了解 virtual destructors，并不一定表示你了解 inline virtual destructors。身经百战的开发人员都认识到，理解 C++ 各个特性之间的互动关系，才是有效使用这个语言最重要的关键之处。本书的组织结构反映出了这一基本事实。

这种做法的缺点是，你恐怕必须前后交叉参考而非只看一个地方，才能发现我所说的某个 C++ 架构的全貌。为了将不方便性降至最低，我在书中各处放了许多交叉索引，书后并有一份涵盖全部范围的索引。（**译注**：为了协助读者更容易掌握 *Effective C++* 和 *More Effective C++* 二书，我以 *Effective C++ CD* 为范本，为两书的中文版额外加上两书之间的交叉索引。此乃原书所无。如果文中出现像条款 **M5** 这样的参考指示，**M** 便是代表 *More Effective C++*）

筹划第二版期间，我改写此书的雄心一再被恐惧所取代。成千上万的程序员热情拥抱 *Effective C++* 第一版，我不希望破坏吸引他们的任何东西。但是自从我写了第一版之后，六年过去了，C++ 有了变化，C++ 程序库有了变化（见条款 49），我对 C++ 的了解也有了变化，乃至于 C++ 的用途也有了变化。许许多多的变化。对我而言，重要的是我必须修订 *Effective C++* 以反映那些变化。我尝试一页一页地修改内容，但是书籍和软件十分类似，局部加强是不够的，唯一的机会就是系统化地重写。本书就是重写后的结果：*Effective C++* 2.0 版。

熟悉第一版的读者可能有兴趣知道，书中的每一项条款都经过重新检验。然

而我相信第一版的结构至今仍是流畅的，所以整本书的结构并没有改变。50 项条款中，我保留了 48 项，其中某些标题稍有变化（附随的讨论内容亦复如此）。被取代的两个条款是 32 和 49，不过原条款 32 的许多信息被我移到如今焕然一新的条款 1 中。我将条款 41 和 42 的次序做了对调，因为这样更能够适当地呈现它们修订后的内容。最后，我把上一版继承体系图所采用的箭头方向颠倒过来，以符合目前几乎已经一致的习惯：从 derived classes 指往 base classes。我的 *More Effective C++* 一书也采用相同习惯（本书最后列有该书摘要）。

本书提供的准则，离巨细靡遗的程度还很远，但是完成一个好的准则 —— 一个几乎可于任何时间应用于任何程序的准则，动手远比动嘴困难得多。如果你知道其它准则，可以协助撰写有效的 C++ 程序，我非常乐意听到你告诉我它们的故事。

此外，说不定你会觉得本书的某些条款不适合成为一般性忠告；或许你认为另有比较好的方法来完成书中所说的任务；或许你认为某些条款在技术讨论方面不够清楚，不够完全，抑或有误导之嫌。我衷心盼望你也能够让我知道你的这些想法。

Donald Knuth（译注：经典书籍 *The Art of Computer Programming*, Volume I,II,III 的作者）长久以来为挑出其书错误的热心读者准备有一份小小的报酬。这个故事传为美谈。追求完美的精神令人佩服。看过那么多仓促上市错误累累的 C++ 书籍后，我更是特别强烈地希望追随 Knuth 的风范。因此，如果有人挑出本书的任何错误并告诉我 —— 不论是技术、文法、错别字或任何其它东西 —— 我将在本书重新印刷的时候，把第一位挑出错误的读者大名加到致谢名单中。

请将你的建议、你的见解、你的批评，以及(喔，真糟……)你的臭虫报告，寄至：

```
Scott Meyers
c/o Publisher, Corporate and Professional Publishing
Addiscn Wesley Longman, Inc.
1 Jaccb Way
Reading, MA 01867
U. S. A.
```

或者传送电子邮件到 ec++@awl.com。

我维护有本书第一次印刷以来的修订记录，其中包括错误更正、文字修润，以及技术更新。你可以从 Effective C++ 网站取得这份记录。如果你希望拥有这份资料，但无法上网，请寄申请函到上述地址，我会邮寄一份给你。

Scott Douglas Meyers

Stafford, Oregon
July 1997

致谢

中文版 略

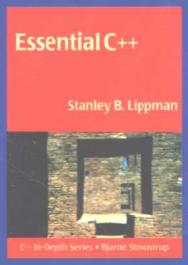
译注：藉此版面提醒读者，本书之中如果出现“条款 5”这样的参考指示，指的是本书条款 5。如果出现“条款 **M**5”这样的参考指示，**M** 是指 *More Effective C++*。

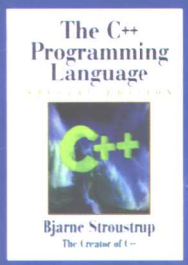
C++/OOP/OOD 经典好书（1）

以下为 C++/OOP/OOD 世界级经典好书。循序渐进，可收宏效。

Essential C++

旁枝暂略、主攻核心。轻薄短小，附习题与解答。适合初学者。



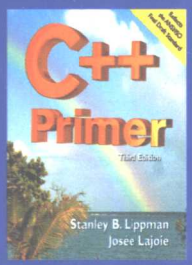


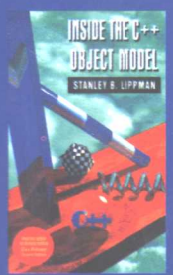
The C++ Programming Language

技术权威，用词深峻、思想深远。C++ 百科全书代表。

C++ Primer

纵横书市十数年不坠，内容巨细靡遗，被誉为 C++ 最佳教本。C++ 百科全书代表。



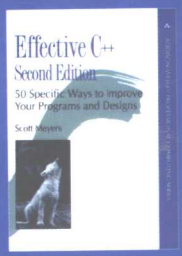


Inside The C++ Object Model

揭示C++ 的底层机制，揭开C++ 背后的重重布幔。（注意：语言机制的底层实现细节因不同的编译器而略有不同）

C++/OOP/OOD 经典好书（2）

以下为 C++/OOP/OOD 世界级经典好书。循序渐进，可收宏效。

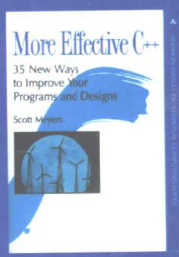


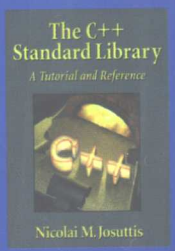
Effective C++

通过50个编程准则，向你灌输专家经验。行文幽默，深入浅出。

More Effective C++

通过35个编程准则，向你灌输专家经验。行文幽默，深入浅出。



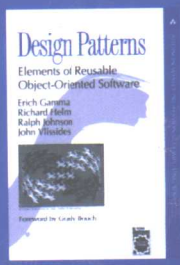


The C++ Standard Library

C++标准程序库的百科全书。对于资料的收集、整理、剖析、范例，均有极优越的表现。

Design Patterns

严谨规范23个OO 设计领域中的惯用手法。开创patterns书籍先河，影响深远。读者需具备高技术水准以及丰富的编程经验。



Effective C++ 第一版的荣耀：

我必须真心地赞美 Meyers 的书，……这本书在内存管理架构等主题上给予读者卓越的引导，并对不同形式之 C++ 继承机制有极佳的解释。

—— *New York Computerist*

在你开始着手第一个真正的 C++ 项目之前，你应该阅读本书；在你获得一些实务经验之后，你应该再读一遍。

—— comp.lang.c++

本书有一个副标题“改善你的程序技术与设计思维的 50 个有效做法”。作者不只提供你撰写 C++代码时应该遵循的明白的规则，也提供了深入的解释与范例。

—— *Sun Expert*

我郑重推荐 *Effective C++* 给任何渴望在中高阶层面精通 C++ 的人。

—— *The C User's Journal*

在各种写给中高阶程序员看的 C++ 书籍中，这是我所见过的最棒的一本。作者以一系列短文叙述 C++ 程序员遭遇的常见问题。……本书既有趣又有用，在程序设计书籍中诚属罕见。

—— comp.lang.c++

本书在取材范围和风格上类似另一本由 William Strunk 和 E.B. White 合著的小书 *The Elements of Style*；至少二者在我的书架上距离不远。……这是一本十分严谨而适用的小书，有着清楚的目标，并且完成了它们。

—— *C++ Report*

这本书内含对 C++ 开发工作的许多实用性忠告。

—— *DEC Professional*

C++ 程序员不只应该拥有这本书，而且应该切实运用这本书。书中的文字极易拿来实际运用。交叉参考与索引的功夫做得很好。

—— *Computer Language*

这是一本 193 页的杰作。……我保证 50 项条款中必定有某一些会攫取你的注意力，并对你产生启蒙作用，你的谨慎投资将获得回报。……这是一本文笔优越、真材实料的书籍，目标瞄准重视流畅与效率的所有 C++ 程序员。

—— Stan Kelley-Bootle, *UNIX Review*

目录