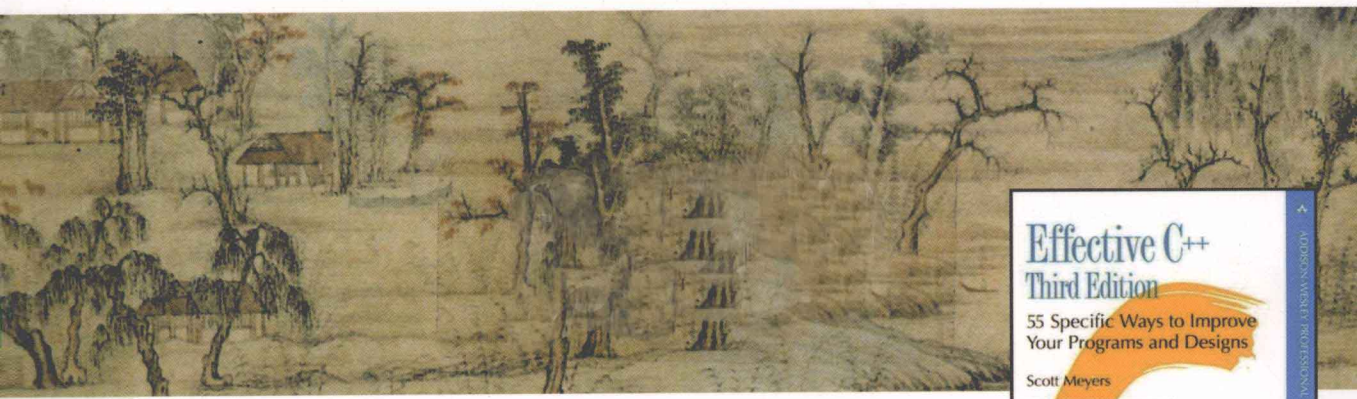
Broadview
www.broadview.com.cn
PEARSON
传世经典书丛
Eternal Classics

Effective C++ (评注版)

改善程序与设计的55个具体做法 (第三版)

[美] Scott Meyers 著

评注 云风



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

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Effective C++

改善程序与设计的55个具体做法（第三版）（评注版）

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

[美] Scott Meyers 著

云 风 评注

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

“C++程序员可以分成两类，读过 Effective C++的和没读过的。”世界顶级 C++大师 Scott Meyers 这部成名之作，与这句话一道在全球无数读者间广为传颂。几乎所有 C++书籍推荐名单上，本书都会位列三甲。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。

对于国外技术图书，选择翻译版还是影印版，常让人陷入两难。本评注版力邀国内资深专家执笔，在英文原著基础上增加中文点评与注释，旨在融合二者之长，既保留经典的原创文字与味道，又以先行者的学研心得与实践感悟，对读者阅读与学习加以点拨、指明捷径。

经过评注的版本，更值得反复阅读与体会。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风光，做一个成功而快乐的 C++程序员。

版权贸易合同登记号：图字：01-2011-0370

图书在版编目（CIP）数据

Effective C++：改善程序与设计的 55 个具体做法：第 3 版 = Effective C++：55 Specific Ways to Improve Your Programs and Designs, 3E：评注版 /（美）梅耶（Meyers,S.）著；云风评注. —3 版. —北京：电子工业出版社，2011.6

（传世经典书丛）

ISBN 978-7-121-13376-3

Ⅰ. ①E… Ⅱ. ①梅… ②云… Ⅲ. ①C 语言－程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 075128 号

责任编辑：许　艳

印　　刷：

装　　订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱　邮编 100036

开　　本：787×980　1/16　印张：21.25　字数：417.7 千字

印　　次：2011 年 6 月第 1 次印刷

定　　价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

悦读上品　得乎益友

孔子云："取乎其上，得乎其中；取乎其中，得乎其下；取乎其下，则无所得矣"。

对于读书求知而言，这句古训教我们去读好书，最好是好书中的上品——经典书。其中，科技人员要读的技术书，因为直接关乎客观是非与生产效率，阅读选材本更应慎重。然而，随着技术图书品种的日益丰富，发现经典书越来越难，尤其对于涉世尚浅的新读者，更为不易，而他们又往往是最需要阅读、提升的重要群体。

所谓经典书，或说上品，是指选材精良、内容精练、讲述生动、外延丰盈、表现手法体贴入微的读品，它们会成为读者的知识和经验库中的重要组成部分，并且拥有从不断重读中汲取养分的空间。因此，选择阅读上品的问题便成了有效阅读的首要问题。当然，这不只是效率问题，上品促成的既是对某一种技术、思想的真正理解和掌握，同时又是一种感悟或享受，是一种愉悦。

与技术本身类似，经典 IT 技术书多来自国外。深厚的积累、良好的写作氛围，使一批大师为全球技术学习者留下了璀璨的智慧瑰宝。就在那个年代即将远去之时，无须回眸，也能感受到这一部部厚重而深邃的经典著作，在造福无数读者后从未蒙尘的熠熠光辉。而这些凝结众多当今国内技术中坚美妙记忆与绝佳体验的技术图书，虽然尚在国外图书市场上大放异彩，却已逐渐淡出国人的视线。最为遗憾的是，迟迟未有可以填补空缺的新书问世。而无可替代，不正是经典书被奉为圭臬的原因？

为了不让国内读者，尤其是即将步入技术生涯的新一代读者，就此错失这些滋养过先行者们的好书，以出版 IT 精品图书，满足技术人群需求为己任的我们，愿意承担这一使命。本次机遇惠顾了我们，让我们有机会携手权威的 Pearson 公司，精心推出"传世经典书丛"。

在我们眼中，"传世经典"的价值首先在于——既适合喜爱科技图书的读者，也符合专家们挑剔的标准。幸运的是，我们的确找到了这些堪称上品的佳作。丛书带给我们的幸运颇多，细数一下吧。

得以引荐大师著作

有恐思虑不周，我们大量参考了国外权威机构和网站的评选结果，并得到了 Pearson 的专业支持，又进

一步对符合标准之图书的国内外口碑与销售情况进行细致分析，也听取了国内技术专家的宝贵建议，才有幸选出对国内读者最富有技术养分的大师上品。

向深邃的技术内涵致敬

中外技术环境存在差异，很多享誉国外的好书未必适用于国内读者；且技术与应用瞬息万变，很容易让人心生迷惘或疲于奔命。本丛书的图书遴选，注重打好思考方法与技术理念的根基，旨在帮助读者修炼内功，提升境界，将技术真正融入个人知识体系，从而可以一通百通，从容面对随时涌现的技术变化。

翻译与评注的双项选择

引进优秀外版著作，将其翻译为中文供国内读者阅读，较为有效与常见。但另有一些外语水平较高、喜好阅读原版的读者，苦于对技术理解不足，不能充分体会原文表述的精妙，需要有人指导与点拨。而一批本土技术精英经过长期经典熏陶及实践锤炼，已足以胜任这一工作。有鉴于此，本丛书在翻译版的同时推出融合英文原著与中文点评、注释的评注版，供不同志趣的读者自由选择。

承蒙国内一流译（注）者的扶持

优秀的英文原著最终转化为真正的上品，尚需跨越翻译鸿沟，外版图书的翻译质量一直屡遭国内读者诟病。评注版的增值与含金量，同样依赖于评注者的高卓才具。好在，本丛书得到了久经考验的权威译（注）者的认可和支持，首肯我们选用其佳作，或亲自参与评注工作。正是他们的参与保证了经典的品质，既再次为我们的选材把关，更提供了一流的中文表述。

期望带给读者良好的阅读体验

一本好书带给人的愉悦不止于知识收获，良好的阅读感受同样不可缺少，且对学业不无助益。为让读者收获与上品相称的体验，我们在图书装帧设计与选材用料上同样不敢轻率，惟愿送到读者手中的除了珠玑章句，还有舒适与熨帖的视觉感受。

所有参与丛书出版的人员，尽管能力有限，却无不心怀严谨之心与完美愿望。如果读者朋友能从潜心阅读这些上品中偶有获益，不啻为对我们工作的最佳褒奖。若有阅读感悟，敬请拨冗告知，以鼓励我们继续在这一道路上贡献绵薄之力。如有不周之处，也请不吝指教。

电子工业出版社博文视点

评注者序

2010 年秋，电子社编辑侠少寄给我一本 Effective C++（第三版）英文原版书，并托我为这本书写一些评注，希望做成评注版在国内出版。经慎重考虑后我受领了这一任务。

与 Effective C++的渊源

回想起来，通过更深入地理解 C++而获得一种喜悦感，已是十多年前的事情了。2000 年前后我刚刚从 C 语言迁移到 C++来做实际的项目，同一时期国内涌现出一大批 C++语言相关著作。我读了一本便一发不可收拾，几乎阅遍当时可以找到的相关书籍。从纯粹学习语言的角度来讲，Effective C++是相当重要的一本书，也是作为过来人的我最想推荐给大家的。阅读其时（当时是第二版），我已经用 C++编写过一个开源的游戏引擎，有了数万行代码的经验，书中总结的条款读来心有戚戚焉。犹记得当年是在书店驻足读完大半本书，才想起来买下来带回去读。

时光堆积的反思

之后的几年里，我用 C++编写了数十万行代码。写得越多，对之前的作品越不满意，代码风格也随之变化，而且慢慢产生怀疑，到底是否存在一种普遍合理、高效的 C++使用方法——它可以让其他程序员，或是将来的自己审阅代码后，表示一致赞赏，而不需要用无休止的重构来满足自己的完美主义倾向。其实也就是本书作者 Scott Meyers 这些年来探讨的主题。C++伴随软件工业发展这些年，修修补补。由于工程需要，它必须保持向前兼容，并尽量满足每个时期的工程需求。这注定它不可能是一门完美的语言，也注定它是一门备受争议的语言。

尽管对 C++的批评和质疑从未间断，但毋庸置疑，它绝不会销声匿迹。而且，C++到底是不是一门高效语言，并不是本书探讨的重点。本书的着眼点在于，如果你选择了 C++，如何使它变得更高效，也即如何有效使用 C++。其实，一开始我是有些好奇，大牛 Scott Meyers 这个主题写了十多年，就没有什么可抱怨的？当这次为点评而重新读到 Item 25——由 std::swap 的扩展问题（本书第 117 页）引申到特例化 std 名字空间里的方法时，一

句“Alas, the form of the prohibition may dismay you”惹我会心一笑，从此释然。

再读经典的别样感触

再读这本书的新版，速度很慢，有时还会和第二版对照一下，体会作者思想的变迁。作为参考，还重温了《C++语言的设计和演化》的几个章节。毕竟文字最终要印成铅字，不由得慎重起来，似乎从来没有如此耐心地逐字读英文句子。读到细节处，发现作者把每个问题讲得都很透，表达流畅，前后反复呼应，即使无太多英文阅读经验的人，也可以轻松读懂，不愧是写了十年的精品。十分钦佩之余，却也为无从下笔而犯难，似乎能任意发挥之处，只剩对 C++的争议。初稿发给出版社后，经编辑提醒又增加一些“帮助初学者解惑和提速”的内容，希望不致误导 C++初学者放弃学习这门有趣的语言。

不安的评注者

评注这个工作比翻译难做，尤其是评 Effective C++这样的经典。写得太多有狗尾续貂之嫌，太少又愧对读者的期望。而且，无论怎样写，都会带有特定时期个人观点的局限性。要特别提醒的是，不要把某些评注看作是对 C++的批评。作为用了多年 C++并一度沉迷其中的程序员，那只是爱之深后的责之切。

本书除了少部分评注是针对个别代码段或关键词外，尚有不少篇幅为对原书篇章、段落主题的拓展思考，可抛开原文独立阅读。限于水平，有很多地方，想表达的东西没能讲透，是一种遗憾。相对于 Scott Meyers 积累了十数年的精华章句，我这个后学晚辈仓促成文忝列其间，实在诚惶诚恐。评注中错误之处，望方家或不吝赐教，或一笑了之。

祝各位读者拥有和我一样愉快的阅读体验。

云风 2011 年春 于 杭州

序 言

1991 年我写下 *Effective C++*第一版。1997 年撰写第二版时我更新了许多重要内容，但为了不让熟悉第一版的读者感到困惑，我竭尽所能保留原始结构：原先 50 个条款中的 48 个标题基本没变。如果把书籍视为一栋房屋，第二版只是更换地毯、灯饰，重新粉刷一遍而已。

到了第三版，修缮工作进一步深入壁骨墙筋（好几次我甚至希望能够翻新地基）。1991 年起 C++世界经历了巨大变革，而本书目标——在一本小而有趣的书中确认最重要的一些 C++编程准则——却已不再能够由 15 年前建立的那些条款体现出来。“C++程序员拥有 C 背景”这句话在 1991 年是个合理假设，如今 C++程序员却很可能来自 Java 或 C#阵营。继承（inheritance）和面向对象编程（object-oriented programming）在 1991 年对大多数程序员都很新鲜，如今程序员已经建立良好概念，异常（exceptions）、模板（templates）和泛型编程（generic programming）才是需要更多引导的领域。1991 年没人听过所谓设计模式（design patterns），如今少了它很难讨论软件系统。1991 年 C++正式标准才刚要上路，如今 C++标准规范已经 8 岁，新版规范蓄势待发[†]。

为了对付这些改变，我把所有条款抹得一干二净，然后问自己“2005 年什么是对 C++

---

† C++标准在 1998 年做过修订，后来在 2003 年做了一次。之后，新的标准一直难产。在很长时间，C++社区都没能确定新标准会在什么时间制订出来，只能将其暂时称为 C++ 0x，以期可以在新世纪的第一个十年内确定下来。可惜最后到了 2010 年还争论不休，以至于大家笑谈 0x 指的是 16 进制。

值得庆幸的是，争议终于在 2011 年得到终结，我们应该能在 2011 年的夏天看到正式的 C++ 2011 标准文本。这个冗长的规范制订过程已经让 C++社区等了太久。

C++ 2011 增加了许多新东西，或许会让 C++程序员们感觉到它更像是一门新的语言。如果需要了解新标准，可以在 C++之父 Stroustrup 的个人主页上找到他维护的 C++0x FAQ 一读。我想，大部分 C++程序员都能从中找到令人兴奋的东西。

我们也期待本书可以再有一个新版本。

程序员最重要的忠告？”答案便是第三版中的这些条款。本书有两个新章，一个是资源管理（resource management），一个是模板编程（programming with templates）。实际上 template 这东西遍布全书，因为它们几乎影响了 C++的每个角落。本书新素材还包括在 exceptions 概念下编程、套用设计模式，以及运用新的 TR1 程序库设施（TR1 于条款 54 描述）。本书也告诉大家在单线程系统（single-threaded systems）中运行良好但可能不适用于多线程系统（multithreaded systems）的某些技术和做法。本书半数以上内容是新的。在此同时第二版大部分基础信息仍然很重要，所以我找出一个保留它们的办法：你可以在附录 B 找到第二、第三两版的条款对应表。

我努力让本书达到我所能够达到的最佳状态，但这并不表示它已臻完美。如果你认为某些条款不适合作为一般性忠告，或你有更好的办法完成本书所谈的某件工作，或书中某些技术讨论不够清楚不够完全，甚或有所误导，请告诉我。如果你找出任何错误——技术上的、文法上的、排版印刷上的，不论哪一种——也请告诉我。我很乐意将第一位提出问题并吸引我注意的朋友加入下次印刷的致谢名单中。

即使本书条款个数扩充为 55，这一整组编程准则还谈不上完备。然而毕竟整理出优良准则——几乎任何时间适用于任何应用程序的准则——比想象中困难得多。如果你有其他编程准则的想法或建议，我将乐以与闻。

我手上维护本书第一印以来的变化清单，其中包括错误修订、进一步说明和技术更新。这份清单放在网址为 http://aristeia.com/BookErrata/ec++3e-errata.html 的"*Effective C++ Errata*"网页上。如果你希望在这份清单更新时获得通知，请加入我的邮件列表。这份列表用来发布消息给可能对我的专业工作感兴趣的人士，详情请见 http://aristeia.com/MailingList/。

Scott Douglas Meyers
http://aristeia.com/

Stafford.Oregon
2005年4月
侯捷　译

致 谢

*Effective C++*已经面世 15 年了，我开始学习 C++ 则是在书写此书的前 5 年。也就是说 "*Effective C++*项目" 已经发展两个年代了。此期间我得益于数百（数千？）人的深刻知识、对我的建议与修正，以及偶发的一些目瞪口呆的事绩。这些人帮助我更加完善 *Effective C++*，我要对他们全体表示感谢。

我已经放弃追踪“在哪儿学到什么”的历史，但永远记得有个公众信息源源不断带给我帮助：Usenet C++ newsgroups，特别是 comp.lang.c++.moderated 和 comp.std.c++。本书许多——也许是大多数——条款得益于这些讨论群所突出的若干技术想法和后续调查与诊疗。

关于第三版新内容，Steve Dewhurst 和我一起讨论了最初的条款名单。条款 11 中关于“藉由 copy-and-swap 实现 `operator=`”的构想来自 Herb Sutter 在此主题的作品，像是 *Exceptional C++*（Addison-Wesley, 2000）条款 13。RAII（见条款 13）源自 Bjarne Stroustrup 的 *The C++ Programming Language*（Addison-Wesley, 2000）。条款 17 背后的想法来自 Boost `shared_ptr` 网页上的“Best Practices”节区（http://boost.org/libs/smart_ptr/shared_ptr.htm#BestPractices），又得到 Herb Sutter 的 *More Exceptional C++*（Addison-Wesley, 2002）条款 21 的琢磨。条款 29 强烈受到 Herb Sutter 在此主题上的广泛作品的影响，像是 *Exceptional C++*中条款 8~19，*More Exceptional C++*中条款 17~23，以及 *Exceptional C++ Style*（Addison-Wesley, 2005）条款 11~13；David Abrahams 帮助我更好地了解三个异常安全性保证。条款 35 的 NVI 手法来自 Herb Sutter 写于 *C/C++ Users Journal*2001 年 9 月份的 "Virtuality" 专栏。同一条款中的 Template Method 和 Strategy 设计模式来自 *Design Patterns*（Addison-Wesley, 1995），作者是 Erich Gamma, Richard Helm, Ralph Johnson 和 John Vlissides。条款 37 所说的 NVI 使用手法，概念来自 Hendrik Schober。David Smallberg 给了我在条款 38 写出一个定制型 `set` 实现品的动机。条款 39 提到 EBO 通常只在多重继承中才可用，这个构想源自 David Vandevoorde 和 Nicolai M. Josuttis 合著的 *C++ Templates*（Addison-Wesley, 2003）。条款 42 中我对 `typename` 的最初理解来自 Greg Comeau 主持的 "C++ and C FAQ"（http://www.comeaucomputing.com/techtalk/#typename），Leor Zolman 则帮助我认识我的最初理解并不正确（是我的错，

和 Greg 无关）。条款 46 的本质源自于 Dan Sak 的谈话，"Making New Friends"。条款 52 末尾的那个想法“如果你声明一版 `operator new`，你也应该声明其对应的 `delete` 伙伴”源自 Herb Sutter 的 *Exceptional C++ Style* 中条款 22。我从 David Abrahams 身上更多了解了 Boost 的检评过程（条款 55 有一份摘要）。

上面所说关于我向谁或从某处学习某一技术，并不必然表示谁或某处就是该技术的发明人或发表处。

我的笔记还告诉我，我也使用了来自 Steve Clamage, Antoine Trux, Timothy Knox 和 Mike Kaelbling 的信息，可惜这份笔记没有提到如何以及在哪儿学到什么。

第一版草稿由 Tom Cargill, Glenn Carroll, Tony Davis, Brian Kernighan, Jak Kirman, Doug Lea, Moises Lejter, Eugene Santos, Jr., John Shewchuk, John Stasko, Bjarne Stroustrup, Barbara Tilly 和 Nancy L. Urbano 共同检阅。我收到了一些改善建议并纳入后来印次，这些建议来自 Nancy L. Urbano, Chris Treichel, David Corbin, Paul Gibson, Steve Vinoski, Tom Cargill, Neil Rhodes, David Bern, Russ Williams, Robert Brazile, Doug Morgan, Uwe Steinmüller, Mark Somer, Doug Moore, David Smallberg, Seth Meltzer, Oleg Shteynbuk, David Papurt, Tony Hansen, Peter McCluskey, Stefan Kuhlins, David Braunegg, Paul Chisholm, Adam Zell, Clovis Tondo, Mike Kaelbling, Natraj Kini, Lars Nyman, Greg Lutz, Tim Johnson, John Lakos, Roger Scott, Scott Frohman, Alan Rooks, Robert Poor, Eric Nagler, Antoine Trux, Cade Roux, Chandrika Gokul, Randy Mangoba 和 Glenn Teitelbaum。

第二版草稿由以下人士共同检阅：Derek Bosch, Tim Johnson, Brian Kernighan, Junichi Kimura, Scott Lewandowski, Laura Michaels, David Smallberg, Clovis Tondo, Chris Van Wyk 和 Oleg Zabluda。我收到来自以下人士的意见并因此对新印版本有所帮助：Daniel Steinberg, Arunprasad Marathe, Doug Stapp, Robert Hall, Cheryl Ferguson, Gary Bartlett, Michael Tamm, Kendall Beaman, Eric Nagler, Max Hailperin, Joe Gottman, Richard Weeks, Valentin Bonnard, Jun He, Tim King, Don Maier, Ted Hill, Mark Harrison, Michael Rubenstein, Mark Rodgers, David Goh, Brenton Cooper, Andy Thomas-Cramer, Antoine Trux, John Wait, Brian Sharon, Liam Fitzpatrick, Bernd Mohr, Gary Yee, John O'Hanley, Brady Patterson, Christopher Peterson, Feliks Kluzniak, Isi Dunietz, Christopher Creutzi, Ian Cooper, Carl Harris, Mark Stickel, Clay Budin, Panayotis Matsinopoulos, David Smallberg, Herb Sutter, Pajo Misljencevic, Giulio Agostini, Fredrik Blomqvist, Jimmy Snyder, Byrial Jensen, Witold Kuzminski, Kazunobu Kuriyama, Michael Christensen, Jorge Yáñez Teruel, Mark Davis, Marty Rabinowitz, Ares Lagae 和 Alexander Medvedev.

第三版早期部分草稿由以下人士共同检阅：Brian Kernighan, Angelika Langer, Jesse

Laeuchli, Roger E. Pedersen, Chris Van Wyk, Nicholas Stroustrup 和 Hendrik Schober。完整草稿由以下人士共同检阅：Leor Zolman, Mike Tsao, Eric Nagler, Gene Gutnik, David Abrahams, Gerhard Kreuzer, Drosos Kourounis, Brian Kernighan, Andrew Kirmse, Balog Pal, Emily Jagdhar, Eugene Kalenkovich, Mike Roze, Enrico Carrara, Benjamin Berck, Jack Reeves, Steve Schirripa, Martin Fallenstedt, Timothy Knox, Yun Bai, Michael Lanzetta, Philipp Janert, Guido Bartolucci, Michael Topic, Jeff Scherpelz, Chris Nauroth, Nishant Mittal, Jeff Somers, Hal Moroff, Vincent Manis, Brandon Chang, Greg Li, Jim Meehan, Alan Geller, Siddhartha Singh, Sam Lee, Sasan Dashtinezhad, Alex Marin, Steve Cai, Thomas Fruchterman, Cory Hicks, David Smallberg, Gunavardhan Kakulapati, Danny Rabbani, Jake Cohen, Hendrik Schober, Paco Viciana, Glenn Kennedy, Jeffrey D. Oldham, Nicholas Stroustrup, Matthew Wilson, Andrei Alexandrescu, Tim Johnson, Leon Matthews, Peter Dulimov 和 Kevlin Henney。某些个别条款的草稿由 Herb Sutter 和 Attila F. Feher 检阅。

检阅一份不够洗练（而且可能尚未完成）的手稿是件吃力的工作，在时间压力之下进行只会使得它更困难。我要感谢这么多人愿意为我做这件事。

如果对讨论素材缺乏背景，而又期望捕捉手稿中的每一个问题，检阅工作将更加困难。令人惊讶的是还是有人选择成为文字编辑。Chrysta Meadowbrooke 是本书的文字编辑，她的周密工作揭露出许多逃过其他每一双眼睛的问题。

Leor Zolman 在正式检阅前先以多种编译器检验所有代码，在我校订手稿之后又做一次。如果书中仍然存在任何错误，全是我的责任。Karl Wiegers 和（特别是）Tim Johnson 为我提供快速而有帮助的反馈。

John Wait 是我的前两版编辑，有点傻傻地又签下这一份责任约。他的助理 Denise Mickelsen 熟练地处理我频繁的纠缠，带着愉快的微笑（至少我认为她是。呃，我从未见过她）。Julie Nahil 向来扮演救火队角色并因此成为我的产品经理。她以非凡的镇定指挥产品计划内的六周通宵工作。John Fuller（她的老板）和 Marty Rabinowitz（他的老板）也协助解决了产品发行量问题。Vanessa Moore 的正式工作是提供 FrameMaker 和 PDF 支持，但她也协助制作附录 B 的条目并格式化打印于封底里。Solveig Haugland 协助将索引格式化。Sandra Schroeder 和 Chuti Prasertsith 负责封面设计，Chuti 总是在每次我说“如果把这张相片加上那个颜色的线条会怎样？”时修订封面。Chanda Leary-Coutu 对市场营销举重若轻。

在我忙于手稿的数月之中，电视剧集 *Buffy the Vampire Slayer* 常常帮助我在每天工作结束后解压。带着极大的克制我才能够不让 Buffyspeak 的身影进入本书。Kathy Reed 于 1971 年教我写程序，我很高兴我们保持友谊至今。Donald French 雇用我和 Moises Lejter 于 1989 年建立起 C++ 培训教材（这项计划诱使我真正了解 C++），1991 年他又聘我在 Stratus

Computer 体现它们。该班学生鼓励我写下最终成为本书第一版的东西。Don 也将我介绍给 John Wait，他同意出版它们。

我的妻子 Nancy L. Urbano 持续鼓励我写作，即使在我完成了七本书、一张 CD 改写版、一篇论文之后。她有令人难以置信的忍耐、自制与宽容。没有她我无法完成我所完成的任何事情。

自始至终，我们的狗儿 Persephone 是我们无可取代的同伴。令人悲伤的是，在这个项目的大部分时间里，她和我们之间的交往关系已经改为办公室内的一坛骨灰瓮。我们十分怀念她。

For Nancy,
without whom nothing
would be much worth doing

Wisdom and beauty form a very rare combination.

— Petronius Arbiter
Satyricon, XCIV

And in memory of Persephone,
1995–2004

Contents