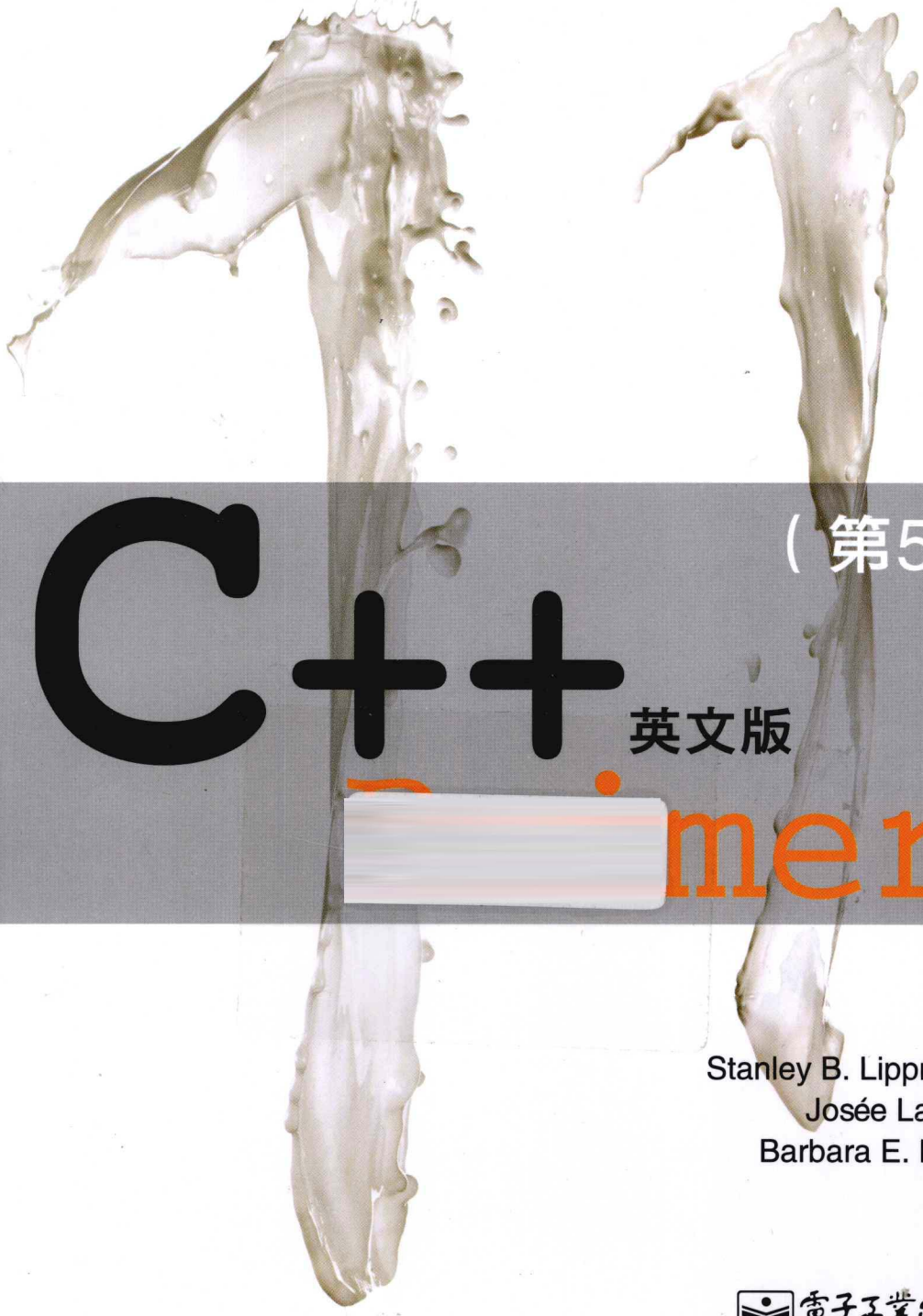
Broadview
www.broadview.com.cn

PEARSON

本书各版全球总销量达1300万册
为新的C++11标准重新撰写



C++ Primer（第5版）

英文版

Stanley B. Lippman
Josée Lajoie
Barbara E. Moo
著

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

C++ Primer（第5版）

英文版

Stanley B. Lippman
Josée Lajoie 著
Barbara E. Moo

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

内 容 简 介

这本久负盛名的C++经典教程，时隔八年之久，终于迎来史无前例的重大升级。除令全球无数程序员从中受益，甚至为之迷醉的——C++大师 Stanley B. Lippman 的丰富实践经验，C++标准委员会原负责人 Josée Lajoie 对 C++标准的深入理解，以及 C++先驱 Barbara E. Moo 在 C++教学方面的真知灼见外，更是基于全新的 C++11 标准进行了全面而彻底的内容更新。非常难能可贵的是，书中所有示例均全部采用 C++11 标准改写，这在经典升级版中极其罕见——充分体现了 C++语言的重大进展极其全面实践。书中丰富的教学辅助内容、醒目的知识点提示，以及精心组织的编程示范，让这本书在 C++领域的权威地位更加不可动摇。无论是初学者入门，或是中高级程序员提升使用，本书均为不容置疑的首选。

图书在版编目（CIP）数据

C++ Primer：第 5 版：英文 /（美）李普曼（Lippman,S.B.），（美）拉乔伊（Lajoie,J.），（美）默（Moo,B.E.）著. —北京：电子工业出版社，2013.5

ISBN 978-7-121-20038-0

Ⅰ. ①C… Ⅱ. ①李… ②拉… ③默… Ⅲ. ①C 语言－程序设计－教材－英文 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2013）第 061239 号

策划编辑：张春雨
责任编辑：刘　舫
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
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：60.25　　字数：1388 千字
印　　次：2013 年 5 月第 1 次印刷
印　　数：4000 册　　定价：128.00 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。

To Beth,
who makes this,
and all things,
possible.

To Daniel and Anna,
who contain
virtually
all possibilities.
—SBL

To Mark and Mom,
for their
unconditional love and support.
—JL

To Andy,
who taught me
to program
and so much more.
—BEM

前言

难以计数的程序员已经通过旧版的*C++ Primer*学会了C++语言。而在这段时间中，C++本身又已成熟了许多：语言本身的关注点和程序设计社区的关注点都已大大开阔，已经从主要关注机器效率转变为更多地关注编程效率。

2011年，C++标准委员会发布了ISO C++标准的一个重要修订版。此修订版是C++进化过程中的最新一步，延续了前几个版本对编程效率的强调。新标准的主要目标是：

- 使语言更为统一，更易于教学
- 使标准库更简单、安全、使用更高效
- 使编写高效率的抽象和库变得更简单

因此，在这个版本的*C++ Primer*中，我们进行了彻底的修改，使用了最新的C++标准。为了了解新标准是如何全面影响C++语言的，你可以看一下xxiii页至xxv页的新特性列表，其中列出了哪些章节涉及了C++的新特性。

新标准增加的一些特性是具有普适性的，例如用于类型推断的`auto`。这些新特性使本书中的代码更易于阅读和理解。程序（以及程序员！）可以忽略类型的细节，从而更容易集中精力于程序逻辑上来。其他一些新特性，例如智能指针和允许移动的容器，允许我们编写更为复杂的类，而又不必与错综复杂的资源管理做斗争。因此，在本书中开始讲授如何编写自己的类，会比第4版简单得多。旧标准中阻挡在我们前进路上的很多细节，你我都不必再担心了。

对于本书中涉及新标准定义的新特性的那些部分，我们都已用一个特殊的图标标记出来了。我们希望这些提示标记对那些已经熟悉C++语言核心内容的读者是有帮助的，可以帮助他们决定将注意力投向哪里。对于那些可能尚不支持所有新特性的编译器，我们还希望这些图标能有助于解释这类编译器所给出的编译错误信息。这是因为虽然本书中几乎所有例子都已经用最新版本的GNU编译器编译通过，但我们知道一些读者可能尚未将编译器更新到最新版本。虽然新标准增加了大量新功能，但核心C++语言并未变化，这构成了本书的大部分内容。读者可以借助这些图标来判断哪些功能可能还没有被自己的编译器所支持。

C++ 11

为什么选择这本书?

现代 C++语言可以看作是三部分组成的:

- 低级语言,大部分继承自 C 语言。
- 现代高级语言特性,允许我们定义自己的类型以及组织大规模程序和系统。
- 标准库,它利用高级特性来提供有用的数据结构和算法。

大多数 C++教材按照语言进化的顺序来组织其内容。首先讲授 C++的 C 子集,然后将 C++的更为抽象的一些特性作为高级话题在书的最后进行介绍。这种方式存在两个问题:读者会陷入那些继承自低级程序设计的细节,从而由于挫折感而放弃;读者被强加学习一些坏习惯,随后又需要忘记这些内容。

我们采用一种相反的方法:从一开始就介绍一些语言特性,能让程序员忽略那些继承自低级程序设计的细节。例如,在介绍和使用内置的算术和数组类型时,我们还连同介绍和使用标准库中的类型 `string` 和 `vector`。使用这些类型的程序更易写、易理解且更少出错。

太多时候,标准库被当作一种“高级”话题来讲授。很多教材不使用标准库,而是使用基于字符数组指针和动态内存管理的低级程序设计技术。让使用这种低级技术的程序正确运行,要比编写相应的使用标准库的 C++代码困难得多。

贯穿 *C++ Primer* 全书,我们都在强调好的风格:我们想帮助读者直接养成好的习惯,而不是在获得很多很复杂的知识后再去忘掉那些坏习惯。我们特别强调那些棘手的问题,并对常见的错误想法和陷阱提出警告。

我们还注意解释规则背后的基本原理——使读者不仅知其然,还能知其所以然。我们相信,通过体会程序的工作原理,读者会更快地巩固对语言的理解。

虽然你不必为了学习本书而掌握 C 语言,但我们还是假定你了解足够多的程序设计知识,了解至少一门现代分程序结构语言,知道如何用这门语言编写、编译以及运行程序。特别是,我们假定你已经使用过变量,编写、调用过函数,也使用过编译器。

第 5 版变化的内容

这一版 *C++ Primer* 的新特点是用边栏图标来帮助引导读者。C++是一种庞大的编程语言,它提供了一些为特定程序设计问题定制的功能。其中一些功能对大型项目团队有很重要的意义,但对于小型项目开发可能并无必要。因此,并非每个程序员都需要了解每个语言特性的所有细节。我们加入这些边栏图标来帮助读者弄清哪些内容可以随后再学习,而哪些主题是更为重要的。

对于包含 C++语言基础内容的章节，我们用一个小人正在读书的图标加以标记。用这个图标标记的那些章节，涵盖了构成语言核心部分的主题。每个人都应该阅读并理解这些章节的内容。

对于那些涉及高级主题或特殊目的主题的章节，我们也进行了标记。在首次阅读时，这些章节可以跳过或快速浏览。我们用一叠书的图标标记这些章节，指出在这些地方，你可以放心地放下书本。快速浏览这些章节可能是一个好主意，这样你就可以知道有这些特性存在。但在真正需要在自己的程序中使用这些特性之前，没有必要花费时间仔细学习这些主题。

为了进一步引导读者的注意力，我们还用放大镜图标标记了特别复杂的概念。我们希望读者对有这种标记的章节能多花费一些时间彻底理解其中的内容。在这些章节中，至少有一些，其主题的重要性可能不是那么明显；但我们认为，你会发现这些章节涉及的主题对理解 C++语言原来至关重要。

交叉引用的广泛使用，是本书采用的另外一种阅读帮助。我们希望这些引用能帮助读者容易地翻阅书中的内容，同时还能在后面的例子涉及到前面的内容时容易地跳回到前面。

没有改变的是，*C++ Primer* 仍是一本清晰、正确、全面的 C++入门教材。我们通过给出一系列复杂度逐步增加的例子来讲授这门语言，这些例子说明了语言特性，展示了如何充分用好 C++语言。

本书的结构

我们首先在第 I 部分和第 II 部分中介绍了 C++语言和标准库的基础内容。这两部分包含的内容足够你编写出有意义的程序，而不是只能写一些玩具程序。大部分程序员基本上都需要掌握本书这两部分所包含的所有内容。

除了讲授 C++的基础内容，第 I 部分和第 II 部分还有另外一个重要目的：通过使用标准库中定义的抽象设施，使你更加适应高级程序设计技术。标准库设施本身是一组抽象数据类型，通常用 C++编写。用来设计标准库的，就是任何 C++程序员都可以使用的用来构造类的那些语言特性。我们讲授 C++语言的一个经验是，在先学习了使用设计良好的抽象类型后，读者会发现理解如何构造自己的类型更容易了。

只有在经过全面的标准库使用训练，并编写了各种标准库所支持的抽象程序后，我们才真正进入到那些允许你编写自己的抽象类型的 C++特性中去。本书的第 III 部分和第 IV 部分介绍了如何编写类的形式的抽象类型。第 III 部分包含基础内容，第 IV 部分介绍更专门的语言特性。

在第 III 部分中，我们将介绍拷贝控制问题，以及其他一些使类能像内置类型一样容易使用的技术。类是面向对象编程和泛型编程的基础，第 III 部分也会介绍这些内容。第 IV 部分是 *C++ Primer* 的结束部分，它介绍了一些在组织大型复杂系统时非常有用的语言特性。此外，我们将在附录 A 中总结标准库算法。

读者帮助

本书的每一章均以一个总结和一个术语表结束，两者一起扼要回顾了这一章的大部分学习重点。读者应该将这些部分作为个人备忘录：如果你不理解某个术语，可以重新学习这一章的相应部分。

在本书中我们还使用了其他一些学习辅助：

- 重要的术语用**加粗字体**显示；我们假定读者已经熟悉的重要术语用***粗斜体***显示。每个术语都会列在章末尾的术语表中。
- 贯穿全书，我们用高亮显示来提醒读者注意语言的重要部分，对常见的陷阱提出警告，建议好的程序设计习惯，以及提供一般性的使用提示。
- 为了更好地理解语言特性间和概念间的联系，我们提供大量向前的和向后的交叉引用。
- 对重要的概念和 C++新程序员常常觉得最困难的主题，我们提供边栏讨论。
- 学习任何程序设计语言都需要编写程序。为此，贯穿全书我们提供大量程序示例。扩展示例的源码可从下面的网址获得：

```
http://www.informit.com/title/0321714113
```

关于编译器的注意事项

在撰写本书时（2012 年 7 月），编译器提供商正在努力工作，升级编译器以匹配最新的 ISO 标准。我们使用最多的编译器是 GNU 编译器 4.7.0。本书中只有一小部分特性在此编译器中尚未实现：继承构造函数、成员函数的引用限定符以及正则表达式库。

致谢

我们要特别感谢标准委员会几位现任和前任委员：Dave Abrahams、Andy Koenig、Stephan T. Lavavej、Jason Merrill、John Spicer 和 Herb Sutter 在准备本书的过程中提供的帮助。在理解新标准的一些更微妙之处，他们为我们提供了宝贵的帮助。我们还要感谢很多致力于升级 GNU 编译器以实现新标准的人们。

与旧版 *C++ Primer* 中一样，我们要感谢 Bjarne Stroustrup 不知疲倦地为 C++ 工作以及他和作者长时间的友谊。我们还要感谢 Alex Stepanov 的非凡洞察力，催生了标准库核心的容器和算法。最后，我们要感谢 C++标准委员会的所有委员，感谢他们这么多年来在净化、精炼和改进 C++语言方面的辛苦工作。

我们衷心感谢审稿人：Marshall Clow、Jon Kalb、Nevin Liber、Dr. C. L. Tondo、Daveed Vandevoorde 和 Steve Vinoski，他们建设性的意见帮助我们对全书做出了大大小小的改进。

本书是用 Latex 及其发行版本中的很多包来进行排版的，我们应该感谢 Latex 社区成员创造出如此强大的排版工具。

最后，我们要感谢 Addison-Wesley 公司的优秀员工，他们指导了本书的整个出版过程：Peter Gordon，我们的编辑，他给了我们动力再次修改 *C++ Primer*；Kim Boedigheimer，保证了一切按计划进行；Barbara Wood，她在编辑过程中找到了大量编辑错误；还有 Elizabeth Ryan，很高兴再次和她共同工作，她指导我们完成了整个设计和生产流程。

Contents