

图书基本信息

书名：<<C++ Primer 英文版 (第5版)>>

13位ISBN编号：9787121200380

10位ISBN编号：7121200384

出版时间：2013-5

出版时间：电子工业出版社

作者：[美] Stanley B. Lippman,[美] Jos é e Lajoie,[美] Barbara E. Moo

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

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。

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。

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。

内容概要

这本久负盛名的C++经典教程，时隔八年之久，终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Jos é e 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，本书均为不容置疑的首选。

作者简介

Stanley B. Lippman目前是微软公司 Visual C++ 团队的架构师。

他从1984年开始在贝尔实验室与C++的设计者Bjarne Stroustrup一起从事C++的设计与开发。

他在迪士尼和梦工厂从事动画制作，还担任过JPL的高级顾问。

Jos é e Lajoie曾经是IBM加拿大研究中心C/C++编译器开发团队的成员，在ISO C++标准委员会工作了7年，担任过ISO核心语言工作组的主席和C++ Report杂志的专栏作家。

Barbara E. Moo是拥有25年软件经验的独立咨询顾问。

在AT&T，她与Stroustrup、Lippman一起管理过复杂的C++开发项目。

书籍目录

Chapter 1 Getting Started 1.1 Writing a Simple C++ Program 1.1.1 Compiling and Executing Our Program 1.2 A First Look at Input/Output 1.3 A Word about Comments 1.4 Flow of Control 1.4.1 The while Statement 1.4.2 The for Statement 1.4.3 Reading an Unknown Number of Inputs 1.4.4 The if Statement 1.5 Introducing Classes 1.5.1 The sales item Class 1.5.2 A First Look at Member Functions 1.6 The Bookstore Program Chapter Summary Defined Terms Part The Basics Chapter 2 Variables and Basic Types 2.1 Primitive Built-in Types 2.1.1 Arithmetic Types 2.1.2 Type Conversions 2.1.3 Literals 2.2 Variables 2.2.1 Variable Definitions 2.2.2 Variable Declarations and Definitions 2.2.3 Identifiers 2.2.4 Scope of a Name 2.3 Compound Types 2.3.1 References 2.3.2 Pointers 2.3.3 Understanding Compound Type Declarations 2.4 const Qualifier 2.4.1 References to const 2.4.2 Pointers and const 2.4.3 Top-Level const 2.4.4 constexpr and Constant Expressions 2.5 Dealing with Types 2.5.1 Type Aliases 2.5.2 The auto Type Specifier 2.5.3 The decltype Type Specifier 2.6 Defining Our Own Data Structures 2.6.1 Defining the Sales_data Type 2.6.2 Using the Sales_data Class 2.6.3 Writing Our Own Header Files Chapter Summary Defined Terms Chapter 3 Strings, Vectors, and Arrays 3.1 Namespace using Declarations 3.2 Library string Type 3.2.1 Defining and Initializing strings 3.2.2 Operations on strings 3.2.3 Dealing with the Characters in a string 3.3 Library vector Type 3.3.1 Defining and Initializing vectors 3.3.2 Adding Elements to a vector 3.3.3 Other vector Operations 3.4 Introducing Iterators 3.4.1 Using Iterators 3.4.2 Iterator Arithmetic 3.5 Arrays 3.5.1 Defining and Initializing Built-in Arrays 3.5.2 Accessing the Elements of an Array 3.5.3 Pointers and Arrays 3.5.4 C-Style Character Strings 3.5.5 Interfacing to Older Code 3.6 Multidimensional Arrays Chapter Summary Defined Terms Chapter 4 Expressions 4.1 Fundamentals 4.1.1 Basic Concepts 4.1.2 Precedence and Associativity 4.1.3 Order of Evaluation 4.2 Arithmetic Operators 4.3 Logical and Relational Operators 4.4 Assignment Operators 4.5 Increment and Decrement Operators 4.6 The Member Access Operators 4.7 The Conditional Operator 4.8 The Bitwise Operators 4.9 The sizeof Operator 4.10 Comma Operator 4.11 Type Conversions 4.11.1 The Arithmetic Conversions 4.11.2 Other Implicit Conversions 4.11.3 Explicit Conversions 4.12 Operator Precedence Table Chapter Summary Defined Terms Chapter 5 Statements 5.1 Simple Statements 5.2 Statement Scope 5.3 Conditional Statements 5.3.1 The if Statement 5.3.2 The switch Statement 5.4 Iterative Statements 5.4.1 The while Statement 5.4.2 Traditional for Statement 5.4.3 Range for Statement 5.4.4 The do while Statement 5.5 Jump Statements 5.5.1 The break Statement 5.5.2 The continue Statement 5.5.3 The goto Statement 5.6 try Blocks and Exception Handling 5.6.1 A throw Expression 5.6.2 The try Block 5.6.3 Standard Exceptions Chapter Summary Defined Terms Chapter 6 Functions 6.1 Function Basics 6.1.1 Local Objects 6.1.2 Function Declarations 6.1.3 Separate Compilation 6.2 Argument Passing 6.2.1 Passing Arguments by Value 6.2.2 Passing Arguments by Reference 6.2.3 const Parameters and Arguments 6.2.4 Array Parameters Part The C++ Library Part Tools for Class Authors Part Advanced Topics Index

章节摘录

版权页： We start by initializing it to denote the first element in text. The loop continues until either we process every element in text or we find an element that is empty. So long as there are elements and we haven't seen an empty element, we print the current element. It is worth noting that because the loop reads but does not write to the elements in text, we use `cbegin` and `cend` to control the iteration. Some vector Operations Invalidate Iterators In 3.3.2 (p.101) we noted that there are implications of the fact that vectors can grow dynamically. We also noted that one such implication is that we cannot add elements to a vector inside a range for loop. Another implication is that any operation, such as `push_back`, that changes the size of a vector potentially invalidates all iterators into that vector. We'll explore how iterators become invalid in more detail in 9.3.6 (p.353). **WARNING** For now, it is important to realize that loops that use iterators should not add elements to the container to which the iterators refer. **EXERCISES SECTION 3.4.1** Exercise 3.21: Redo the first exercise from 3.3.3 (p.105) using iterators. Exercise 3.22: Revise the loop that printed the first paragraph in text to instead change the elements in text that correspond to the first paragraph to all uppercase. After you've updated text, print its contents. Exercise 3.23: Write a program to create a vector with ten int elements. Using an iterator, assign each element a value that is twice its current value. Test your program by printing the vector.

编辑推荐

《C++ Primer英文版(第5版)》编辑推荐：最新标准C++11发布，距上一版本已10年；《C++ Primer英文版(第5版)》是唯一持续更新的全球顶级C++读本。
多位深孚众望的大师组合堪称绝无仅有；经过前四个版本积累，第5版的体例堪称完美。
这一版本作者历时3年完成，极力避免在原版上升级，而是将C++11的新特性真正融入各章节；更将所有代码示例用C++ 11的简化写法完成，而不是仅单独增加内容。

版权说明

本站所提供下载的PDF图书仅提供预览和简介, 请支持正版图书。

更多资源请访问:<http://www.tushu007.com>