PROGRAMMER TO PROGRAMMER™

这是一本曾成就无数 C++ 程序员的经典名著，厚而不“重”，可帮助您轻松掌握 C++ 的各种编程知识，为您的职业生涯打下坚实的基础。

FULLY REVISED AND UPDATED for the Latest ANSI Standard

Ivor Horton's Beginning ANSI C++ The Complete Language, Third Edition

C++ 入门经典（第 3 版）

(美) Ivor Horton 著
李予敏 译

清华大学出版社

C++入门经典

(第3版)

(美) Ivor Horton 著

李予敏 译

清华大学出版社

北 京

内 容 简 介

C++在几乎所有的计算环境中都非常普及，而且可以用于几乎所有的应用程序。C++从C中继承了过程化编程的高效性，并集成了面向对象编程的功能。C++在其标准库中提供了大量的功能。有许多商业C++库支持数量众多的操作系统环境和专业应用程序。但因为它的内容太多了，所以掌握C++并不十分容易。本书详述了C++语言的各个方面，包括数据类型、程序控制、函数、指针、调试、类、重载、继承、多态性、模板、异常和输入输出等内容。每一章都以前述内容为基础，每个关键点都用具体的示例进行详细的讲解。

本书基本不需要读者具备任何C++知识，书中包含了理解C++的所有必要知识，读者可以从头开始编写自己的C++程序。本书也适合于具备另一种语言编程经验但希望全面掌握C++语言的读者。

EISBN：1-59059-227-1
Ivor Horton's Beginning ANSI C++: The Complete Language, Third Edition
Ivor Horton
Original English language edition published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA.

北京市版权局著作权合同登记号　图字：01-2005-4579

图书在版编目(CIP)数据

C++入门经典(第3版)/(美)霍顿(Horton, I.)著；李予敏　译. —北京：清华大学出版社，2006.1
书名原文：Ivor Horton's Beginning ANSI C++: The Complete Language, Third Edition
ISBN 7-302-12062-5
Ⅰ. C…　Ⅱ. ①霍…②李…　Ⅲ. C语言—程序设计　Ⅳ. TP312
中国版本图书馆CIP数据核字(2005)第127516号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：曹　康
文稿编辑：徐燕华
封面设计：康　博
版式设计：康　博
印 刷 者：北京国马印刷厂
装 订 者：三河市新茂装订有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：50　**字数**：1280千字
版　　次：2006年1月第1版　2006年1月第1次印刷
书　　号：ISBN 7-302-12062-5/TP·7808
印　　数：1～4000
定　　价：98.00元

作者简介

Ivor Horton是世界著名的计算机图书作家，主要从事与编程相关的顾问及撰写工作，曾帮助无数程序员步入编程的殿堂。他曾在IBM工作多年，能使用多种语言进行编程(在多种机器上使用汇编语言和高级语言)，设计和实现了实时闭环工业控制系统。Horton拥有丰富的教学经验(教学内容包括C、C++、Fortran、PL/1、APL等)，同时还是机械、加工和电子CAD系统、机械CAM系统和DNC/CNC系统方面的专家。Ivor Horton还著有*Beginning Visual C++ 6*、*Beginning C Programming*和*Beginning Java 2*等多部入门级好书。

译者简介

李予敏，男，计算机科学及应用专业博士，某研究院高级研究员，拥有丰富的C、C++编程经验，在核心期刊、国际国内会议上发表多篇文章，拥有著作3本、译著2本。

译 者 序

C++自诞生以来，已成为使用最广泛的一种编程语言。C++从C中继承了过程化编程的高效性，集成了面向对象编程技术。C++还在其标准库中提供了大量的功能。它有着极大的灵活性、强大的功能和非常高的效率，常常用于专业应用程序的开发，由于其内容较多，掌握起来也不易。

本书是《C++入门经典》的第3版，采用的编写方法与前两版相同，第3版在第1、2两版基础上又做了修订和更新，包括示例，对于自学的学生来说也是理想的选择。

本书详细介绍了C++语言的各个方面，包括数据类型、程序控制、函数、指针、调试、类、重载、继承、多态性、异常和输入输出等内容。本书还深入讨论了类模板，包括标准模板库(STL，Standard Template Library)。每章都以前述内容为基础，每个关键点都通过具体的示例进行讲解。每章的最后都提供了练习题。

本书主要介绍标准的C++编程语言，涉及C++的语法、面向对象的功能和标准库等所有基本内容。通过本书可以获得编写C++应用程序的所有必要知识。

对于初学者来说，C++语言似乎比其他语言更难，但其功能和适用范围要远远超过其他编程语言。读者只要抱以正确的态度、具备编程的基本知识以及掌握C++的热情，在C++的学习和应用上就不会有太大的问题。学习本书，读者基本不需要具备任何编程语言的知识，只需了解基本的编程概念，就可以读懂本书，编写自己的C++程序，这也是学好C++的惟一方式。如果您了解像分支和循环这样的概念，那本书就非常适合您。

本书也适合于已有其他语言编程经验但希望全面掌握C++语言的读者。

作者Ivor Horton以善于教学而著称，他的C、C++等书都拥有大量忠实读者。

由于受时间和译者自身水平的限制，翻译过程中难免出现错误和疏漏，敬请读者多多批评指正，反馈信息请发至fwkbook@tup.tsinghua.edu.cn信箱。

译者

2005.7

前　言

本书主要介绍标准的C++编程语言，涉及C++的语法、面向对象的功能和标准库等所有基本内容。阅读本书将获得编写C++应用程序的所有必要知识。

为什么要学习C++

C++自问世以来，已成为应用最广泛的一种编程语言。C++由于其极高的灵活性、强大的功能和非常高的效率，常常用于专业应用程序的开发，C++非常适合于编写各种编程环境下的高性能代码。

它要比许多人想像的更容易理解。只要有正确的引导，掌握C++编程语言是比较容易的。开发C++技巧，学习许多人已在使用的语言，在自己的编程工具箱中就会多一种功能强大的新工具。

C++的标准

1998年，C++的国际化标准ISO/IEC 14882最终定稿，并被美国国家标准协会ANSI和信息技术标准国际协会INCITS采纳。这是ANSI/ISO小组9年工作的成果，其目的是为C++编程语言开发一种世界标准。尽管编写本书时1998年的标准仍在使用，但改进该语言的工作一直在进行，因此将来C++一定会添加新特性。

C++的1998标准为编译器的编写人员提供了一幅蓝图，所以，目前许多(但不是全部)编译器都遵循该标准。如果使用遵循该标准的编译器，代码的可移植性将非常高，将来，还可以避免非标准语言元素带来的麻烦。

当然，C++的标准定义为开发在任何硬件或操作系统环境下运行的编译器的参考框架。另外，它还将试图在任何开发环境下尽可能地提高性能。也就是说，编译器编写人员在许多领域都有非常大的灵活性，以包容机器体系之间的差异。例如，该标准定义了数字数据和算术操作，这样编译器编写人员就可以充分利用各种机器的不同特性，优化执行性能。编译器编写人员还可以选择用于定义C++程序的字符编码。这样，就可以包容默认字符编码在不同操作系统上的变化。没有这种灵活性，在某些机器上该标准就会有一定的局限性，导致性能较差，这非常不利于一般目的的编程语言。

本书将指出机器之间重要的、潜在的不同。但是，这需要一个实际有效的环境来显示本书中各个例子的输出。因此，所有的例子都在一台安装了Intel处理器体系结构的PC上运行。

错误和更正

作者和 Apress 的编辑们已经尽最大努力确保本书中的文本和代码没有错误，但是错误仍

然在所难免。如果您发现本书存在错误，请进入 Apress 网站的下述 Web 页面：

http://www.apress.com/book/download.html

如果在这个页面的列表中选择本书的书名，就可以下载勘误表和本书所有例子的代码，还可以记录下您找到的其他错误。下载的代码也包含所有练习的答案，但读者最好在完成了练习后再看答案。

使用本书

要通过本书学习 C++，需要一个与 ANSI/ISO 兼容的编译器和一个适合于编写程序代码的文本编辑器。目前，大多数专业 C++开发环境所附带的编译器都遵循这个标准，但在购买之前最好检查一下。另外，Internet 上的一些免费软件和开放源代码的 C++编译器也遵循 C++标准。可以使用其中一个编译器和免费的程序文本编辑器，建立起一个经济、可行的学习环境。

本书的内容循序渐进，所以读者应从头开始一直阅读到最后。但是，没有人能仅从一本书中获得所有的编程技巧。本书仅介绍了如何使用 C++编程，读者应自己输入所有的例子，而不是从下载文件中复制它们。再编译和执行输入的代码，这似乎很麻烦，但输入 C++语句可以帮助理解 C++，特别是觉得某些地方很难掌握时，自己输入代码就显得非常有帮助。如果例子不工作，不要直接从书中查找原因，而应在自己输入的例子代码中找原因，这是编写 C++代码时必须做的一个工作。

犯错误也是学习过程中不可避免的，练习应提供大量犯错误的机会，犯的错误越多，对 C++的功能和错误的原因认识得就越深刻。读者应完成所有的练习，记住不要看答案，直到肯定不能自己解决问题为止。许多练习都涉及某章内容的一个直接应用，换言之，它们仅是一种实践，但也有一些练习需要多动脑子，甚至需要一点灵感。

希望每个人都能成功驾驭 C++。

目　录

第1章 基本概念

初看起来，学习C++编程与家禽不可能有关系，但它们还是有关系的——这是一个鸡生蛋、蛋生鸡的问题。特别是在学习C++的初期，我们常常要利用还不能理解的例子来学习C++。本章将概述C++语言，介绍C++的各种特性，解决这个鸡生蛋、蛋生鸡的问题，再讨论几个要使用的概念，以后会详细论述它们。

这里介绍的所有概念都将在后面的章节详细讨论。大多数内容仅是在编写C++程序之前介绍一下基本概念。我们将列举一个简单的C++程序，并说明它的组成部分。本章还将介绍C++中编程的一些主要概念，以及如何根据所编写的源代码文件创建可执行的程序。

不必努力记住本章介绍的概念，而应把注意力集中在涉及到的理念上。本章提到的所有概念都将在后面的章节中详细论述。

本章主要内容

- C++的哪些特性使之这么普及
- C++程序的基本元素
- 如何注释程序的源代码
- 源代码如何变成可执行程序
- 面向对象的编程方式与过程编程方式的区别

1.1 编程语言

读者可能很熟悉编程和编程语言的基本概念，下面从普遍的意义上简要描述一下本书将用到的一些术语，并阐述C++与其他编程语言的关系。

目前有许多编程语言，每一种语言都有其优缺点，都有其吹捧者和批评者。除了C++之外，读者一定还听说过Java、BASIC(Beginner's All-purpose Symbolic Instruction Code的首字母缩写)、COBOL(Common Business-Oriented Language的首字母缩写)、FORTRAN(formula translator的前几个字母缩写)、PASCAL(以一位法国数学家Blaise Pascal命名)和C(只是因为它是B语言的后续语言)等编程语言。所有这些统称为高级语言，因为它们可以比较容易地表达出要计算机完成的工作，而且不针对某台计算机。高级语言中的每个源语句一般映射为几个内部机器指令，低级语言比较接近内部机器指令，通常称为汇编语言，一种汇编语言专门用于一种硬件设计，一般一个汇编指令映射为一个内部机器指令。

1.1.1 编程语言简史

FORTRAN是第一种开发出来的高级语言，第一个FORTRAN编译器是在上个世纪50年代后期开发出来的。FORTRAN已有40多年的历史了，目前仍广泛应用于科学和工程计算中，但C++和其他语言也逐渐进入这些领域。

COBOL语言专门用于商务数据处理应用程序，它的历史几乎与FORTRAN语言一样长。

目前几乎不用 COBOL 编写新代码，而是多年前编写的大量代码仍在使用，所以必须维护它们。C++也逐渐成为许多商务数据处理程序的可选语言。

BASIC 在上个世纪 70 年代诞生，那时已经有了个人计算机的概念。有趣的是，Microsoft 销售的第一个产品是一个 BASIC 解释程序。这种语言所固有的易用性使之很快普及，直到今天仍非常流行。

Java 是在上个世纪 90 年代开发的，它最初开发为 Oak 语言，用于给小型电子设备编程。1995 年，Oak 演变为 Java 语言，可以在 Web 页面中内嵌代码，从那时起直到现在，这已经成为 Java 的主要用途。Java 成功的主要原因是它的可移植性。Java 程序可以在任何支持它的硬件平台上运行，而且不需要任何修改。Java 语言的语法有许多特性，使它看起来很象 C++，但有很大的区别。Java 在可移植性方面比 C++好，但执行性能比不上 C++。

C 在上个世纪 70 年代被开发为一种高级语言，用于低级编程，例如实现操作系统。大多数 Unix 操作系统就是用 C 编写的。

C++是 Bjarne Stroustrup 在上个世纪 80 年代早期开发的，是一种基于 C 的面向对象语言。顾名思义，C++表示 C 的累加。由于 C++基于 C，所以这两种语言有许多共同的语法和功能，C 中所有低级编程的功能都在 C++中保留下来。但是，C++比其前身丰富得多，用途也广泛得多。C++对内存管理功能进行了非常大的改进，C++还具有面向对象的功能，所以 C 在功能上只是 C++的一个很小的子集。C++在适用范围、性能和功能上也是无可匹敌的。因此，目前大多数高性能的应用程序和系统仍使用 C++编写。

1.1.2 解释性程序和编译性程序的执行过程

无论使用哪种编程语言，编写出来的程序都是由各个指令或源语句构成的，它们描述了希望计算机执行的动作。这些指令或源语句统称为源代码，存储在磁盘的源文件中。任何规模的 C++程序都是由若干个源文件组成的。

编程语言的目的是，与计算机可以执行的程序相比，能够更简单地描述希望计算机执行的动作。计算机只能执行包含机器指令(也称为机器代码)的程序，不能直接执行我们编写的程序。用前面提到的语言编写的程序基本上有两种执行方式，在大多数情况下，一种语言会选择其中一种执行方式。例如，用 BASIC 语言编写的程序通常是解释性的，也就是说，另一个称为解释器的程序会检查 BASIC 源代码，确定该程序要做什么，再让计算机完成这些动作。如图 1-1 所示。

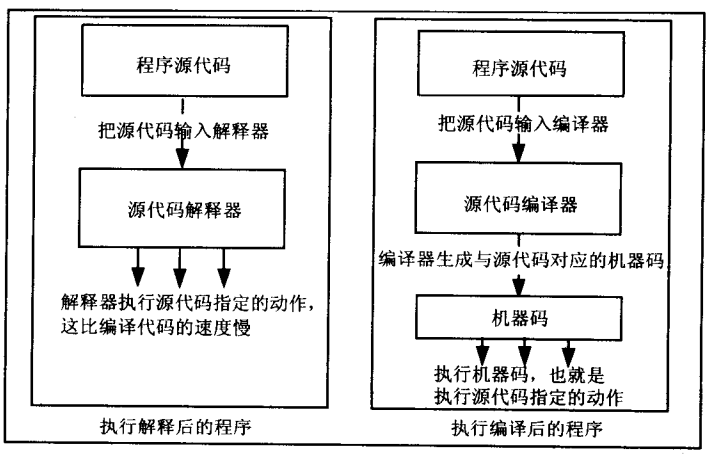

图 1-1 解释性程序和编译性程序的执行过程

而C++是一种编译语言。在执行C++程序之前，必须用另一个程序(即编译器)把它转换为机器语言。编译器会检查并分析C++程序，并生成机器指令，以执行源代码指定的动作。当然，解释和编译都不像这里描述的那样简单，但其工作原理就是这样。

使用解释性语言，执行过程是间接的，也就是说，每次执行程序时，都需要确定源代码的意图。因此，这种语言比编译语言的对应程序的执行速度慢得多，有时要慢100倍。其优点是在运行之前，不必等待程序的编译。使用解释性语言，一旦输入代码，就可以立刻执行程序。任何一种语言要么是解释性的，要么是编译性的，这通常由该语言的设计和用途来决定。前面说过BASIC是一种解释性语言，但这不是绝对的，目前有许多BASIC语言的编译器。

于是，就有了“哪种语言比较好”这个问题。简单地说，没有所谓“最好”的语言，因为这取决于环境。例如，用BASIC编写程序通常比使用其他语言快得多，所以，如果开发速度比较重要，但执行性能不是很重要，BASIC就是一种非常好的选择。另一方面，如果程序要求具有C++提供的执行性能，或者应用程序需要C++中的功能而不是BAISC，显然就应使用C++。如果应用程序必须在许多不同的计算机上执行，而且执行性能不是很重要，可能Java就是最佳选择。

当然，不同的语言学习起来，所需的时间和难度也不一致。根据学习一种语言所需的时间，C++可能是比较难的，但是不应放弃，这并不是说C++非常难，而是说C++比其他语言难一些，需要较长的学习时间。

学习哪种语言的最后一个要点是，任何专业程序员都需要掌握几种编程语言。如果读者是一位初学者，这可能会使人觉得沮丧，但一旦学习并掌握了一两种编程语言，再掌握其他语言就非常容易了。第一种编程语言总是最难的。

1.1.3 库

每次编写程序时，如果总是要从头开始编写，就相当繁琐。在许多程序中，常常需要某种相同的功能，例如从键盘上输入数据，或在屏幕上显示信息，或按照指定的顺序对数据记录排序。为了解决这个问题，编程语言通常提供了大量预先编写好的代码，以执行标准的操作，这样就不必重新编写这些代码了。

可用于任意程序的标准代码都保存在一个库中。编程语言附带的库跟语言本身一样重要，因为库的质量和使用范围对完成某一编程任务所需的时间有非常大的影响。

1.2 C++是一种强大的语言

C++在几乎所有的计算环境中都非常普及：个人电脑、Unix工作站和大型计算机。如果考察一下新编程语言的发展史，就可以看出C++的这种普及率是非常高的。用以前的语言编写的程序量非常大，这无疑会降低对新语言的接受程度。除此以外，大多数专业程序员总是愿意使用他们已熟知的、使用起来得心应手的语言，而不是转而使用新的、不熟悉的语言，花大量的时间来研究其特性。当然，C++是建立在C的基础之上(在C++出现之前，许多环境都使用C语言)，这对于C++的普及有很大的帮助，但是C++的流行远不只这一个原因。C++有许多优点：

- C++适用的应用程序范围极广。C++可以用于几乎所有的应用程序，从字处理应用程序到科学应用程序，从操作系统组件到计算机游戏等。
- C++可以用于硬件级别的编程，例如实现设备驱动程序。
- C++从C中继承了过程化编程的高效性，并集成了面向对象编程方式的功能。
- C++在其标准库中提供了大量的功能。
- 有许多商业C++库支持数量众多的操作系统环境和专门的应用程序。

因为几乎所有的计算机都可以使用C++编程，所以C++语言普及到几乎所有的计算机平台上。也就是说，把用C++编写的程序从一台机器迁移到另一台机器上不需要费什么力气。当然，如果这个过程真的非常简单，那么编写在另一台机器上运行的程序时就需要考虑使用C++语言了。

C++的ANSI/ISO标准

C++的国际标准由ISO/IEC 14882文档定义，该文档由美国国家标准协会ANSI发表。读者可以获得该标准的副本，但要记住，该标准主要由编译器编写人员使用，而不是学习该语言的人使用。如果读者不在意这一点，就可以从http://webstore.ansi.org/ansidocstore/default.asp上用合理的费用下载该标准的副本。

标准化是把所编写的程序从一种类型的计算机迁移到另一种类型的计算机上的基础。标准的建立使语言在各种机器上的实现保持一致。在所有相容编程系统上都可用的一组标准功能意味着，用户总是能确定下一步会获得什么结果。C++的ANSI标准不仅定义了语言，还定义了标准库。使用ANSI标准后，C++使应用程序可以轻松地在不同的机器之间迁移，缓解了在多个环境上运行的应用程序的维护问题。

C++的ANSI标准还有另一个优点：它对用C++编程所需要学习的部分进行了标准化。这个标准将使后续的程序具有一致性，因为它只为C++编译器和库提供了一个定义参考。在编写编译器时，该标准的存在也使编写人员不再需要许可。读者在购买遵循ANSI标准的C++编译器时，就知道会得到什么语言和标准库功能。

1.3 一个简单的C++程序

下面介绍一个非常简单的C++程序，了解C++程序的组成。现在读者不需要输入代码，只是了解一下建立程序的过程。这里也不详细介绍所有的细节，因为这些内容将在后面的章节中探讨。如图1-2所示。

图1-2中所示的程序会显示如下消息：

```
The best place to start is at the beginning
```

这个程序不是很有用，但说明了几点。该程序由一个函数main()组成。函数是代码的一个自包含块，用一个名称表示，在本例中是main。程序中还可以有许多其他代码，但每个C++程序至少要包含函数main()，且只能有一个main()函数。C++程序的执行总是从main()中的第一条语句开始。