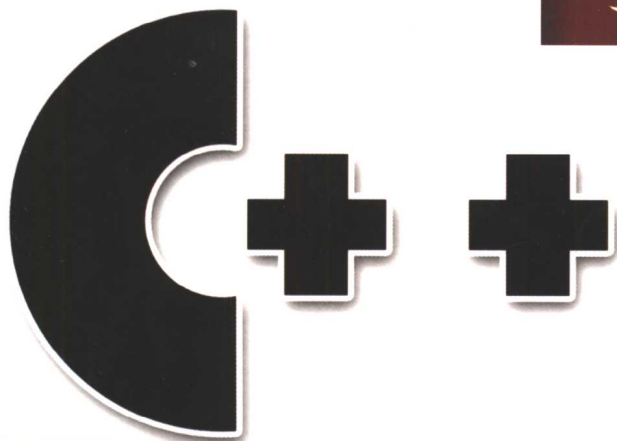


让我们跟随权威专家
学习C++程序设计!



第7版

大学自学教程

· Teach yourself C++
· Teach

[美] Al Stevens 著

Dr. Dobb's Journal 专栏作家

林 瑶 蒋晓红 彭卫宁 等译



电子工业出版社

Publishing House of Electronics Industry
<http://www.phei.com.cn>

C++ 大学自学教程

第 7 版

Wiley's Teach Yourself C++

[美] Al Stevens 著

林 瑶 蒋晓红 彭卫宁 等译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是一本供读者自学C++语言的指导用书。C++自1980年问世以来,已发展成在世界范围内有广泛影响的面向对象的编程语言。全书共5个部分,40章。第1部分介绍C++语言的基本元素,主要是从C语言中发展而来的部分。第2部分讲解C++类机制,这是大多数标准C++库实现都用到的语言特性,也是支持用户进行面向对象程序开发的机制。第3部分详细介绍标准C++库的组成部分,包括从C语言继承的标准C库函数和标准的C++字符串类、复数类、I/O流类等。第4部分介绍实现通用容器、算法和迭代器的标准模板库,读者将学习如何进行通用编程。第5部分讨论标准C++支持的高级特性,包括异常处理、名字空间、类型强制转换、运行时类型信息、区域表示等。

本书所带光盘提供了Quincy 2002集成开发环境和书中所有示例程序的源代码,读者可通过装入并运行示例程序加深对书中内容的理解。

本书自成体系,叙述简洁,深入浅出,是学习完整C++语言的优秀读物。本书适合希望使用C++进行编程的程序员,也可供从事面向对象程序设计的研究、开发人员参考。



Copyright ©2003 by Publishing House of Electronics Industry. Original English language edition copyright ©2003 by Wiley Publishing, Inc. All rights reserved including the right of reproduction in whole or in part in any form. This translation published by arrangement with Wiley Publishing, Inc.

本书中文简体专有翻译出版版权由美国Wiley Publishing, Inc.授予电子工业出版社及其所属今日电子杂志社。未经许可,不得以任何手段和形式复制或抄袭本书内容。该专有出版版权受法律保护,侵权必究。

著作权合同登记号 图字:01-2003-1534

图书在版编目(CIP)数据

C++大学自学教程(第7版)/(美)史蒂文斯(Stevens, A.)著;林瑶,蒋晓红,彭卫宁译.

—北京:电子工业出版社,2003.11

书名原文:Wiley's Teach Yourself C++

ISBN 7-5053-9260-3

I.C... II.①史...②林...③蒋...④彭... III.C语言-程序设计-高等学校-教材 IV.TP312

中国版本图书馆CIP数据核字(2003)第095877号

责任编辑:徐津平 王彦

排版制作:今日电子公司制作部

印刷:北京市增富印刷有限责任公司

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编:100036

经销:各地新华书店

开本:787×1092 1/16 印张:41.75 字数:1069千字

版次:2004年1月第1版 2004年1月第1次印刷

定价:69.00元(含光盘)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010) 88211980 68279077

译 者 序

就计算机编程来说，用“天下谁人不识君”来评价C++，恐怕不算夸张。自1980年被贝尔实验室公布以来，C++引起了计算机业界的广泛关注。正如本书所言，C++已发展成为具有世界级影响的面向对象语言，成为几代程序员的首选。在几乎所有的计算机和操作系统上都有C++的编译器，当代大多数小型机和微型机的应用程序大都用C++语言写成。因此，学习C++对计算机从业人员，尤其是程序员，是非常必要的。

本书的主题是讲解标准C++程序设计。作者Al Stevens具有丰富的C++编程、教学和写作经验。因此，在内容讲述和结构编排上很适合学习。内容丰富是本书一大特点。本书既包括C++语言的基本组成、类机制、C++库等基础内容，还包括标准模板库、异常处理、名字空间、RTTI、国际化等高级内容。自成体系、循序渐进是本书的又一特点。它不要求读者具备C语言背景知识，在内容安排上先易后难。第3个特点是程序示例丰富、易懂。作者尽量使用接近生活、可供程序员参考的示例，每个程序都展现了C++的特性，而且都能独立编译和链接，是完整程序。不愧为学习C++的优秀读物。

本书的中译本力求保留原版书的写作风格，同时尊重我国读者的语言习惯，力求译文通畅，技术术语准确。虽然有关C++的中译本很多，但不少技术词汇目前还没有标准译法，这给翻译造成不少困难。我们尽量挑选被大多数人公认的译法，例如把 locale 翻译为“区域表示”，把 facet 翻译为“刻画”，把 deference 翻译为“提领”，一些术语的译法还根据上下文加以调整。

参加本书翻译的有林瑶、蒋晓红、彭卫宁、赵刚、肖斌、赵滨华、杨宏、董韶军、康平军、韩锋、钱卫等。限于水平，错误和不妥之处在所难免，敬请广大读者批评指正。

译者

2003年8月

作者简介

自 1988 年以来，Al Stevens 一直为一流程序员杂志《Dr. Dobb's Journal》的“C 语言程序设计”专栏撰写文章，也是 MIS 出版社和 M&T Books 出版发行的十多本畅销书的作者。他自 1958 年起就担任专业程序员，更从 1978 年起成为独立的程序设计顾问和作家。凭借独到的洞察力和清晰的文风，Al Stevens 一直拥有大批忠实读者。

前 言

“观念在变，范围在变，没有什么一成不变。”

——吉米·巴菲特

本书是《C++ 大学自学教程》的第7版，第1版惠蒙 MIS 出版社1990年出版发行。这是一本指导用书，读者一边在PC机上运行程序示例，一边参照案头的《C++ 大学自学教程》，达到学通C++的目的。

随着程序设计的长足发展，程序设计已不同以往。计算机图书行业今非昔比，笔者本人也有了变化。这些先按下不表，先谈谈本书适合哪些读者，读者应该如何使用本书。

本书的读者群

本书假设读者对计算机程序设计已有一定了解，适合那些希望用C++编程的程序员。本书读者应该知道：

- ◆ 程序设计语言由程序员通过文本编辑器编写为文本文件，再由编译器或解释器处理为机器指令。
- ◆ 操作系统存储程序的机器代码(通常存储在磁盘上)，并将其读入内部存储器执行，从而达到运行程序的目的。
- ◆ 计算机不仅利用内部存储器临时存放程序，也存储数据。这些程序和数据也永久性地存储在其他地方，通常是磁盘上。
- ◆ 操作系统、编译器、程序员使用的编辑器等本身也是人们编写的程序，用来支持计算机用户和程序员的工作。

如果读者还不了解上述内容，请在学习C++或其他计算机程序设计语言之前先补上。

对非C语言程序员的用处

您是不是C语言程序员无关紧要，这本书的读者并不需要了解C语言。本书自成体系，指导读者学习C++语言。早先出版的许多C++书籍都假定读者已经了解C语言，《C++大学自学教程》一书的前4版也曾如此。自第5版起，此书讲授完整的C++语言，而第7版专为大多数对两种语言都不了解而又希望学习C++的程序员编写。

按照常规做法，学习C++前先要学习C语言，但事实证明这种做法不明智。我们现在已不再讲授所谓“C风格”的程序设计，避免让学生学习那些有害无益、学过后还必须忘掉的规矩。如果

读者还不了解C语言，那就按照目前流行的做法，精神饱满地从头学习C++吧。因此，笔者决定在本书中全面讲述C++。此书面向了解程序设计的人员，读者没必要掌握任何具体的程序设计语言。

对C语言程序员的用处

在C语言程序员看来，本书第1部分大都在介绍C++包含的C语言组件。C语言程序员也许会跳过这些章节，但我建议读者认真阅读这些内容。例如，第2章介绍了C++控制台输入/输出iostream对象，C++程序员用它来取代getchar，putchar，gets，puts，printf和scanf等标准C函数。

在阅读中，请注意观察C++语言的细微差别。例如，第4章中C++的goto语句行为与C语言的goto语句不同；C++中的void指针只有经过类型重载，才能被赋值给某个类型的指针；main函数必须被声明为返回int，并且不能在C++程序中被递归调用；必须对const对象进行初始化；变量不能被隐式说明为int类型；每个枚举成员是一个类型而不是一个int……如此种种，不一而足。我并未在书中每一处指出这些差别，而是把C++当成一个全新的主题来对待。所以，不要仅仅依赖所掌握的C语言知识而忽略本书第1部分的内容。请利用第1部分来复习已掌握的知识，找出差异，从而加强对C和C++之间异同点的理解。

本书内容

本书并不讲授针对某一特定平台的程序设计。它不讲Windows程序设计、UNIX操作系统或Linux程序设计，也不讲Macintosh程序设计或是针对任何其他特定平台的程序设计。本书只讲解标准C++程序设计。标准C++是独立于平台的语言规范，而标准C++编译器适用于前面提到的所有平台。本书所带的光盘包含了一套可在Win 32平台上运行的编译器。之所以采用Win 32是考虑到两个原因：一是我为Win 32开发软件，二是大多数读者都拥有Win 32。如果读者有不同的开发平台，可以将光盘中的程序示例装入该平台。如果该平台本身带有标准C++编译器，读者就可以在平台上运行并测试程序示例。

作者的变化

我先简要介绍一下自己，然后谈谈撰写《C++大学自学教程》第1版后自己的变化。为什么要说这些？很多读者可能并不关心（请跳过这一节），但有些读者多年来一直关注着我的工作，因此会有兴趣。

1990年以来，通过长期的程序设计和项目开发，我对C++的理解日益丰富。多年来，除了从事许多有关C++的项目外，我基本上每月都在《Dr. Dobbs's Journal》杂志的“C语言程序设计专栏”发表小型程序。

发表代码就像在大庭广众下公开编程，不能隐藏任何东西。你不仅要注意如何用代码解决问题，还要顾及到如何让别人轻松地理解代码。你必须学习和应用程序设计者公用的编码约定，必

须理解这些约定得以定义和接受的原因。你必须了解哪些常规做法安全，哪些先天就有危险。随着约定的改进，人也在变化。我一直在改变自己，努力朝好的方向发展。

十多年前撰写本书第1版时，我并不是C++的忠实信徒，当时编写的大多数程序都是用C语言完成的。然而，今非昔比。许多用其他语言成功编程的程序员看不出改变的必要，我却认为C++的年轻支持者数量与日俱增。作为作者，我有责任满足读者的需要。在投入这项工作后，我才真正发现C++和面向对象程序设计的魅力所在。

这些年来，当其他类似语言被大众接受时，我一直在冷眼旁观。Java是继C++之后第一个被接受的。Java针对的开发环境与C++不同，它基于Web应用，我目前尚无必要为此目的编写程序。C#是最近亮相的语言。这两种语言我都读过，并未从中发现能让我改变信仰的东西。对于C++不能实现或能够实现得很好的程序，我都没有创作欲。

以上观点仅代表笔者一家之言，并非对Java或C#的批评。这两种语言各有长处和拥护者。我的观点只反映出一种实际的选择，因为相比使用其他语言的程序员，C++程序员的需求量要大得多。但我不能承诺这种需求还将持续多久，十多年前我也对C语言说过同样的话。

计算机书籍的变化

MIS出版社的出版标志改变了。1990年，MIS出版社还是位于俄勒冈州Portland的一个新兴小出版社，由创办者Bob Williams经营。当时，谁都知道Bob曾挨家挨户地推销过计算机书籍。但Bob最终创建起自己的计算机图书事业。他本人撰写并出版了一本某种流行电子表格程序的用户手册。这本手册使人们能够非授权地复制该电子表格软件，而且用户在运行程序时不需要正式的用户指南（正式指南本来只能在购买正版程序后才能得到）。当时，Bob的书并不受软件公司欢迎，他们起诉了他，但Bob胜诉。由于Bob的开创性努力和官司带来的知名度，商用PC应用程序的书在当地蓬勃发展起来。Bob在创建名为MIS出版社的小型出版公司后，拥有了一批稳定作者和一系列出版书籍。然后，他将出版标志卖给Henry Holt，后者卖给IDG图书公司，而IDG又卖给Wiley。到目前为止，Wiley还保持该标志的拥有权。老板几易，我为MIS出版公司撰写书籍的事情也几经变化，以符合各位老板的要求和喜好。当然，这些变化有好有坏。在此期间，对计算机程序设计图书的需求也发生了变化，这些变化同样有好有坏。

C++ 语言的变化

C/C++程序设计语言的历史比许多使用C/C++的程序员的年龄还老。C语言在20世纪70年代初期问世，C++问世于1980年。自那以后，C和C++从实验室程序员专为个人使用而开发的小型工具，发展成有世界影响的面向对象语言，成为几代程序员的首选。C++编译器几乎可以用于所有计算机和操作系统，当代大多数小型机和微型机的应用程序都是用C++语言编写的，因而C++享有盛誉。

1990年，C++只是一种小型的程序设计语言，它在久负盛名的C语言上增加了面向对象的应用扩展。《C++大学自学教程》的第1版不到300页，几乎涵盖了C++的所有内容。10年后，随着国际标准化进程，C++有了长足的发展。如今，第7版的内容已经是第1版的两到三倍。

1998年，C++ 被批准为国际标准程序设计语言，发布文档很长，题为《国际标准 ISO/IEC: 14882(E)，程序设计语言 --C++》。这是志愿人员长达9年辛勤工作的结晶，他们组成了所谓“X3J16委员会”，简称为“X3J16”或“委员会”，代表了工业界、政府、编译器公司以及几乎所有开发C++ 程序和相关素材的企业。

按惯例，程序设计语言标准委员会用准确清晰的术语定义了这门语言，他们称之为“艺术加工之前的整理”。他们关注那些被时间和实践证明为有用的语言特性，只有当编译器的实现与语言特性有冲突时才对特性做出修改。“X3J16”委员会并不限制自己的工作，委员会成员抓住机会解决现有语言中的问题，同时增加新特性。

“程序设计语言标准化”一词意味着“程序设计语言发明”，委员会增加的许多新特性都未经测试和证明。“X3J16”这种非常规做法引起了极大争议。与传统C++相比，标准C++是如此之大，如此不同，以致在正式发布标准定义4年后，C++的某些特性仍然未被编译器充分实现。

此外，与小巧整洁的C语言相比，标准C++需要更多硬件设备，编译时间更长，生成的可执行二进制文件更大。当然，现在的计算机容量更大、速度更快，某些负面性能因素已被排除，程序员有更丰富的编码方法可供选择。

“程序设计语言标准化”到底是好事还是坏事？我认为在很大程度上是好事。“委员会”发明的许多已实现和使用的特性，大多数通过了实践检验。人们正像“委员会”所希望的那样利用这些特性，而这些特性也达到了预期目的。另一方面，如果有人不需要、不关心某一语言特性，大可不使用它。只要编译器设计优良，用户不会付出任何代价。

编程的变化

自1990年以来，主流编程方法变化巨大。当时，C是PC机程序设计语言的选择之一，大多数主流应用程序都为文本模式的运行环境编写。C++虽然已经问世，但还未流行。1990年发生的一起事件促使情况改变。微软公司开发了Windows 3x，即图形用户界面运行环境，它的问世注定要改变人们使用个人计算机的方式。PC机用户需要在新型图形环境下运行应用程序，因此，程序员开始着手研究如何编写满足人们需求的程序。

用C语言进行Windows程序设计是一项令人生畏的工作。程序员需要编写基于事件和消息的特殊编程模型，需要学习大量的应用编程接口(API)，其中包含成百上千的消息代码、函数和结构。有人估计，一个优秀的程序员至少要花一年时间，经过大量研究和实践，才能成为一名胜任的Windows C程序员。

Windows程序的市场需求广阔，因而程序员愿意编写Windows程序，但Windows编程难度很大。开发编译器的公司，如著名的微软和Borland，将Windows API打包装入所谓“应用程序框架”中，以此解决这一难题。Windows基于消息、事件驱动的编程模型是C++类与生俱来的。框架库(如微软基础类MFC)和Borland的对象Windows库(OWL)很快在程序员中风靡一时，他们所要做的就是学习C++语言。由于其中大多数人已经掌握了C语言，所以学习过程并不艰难。通过使用C++，程序员很可能迅速成为高产的Windows程序员。虽然这并不一定准确，但诸多巧合汇集起来，使C++逐渐成为人们的选择。

使用其他平台的程序员可能不同意我的观点，或者至少对我的看法不满，即 Windows 编程促成了 C++ 在主流程序员中的成功。这个问题见仁见智。为 Windows 平台编写的程序数量之大，Windows 的用户人数之多，足以证明我的观点。

关于 Wiley 公司的《C++ 大学自学教程》

本书从第 1 章起用一系列程序示例引导读者进行 C++ 学习。书和光盘中的程序都含有 C++ 源代码，读者可以编译和执行。读者在学习的过程中，应装入并运行这些程序示例，从而最大程度地受益。

程序示例按照先易后难的顺序编排，先介绍简单概念，然后进入更复杂的主题。每一个程序示例都建立在前例的基础上，读者要按部就班地学习。

程序示例很小，不是非常完善、实用的程序，我也不打算让它们成为正式使用的程序。每个程序都展现了 C++ 的一个专门特性。不过，从独立编译和链接的意义上讲，这些示例都是完整的程序。

在讲课过程中，我尽量使用程序员能够参考的示例。有些程序设计的书中常使用毫无意义、晦涩难懂的示例来解释语言特性，比如使用代表昆虫种类、家具等级和某人衣柜中衣服清单的类库例子。现实中谁也不会为那些示例编写程序。读者可在某些书中看到实现计算机屏幕显示、串和对象容器的示例，而这些内容早已由语言和操作系统提供了。我曾见过完全抽象的示例，使用如 X 等毫无意义的名字，根本不能很好地说明语言特性。许多书中除了代码段根本没有完整的程序，读者不能对其编译、测试和观察。

尽管上述罗列的程序设计书籍不乏成功者，我还是尽量避免采用其做法。我的目标是提供一些根据实物编写的小型、简明的代码示例，而且能够被程序员理解，如日期、工资和图形形状等。这样，当读者学习语言时，不必费力理解问题的定义域。读者也许永远不会为工资管理系统编写代码，但如果读者有工作，他至少可以明白这是怎么回事。不过，尽管有良好的意愿，我的书中也难免偶尔采用深奥难懂的程序示例。但总体来说，简明扼要的示例可以防止读者误入歧途，不得要领。

本书所附编译器

本书所附光盘中带有 Quincy 2002，这是基于 Windows (Windows 95 或更高版本) 的 C++ 程序开发系统，读者可以利用它编译和运行程序示例。Quincy 是所谓的“集成开发环境 (IDE)”，将程序员的编辑器、编译器和调试器集成在一个程序包中。本环境提供 Windows 界面。附录 C 是“Quincy 用户快速入门指南”，其中说明了如何让 Quincy 和编译系统升级。

循环的学习过程

程序设计语言这类复杂问题常常将读者带入周而复始的学习过程中。要学习某项课程，就要先具备某种知识，而这种知识又以其他知识为前提。C++ 的 `std::iostream` 类就是一个很好的例子，

它定义了对标准输入 / 输出设备（典型的设备是系统控制台）进行读写的对象。为充分理解这些知识，读者必须先了解 C++ 名字空间、类、对象和运算符重载，所有这些都是高级 C++ 的内容。而为了学习这些高层次的知识，读者必须运行一些练习程序，这些程序要用到 `std::iostream` 类支持的键盘和屏幕控制台设备。

要相信本书，要有耐心，功到自然成。某些章节可能会提前引用还没有学到的内容。如果对某个内容的讨论不甚清楚，建议你做笔记，等到从其他章节中学到更多知识后，再回到这项讨论上来。

本书的结构

本书第 1 部分的第 1 章至第 13 章介绍了 C++ 语言的基本组成，主要是与 C 语言基本相似、同时也包含新成份的内容。

第 1 章介绍 C++ 编程。本章包括 C++ 简史和 C++ 程序设计语言综述，其中专门介绍了 `main` 函数，这是一切 C++ 程序的入口点。

从第 2 章开始学习编写程序。讲解有关 `main` 函数的详细内容，如何在编码中加入注释，如何包含标准库的头文件，如何执行简单的控制台输出以观察程序结果。还将学习 C++ 表达式和赋值语句。读者将了解如何从键盘和显示屏幕读数据。本章还讲授 C++ 数据类型，包括字符、整数和浮点数。

第 3 章的重点是函数。本章介绍 C++ 如何声明、定义和调用函数，如何传递形参对应的实参并获得返回值。读者将学习如何将 C++ 函数组织成嵌套代码块，如何将 C++ 程序模块与用其他语言（如 C 语言）编写的模块进行链接。

第 4 章介绍程序流程控制。将学习 `if...else`, `do`, `while`, `for`, `goto`, `switch`, `break`, `continue` 和 `return` 语句。

第 5 章涉及程序变量的组织。本章讲授变量作用域，将变量转换为常量的限定符和控制计算机在何处存储变量的限定符。

第 6 章讲授结构、类和联合。它们能够将数据组织成存储器中的数据记录格式。本章还通过对 `struct` 的扩展，介绍类的机制。读者可以学习内部和用户定义类型的数据抽象特性，数据成员和成员函数，以及访问说明符。本章介绍了 C++ 与 C 语言的主要区别。

第 7 章讲授有关 C++ 数据的详细知识，如枚举常量、数组定义、程序堆和栈、C 风格的强制类型转换等。本章还介绍 C++ 强制类型转换的基础知识。有关强制类型转换机制的高级课程将在第 37 章中讲授。

第 8 章讲授程序的指针变量和存储器地址。

第 9 章介绍引用变量。这是 C++ 的特性，与指针类似，但不完全相同。

第 10 章讲授递归，这是当今大多数程序设计语言的特性，这个特性允许函数自我调用。本章解释其工作原理、使用价值、何时使用、为何在某些程序设计情况下要避免使用，以及怎样避免。

第11章的重点是C++预处理器。利用它可以定义宏，可编写编译时条件表达式，该条件表达式被用来控制程序的编译方式。

第12章解释函数模板。利用它可以生成适用于不同数据类型的通用函数。本章将学习模板的形参和实参，以及更高级的模板主题，如模板重载等。

第13章讲授程序的组织结构。它的命令行实参将多个已编译模块链接成一个可执行模块，还将说明如何在一个程序中包含C和C++两种源代码文件。

本书第2部分从第14章至第25章，讲授C++类机制，这是大多数标准C++库实现都用到的语言特性，也是用户扩展C++的一种方法。

第14章讨论类。说明如何设计含有数据成员和成员函数的类。

第15章介绍类的构造函数和析构函数。它们是两个特殊的成员函数：构造函数是使用类的程序对该类的对象进行实例化时执行的特殊成员函数，析构函数是程序结束一个类的对象时执行的特殊成员函数。

第16章是关于C++类型转换的内容。通过使用程序中的转换函数，一个类的对象可以自动转换为其他类的对象。

第17章介绍类对象数组，以及为满足其要求在类设计中应有的特别考虑。

第18章详细讲解类成员。介绍它们如何工作，如何对声明方式做出反映。

第19章讲解C++“友元”函数的特性。它使函数可以访问类的隐藏成员。

第20章说明类如何使用堆存储器，以及程序如何修改这一行为。

第21章介绍重载运算符。当这一特性用在包含算术、关系和其他运算符的表达式中时，类对象能够模仿内部数据类型的行为。

第22章说明如何创建和使用类库。本章构建了两个供后续章节使用的类库。

第23章介绍类继承。这一特性使用户能够创建由基类和派生类组成的面向对象的类层次结构。本章还学习被称为多态的面向对象属性。

第24章主要介绍多重继承。这是C++的特性，可使派生类继承多个基类的特性。

第25章讨论类模板。这是C++的特性，可使用户创建通用的参数化类。

本书第3部分从第26章至第29章，详细介绍标准C++库的各个组成部分，包括C++继承自标准C的部分。

第26章讨论标准C库函数。本章将学习如何为程序所调用的标准函数调用头文件，还将学习字符串函数、存储器分配函数、数学函数等。

第27章介绍标准C++库。通过学习本章，可增加对输入/输出类*iostream*的了解。本章也介绍标准的*string*和*complex*类。

第28章介绍*iostream*和*stringstream*类中字符串数据的管理和格式化。

第29章讲解使用C++流来操纵磁盘文件，包括读写文件的流类。还将学习文本和二进制文件之间的差别。

本书第4部分从第30章至第34章，介绍标准模板库 (STL)，这是一个模板类的库，实现通用容器、算法和迭代器。读者在第4部分中将学习通用编程。

第30章介绍在STL中定义的类的各种类型，包括序列、结合容器、算法和迭代器等对象的类。读者将学习由STL实现并支持的通用编程模型。

第31章讲授如何用STL序列类进行程序设计。这一功能实现链表、队列、栈、向量等。

第32章说明如何使用结合容器类。结合容器类实现 map, set 和 bitset。

第33章主要解释通用算法。这是一个函数库，可使程序处理STL对象定义的数据。这种处理包括计算、排序和分组。

第34章解释STL迭代器。这是指针的一个特殊类型，可使程序对容器进行前向或反向迭代访问。本章还学习如何使用迭代器获取在容器中存储的元素信息。

本书第5部分从第35章至第40章，讨论标准C++支持的高级特性。

第35章讲授异常处理。这一特性可使程序以有序的方式抛出或捕捉异常。

第36章介绍名字空间，包括关于如何定义名字空间的讨论。本章还解释名字空间的作用域解析、未命名的名字空间和名字空间别名。

第37章介绍高级的C++类型强制转换机制，包括动态强制、静态强制和常量强制。

第38章介绍C++运行时类型信息 (RTTI)。这是程序在运行时确定对象有关信息的方法。

第39章介绍C++区域表示 (locale) 类，以及如何使用区域表示类让应用程序国际化。读者将学习设置区域表示，以及如何按区域表示所需的格式显示信息。

第40章介绍面向对象设计和编程的技巧，包括抽象、封装、继承和多态等。

附录A包含本书示例程序使用的类库源代码清单。

附录B介绍本书附带的光盘。

附录C是Quincy 2002集成开发环境的简要使用指南。读者可用该环境编译、调试和运行程序练习，并可用其编写自己的C++程序练习。

附录D用便于参照的表格形式介绍C++语言的组成。

附录E是文献参考书目。

词汇表定义了部分常用的C++程序设计专用术语。如果读者对本书中某个词汇不熟悉，可查询词汇表。如果表中没有，可发送邮件至 astevens@ddj.com 向笔者查询。

Al Stevens

2003年3月

目 录

第 1 部分 C++ 语言	1
第 1 章 C++ 程序设计概述	3
1.1 C++ 简史	3
1.2 C++ 介绍	4
1.2.1 核心 C++ 语言	4
1.2.1.1 函数	4
1.2.1.2 类	6
1.2.1.3 main 函数	6
1.2.1.4 剖析函数	7
1.2.1.5 语句	7
1.2.1.6 程序流控制	10
1.2.1.7 异常处理	10
1.2.1.8 模板机制	10
1.2.1.9 名字空间	10
1.2.1.10 声明次序	10
1.2.2 标准 C++ 库	11
1.2.3 源代码文件	12
1.3 小结	12
第 2 章 编写简单的 C++ 程序	13
2.1 第一个程序	13
2.1.1 #include 指令	13
2.1.2 空白	14
2.1.3 main 函数声明	14
2.1.4 main 函数语句块	14
2.1.5 源代码注释	15
2.1.6 向控制台写入信息	15
2.1.7 return 语句	15
2.1.8 终止语句块	16
2.2 标识符	16
2.3 关键字	16
2.4 标准输出流	17
2.5 变量	18
2.5.1 布尔型	18
2.5.2 字符型	19

2.5.3	wchar_t 型	20
2.5.4	整型	21
2.5.5	浮点数	22
2.6	常量	22
2.6.1	字符常量	23
2.6.2	转义序列	23
2.6.3	整数常量	24
2.6.4	浮点常量	24
2.6.5	地址常量	25
2.6.6	字符串常量	25
2.7	表达式	26
2.8	赋值	26
2.9	以逗号分隔的声明	27
2.10	表达式中的运算符	28
2.10.1	算术运算符	28
2.10.2	逻辑运算符	29
2.10.3	位逻辑运算符	30
2.10.4	移位运算符	31
2.10.5	关系运算符	32
2.10.6	增量运算符和减量运算符	33
2.10.7	赋值运算符	34
2.10.8	复合赋值运算符	35
2.10.9	条件运算符	36
2.10.10	逗号运算符	38
2.11	结合性和优先级	38
2.11.1	结合性	39
2.11.2	优先级	40
2.12	表达式何时不被求值	40
2.13	初始式	41
2.14	类型转换	42
2.15	控制台输入/输出	43
2.15.1	标准输出流	43
2.15.2	格式化输出	44
2.15.3	标准错误流	45
2.15.4	标准输入流	45
2.16	一个问题	46
2.17	小结	46
第3章	函数	47
3.1	再论 main 函数	48
3.2	实参与形参	48

3.3 用原型声明函数	49
3.3.1 未命名的形参类型	49
3.3.2 典型的原型	49
3.3.3 返回 void 的函数	49
3.3.4 没有形参的函数	50
3.3.5 形参表可变的函数	50
3.3.6 不返回值且没有形参的函数	50
3.3.7 标准库原型	50
3.3.8 没有原型的函数	50
3.4 定义和调用函数	51
3.5 函数返回	51
3.6 函数返回值	53
3.7 实参的传递和使用	53
3.7.1 标识符作用域	54
3.7.2 利用函数调用实现初始化	55
3.7.3 初始式的执行顺序	55
3.7.4 传递多个实参	55
3.7.5 用做实参的函数调用	56
3.7.6 值传递	56
3.7.7 实参和返回值的类型转换	56
3.8 未命名的函数形参	56
3.9 默认函数实参	57
3.10 内联函数	58
3.11 递归	59
3.12 函数重载	59
3.12.1 不同操作的重载	59
3.12.2 不同格式的重载	60
3.13 小结	61
第 4 章 程序流程控制	63
4.1 语句块	63
4.1.1 嵌套深度	63
4.1.2 缩进风格	64
4.2 条件测试	65
4.2.1 if 语句	65
4.2.2 if...else 语句	67
4.2.3 else if 语句	68
4.2.4 switch...case 语句	69
4.2.5 if 条件表达式内的声明	70
4.3 循环迭代	70
4.3.1 while 语句	71

4.3.2 do...while 语句	72
4.3.3 复合条件运算符	73
4.3.4 for 语句	74
4.3.5 for 条件表达式中的声明	75
4.4 循环控制	75
4.4.1 break 语句	75
4.4.2 continue 语句	76
4.5 跳转语句	77
4.5.1 goto 的非法使用	77
4.5.2 C++ 的 goto 与 C 的 goto	78
4.5.3 修改非法 goto	78
4.5.4 使用 goto 的争议	79
4.6 小结	80
第 5 章 C++ 数据类型结构	81
5.1 作用域	81
5.1.1 全局作用域	81
5.1.2 局部作用域	82
5.1.3 全局作用域解析运算符	83
5.1.4 文件作用域	84
5.1.5 作用域、可见性和生存期	85
5.2 存储类	86
5.2.1 auto 存储类	86
5.2.2 static 存储类	86
5.2.3 extern 存储类	87
5.2.4 register 存储类	89
5.3 初始默认值	89
5.4 类型限定词	89
5.4.1 const 类型限定词	90
5.4.2 volatile 类型限定词	90
5.5 程序存储器体系结构	91
5.5.1 外部变量存储器和静态变量存储器	92
5.5.2 堆	92
5.5.3 栈	92
5.6 小结	93
第 6 章 结构和联合	95
6.1 结构	95
6.1.1 声明结构	96
6.1.2 定义结构变量	96