

标准C++ 程序设计教程

林丽闽 褚尚军 等编著



電子工業出版社

Publishing House of Electronics Industry

www.phei.com.cn

标准 C++ 程序设计教程

林丽闽 褚尚军 等编著

電子工業出版社
Publishing House of Electronics Industry
北京 · Beijing

内 容 简 介

C++的国际性标准版本“标准C++”于1998年正式推出,本书以此标准为基础,对C++语言不同层次的知识进行了全面介绍。

本书以知识单元为基本构件,共分为七个单元:C++函数、控制结构与数据类型;C++库函数与函数模板;C++类与类模板;C++标准库;C++ STL;深入学习C++;附录。本书从最基本的内容讲起,对C++强大的功能做了详细介绍,并结合C++的最新功能,详细论述了面向对象编程的思想。本书针对那些比较难懂的高深主题也做了深入浅出的介绍,比如模板(包括标准模板库)、名字空间、运行时类型信息和区域表示等,这些功能对开发大型复杂系统的程序设计人员非常重要。因此,无论是新手还是有经验的程序员,都可以从本书中找到丰富的信息。使用本书,读者不仅可以学到技术,还可以得到最好的练习,使自己的程序设计水准达到专业水平。

本书是学习和使用C/C++的优秀图书,最大的特点是内容综合、完整,叙述清晰、易懂,实例典型、丰富,练习、习题实用。同时,由于本书以知识单元为基本构件,便于拆分和重组,可适合不同层次的教学需要。可作为大专院校计算机专业的教材,也可作为从事计算机工作的科技人员的自学参考书。

本书版权归电子工业出版社所有,未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,翻版必究。

图书在版编目(CIP)数据

标准C++程序设计教程/林丽闽编著.-北京:电子工业出版社,2003.5
ISBN 7-5053-8651-4

I.标... II.林... III.C语言-程序设计-教材 IV.TP312

中国版本图书馆CIP数据核字(2003)第026915号

责任编辑:徐津平

印 刷:北京天竺颖华印刷厂

出版发行:电子工业出版社 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:48.5 字数:1241千字

版 次:2003年6月第1版 2003年6月第1次印刷

定 价:69.00元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。联系电话:88211980 68279077

序 言

C++ 语言是目前软件开发者们最主要的程序设计语言，这也是它受到广泛传播并且成为标准的原因。现在也有新的程序设计语言出现，但这些语言仍在演化之中，C++ 语言仍然是全世界专业程序设计人员进行强大的、可移植程序设计的首选语言。

C++ 已经有了一个公认的标准，并且许多编译器支持该标准中的绝大多数特性，所以应该抓紧时间深入学习这门内容丰富、功能强大的语言，并且用它进行程序设计。本书几乎完全是在讨论 ANSI/ISO C++，即标准 C++，也就是说书中对 C++ 语言的定义和介绍，完全遵循 ANSI/ISO 国际标准。市面上有许多介绍 C++ 语言的书籍，可谓琳琅满目、各有千秋。但是，本书是一本将 C++ 语言以全面而实用的方式介绍给读者的优秀书籍。

本书的读者对象

本书的主要读者是那些对 C 和 C++ 都不懂，但是希望学习 C++ 的人员。

按照传统的观念，在学习 C++ 之前，首先应该学习 C。在过去的几年中，所出版的大多数关于 C++ 的书籍，都假设读者已经了解了 C 语言（到目前为止，大多数 C 程序员都已经学习过 C++）。但这种观念现在已经过时了，因为学习了 C 风格的程序设计，未免会带有一些不必要的、不合理的偏见。如果还没有学习 C，那么就应该直接学习 C++，这是目前流行的比较合理的观念。因此我们在编写本书时，将多年教学的经验与这种新观念结合起来，使本书自成体系，成为 C++ 语言的完整教程，不要求读者预先掌握 C 语言。作为 C++ 的一个完整的学习教程，主要读者对象是那些了解程序设计的人，但是不要求一定掌握某个具体的程序设计语言。当然，C 程序员也可以使用本书学习 C++，不过由于书中 C++ 是作为一个完整的新主题介绍的，因此 C 程序员不要完全依赖于已经掌握的 C 的有关知识，因为 C++ 语言与 C 语言还是有一些细微差别的。

实例程序

本书自始至终在讲述 C++ 的过程中提供了一系列实例程序，每一个程序都包含了 C++ 程序源代码，可以对这些源代码进行编译和执行。我们根据多年学习和教授程序设计语言的实际经验，并没有像其他图书那样将这些源代码以电子文件的形式（通常保存在 CD-ROM 光盘上附在图书的后面）给出，而是要求读者在阅读本书学习 C++ 语言时，一边学习一边将书中的程序源代码亲手输入，这样就不仅仅学会了在程序中如何使用正确的方法和程序设计机制，而且印象会很深，学到的知识会很扎实，同时学会了程序的编写格式、风格和程序的调试，学会了如何创建易于阅读、调试和更改的程序。从而，从这个教程中获得最大的收益。

本书的实例程序按照内容顺序编排，先从比较简单的概念开始，循序渐进，一直到比较复杂的内容。后面的程序是建立在前面程序所介绍知识的基础之上的，因此应该按照实例程序出现的先后顺序阅读。实例程序比较小，但对学习和掌握 C++ 语言很实用。每个程序对 C++ 的某个特定的功能做了例证，同时每个实例程序又是可以进行独立编译和链接的完整程序。

对于本书的所有实例程序，可以使用任何标准 C++ 编译器来进行编译。我们在编写本书时，所用的是基于 Windows 的 C++ 编译系统 Quincy 99 (www.ddj.com/ftp/packages/quincy99/quincy99exe.zip/)。

使用标准 C++ 的原因

前面提到，本书几乎完全是在讨论 ANSI/ISO C++，即标准 C++。这是因为，通过坚持使用标准 C++，可以在最大程度上使代码具有可移植性。本书并不是只面向 Windows C++ 程序员、DOS C++ 程序员或 UNIX C++ 程序员，而是面向每个人的。本书的程序都采用标准 C++ 编写，因此，它面向所有的平台。

使用本书学习 C++ 的方法

程序设计是一门复杂的课程，需要通过循序渐进的方法进行学习。使用本书时，要学好每一章的内容，都依赖于对前面一章内容的掌握，有时还依赖于新的没有讲到的内容。比如 C++ 中的 `iostream` 类，这个类定义了一些对象，实现从控制台的输入和输出，要充分理解这个概念，需要掌握 C++ 中有关类和重载运算符的概念，这些都是 C++ 的高级内容。然而，要学习这些高级内容，必须学习有关键盘和显示设备的知识。`iostream` 类集合实现了这些设备的功能，所以在使用系统定义的 `cin` 和 `cout` 对象时，并不要求知道其实现的细节，但是应该相信其操作的正确性，最终在后面的学习中把它弄懂。

本书最终都会介绍预先使用而未讲述的内容，在学习中要有信心并且保持耐心，所有的知识都会讲述清楚。在本书中会出现这样的情况，就是有许多提前用到但是尚未讲述的知识。如果对这些知识不够明白，可以做一个记号，在后面学习到有关内容之后，再回到所标记之处，重新温习。

本书内容浏览

本书以 1998 年正式推出的“标准 C++”为基础，从最基本的内容讲起，对 C++ 强大的功能做了详细介绍，并结合 C++ 的最新功能，详细论述了面向对象的程序设计思想。本书对那些比较难懂的高深主题也做了深入浅出的介绍，比如模板（包括标准模板库）、名字空间、运行时类型信息和区域表示等，这些功能对开发大型复杂系统的程序设计人员非常重要。因此，无论是新手还是有经验的程序员，都可以从本书中找到丰富的信息。

本书以知识单元为基本构件，共分为七个单元：C++ 函数、控制结构与数据类型；C++ 库函数与函数模板；C++ 类与类模板；C++ 标准库；C++ STL；深入学习 C++；附录。在每个单元的后面都有大量练习，包括简单的概念复习，编写各种 C++ 语句、函数、类和程序，这些都有助于读者自学，有助于读者检验自己对所学内容的掌握，有助于读者建立自信心。各个单元所有练习的参考答案附在书后。另外，我们在本书的最后收录了一个近 300 道习题的习题集，供授课老师和读者参考使用。

下面详细介绍每个单元和各章的内容。

第 1 单元 C++ 函数、控制结构与数据类型

第 1 章“了解 C++”介绍了计算机语言的概念和 C++ 程序设计概念，指明了读者需要预先掌握的知识。在本章中，对 C 和 C++ 的历史和未来做了简要介绍，对 C++ 程序设计语言进

行了概述。

第2章“编写简单的C++程序”引导读者开始编写简单的C++程序，其中对主函数main()（它是C++程序的入口点）做了详尽介绍，并且介绍了如何为代码加注释，如何把标准库的头文件包含在程序中，如何实现简单的控制台输出以观察程序运行结果。本章还讲述了C++的表达式和赋值语句，以及如何从键盘读入数据并在屏幕上显示。

第3章“C++函数”集中讨论了函数，包括如何在C++程序中声明、定义和调用函数，如何向函数传递参数，如何从函数返回值，以及如何书写C++函数的语句块结构。此外，本章还介绍了如何把C++的程序模块与其他语言（比如C语言）编写的程序模块链接起来。

第4章“C++控制结构”介绍了程序流程控制，其中包括if...else, do, while, for, goto, switch, break, continue 和 return 语句。

第5章“C++数据类型”讲述了C++的数据类型，包括字符型、整型和浮点类型。讲述了如何使用常量，介绍了变量有效范围的概念，以及如何用基本数据类型构造数组、结构和联合等。

第6章“指针、地址与引用”讲述了如何在程序中使用指针，如何使用typedef操作符定义新的数据类型标识符。还介绍了有关递归的概念，以及C++引用型变量。

第2单元 C++ 库函数与函数模板

第7章“C++库函数和宏”讨论了标准C库函数中的一部分内容。介绍了如何使用头文件实现在程序中调用标准函数，还介绍了字符串函数、内存分配函数、数学函数以及宏等。

第8章“C++程序的预处理”介绍了C++中的预处理功能，包括定义宏、编写条件编译表达式来控制程序编译过程。

第9章“C++结构与类”对结构体进行扩展并引入类机制，介绍了内部数据类型和用户定义数据类型的数据抽象特性，还介绍了数据成员、成员函数和访问控制符。

第10章“C++函数模板”介绍了函数模板。使用函数模板可以创建通用函数，用于不同类型的数据。本章介绍了有关模板参数的内容，还介绍了关于模板的高级主题，比如模板重载。

第3单元 C++ 类与类模板

第11章“C++类”继续讨论类的相关内容，包括构造函数、析构函数、转换函数、赋值函数、类对象数组、类对象的内存分配等。

第12章“运算符重载”讨论了重载运算符，重载运算符给类对象赋予了行为，当其用于含有算术、关系等运算符的表达式中时，重载运算符使类对象可以模仿内部数据类型的行为。

第13章“类的继承”介绍了类的继承。使用类的继承，可以建立面向对象的类层次结构，层次结构由基类和派生类构成。本章还介绍了多态的概念，它是一种面向对象的特性。

第14章“类的多重继承”讲述了多重继承，它是一种语言特性，允许派生类从多个基类继承属性。

第15章“类模板”讨论了类模板，它是C++语言的特性，可以根据参数生成类。

第16章“面向对象的程序设计”描述了面向对象程序设计的技术，包括数据抽象、封装、继承和多态性等。

第4单元 C++ 标准库

第17章“标准库简介”讨论了标准C++的库，其中对iostream控制台输入/输出类进行了扩充介绍，还介绍了标准字符串类和复数类，以及Standard Template Library（标准模板库）的容器类集合。

第 18 章“格式化输出与字符串处理”介绍了 iostreams 的管理和格式化。

第 19 章“C++ 输入/输出流”详细介绍了 C++ 流的概念和磁盘文件操作。在读写文件时可以使用不同的流类。本章还介绍了如何处理文本文件和二进制文件。

第 5 单元 C++ STL

第 20 章“STL 概述”介绍了 STL（标准模板库）中定义的各种类型的类，包括序列、结合容器、算法和指示器。

第 21 章“序列在程序中的使用”讲述了如何使用 STL 的序列类进行编程。序列类可以实现链表、队列、堆栈、向量等。

第 22 章“结合容器”介绍了如何使用结合容器类。结合容器类可以实现映射、集合和位集合。

第 23 章“通用算法”讨论了函数库中的通用算法。这些算法可以使程序处理定义在 STL 中的对象数据，处理过程包括计数、排序和分组。

第 24 章“指示器”介绍了 STL 指示器，它是一种特殊类型的指针，可以使程序在容器中沿正序或者倒序移动。本章还介绍了如何使用指示器访问存储在容器中元素的信息。

第 25 章“C++ 异常处理”详细介绍了异常处理，包括如何使程序以有序的方式处理异常。

第 6 单元 深入学习 C++

第 26 章“标准 C++ 语言补充”介绍了名字空间，包括如何定义名字空间、名字空间的范围、没有名称的名字空间、名字空间的别名。另外，还介绍了 C++ 的运行时类型信息（RTTI）和新风格类型转换，包括动态类型转换、静态类型转换和常类型转换。

第 27 章“区域表示”讲述了 C++ 中的 locale 类，包括如何使用 locale 类实现应用程序的国际化，如何对 locale 进行设置，以及如何根据 locale 要求的形式显示信息。

第 7 单元 附录

附录 A“数值系统”介绍了二进制、八进制、十进制和十六进制数值系统，并考虑如何转换进制和介绍反码与补码的二进制表示。

附录 B“运算符的优先级与结合律”将运算符优先级和结合律排列成表的形式，便于查阅和使用。

附录 C“ASCII 字符集”将表示各种语言字符的 Unicode 字符集的子集“ASCII 字符集”排列成表的形式，便于查阅和使用。

附录 D“C 与 C++ 的区别”介绍了 C 与 C++ 之间的细微差别。

附录 E“标准 C++ 特性总结”对标准 C++ 的特性进行了概括性总结。

附录 F“标准异常”对 C++ 中的标准异常以表格的形式列出，便于学习和使用。

附录 G“C++ 术语与概念”收录了 C++ 语言中的一些常用术语和基本概念。

本书的作者

本书的编写是集体劳动的结晶。本书第 1 章~第 8 章由林丽闽编写，第 9 章~第 15 章由褚尚军编写，其余部分由薛向东、张艳斌、文达、刘宇键、杨龙、戴伟、李梦想、汪华、韩军、李茉莉、张朝利、李铁强、刘建军、王海燕、张宇丰、左静、苏维义、赵卫滨、纪一平、陈文、严敏、魏玉明、蒋晓军、单东淇、杨阳、马平初、吴燕燕、宋春雨、李华、陈晓文、孙昆鹏、华云、赵晓强、王萍、陈波、李方义、皮鲲、叶枫冀、曹智君、李振林、李志刚、梅雪峰、潘同利、万绍斌等编写。

因作者水平有限，错误之处在所难免，敬请读者和同行批评指正。

目 录

序言	1
----------	---

第 1 单元 C++ 函数、控制结构与数据类型

第 1 章 了解 C++	3
--------------------	---

1.1 计算机语言	4
1.2 C 语言和 C++ 语言简史	6
1.3 从 C 语言到 C++ 语言的转换	8
1.4 C++ 的基本概念	9
1.5 C++ 语言及其增强类型	11
1.6 C++ 环境	12
1.7 C 语言与 C++ 语言的未来	13
1.8 C++ 基本知识点小结	13

第 2 章 编写简单的 C++ 程序	15
--------------------------	----

2.1 主函数 main()	15
实例 2-1 一个最小的 C++ 程序	15
2.2 简单的 C++ 程序	16
实例 2-2 自己编写的第一个 C++ 程序	16
2.2.1 #include 指令	17
2.2.2 空白	17
2.2.3 main()函数的声明	17
2.2.4 main()函数的作用范围	18
2.2.5 源程序中的注释	18
2.2.6 控制台输出	18
2.2.7 return 语句	18
2.2.8 程序的结束	18
2.3 标识符	18
2.4 关键字	19
2.5 标准输出流	19
2.6 变量	20
2.6.1 布尔型	21
实例 2-3 应用布尔型变量	21
2.6.2 字符型	21
实例 2-4 字符型变量的应用	21
2.6.3 wchar_t 型	22
实例 2-5 wchar_t 型变量的使用	22
2.6.4 整型	23

实例 2-6 引用 int 数据类型	24
2.6.5 浮点类型	24
实例 2-7 声明和引用浮点型变量	25
2.7 常量	25
2.7.1 字符型常量	25
2.7.2 转义字符	26
2.7.3 整型常量	26
2.7.4 浮点型常量	27
2.7.5 地址常量	27
2.7.6 字符串常量	27
实例 2-8 字符串常量的合并	28
2.8 表达式	28
2.9 赋值语句	29
实例 2-9 应用赋值语句和表达式	29
2.10 用逗号分隔开的声明语句	30
2.11 表达式中的运算符	31
2.11.1 算术运算符	31
实例 2-10 表达式赋值	31
2.11.2 逻辑运算符	32
2.11.3 位逻辑运算符	33
2.11.4 位移运算符	34
2.11.5 关系运算符	34
2.11.6 自增和自减运算符	36
实例 2-11 自增和自减运算符的位置	36
2.11.7 赋值运算符	37
实例 2-12 赋值运算符	38
2.11.8 复合赋值运算符	39
实例 2-13 复合赋值运算符应用	39
2.11.9 条件运算符	40
实例 2-14 计算罚款数目	40
2.11.10 逗号运算符	41
实例 2-15 逗号运算符应用	41
2.12 优先级和计算顺序	42
2.12.1 结合律	43
2.12.2 优先级	43
2.13 表达式不参加运算的情况	44
2.14 初始化	44
实例 2-16 不同变量的初始化	44
2.15 类型转换	45
实例 2-17 几种类型转换	46
2.16 控制台输入与输出	47
2.16.1 标准输出流	47
实例 2-18 利用 std::cout 优化代码	47

2.16.2 格式化输出	48
实例 2-19 用三种进制形式显示整数	48
2.16.3 标准错误流	49
2.16.4 标准输入流	49
实例 2-20 使用 cin 从键盘读入一个整数	49
实例 2-21 使用 std::cin 设备从键盘把字符串读入到字符串数组中	50
第 3 章 C++ 函数	51
3.1 函数简介	51
3.2 函数原型	52
3.2.1 无标识符的参数类型	52
3.2.2 函数原型实例	53
3.2.3 void 型函数	53
3.2.4 无参数的函数	53
3.2.5 参数个数可变的函数	53
3.2.6 既没有返回值也不带参数的函数	54
3.2.7 标准库函数的原型	54
3.2.8 不需要原型的函数	54
3.3 函数的定义与调用	54
实例 3-1 定义一个函数	54
3.4 函数的返回	55
实例 3-2 return 语句可在函数体内的任何位置	56
3.5 函数的返回值	57
实例 3-3 计算宽度	57
3.6 参数的传递和使用	58
实例 3-4 计算宽度——改进版本	58
3.6.1 标识符的作用域	59
3.6.2 通过函数调用来初始化	60
3.6.3 初始化执行的顺序	60
3.6.4 传递多个参数	60
实例 3-5 计算砖块的体积	60
3.6.5 用做实参的函数调用	61
3.6.6 值传递	61
3.6.7 实参和返回值的类型转换	61
3.7 无名的函数形参	62
3.8 函数的默认参数	62
实例 3-6 默认参数的用法	62
3.9 内联函数	63
实例 3-7 指定为内联函数	63
3.10 递归	64
3.11 函数重载	64
3.11.1 为完成不同的操作而重载	64
实例 3-8 复制字符串	65

3.11.2 为不同的格式而重载	66
实例 3-9 显示时间	66
3.12 安全连接	67
3.13 连接说明	67
实例 3-10 使用 C 的连接约定	68
第 4 章 C++ 控制结构	71
4.1 语句块	71
4.1.1 嵌套深度	71
4.1.2 缩进风格	71
4.2 条件语句	72
4.2.1 if 语句	72
实例 4-1 检验条件是否为零	72
实例 4-2 检验变量是否为零	73
实例 4-3 条件为真时执行	74
4.2.2 if...else 语句	75
实例 4-4 有条件地执行程序块	75
4.2.3 else...if 语句	75
实例 4-5 屏幕菜单程序	75
4.2.4 switch...case 语句	77
实例 4-6 检验整型变量的取值	77
4.2.5 在 if 语句的条件表达式里声明变量	78
4.3 循环语句	79
4.3.1 while 语句	79
实例 4-7 菜单选择与退出	79
4.3.2 do...while 语句	81
实例 4-8 猜谜游戏	81
4.3.3 for 语句	82
实例 4-9 限次循环显示	82
实例 4-10 遍历数组中的每一个元素	83
4.3.4 在 for 语句的条件表达式内声明变量	84
4.4 循环控制	84
4.4.1 break 语句	84
实例 4-11 终止并跳出循环	84
4.4.2 continue 语句	85
实例 4-12 中断但不跳出循环	85
4.5 跳转	86
实例 4-13 跳转到任意可执行语句	86
4.5.1 goto 语句的错误用法	87
4.5.2 C++ 的 goto 和 C 的 goto	88
4.5.3 goto 语句错误的改正	88
实例 4-14 优化的跳转程序之一	88
实例 4-15 优化的跳转程序之二	88
4.5.4 是否可以使用 goto 语句	89

第 5 章 C++ 数据类型	91
5.1 作用域	91
5.1.1 全局作用域	91
实例 5-1 具有全局作用域的外部变量	91
5.1.2 局部作用域	92
实例 5-2 具有局部作用域的变量	93
实例 5-3 作用域的屏蔽	93
5.1.3 全局作用域解析符	94
实例 5-4 使用全局作用域解析符	94
5.1.4 文件作用域	95
实例 5-5 static 存储类型修饰符	95
5.1.5 作用域和生存周期	96
实例 5-6 变量的生存周期	96
5.2 存储类型	97
5.2.1 自动存储类型	98
5.2.2 静态存储类型	98
实例 5-7 静态局部变量	98
5.2.3 外部存储类型	99
实例 5-8 使用外部变量	99
5.2.4 寄存器存储类型	100
实例 5-9 使用寄存器变量	100
5.3 初始化的默认值	101
5.4 类型修饰符	101
5.4.1 const 类型修饰符	101
实例 5-10 使用 const 型变量作为循环的上限	102
5.4.2 volatile 类型修饰符	102
实例 5-11 声明 volatile 型变量	102
5.5 用户定义的数据类型	103
5.5.1 结构的声明	103
5.5.2 结构变量的定义	103
5.5.3 对结构成员的引用	104
实例 5-12 结构的声明与使用	104
5.5.4 结构的初始化	105
实例 5-13 初始化结构变量	105
5.5.5 结构的嵌套	105
实例 5-14 声明两个结构	105
5.5.6 向函数传递结构和从函数返回结构	106
实例 5-15 传递和返回结构	107
5.6 联合数据类型	108
实例 5-16 使用联合	108
5.6.1 联合的初始化	109
实例 5-17 初始化一个包含结构的联合	109
5.6.2 匿名联合	110

实例 5-18 使用匿名联合	110
5.6.3 枚举常量	111
实例 5-19 在 switch 语句中使用枚举类型	111
5.7 数组	112
5.7.1 数组的声明	112
5.7.2 利用下标访问数组元素	113
5.7.3 数组的初始化	113
实例 5-20 一个简单的 7 元素整型数组	113
5.7.4 结构数组	114
实例 5-21 一个简单的结构数组	114
5.7.5 多维数组	115
实例 5-22 二维数组——税收数据表	115
5.7.6 字符型数组	116
实例 5-23 初始化字符型数组	117
第 6 章 指针、地址与引用	119
6.1 指针和地址	119
6.1.1 指向内部数据类型的指针	120
实例 6-1 指向 C++ 内部数据类型的指针	120
6.1.2 指针的算术运算	121
实例 6-2 指针运算	121
实例 6-3 利用指针变量对数组进行遍历	123
6.1.3 指针和数组	123
实例 6-4 一种新的数组地址表示法	124
实例 6-5 指针和数组的几种常见用法	125
6.1.4 结构指针	126
实例 6-6 利用结构指针遍历结构数组	126
6.1.5 用做函数参数的指针	127
实例 6-7 指针作为参数使用	128
实例 6-8 指向多维数组的指针参数	128
6.1.6 以地址作为函数的返回值	130
实例 6-9 返回地址的函数	130
实例 6-10 利用指针遍历数组	131
6.1.7 函数指针	132
实例 6-11 函数指针的用法	132
实例 6-12 菜单管理器	133
6.1.8 指针的指针	135
实例 6-13 检索指针的指向	135
6.1.9 指向指针数组的指针	136
实例 6-14 字符串表	136
6.1.10 指向 const 型变量的指针	137
实例 6-15 标准 C 的 std::strcpy() 函数实现	138
6.1.11 const 型指针变量	139
实例 6-16 使用 const 型指针	139

6.1.12 void 型指针	140
实例 6-17 标准 C 在<cstdlib>头文件中声明的内存管理函数	140
6.2 sizeof 运算符	142
实例 6-18 动态地计算数组元素的个数	143
6.3 C 风格的强制类型转换与 typedef	143
实例 6-19 改变记录计数器的数值类型	144
6.4 命令行参数	145
实例 6-20 命令行参数的用法	146
6.5 程序的内存结构	146
6.5.1 堆	147
6.5.2 new 和 delete 运算符	147
实例 6-21 new 和 delete 运算符的用法	147
6.5.3 为固定大小的数组分配内存	148
实例 6-22 用 new 和 delete 为数组分配和释放内存	148
6.5.4 为动态数组分配内存	149
实例 6-23 用 new 运算符为可变大小的数组分配内存	149
6.5.5 堆耗尽时的处理	150
6.5.6 栈	150
6.6 递归	151
6.6.1 一个简单递归的例子	151
实例 6-24 逆序显示姓名	151
6.6.2 一个使用了递归的计算器程序	152
实例 6-25 一个计算数字表达式的计算器程序	152
6.7 引用型变量	156
6.7.1 引用是一个别名	156
实例 6-26 引用型变量的用法	157
实例 6-27 比较实际变量地址和引用地址	157
6.7.2 引用的初始化	158
6.7.3 利用引用来简化复杂的表示法	158
实例 6-28 利用引用来简化复杂的表示法	158
6.7.4 作为函数形参的引用	160
实例 6-29 以引用代替复制	160
6.7.5 以引用方式调用	161
实例 6-30 把引用作为参数传递	161
6.7.6 const 型引用形参	163
6.7.7 以引用作为返回值	163
实例 6-31 引用函数返回值	163
6.7.8 以 const 型引用作为返回值	164
6.7.9 返回指向自变量的引用	164
6.7.10 指针和引用	165
第 1 单元练习	167

第 2 单元 C++ 库函数与函数模板

第 7 章 C++ 库函数和宏	179
7.1 <cassert>	180
实例 7-1 assert 宏的用法	180
7.2 <cctype>	181
7.3 <cerrno>	182
实例 7-2 errno 变量的用法	182
7.4 <cmath>	183
7.5 <csetjmp>	184
实例 7-3 setjmp()和 longjmp()函数的用法	184
7.6 <cstdarg>	186
实例 7-4 处理参数个数可变的函数	186
7.7 <stdio>	187
7.8 <stdlib>	187
7.8.1 数字函数	187
7.8.2 内存管理函数	188
7.8.3 系统函数	188
7.8.4 随机数发生器函数	189
实例 7-5 改进的猜谜游戏	189
7.9 <string>	190
实例 7-6 比较和复制字符串	190
7.10 <ctime>	192
实例 7-7 显示当前的格林威治标准时间	193
第 8 章 C++ 程序的预处理	195
8.1 预处理指令	195
8.2 文件的包含与 #include 指令	196
8.3 宏	197
8.3.1 #define 指令	197
8.3.2 带参数的 #define 指令	198
实例 8-1 计算工资	199
实例 8-2 改进后的计算工资程序	201
8.3.3 格式和对齐	203
8.3.4 # 运算符	203
实例 8-3 字符串化运算符的用法	204
8.3.5 ## 运算符	204
实例 8-4 符号连接运算符的用法	204
实例 8-5 符号连接运算符的高级用法	205
8.3.6 #undef 指令	205
8.4 条件编译指令	206
8.4.1 #if 指令	206

8.4.2 #endif 指令	206
实例 8-6 #if 和 #endif 的用法	206
8.4.3 #if defined 指令	206
实例 8-7 检测宏是否被定义	207
8.4.4 #ifdef 和 #ifndef 指令	207
8.4.5 #else 指令	207
实例 8-8 #else 预处理指令的用法	207
8.4.6 #elif 指令	208
实例 8-9 #elif 预处理指令的用法	208
8.4.7 #error 指令	208
实例 8-10 检测宏表示的值	209
8.5 其他标准指令	209
8.5.1 #line 指令	210
8.5.2 #pragma 指令	210
第 9 章 C++ 结构与类	211
9.1 C++ 的类	211
9.2 数据类型的属性	211
9.2.1 数据描述	212
9.2.2 实现	212
9.2.3 性质	212
9.2.4 接口	212
9.3 用户定义数据类型	212
9.3.1 抽象	212
9.3.2 对象是什么	212
9.3.3 数据抽象和面向过程编程	213
实例 9-1 显示日期	213
9.3.4 数据抽象和封装	214
9.4 带有函数的结构体	215
9.4.1 向结构体添加函数	215
实例 9-2 显示一个人的生日	215
9.4.2 同一结构的多个实例	216
实例 9-3 显示几个人的生日	216
9.4.3 具有相同函数名的不同结构	217
实例 9-4 显示日期和时间	217
9.5 访问控制符	219
9.6 面向过程的编程方法	220
9.7 类与结构的比较	220
9.8 联合体	220
第 10 章 C++ 函数模板	221
10.1 函数模板基础	221
实例 10-1 一个简单的函数模板	222

实例 10-2 显示最小数	223
10.2 函数的定制	223
实例 10-3 比较两个字符串	224
实例 10-4 定制函数模板	225
10.3 使用模板排序	226
实例 10-5 定义函数模板	226
实例 10-6 对整数进行排序	227
第 2 单练习	229

第 3 单元 C++ 类与类模板

第 11 章 C++ 类	233
11.1 类的设计	233
11.1.1 类的声明	233
实例 11-1 计算立方体的体积	234
11.1.2 类的成员	235
11.1.3 类成员的可见性	235
11.1.4 数据成员	236
11.1.5 初始化	236
11.1.6 成员函数	236
11.1.7 面向对象的类的设计方法	237
11.1.8 类对象的作用域	237
11.1.9 内联函数	237
实例 11-2 使用内联函数计算立方体的体积	237
11.2 构造函数	239
11.2.1 使用默认参数的构造函数	239
实例 11-3 用默认值来初始化	240
11.2.2 默认构造函数	240
11.2.3 重载构造函数	241
实例 11-4 具有两个构造函数的类	241
11.3 析构函数	242
11.4 类的转换	243
11.4.1 转换函数	243
11.4.2 转换构造函数	243
实例 11-5 从 std::time_t 对象到 Date 类对象的转换	243
11.4.3 成员转换函数	245
实例 11-6 计算从 20 世纪开始至今的天数	245
11.4.4 类的转换	246
实例 11-7 对两种日期进行转换	246
11.4.5 转换函数的调用	248
实例 11-8 三种类转换形式	248
11.4.6 其他转换方法	250