

Computer Science: A Structured
Approach Using C++, Second Edition

计算机科学

——C++软件工程方法

(第2版)

Behrouz A. Forouzan

Richard F. Gilberg

著

李海峰 张耀辉 等译



国外计算机科学经典教材

计算机科学

C++软件工程方法

(第2版)

Behrouz A. Forouzan

Richard F. Gilberg

著

李海峰 张耀辉 等译



清华大学出版社

北京



0406287

Behrouz A. Forouzan, Richard F. Gilberg.

Computer Science: A Structured Approach Using C++, Second Edition

EISBN: 0-534-37480-8

Copyright © 2004 by Course Technology, a division of Thomson Learning.

Original Language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All Rights reserved.

本书原版由汤姆森学习出版集团出版。版权所有，盗印必究。

Tsinghua University Press is authorized by Thomson Learning to publish and distribute exclusively this Simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本中文简体字翻译版由汤姆森学习出版集团授权清华大学出版社独家出版发行。此版本仅限在中华人民共和国境内（不包括中国香港、澳门特别行政区及中国台湾）销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可，不得以任何方式复制或发行本书的任何部分。

北京市版权局著作权合同登记号 图字：01-2004-7365 号

版权所有，翻印必究。举报电话：010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

本书防伪标签采用特殊防伪技术，用户可通过在图案表面涂抹清水，图案消失，水干后图案复现；或将表面膜揭下，放在白纸上用彩笔涂抹，图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

计算机科学——C++软件工程方法(第2版)/(美)弗罗占(Forouzan, B. A.), (美)吉尔博格(Gilberg, R. F.)著;李海峰,张耀辉等译.—北京:清华大学出版社,2006.1

书名原文: Computer Science: A Structured Approach Using C++

ISBN 7-302-12041-2

I. 计… II. ①弗… ②吉… ③李… ④张… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2005)第124731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

责任编辑:冯志强

印刷者:清华大学印刷厂

装订者:北京市密云县京文制本装订厂

发行者:新华书店总店北京发行所

开本:185×260 印张:47 字数:1172千字

版次:2006年1月第1版 2006年1月第1次印刷

书号:ISBN 7-302-12041-2/TP·7795

印数:1~3000

定价:85.00元

出版说明

会员委审编

近年来,我国高等学校的计算机学科教育进行了较大的改革,急需一批门类齐全、具有国际水平的计算机经典教材,以适应当前的教学需要。引进国外经典教材,可以了解并吸收国际先进的教学思想和教学方法,使我国的计算机学科教育能够与国际接轨,从而培育更多具有国际水准的计算机专业人才,增强我国信息产业的核心竞争力。Pearson、Thomson、McGraw-Hill、Springer、John Wiley等出版集团都是全球最有影响的图书出版机构,它们在高等教育领域也都有着不凡的表现,为全世界的高等学校计算机教学提供了大量的优秀教材。为了满足我国高等学校计算机学科的教学需要,我社计划从这些知名的国外出版集团引进计算机学科经典教材。

为了保证引进版教材的质量,我们在全国范围内组织并成立了“清华大学计算机外版教材编审委员会”(以下简称“编委会”),旨在对引进教材进行审定、对教材翻译质量进行评审。“编委会”成员皆为全国各类重点院校教学与科研第一线的知名教授,其中许多教授为各校相关院、系的院长或系主任。“编委会”一致认为,引进版教材要能够满足国内各高校计算机教学与国际接轨的需要,要有特色风格,有创新性、先进性、示范性和一定的前瞻性,是真正的经典教材。为了保证外版教材的翻译质量,我们聘请了高校计算机相关专业教学与科研第一线的教师及相关领域的专家担纲译者,其中许多译者为海外留学回国人员。为了尽可能地保留与发扬教材原著的精华,在经过翻译和编辑加工之后,由“编委会”成员对文稿进行审定,以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和能力所限,本套外版教材在出版过程中还可能存在一些不足和遗憾,欢迎广大师生批评指正。同时,也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材,共同为我国高等学校的计算机教育事业贡献力量。

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

清华大学出版社

国外计算机科学经典教材

编审委员会

主任委员:

孙家广

清华大学教授

副主任委员:

周立柱

清华大学教授

委员 (按姓氏笔画排序):

王成山

天津大学教授

王 珊

中国人民大学教授

冯少荣

厦门大学教授

冯全源

西南交通大学教授

刘乐善

华中科技大学教授

刘腾红

中南财经政法大学教授

吉根林

南京师范大学教授

孙吉贵

吉林大学教授

阮秋琦

北京交通大学教授

何 晨

上海交通大学教授

吴百锋

复旦大学教授

李 彤

云南大学教授

杨宗源

华东师范大学教授

沈钧毅

西安交通大学教授

邵志清

华东理工大学教授

陈 纯

浙江大学教授

陈 钟

北京大学教授

陈道蓄

南京大学教授

周伯生

北京航空航天大学教授

孟祥旭

山东大学教授

姚淑珍

北京航空航天大学教授

徐佩霞

中国科学技术大学教授

徐晓飞

哈尔滨工业大学教授

秦小麟

南京航空航天大学教授

钱培德

苏州大学教授

曹元大

北京理工大学教授

龚声蓉

苏州大学教授

谢希仁

中国人民解放军理工大学教授

前言

本书有两个主要目标：按照 ACM 课程为 CS1 班级学生讲授基本的编程原则；介绍 C++ 语言的基本构件。本书将这些目标放在已经在业界和学术界发展了 30 多年的良好的软件工程概念环境中。

使用 C++ 语言

虽然 C++ 是一种复杂而专业的语言，但是我们的经验证明初学者很容易理解这种语言。事实上，它的输入/输出和引用传递功能使它比传统的 C 语言更容易掌握。我们认为，如果 C++ 能够让学生理解它的设计，那么这种语言并不复杂。

C++ 与其他大多数语言的区别在于 3 个方面：表达式、指针和类。C 语言和 C++ 语言所使用的表达式概念是独一无二的。本书前半部分让学生牢固掌握表达式和基本的结构化编程构件。牢固地掌握了表达式和基本构件后，C++ 就不那么神秘了。第二个强大的 C++ 工具是指针。指针将在第 9 章讨论，然后本书剩下的内容都要使用它。

虽然现在提到有点早，但是从第 10 章开始介绍的类的开发内容首先讨论了基本的类概念，包括定义和访问类对象、构造函数和析构函数、类函数、不变量和复杂的类函数。第 11 章继续介绍类的开发，讨论初始化列表、重载、静态成员、友元类、类和指针、对象数组，以及从 C 语言继承的结构。第 12 章和第 13 章结束了类概念的讨论，但是在后面介绍异常处理程序、高级 I/O 和链表（该章把链表实现为类对象）的各章中也讨论了类的概念。

本书新增内容

本书的首要原则是保留第一版中包含的基本设计概念。因此，本书的基本结构和风格并没有改变。我们仍然首先介绍编程原则，然后用 C++ 实现这些原则。我们增强了与大量示例相结合的可视化方法。虽然这些程序很像 C++，但是本书仍然强调可读性和一致性风格。

虽然如此，本书仍存在很多不同之外，尤其是后半部分。这些变化主要体现在两个重要方面。

(1) 本书进行了全面修改，以反映标准。

(2) 我们对类的介绍更加全面，特别是模板、继承和异常处理。这些主题分别扩展成了整章的内容。

总体上，本书重新编写了书中所有程序，以反映这种语言的风格和功能，及其实现方式。本书在每章结尾增加了练习题，并增加了一些新的复习题、练习和项目。本书重新编写了一些编程练习，最明显的是把 Morse Code 程序重新设计成了类。

除了第 7 章以外，本书前半部分的变动较少。从第 10 章开始，我们对内容进行了全面地设计和重新编排。

- 第10章“类”，结合了在第11章介绍的材料。我们对第10章的软件工程部分进行了修改，在介绍类时包含了UML的基本用法。此外，从第10章开始，我们使用UML用图例描绘类的结构。由于这些变化，我们移动了本书后半部分的几个软件工程主题。
- 第11章“高级类功能”除了继承外，还结合了在第12章介绍材料。此外，还增加了一节新的内容“静态成员”。第11章还简单介绍了结构、枚举和联合。
- 第12章介绍了继承、多态性、抽象类和组成的概念。
- 第13章讨论了模板及其实现。
- 我们修改了对字符串的介绍，以反映标准中新建立的字符串类型。字符串被放在第14章介绍，因为后面的章节需要用到它。
- 第15章是新增内容，介绍了C++异常处理程序中实现的概念。
- 第16章被重新命名为“高级I/O概念”。它仍然介绍了二进制文件，但是内容已经被扩展了，而且它面向的是流对象。第16章结束时，在软件工程部分介绍了使用经典顺序文件更新模型来更新文件的概念。
- 链表被作为类实现和迁移到数据结构的示例，放在了本书的最后。
- 除了表面的改进以外，许多附录内容都未变，比如修改了编程风格，以使它们更多地面向C++。新增的两个附录介绍了C++语言的两种功能：名字空间和STL库。

本书特色

本书的几个特色使本书别具一格，易于初学者学习。

结构和格式

我们的一个基本原则是尽早养成良好的习惯。不良习惯很难再更改。因此，我们一贯强调结构化编程和软件工程原则。每一个完整的程序都使用一致的格式。在分析程序时，进一步解释了格式和标准。我们并不是说其他的格式不好，但是我们的经验证明，如果学生学会了良好的格式并实现这种格式，他们就会更容易适应其他的良好格式。另一方面，很难改变随意的习惯。

实践的原则

我们尽可能地首先介绍某个主题的原则，然后再介绍语言的实现。例如，在第5章，我们首先介绍了逻辑数据和选择的概念，然后才介绍if...else和switch语句。这种方法在介绍语言的细微差别之前，能先使学生理解选择的概念。

可视化方法

简单浏览一下本书就可以发现我们的方法是可视化的。本书有440多幅图、110个表和325个程序示例。虽然这些内容增加了本书的篇幅，但是可视化方法却使学生易于学习。

示例

虽然编程示例的复杂性各不相同，但是每个示例使用的格式都一致。我们拥有长达 10~20 年使用生产程序的经验，使我们确信易读和易于理解的程序比以简洁、隐含方式编写的程序更易于使用。我们始终如一地坚持这样的格式，即在文档中穿插注释和易于理解的数据名称；我们还规定各代码行上的声明、定义和语句最多为一个。学生不仅可以运行这些程序，还可以通过修改和实验来学习一些相关的主题。

软件工程

在各章结束时，我们讨论了软件工程的原理。我们的目的并不是代替专门的软件工程课程。相反，我们坚信，通过早早地学习基本的软件工程原理。学生可以为正式学习该课程做好准备。更重要的是，通过从头开始编写设计良好的程序，学生就不会再忘了学，学了忘。

虽然每章结束时都介绍一些软件工程原理，但是将它们作为章节说明可以更好地掌握它们。因此，在各章结束时简短回顾可以总结本章所讲述的原理。本书的其余部分不包含软件工程原理的内容。你可以把它们作为课外阅读材料来学习，也可以完全不理睬这些内容。

练习题

练习题包括 3 部分，复杂程序依次增大。

“复习题”以简短的正确/错误问答，以及多项选择题来测试学生对所学内容的理解。

“练习”是包含本章所讲内容的简短问题。所选问题的答案可以从 Brooks/Cole 站点得到。

“项目”是较长、较大的任务，它可能平均需要花费 6~9 个小时来完成。

Instructor's Solution Manual（联机可得）包含所有复习题、练习的完整答案。

致 谢

感谢参与此过程的每一个人，我对同行的审阅感激不尽。快速编写图书往往缺乏远见。因此我们非常感谢本书的评审人员的重要指导。还是那句老话，他们的帮助是不可估价的。我们将特别感谢下列评审人员：Theresa Beauboeuf, South Eastern Louisiana University; Charlotte Busch, Texas A&M University; Robert Cubert, University of Florida; Eric Nagler, Lawrence Technological and ITT; 以及 Ron Schwartz, Florida Atlantic University。另外，Brooks/Cole 的编辑和制作人员为本书的编写提供了极其重要的指导和支持。我们要特别感谢编辑 Kallie Swanson，以及其他的幕后工作人员。

最后，真诚地感谢我们的家人和朋友。虽然我们承担了整个编写工作，但是家人和朋友却承受了与我们的分别。

Behrouz A. Forurzan

Richard F. Gilberg

目 录

第1章 计算机导论	1	2.2.3 第一个程序	20
1.1 计算机系统	1	2.2.4 注释	22
1.2 计算机硬件	1	2.3 标识符	23
1.3 计算机软件	2	2.4 数据类型	24
1.3.1 系统软件	2	2.4.1 空型数据	25
1.3.2 应用软件	3	2.4.2 整型数据	25
1.4 计算环境	3	2.4.3 字符型数据	26
1.4.1 个人计算环境	3	2.4.4 浮点型数据	26
1.4.2 分时环境	4	2.4.5 C++中的逻辑数据	27
1.4.3 客户机/服务器环境	4	2.5 变量	28
1.4.4 分布式计算	4	2.5.1 变量声明和定义	28
1.5 计算机语言	5	2.5.2 变量初始化	29
1.5.1 机器语言	5	2.6 常量	30
1.5.2 符号语言	6	2.6.1 整型常量	30
1.5.3 高级语言	7	2.6.2 浮点常量	31
1.5.4 自然语言	7	2.6.3 字符常量	31
1.6 编写、编辑、编译和连接程序	8	2.6.4 字符串常量	32
1.6.1 编写和编辑程序	8	2.6.5 布尔常量	32
1.6.2 编译程序	8	2.7 编码常量	33
1.6.3 连接程序	8	2.7.1 字面量常量	33
1.7 程序的执行	9	2.7.2 定义常量	33
1.8 系统开发	9	2.7.3 内存常量	33
1.8.1 系统开发生存周期	10	2.7.4 小结	34
1.8.2 程序开发	10	2.8 读写数据	34
1.9 软件工程和编程风格	14	2.8.1 数据源和目标	35
1.10 要点和常见的编程错误	15	2.8.2 标准流	35
1.11 练习题	16	2.8.3 往标准输出中写数据 (cout)	36
1.11.1 复习题	16	2.8.4 写入错误 (cerr 和 clog)	37
1.11.2 练习	17	2.8.5 格式化输出	37
第2章 C++语言简介	18	2.8.6 从标准输入中读数据 (cin)	45
2.1 背景知识	18	2.9 编程举例	46
2.2 C++程序	19	2.9.1 打印 “Nothing!”	46
2.2.1 C++程序的结构	19	2.9.2 打印字符值	47
2.2.2 一个简单的程序	19	2.9.3 计算面积和周长	48
		2.10 软件工程和编程风格	50

2.10.1 程序文档	50	3.7.7 示例: 计算学生分数	83
2.10.2 数据名称	51	3.8 软件工程和编程风格	85
2.10.3 数据隐匿	52	3.8.1 KISS	86
2.11 提示和常见的编程错误	53	3.8.2 括弧	86
2.12 练习题	53	3.8.3 用户通信	86
2.12.1 复习题	53	3.9 提示与常见的编程错误	87
2.12.2 练习	54	3.10 练习题	88
2.12.3 项目	55	3.10.1 复习题	88
第3章 C++程序的结构	57	3.10.2 练习	89
3.1 表达式	57	3.10.3 项目	90
3.1.1 基本表达式	58	第4章 函数	92
3.1.2 二元表达式	59	4.1 设计结构化程序	92
3.1.3 赋值表达式	61	4.2 C++中的函数	93
3.1.4 后缀表达式	63	4.3 用户定义函数	95
3.1.5 一元表达式	64	4.3.1 没有参数的 void 函数	95
3.2 优先级和结合性	66	4.3.2 带有参数的 void 函数	96
3.2.1 优先级	67	4.3.3 返回值的函数	97
3.2.2 结合性	68	4.3.4 函数定义	99
3.3 副作用	69	4.3.5 原型声明	100
3.4 表达式求值	70	4.3.6 函数调用	101
3.4.1 没有副作用的表达式	70	4.3.7 函数示例	102
3.4.2 有副作用的表达式	70	4.3.8 参数传递	107
3.4.3 警告	72	4.4 默认参数变元	112
3.5 混合型表达式	72	4.5 标准库函数	113
3.5.1 隐式类型转换	73	4.5.1 用于数学运算的标准函数	113
3.5.2 显式类型转换(强制转换)	74	4.5.2 一般库函数	115
3.6 语句	76	4.5.3 换算随机数字	117
3.6.1 表达式语句	76	4.6 作用域	119
3.6.2 复合语句	77	4.6.1 一般规则	119
3.6.3 语句和定义常量	77	4.6.2 全局作用域	119
3.7 示例程序	78	4.6.3 局部作用域	120
3.7.1 示例: 计算商和余数	78	4.7 编程示例——计算器程序	121
3.7.2 示例: 显示整数右边的数字	79	4.8 软件工程和编程风格	122
3.7.3 示例: 计算平均数	79	4.8.1 结构图	122
3.7.4 示例: 把弧度转换为角度	80	4.8.2 结构图规则和符号	123
3.7.5 示例: 将华氏温度转换为 摄氏温度	81	4.8.3 函数内聚性	125
3.7.6 示例: 计算销售总额	82	4.8.4 自上至下的开发	126
		4.9 提示和常见的编程错误	128

4.10 练习题	129	6.3 初始化与更新	175
4.10.1 复习题	129	6.3.1 循环初始化	175
4.10.2 练习	130	6.3.2 循环更新	175
4.10.3 项目	131	6.4 事件控制循环与计数器控制循环	175
第 5 章 选择——决策	134	6.4.1 事件控制循环	176
5.1 逻辑数据和操作符	134	6.4.2 计数器控制循环	176
5.1.1 C++中的逻辑数据	134	6.4.3 循环对比	177
5.1.2 逻辑操作符	134	6.5 C++中的循环	177
5.1.3 逻辑表达式求值	136	6.5.1 while 循环	177
5.1.4 关系操作符	138	6.5.2 for 循环	181
5.2 双向选择	139	6.5.3 do...while 循环	184
5.2.1 if...else 语句	140	6.5.4 逗号表达式	186
5.2.2 无效的 else 语句	141	6.6 循环示例	188
5.2.3 嵌套 if 语句	142	6.6.1 for 循环	188
5.2.4 悬挂 else 问题	143	6.6.2 while 循环	193
5.2.5 简化 if 语句	144	6.6.3 do...while 循环	194
5.2.6 条件表达式	145	6.7 与循环相关的其他语句	197
5.2.7 处理主要错误	146	6.7.1 break 语句	197
5.2.8 双向选择示例	147	6.7.2 continue 语句	198
5.3 多向选择	151	6.8 循环应用	200
5.3.1 switch 语句	151	6.8.1 求和	200
5.3.2 else-if 语句	156	6.8.2 求积	201
5.4 其他标准库函数	157	6.8.3 最小数与最大数	201
5.5 菜单程序	159	6.8.4 查询	202
5.6 软件工程和编程风格	162	6.9 递归	203
5.6.1 从属语句	162	6.9.1 迭代定义	203
5.6.2 否定逻辑	163	6.9.2 递归定义	204
5.6.3 选择语句规则	163	6.9.3 阶乘——迭代的解决方法	204
5.6.4 结构图中的选择	164	6.9.4 阶乘——递归的解决方法	205
5.7 提示及常见的编程错误	165	6.9.5 设计递归函数	205
5.8 练习题	166	6.9.6 Fibonacci 数字	206
5.8.1 复习题	166	6.9.7 递归的限制	208
5.8.2 练习	167	6.9.8 河内塔	209
5.8.3 项目	169	6.9.9 河内塔递归解决方法	209
第 6 章 循环	173	6.10 编程示例——计算器程序	212
6.1 循环的概念	173	6.11 软件工程与编程风格	214
6.2 前测型循环与后测型循环	173	6.11.1 结构图中的循环	214
		6.11.2 确定算法的效率	215
		6.11.3 线性循环	215

6.11.4 对数循环	216	7.11.1 复习题	260
6.11.5 嵌套循环	216	7.11.2 练习	261
6.11.6 效率的标准度量	219	7.11.3 项目	262
6.12 提示与常见的编程错误	219	第8章 数组	264
6.13 练习题	220	8.1 概念	264
6.13.1 复习题	220	8.2 在C++中使用数组	266
6.13.2 练习	222	8.2.1 声明和定义	266
6.13.3 项目	224	8.2.2 在数组中存取元素	267
第7章 文本 I/O	228	8.2.3 在数组中存储值	267
7.1 输入实体和输出实体	228	8.2.4 索引操作符的优先级	270
7.1.1 文件	228	8.2.5 索引范围检查	270
7.1.2 标准输入	229	8.3 数组和函数	273
7.1.3 标准输出	229	8.3.1 传递个体元素	273
7.1.4 标准错误	229	8.3.2 传递整个数组	274
7.2 流	229	8.4 数组应用	275
7.2.1 创建流与连接流	230	8.4.1 频率数组	275
7.2.2 标准流	230	8.4.2 直方图	276
7.2.3 文件流	231	8.4.3 随机数序列	279
7.3 标准库输入/输出函数	231	8.5 排序	281
7.4 格式化输入与格式化输出	234	8.5.1 选择排序法	282
7.4.1 输入输出操作符	234	8.5.2 选择排序算法	282
7.4.2 格式化数据	235	8.5.3 上推(冒泡)排序法	283
7.4.3 小结	244	8.5.4 上推(冒泡)排序算法	284
7.5 文件示例	245	8.5.5 插入排序法	285
7.6 字符输入/输出函数	248	8.5.6 插入排序算法	285
7.6.1 get	248	8.5.7 重新编写的插入排序法	287
7.6.2 put	248	8.5.8 排序总结	288
7.7 字符输入/输出示例	249	8.6 搜索	288
7.7.1 创建文本文件	249	8.6.1 顺序搜索	289
7.7.2 复制文本文件	249	8.6.2 对分搜索	291
7.7.3 计算字符和行数	250	8.7 二维数组	293
7.7.4 计算文件中的字数	251	8.7.1 声明与定义二维数组	294
7.8 检测文件错误	253	8.7.2 把二维数组传递到函数	298
7.8.1 流状态	253	8.7.3 数组示例	299
7.8.2 错误恢复	255	8.8 多维数组	301
7.9 软件工程与编程风格	257	8.9 编程示例——计算行和列的平均数	302
7.10 提示与常见的编程错误	260	8.10 软件工程与编程风格	304
7.11 练习题	260	8.10.1 测试排序	305

8.10.2 测试搜索	305	9.15 内存配置函数	347
8.10.3 分析排序算法	305	9.16 指针数组	353
8.10.4 分析搜索算法	307	9.17 编写应用	354
8.11 提示和常见的编程错误	308	9.17.1 再研究选择排序法	354
8.12 练习题	309	9.17.2 动态数组	356
8.12.1 复习题	309	9.18 软件工程和编程格式	360
8.12.2 练习	310	9.18.1 指针和函数调用	361
8.12.3 项目	311	9.18.2 指针和数组	361
第9章 指针	315	9.18.3 数组索引交换性	361
9.1 概念	315	9.18.4 动态内存：理论和实践	361
9.1.1 指针常量	315	9.19 要点和常见的编程错误	362
9.1.2 指针值	316	9.20 练习题	364
9.2 指针变量	318	9.20.1 复习题	364
9.3 通过指针存取变量	318	9.20.2 练习	366
9.4 指针声明和定义	320	9.20.3 项目	371
9.5 指针变量的初始化	321	第10章 类	375
9.5.1 示例：指针游戏	322	10.1 基本的类概念	375
9.5.2 示例：添加两个数字	323	10.1.1 存取说明符	375
9.5.3 示例：指针弹性	324	10.1.2 创建类	377
9.5.4 示例：一个变量的多个指针	324	10.2 类对象	380
9.6 指针和函数	325	10.2.1 实例化	380
9.6.1 作为形参的指针	325	10.2.2 访问对象成员	380
9.6.2 返回指针函数	326	10.2.3 使用类	381
9.7 指针的指针	327	10.2.4 this 指针	384
9.8 兼容性	329	10.3 管理器函数	384
9.8.1 兼容性和 void 指针	330	10.3.1 构造函数	384
9.8.2 强制转换指针	330	10.3.2 复制构造函数	390
9.9 读写指针值	331	10.3.3 析构函数	393
9.10 左值和右值	332	10.4 增变器和存取器	394
9.11 指针应用	334	10.4.1 增变器函数	394
9.11.1 把秒转换为小时	334	10.4.2 存取器函数	394
9.11.2 二次方程	335	10.5 类常量	394
9.12 数组和指针	339	10.5.1 最大公约数	395
9.13 指针运算和数组	341	10.5.2 构造函数	395
9.13.1 指针和一维数组	341	10.5.3 改变存储函数	396
9.13.2 指针和其他操作符	343	10.5.4 修订的分数类	397
9.13.3 指针和二维数组	344	10.6 复杂的类函数	399
9.14 把数组传递给函数	345	10.6.1 一元类函数	399

10.6.2 二元类函数	401	11.9.1 引用联合	441
10.6.3 比较分数	405	11.9.2 初启程序	441
10.7 编写应用程序	405	11.10 枚举类型	442
10.7.1 account 类	405	11.10.1 声明一个枚举类型	442
10.7.2 time 类	407	11.10.2 为枚举类型赋值	443
10.8 软件工程和编程风格	410	11.10.3 枚举类型的转换	443
10.8.1 UML: 类图	410	11.10.4 初始化枚举常量	444
10.8.2 类之间的关系	411	11.10.5 匿名的枚举: 常量	444
10.8.3 UML: 对象	412	11.11 类型定义 (typedef)	445
10.9 提示和常见编程错误	413	11.11.1 typedef 和移植式程序	445
10.10 练习题	413	11.11.2 typedef 和数组声明	446
10.10.1 复习题	413	11.12 编写应用程序	446
10.10.2 练习	414	11.12.1 升降机的设计	446
10.10.3 项目	416	11.12.2 升降机类	447
第 11 章 更多类功能和其他类型	417	11.13 软件工程和编程风格	453
11.1 内联函数	417	11.13.1 耦合	453
11.1.1 非成员内联函数	418	11.13.2 数据隐藏	456
11.1.2 内联成员函数	418	11.13.3 小结	456
11.2 初始化列表	419	11.14 提示和常见的编程错误	456
11.3 重载	420	11.15 练习题	457
11.3.1 操作符重载	420	11.15.1 复习题	457
11.3.2 最终的分分数类头文件	425	11.15.2 练习	458
11.4 静态成员	429	11.15.3 项目	460
11.4.1 静态数据成员	430	第 12 章 继承和聚集	462
11.4.2 静态函数成员	431	12.1 继承	462
11.4.3 静态成员的示例	431	12.2 私有、保护和公有继承类型	465
11.5 友元类	433	12.2.1 私有继承类型	465
11.6 类和指针	433	12.2.2 保护继承类型	466
11.6.1 包含指针的类	433	12.2.3 公有继承类型	466
11.6.2 对象的指针	435	12.2.4 继承示例	466
11.7 对象数组	436	12.2.5 多边形——三角形的实现	466
11.7.1 对象的初始化	436	12.2.6 重写继承访问说明符	470
11.7.2 访问对象	437	12.3 继承期间的管理器函数	470
11.8 结构	438	12.3.1 构造函数	470
11.8.1 结构应用程序	439	12.3.2 复制构造函数	471
11.8.2 初始化	439	12.3.3 析构函数	472
11.8.3 小结	440	12.3.4 员工类示例	472
11.9 联合	440	12.3.5 员工类程序	474

12.4 重写成员函数	475	13.2.5 专用成员函数	517
12.4.1 重写方法示例	475	13.2.6 类模板继承	517
12.4.2 重写与重载	477	13.3 软件工程和编程风格	519
12.5 多态性	477	13.3.1 原子和复合数据	520
12.5.1 静态绑定	478	13.3.2 数据结构	520
12.5.2 动态绑定和虚函数	479	13.3.3 抽象数据类型	521
12.6 纯虚函数: 抽象类	481	13.3.4 抽象数据类型的模型	522
12.6.1 纯虚函数	481	13.3.5 ADT 和模板	522
12.6.2 抽象类	481	13.4 提示和常见的编程错误	522
12.7 分层类中的类型转换	486	13.5 练习题	523
12.7.1 强制转换对象	486	13.5.1 复习题	523
12.7.2 强制转换指向对象的指针	488	13.5.2 练习	523
12.8 多重继承	489	13.5.3 项目	524
12.9 聚集	491	第 14 章 字符串	526
12.9.1 示例: 学生类	491	14.1 字符串的概念	526
12.9.2 聚集与组合	494	14.1.1 固定长度字符串	526
12.10 软件工程和编程风格	495	14.1.2 可变长度字符串	527
12.10.1 程序设计概念	495	14.2 C++字符串	527
12.10.2 信息隐藏	495	14.3 C++字符串输入/输出	529
12.10.3 内聚度	495	14.3.1 字符串输出 (<<)	529
12.10.4 小结	498	14.3.2 字符串输入 (>>)	530
12.11 提示和常见编程错误	498	14.3.3 赋值操作符	532
12.12 练习题	498	14.3.4 C++字符串示例	533
12.12.1 复习题	498	14.4 字符串数组	536
12.12.2 练习	500	14.5 字符串操作函数	536
12.12.3 项目	501	14.5.1 字符串长度 (length 和 size)	537
第 13 章 模板	503	14.5.2 字符串比较	537
13.1 函数模板	503	14.5.3 连接和附加	539
13.1.1 模板格式	505	14.5.4 提取子字符串	541
13.1.2 重载函数模板	507	14.5.5 搜索子字符串	542
13.1.3 混合参数类型	507	14.5.6 搜索字符	543
13.1.4 多重通用参数类型	508	14.5.7 访问和修改字符	546
13.1.5 函数模板与其他解决方案	510	14.5.8 字符串插入	547
13.2 类模板	511	14.5.9 替换字符串	548
13.2.1 类模板的语法	511	14.5.10 删除字符串	549
13.2.2 定义类	512	14.5.11 交换字符串	550
13.2.3 对象的实例化	513	14.5.12 转换为 C 字符串	551
13.2.4 数组类	513	14.6 程序设计示例: 莫尔斯电码	551

14.7 C 字符串	557	15.6.2 人工调试	594
14.7.1 C 字符串的基本概念	557	15.6.3 测试驱动程序	596
14.7.2 C 字符串输入/输出	558	15.6.4 修改无效输出	596
14.7.3 C 字符串函数库	559	15.7 提示和常见的编程错误	597
14.7.4 把 C 字符串转换成 C++ 字符串	562	15.8 练习题	597
14.8 C 字符串和 C++ 字符串的比较	563	15.8.1 复习题	597
14.9 软件工程和编程风格	563	15.8.2 练习	598
14.9.1 工资单案例分析	563	15.8.3 项目	599
14.9.2 程序设计步骤	564	第 16 章 高级 I/O 概念	600
14.9.3 结构图设计	565	16.1 输入/输出类	600
14.10 提示和常见的编程错误	568	16.2 文件状态	601
14.11 练习题	569	16.2.1 在读状态下打开文件	602
14.11.1 复习题	569	16.2.2 在写状态下打开文件	602
14.11.2 练习	570	16.2.3 I/O 系统标志符	603
14.11.3 项目	572	16.2.4 测试文件是否打开	604
第 15 章 异常处理	574	16.2.5 打开和重载构造函数	604
15.1 处理错误	574	16.3 文本文件和二进制文件	605
15.1.1 传统的错误处理	574	16.4 文件的标准库函数	605
15.1.2 使用异常处理机制	575	16.4.1 块输入/输出函数	606
15.1.3 多重错误	578	16.4.2 定位函数	609
15.1.4 通用处理程序	579	16.4.3 块 I/O 示例: 附加文件	612
15.2 异常处理类	580	16.5 转换文件类型	613
15.3 异常规范	583	16.5.1 从文本文件创建二进制文件	614
15.3.1 抛出指定的异常	584	16.5.2 从二进制文件创建文本文件	616
15.3.2 不抛出异常	584	16.6 字符串流	619
15.3.3 抛出任何异常	585	16.6.1 写入字符串	619
15.3.4 unexpected 异常	585	16.6.2 读取字符串	620
15.4 类中的异常	585	16.7 文件程序示例	621
15.4.1 构造函数中的异常	586	16.7.1 随机文件存取	621
15.4.2 析构函数中的异常	586	16.7.2 合并文件	623
15.5 标准异常	587	16.8 软件工程和编程风格	626
15.5.1 逻辑错误	588	16.8.1 更新文件	626
15.5.2 内存分配故障	589	16.8.2 顺序文件更新	627
15.5.3 I/O (文件) 异常	589	16.8.3 更新程序设计	627
15.5.4 将新类添加到标准异常类中	590	16.8.4 更新错误	628
15.6 软件工程和编程风格	593	16.8.5 更新结构图	629
15.6.1 修改程序故障	593	16.8.6 更新逻辑	629
		16.8.7 小结	630

16.9 提示和常见的编程错误	630	A.1.2 连接符	672
16.10 练习题	631	A.2 主要符号	672
16.10.1 复习题	631	A.2.1 顺序语句	672
16.10.2 练习	632	A.2.2 选择语句	674
16.10.3 项目	636	A.2.3 循环语句	675
第 17 章 链表	638	附录 B 预处理器指令	678
17.1 链表结构	638	B.1 文件包含	678
17.1.1 节点	638	B.2 宏定义	679
17.1.2 链表的声明	639	B.2.1 简单的命令	679
17.1.3 链接列表的顺序	640	B.2.2 参数化的宏	680
17.1.4 指向链表的指针	640	B.2.3 嵌套的宏	681
17.2 基本的链表函数	640	B.2.4 宏语句的延续	682
17.2.1 添加节点	641	B.2.5 字符串命令(#)	683
17.2.2 删除节点	644	B.2.6 合并命令(##)	683
17.2.3 搜索链表	646	B.2.7 取消定义命令	683
17.2.4 遍历链表	647	B.3 条件编译	683
17.3 链表设计	649	B.3.1 if...else 命令	684
17.3.1 节点类	649	B.3.2 有条件多向选择	684
17.3.2 列表类	649	B.3.3 排除代码	685
17.3.3 类的迭代器	654	B.3.4 如果已定义(#ifdef/#ifndef)	685
17.4 编程示例——链表平均值	657	命令	685
17.4.1 构建链表	658	B.4 防止重复的库文件包含	686
17.4.2 打印列表	659	附录 C 与输入输出有关的类	688
17.4.3 打印列表的平均值	660	C.1 ios 类	688
17.5 软件工程和编程风格	662	C.1.1 错误状态标志位	688
17.5.1 质量的定义	662	C.1.2 ios 标志位	689
17.5.2 质量要素	662	C.1.3 定位标志位	689
17.5.3 质量周期	664	C.1.4 格式标志位	689
17.5.4 小结	665	C.1.5 格式参数	690
17.6 提示和常见的编程错误	665	C.2 ios 的派生类	690
17.7 练习题	666	C.2.1 输入流	691
17.7.1 复习题	666	C.2.2 输出流	691
17.7.2 练习	667	C.3 istream 和 ostream 的派生类	691
17.7.3 项目	668	C.3.1 输入文件流	691
附录 A 流程图编制	671	C.3.2 输入/输出流	692
A.1 辅助符号	671	C.3.3 输出文件流	692
A.1.1 流程线	672	C.3.4 I/O 字符串流	692