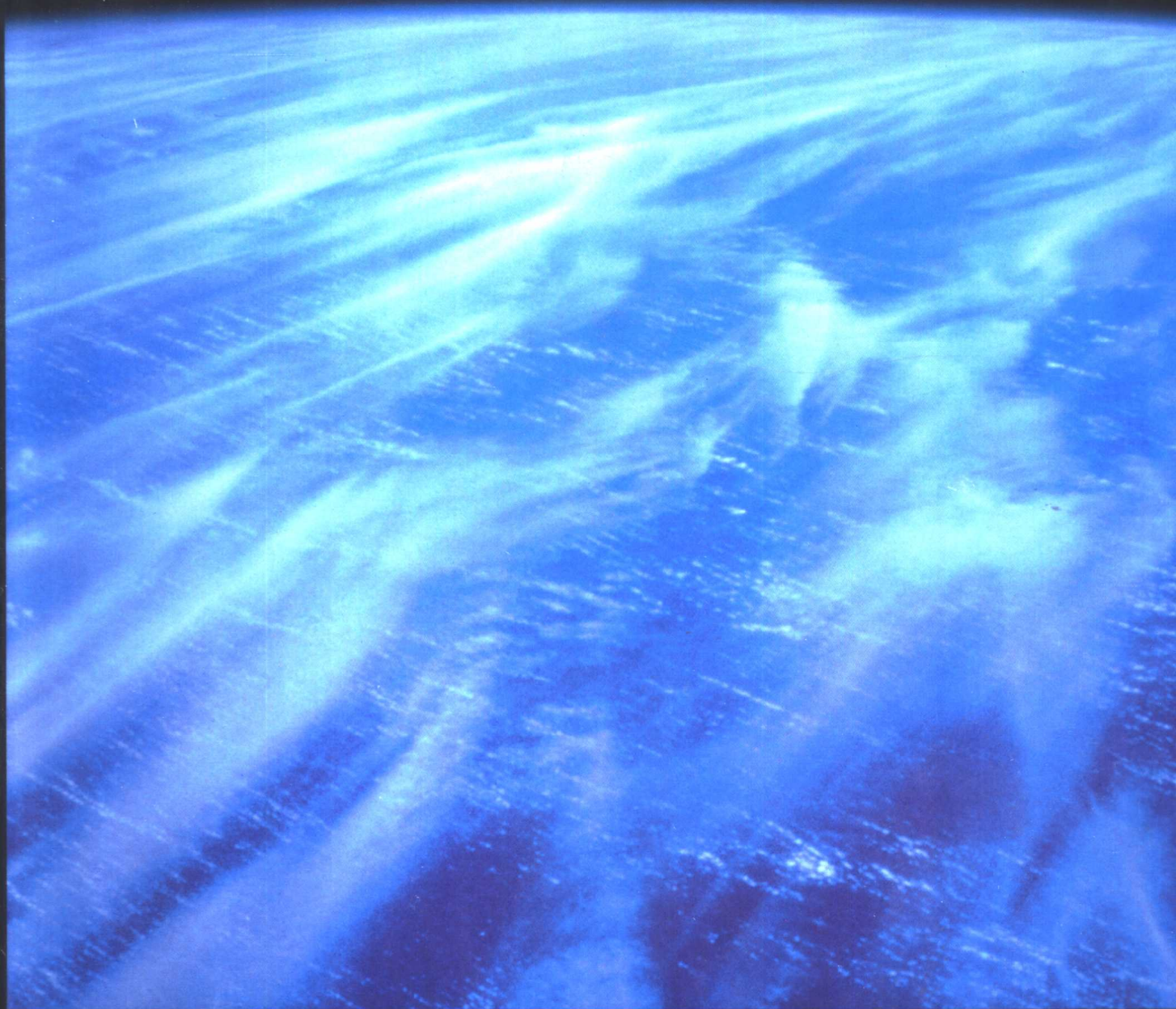


THOMSON



C++ 程序开发与设计

(第二版)



Gary J. Bronson 著
刘勇 译

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

C++ 程序开发与设计

(第二版)

Gary J. Bronson 著

刘勇 译

人民邮电出版社

图书在版编目 (CIP) 数据

C++程序开发与设计/(美)布朗森(Bronson, G.J.)著;刘勇译.—2版.—北京:人民邮电出版社,2002.6

ISBN 7-115-10087-X

I. C... II. ①布...②刘... III. C语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 034004 号

版权声明

Gray J. Bronson: Programming Development and Design Using C++, Second Edition

ISBN: 0-534-37130-2

Copyright © 2001 by Brooks/Cole, a division of Thomson Learning

All Rights Reserved.

Authorized Translation/Adaption of the edition by Thomson Learning and PT Press. No part of this book may be reproduced in any form without the express written permission of Thomson Learning and PT Press.

本书英文原版由汤姆森学习出版集团出版。

人民邮电出版社由汤姆森学习出版集团授权, 出版本书简体中文版。未经原出版者书面许可, 不得以任何形式复制本书任何内容。

版权所有, 翻印必究。

C++ 程序开发与设计 (第二版)

- ◆ 著 Gary J. Bronson
译 刘 勇
责任编辑 陈冀康
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 54.25
字数: 1 762 千字 2002 年 8 月第 1 版
印数: 1-4 000 册 2002 年 8 月北京第 1 次印刷

著作权合同登记 图字: 01-2001-2766 号

ISBN 7-115-10087-X/TP·2771

定价: 78.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内 容 提 要

本书介绍了从 C++ 面向过程编程到面向对象编程的技术方法, 详细分析了 C++ 的各种常用数据结构。

全书共 15 章。第 1 章介绍了计算机和编程的基础知识, 为学习 C++ 编程奠定理论基础。第 2 到第 6 章介绍了面向对象编程的方法和步骤, 目的是使初学者容易接受用 C++ 语言分析和解决问题的思想方法。第 7 到 10 章讲解了面向对象编程的几个范畴的问题, 如类、类函数和转换、继承、动态内存分配、I/O 文件流等等, 全面展示了 C++ 的功能特性。第 11 到 15 章详细分析了 C++ 的常用数据结构, 这对读者进一步提高编程能力有很大帮助。每一章中间和结尾都有大量的练习, 供读者巩固知识或编程实践。附录部分提供了常用参考资料、VC 和 C++Builder 简介、部分习题答案等。

本书适合广大 C++ 语言学习者阅读, 也可作为高校学生学习 C++ 的教学参考书。对于广大 C++ 程序员, 书中丰富详尽的参考资料具有很高的使用价值。

前言

就我的经验而言，最稳妥的方法是自下而上地学习 C++，即首先学习 C++ 为传统的面向过程的程序设计所提供的特性——better-C 子集，然后学习使用和感受其数据抽象特性，随后学习使用类的层次来组织各种相关的类的集合。¹

——Bjarne Stroustrup

对多数人来说，要完全发挥像 C++ 这样的复杂语言的全部潜能（包括面向过程和面向对象的特性），就需要让自己的编程技巧从面向过程（procedural）逐渐改进到面向对象（object-oriented），这正是本书所遵循的途径。本书在编写时力图从教学和实际效果上做到尽快地转向彻底的面向对象的程序设计。

尽管我们现在是要从面向过程转变到完全面向对象，对于 C++ 用户而言，关于“在真正尝试面向对象的程序设计之前要学习多少面向过程的程序设计”这一问题仍然广泛地存在着各种各样的观点。作为一个实际问题，它会引发如下两种争论：面向对象的程序设计是应该在介绍 C++ 语言中间就提及还是应该放在最后呢？

对于这个问题，我个人的回答（同时也是本书坚持的）是来自于我的一个基本观点的：最终教会学生的不是大学的课本，而是大学里的教授。因此，我编写的所有课本都从一个基本点出发，即让课本成为教授们所使用的各种“主角”的“配角”。用通俗的话来说，就是课本要十分灵活，以便那些采用了我的教材的教授们仍然可以按照他们各自的喜好来使用教材中的内容。这一点是用以下方式来实现的。

第 1 章讲述了关于计算机的一些普及性知识，用来帮助那些以往没有接触过计算机的读者。除了这个章节，其余章节讲述的都是 C++ 的有关内容。本书的第 1 部分讲解了基本的面向过程的程序设计语法、流控制（flow control），以及有效实现 C++ 的对象特性所需要的模块化问题。这些关于面向过程的章节中都包含“为对象做准备”这部分内容，这些内容可以用来帮助学员们开始从思想上向面向对象的开发与设计技术转变。

第 1 部分结束后，第 2 部分和第 3 部分则分别介绍了面向对象的程序设计和数据结构，这两个部分的先后次序是可以互换的。这样，如果想尽快学习面向对象的程序设计，你可以按照正常的顺序来学习；另一方面，如果打算继续深入学习更多的面向过程的程序设计知识并将面向对象的程序设计放到最后，你可以在阅读完第 1 部分后提前阅读第 3 部分，最后再学习第 2 部分。无论采取哪种方式，第 10 章中介绍的内容在学习完第 1 部分后可以随时阅读。

各主题的机动性，涉及面向过程的程序设计、抽象数据类型以及继承等的总体上下文关系，可以用下面的主题从属关系图来表示：

在使用本书时，除了要求学生了解基本的代数学以外，没有任何其他要求。第 1 章为有需要的学生介绍一些关于计算机的普及性知识。在这个章节中，我们大量使用了来自日常生活、业务和技术领域的各种例子。因此，教师们可以针对特定的课程重点结合学生的实际来选择适用的素材。

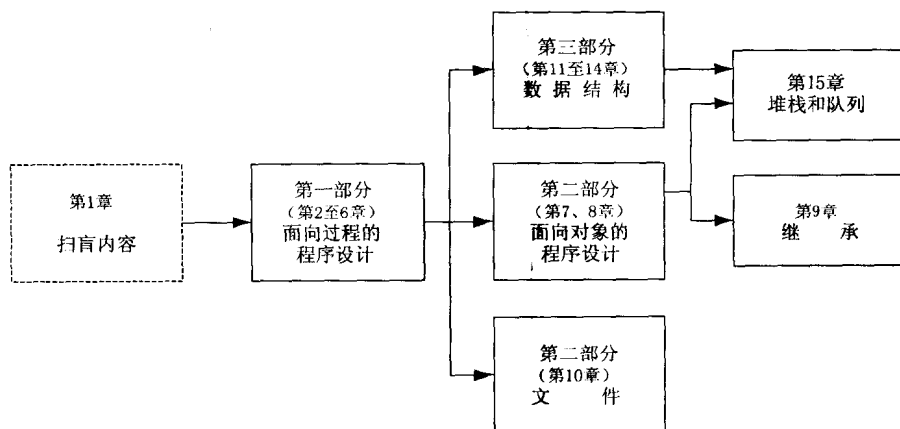
根据本书第一版的读者建议和反馈，第二版中加入了许多新的教学特性和素材。最广泛的修改是将 C++ 的 ANSI 规范结合到本书中了。

此外，第二版所做的修改还包括：

- 包含了关于标准模板库（Standard Template Library）的内容；

¹ *The Design and Evolution of C++*, 1994, 原 AT&T 公司贝尔实验室（现朗讯科技贝尔实验室）版权所有，Addison-Wesley 出版公司出版。

- 包含了关于名称空间 (namespace) 的内容;
- 关于函数模板的部分添加了新内容;
- 添加了关于使用 Visual C++ Version 6.0 和 C++ Builder 的信息;
- 将 Date 类从存储两位年份改为存储 4 位年份。



但是,我们对第二版的基本要求仍然与第一版相同:所有的主题都以一种清楚、明显和可以理解的方式来向那些初学者进行讲解。基于这个目标,第一版的核心组成部分在第二版中基本上没有进行改动。这样,除了根据 ANSI 标准做了修改的部分内容外,第一版中所有的主题、例子、解释和图示在第二版中仍然可以找到。

本书的特点

1. 写作风格

我在自己的教学中总结出的最重要的一件事情就是无论一本教材写的是什么内容,这些内容必须是学生们能够阅读的。一旦教授站在了讲台上,他选用的教材必须能够鼓励、培养和帮助学生们获得和掌握课堂上讲述的内容。我主要关心的(也是本书的特色之一)这本书是为学生们编写的。这样,我首先意识到,作为要讲述的概念的载体,写作风格是本书的一个重要和与众不同的方面。

2. 模块化

要编写出易于阅读和维护的代码,模块化是一个基本的要求。C++生来就是一种模块化语言。因此,在本书的 2.1 节中,我较早地将 C++ 的函数和模块联系在了一起,并且贯穿在全书中。在 3.3 节中,将参数传递到模块中的想法也较早地使用 C++ 的函数库来加以实现。通过这种途径,学生们认识到函数和参数传递是程序设计中的一种固有的技术。这种对模块的强调在第 7 章介绍面向对象的程序设计时得以延续和加强。

3. 软件工程

本书不仅仅是简单地向学生们介绍用 C++ 进行程序设计,还从面向过程和面向对象的角度出发,讲解了软件工程的一些基础知识。这是从 1.3 节介绍软件的开发周期开始的,紧随该小节之后是对一些算法的介绍和第一个“为对象做准备”部分。在第 2 章中,我们阐述了本书的核心主题,它更正式地强调了解决问题的技术。在这里,我们突出了理解一个问题并选择和提炼出解决方案的重要性,介绍了分析、设计、编写代码和代码测试之间的关系。此外,各章节中的“为对象做准备”部分介绍并巩固了面向对象的设计方法所需要的一些概念和技术。

4. 注重解决问题

从第 2 章开始, 每个章节都包含一个“注重解决问题”部分, 平均每个章节中会解决两个完整的问题。每个应用程序都用于演示如何在一个完整的程序解决方案中有效地解决问题, 并分别使用面向过程和面向对象的设计方案加以实现。

5. 对引用 (Reference) 和指针 (Pointer) 的介绍

我以前编写的一本教材在介绍指针时所使用的方式独具特色, 该教材名为 *A First Book of C*。它在讲解指针时先显示变量的地址, 然后用其他的变量来存储这些地址。相对于同期出版的其他书籍中对指针进行间接描述的流行做法而言, 这种方式显得更有逻辑性、更直观。令我感到欣慰的是, 从那时起, 使用一个输出函数来显示地址成为讲解指针的一种标准方式。尽管这种方式不再是我的教材中的独特之处, 我还是为它的表现感到骄傲并在这本教材中加以沿用。引用的主要目的在于将其作为某函数的一个参数来使用, 本书的第 6 章对引用进行了讲解, 而在第 13 章中则对引用和指针做了比较。

6. 程序测试

本书中每个单独的 C++ 程序都成功地在 Borland 公司的 Turbo C++ 编译器下通过编译和运行。

教学方法上的特色

为了帮助 C++ 更易于成为必修课, 本书在教学方法上具有了以下特色:

1. 为对象做准备

由 6 个部分组成 (分别包含在本书的前 6 个章节中), 其内容自成一体, 形成了一个关于面向对象程序设计的概念和设计技术的简短的“迷你课程”。这些部分可以在课程的最初阶段使用, 也可以在课程中讲述面向对象程序设计的阶段中作为浓缩素材使用。

2. 小结练习

本书中的每个章节都几乎包含大量的各种性质的技能培训和编程练习。此外, 本书的附录 K 中还有所选择地对编号为奇数的习题给出了解决方案。

3. 伪代码描述

本书中着重使用了伪代码 (pseudocode)。尽管书中也采用了许多流程图 (flowchart) 标号, 但是它们只用于从视觉上表现流控制的构成。

4. 常见编程错误和章节回顾

在每一章的结尾处都有关于常见编程错误的小节, 还有关于该章节中涉及的主要问题的一個回顾。

5. 注重解决问题

这些小节都含有两个完整的练习, 这两个练习引导学生经历问题确定、分析、解决方案的研究、编写代码、代码测试和代码调试。通过各种应用程序的结合, 采用面向过程和面向对象方式, 这些经过仔细挑选的练习用来提高学生的兴趣, 并巩固程序开发的概念。

6. 浓缩和深入探讨

用多种方式对 C++ 教学中的要点进行强调, 其中包括许多浓缩内容。这些内容让教师为不同的学生和

不同性质的 C++ 课程采取不同的侧重点。

7. 知识要点

这些内容为一些常用和(或)较难理解的概念提供额外的简短说明,例如抽象(abstraction)、左值(lvalue)和右值(rvalue)、值(value)与等式(identity)、标志(flag)、退出队列(dequeue)和流格式化(stream formatting)等。

8. 编程笔记

这些内容突出了其他可用的以及更先进的编程技术。

9. 背景点滴

为了让计算机科学的学习更有收获并增加素材的广度,本书中穿插了这些小的背景资料。这些资料利用历史、传记和其他有趣的相关材料对技术性内容进行了补充。

10. C++快速参考

这个方便分的参考对 C++ 的关键字、运算符、语句语法和类结构进行了汇总,以便为学生提供简单而快速的参考。

11. 函数和头文件参考

这是一个关于 C++ 的常用头文件、库函数和格式标志的快速参考。

附录和补充材料

本书提供了一组在第一版基础上经过扩充的附录,其中包括关于运算符优先、ASCII 码、程序的进入和汇编、I/O 重定向、浮点数存储和附加的 C++ 特性等方面的内容。而在第二版中新加入了关于命名空间、标准模板库以及使用 Visual C++ 和 C++ Builder 开发环境输入和建立 C++ 程序的有关附录。

此外,我们还为使用本书作为教材的教师们提供了教师手册。源代码和可下载的 PowerPoint 演示文档可以在 <http://www.brookscole.com/compsci/bronson/pddcpp> 上找到。

对 PASCAL 程序员的一点说明

用户从 C 和 Pascal 向 C++ 过渡现在已经十分明确了。根据我的经验, C 语言的 3 个基本特点同时也是 C++ 的基础,这在最初会给我们造成一些混淆,因此我在这里要对这些特点做一个粗略的概括。

对于 C 和 C++ 而言,首先一个独特之处就是它们不依靠某种内建的布尔(Boolean)数据类型和能够得出某个确定数字值的测试表达式来进行判断。因此,以下的语法在 C 语言中是看不到的,在 C++ 也不常用。

```
if(boolean expression)
    execute this statement;
```

相反, C 和 C++ 使用另外一种更有包容性的语法:

```
if(expression)
    execute this statement;
```

在这里, expression 是任意一个可以得出数字值的表达式。如果测试表达式得到的是一个非 0 的值,它会将判断的结果视为 true; 只有为 0 的值才会被视为 false。反之,结果为 true 的关系表达式对应的数字

值为 1，而结果为 `false` 的关系表达式对应的数字值为 0。

C 语言和 C++ 的第二个独特之处在于它们对指针的解释和使用。在 C++ 中，一个指针就是一个变量，该变量用来存储一个内存地址。尽管指针变量主要是和间接寻址运算符 (*) 一起用来解除对某个指针的参照并获取想要得到的“指向”目标，C++ 的指针的用法要比 Pascal 的指针丰富得多。这是因为一个指针变量的内容可以像任何其他的变量一样来显示和操作，并且还支持指针的算术运算。

最后，C++ 在语句的位置安排上没有严格的规定。从这个角度来说，C++ 中唯一真正的规定就是一个标识符在使用以前必须先进行声明。尽管合理地安排语句的位置始终不失为一种良好的编程习惯，但 C++ 对于语句放置顺序的要求并没有 Pascal 那样严格。

函数和头文件参考

标准 I/O （需要 `iostream.h` 头文件）

<code>cin</code>	标准输入流
<code>cin.get()</code>	输入单个字符
<code>cin.getline(str, ln, chr)</code>	输入长度为 <code>ln</code> 的字符串，或在检测到 <code>chr</code> 时停止
<code>cout</code>	标准输出流

I/O 操纵符（需要 `iomanip.h` 头文件）

<code>setfill(ch)</code>	将填充字符设定为 <code>ch</code>
<code>setw(n)</code>	设定域宽为 <code>n</code>
<code>setprecision (n)</code>	设定浮点精度为 <code>n</code> 位
<code>setiosflags(flags)</code>	设定格式标志
<code>dec</code>	设定输出为十进制显示
<code>hex</code>	设定输出为十六进制显示
<code>oct</code>	设定输出为八进制显示
<code>endl</code>	插入换行并清空流
<code>flush</code>	清空一个流

`setiosflags()`的格式标志

<code>ios::showpoint</code>	总是显示小数点（缺省值为 6 位小数）
<code>ios::showpos</code>	当数为正数时在前面显示+号
<code>ios::fixed</code>	最多显示 3 位整数位和两位小数位
	对于更大的整数值，转为指数表示法
<code>ios::scientific</code>	输出时使用指数表示法
<code>ios::showbase</code>	输出时显示进制指示符
<code>ios::dec</code>	用十进制格式显示
<code>ios::oct</code>	用八进制格式显示
<code>ios::hex</code>	用十六进制格式显示
<code>ios::left</code>	左对齐输出
<code>ios::right</code>	右对齐输出
<code>ios::stdio</code>	在插入后清空 <code>stdout</code> 和 <code>stderr</code>
<code>ios::skipws</code>	输入时跳过空格

注：这些标志可以用 OR 运算符结合起来。例如，`setiosflags(ios::showpoint || ios::left)` 同时设定了 `showpoint` 和 `left` 标志。

转换例程（需要 `stdlib.h` 头文件）

<code>atof(string)</code>	将 ASCII 字符串转换为浮点数
<code>atoi(string)</code>	将 ASCII 字符串转换为一个整数
<code>itoa(num, string)</code>	将整数转换成 ASCII 字符串

字符例程（需要 ctype.h 头文件）

isalpha(character)	字符是否为文字或数字
isascii(character)	字符是否为一个 ASCII 字符
islower(character)	字符是否为小写
isupper(character)	字符是否为大写
isdigit(character)	字符是否为数字
isspace(character)	字符是否为空格
isprint(character)	字符是否为可见字符
ispunct(character)	字符是否为标点符号
iscntrl(character)	字符是否为控制字符
toupper(character)	将字符转换为大写
tolower(character)	将字符转换为小写

字符串例程（需要 string.h 头文件）

strcat(string1, string2)	连接两个字符串
strcpy(tostring, fromstring)	将 fromstring 复制到 tostring
strlen(string)	确定一个字符串的长度
strchr(string, character)	在一个字符串中查找某个字符
strcmp(string1, string2)	比较两个字符串

文件 I/O（需要 fstream.h 头文件）

file.open(char*, mode)	用给定的模式打开一个 fstream
file.close()	关闭一个 fstream
file.get()	从文件中提取下一个字符
file.getline(str, ln, ch)	从文件中提取一个字符串
file.peek()	不改变文件位置而返回下一个文件字符
file.putback(ch)	将一个字符放回到文件中
file.eof()	如果到达文件末尾，则返回一个 1

允许的文件模式

iso::in	用输入模式打开
iso::out	用输出模式打开
iso::app	用追加模式打开
iso::ate	跳到打开的文件的末尾
iso::binary	用二进制模式打开（缺省情况是文本模式）
iso::trunc	如果文件存在，删除文件的内容
iso::nocreate	如果文件不存在，则打开文件失败
iso::noreplace	如果文件已经存在，则用输出模式打开文件失败

目 录

第 1 章 计算机与程序设计简介	1
1.1 硬件和软件	1
1.1.1 计算机硬件	3
1.1.2 硬件的发展	6
1.1.3 计算机软件	8
练习 1.1	10
1.2 编程语言	11
1.3 问题解决与软件开发	13
1.3.1 阶段一: 开发与设计	14
1.3.2 阶段二: 编写文档	15
1.3.3 阶段三: 维护	16
1.3.4 深入探讨开发与设计	16
1.3.5 备份	20
练习 1.3	21
1.4 算法	21
练习 1.4	24
1.5 为对象做准备: 面向对象的程序设计	25
练习 1.5	26
1.6 深入探讨数字式存储的概念	29
1.6.1 二进制补码	29
1.6.2 字与地址	30
1.7 常见编程错误	30
1.8 章节回顾	31
1.8.1 关键术语	31
1.8.2 概要	31
1.8.3 练习	32
第一部分 C++面向过程的程序设计	35
第 2 章 用 C++解决问题	37
2.1 C++简介	37
2.1.1 main()函数	39
2.1.2 cout 对象	40
练习 2.1	42
2.2 程序设计风格	44
练习 2.2	46
2.3 数据值与算术运算	47
2.3.1 整数值	47
2.3.2 浮点数	48
2.3.3 字符值	49
2.3.4 转义序列	50

2.3.5 算术运算	51
2.3.6 整数的除法	53
2.3.7 一元运算符	53
2.3.8 运算符的优先级和结合性	54
练习 2.3	55
2.4 变量与声明语句	57
2.4.1 声明语句	58
2.4.2 多重声明	60
2.4.3 内存分配	62
2.4.4 显示一个变量的地址	63
练习 2.4	65
2.5 整数限定符	67
2.5.1 数据类型转换	68
2.5.2 确定存储长度	68
练习 2.5	69
2.6 软件开发过程运用	70
2.6.1 步骤一: 分析问题	71
2.6.2 步骤二: 研究解决方案	71
2.6.3 步骤三: 方案的代码化	71
2.6.4 步骤四: 程序的测试与更正	72
练习 2.6	72
2.7 注重解决问题	74
2.7.1 问题一: 钟摆	74
2.7.2 问题二: 直通电话网络	75
练习 2.7	76
2.8 为对象做准备: 关于“抽象”的简介	77
2.9 常见编程错误	80
2.10 章节回顾	81
2.10.1 关键术语	81
2.10.2 概要	81
2.10.3 练习	82
第 3 章 基础知识介绍	86
3.1 赋值操作	86
3.1.1 赋值的各种形式	89
3.1.2 累加和	90
3.1.3 计数	91
练习 3.1	93
3.2 指定输出数的格式	96
练习 3.2	102
3.3 利用数学库函数	104
练习 3.3	107
3.4 用 cin 对象接收程序输入	109
练习 3.4	112
3.5 const 限定符	115

练习 3.5	117
3.6 注重解决问题	118
3.6.1 问题一：酸雨	118
3.6.2 问题二：指数函数的逼近	120
练习 3.6	123
3.7 为对象做准备：程序性能衡量和面向对象技术	125
3.7.1 程序性能的衡量	126
3.7.2 转向面向对象技术	127
练习 3.7	128
3.8 深入探讨错误	128
3.8.1 编译时和运行时错误	128
3.8.2 语法和逻辑错误	128
3.9 常见编程错误	131
3.10 章节回顾	131
3.10.1 关键术语	131
3.10.2 概要	132
3.10.3 练习	133
第 4 章 选择结构	137
4.1 选择标准	137
4.1.1 逻辑运算符	139
4.1.2 数的精度问题	141
练习 4.1	142
4.2 if-else 语句	143
4.2.1 复合语句	145
4.2.2 代码块的作用范围	147
4.2.3 单选择	147
4.2.4 与 if-else 语句相关的问题	149
练习 4.2	150
4.3 嵌套 if 语句	152
练习 4.3	157
4.4 switch 语句	160
练习 4.4	163
4.5 注重解决问题	164
4.5.1 问题一：数据有效性检验	165
4.5.2 问题二：解二次方程	166
练习 4.5	169
4.6 为对象做准备：内部和外部	170
4.6.1 抽象与封装	172
4.6.2 代码重用和扩展	172
练习 4.6	173
4.7 深入探讨程序测试	174
4.8 常见编程错误	175
4.9 章节回顾	176
4.9.1 关键术语	176

4.9.2 概要	176
4.9.3 练习	178
第 5 章 循环结构	181
5.1 引言	181
5.1.1 先判断循环与后判断循环	181
5.1.2 固定计数循环与可变条件循环	181
5.2 while 循环	183
练习 5.2	188
5.3 交互式 while 循环	189
5.3.1 标记	195
5.3.2 break 语句与 continue 语句	196
5.3.3 空语句	197
练习 5.3	197
5.4 for 循环	199
练习 5.4	205
5.5 循环编程技术	208
5.5.1 技术一：在循环中交互输入	208
5.5.2 技术二：在循环中选择	208
5.5.3 技术三：计算单变量函数	210
5.5.4 技术四：交互式循环控制	212
练习 5.5	213
5.6 嵌套循环	214
练习 5.6	216
5.7 do-while 循环	217
练习 5.7	219
5.8 为对象做准备：对象建模技术	220
5.8.1 对象图	221
5.8.2 关系	223
练习 5.8	224
5.9 常见编程错误	225
5.10 章节回顾	225
5.10.1 关键术语	225
5.10.2 概要	226
5.10.3 练习	227
第 6 章 用函数实现模块化	231
6.1 函数声明与参数声明	231
6.1.1 函数原型	232
6.1.2 调用一个函数	233
6.1.3 定义一个函数	234
6.1.4 语句的位置	237
6.1.5 函数占位符	238
6.1.6 带有空参数列表的函数	239
6.1.7 缺省参数	240

6.1.8 函数模板	240
6.1.9 重用函数名(重载)	244
练习 6.1	245
6.2 返回值	246
练习 6.2	251
6.3 变量的作用范围	253
6.3.1 范围解析运算符	256
6.3.2 全局变量的误用	256
练习 6.3	257
6.4 变量存储类型	259
6.4.1 局部变量存储类型	260
6.4.2 全局变量存储类型	263
练习 6.4	264
6.5 使用引用参数的引用传递	265
练习 6.5	272
6.6 递归	273
6.6.1 数学递归	273
6.6.2 计算是如何执行的	275
6.6.3 递归与迭代	276
练习 6.6	277
6.7 注重解决问题	277
6.7.1 产生伪随机数	277
6.7.2 问题一: 模拟抛硬币	279
6.7.3 问题二: 编写一个 HiLo 游戏	281
6.7.4 问题三: 用蒙特卡洛模拟估计曲线下的面积	283
练习 6.7	287
6.8 为对象做准备: 动态 OMT 模型	290
6.8.1 动态模型	290
6.8.2 一个例子	292
练习 6.8	293
6.9 常见编程错误	294
6.10 章节回顾	294
6.10.1 关键术语	294
6.10.2 概要	294
6.10.3 练习	295
第二部分 C++面向对象的程序设计	303
第 7 章 类的介绍	305
7.1 C++中的抽象数据类型(类)	305
7.1.1 抽象数据类型	305
7.1.2 类的构造	307
7.1.3 术语	313
练习 7.1	313
7.2 构造函数	315

7.2.1 构造函数的调用	316
7.2.2 重载和内联构造函数	317
7.2.3 析构函数	320
练习 7.2	321
7.3 注重解决问题	321
7.3.1 问题一: 构造一个电梯对象	322
7.3.2 问题二: 汽油泵对象的仿真	326
练习 7.3	331
7.4 常见编程错误	332
7.5 章节回顾	332
7.5.1 关键术语	332
7.5.2 概要	332
7.5.3 练习	333
第 8 章 类的函数和转换	335
8.1 赋值	335
8.1.1 复制构造函数	338
8.1.2 基础/成员初始化	340
练习 8.1	341
8.2 类的其他特征	341
8.2.1 类的作用范围	341
8.2.2 静态类成员	343
8.2.3 友元函数	346
练习 8.2	348
8.3 运算符函数	349
练习 8.3	355
8.4 用户的两种选择: operator()和 operator[]	356
练习 8.4	359
8.5 数据类型转换	359
8.5.1 内置数据类型转换成内置数据类型	360
8.5.2 内置数据类型转换成类	360
8.5.3 类转换成内置数据类型	362
8.5.4 类转换成类	363
练习 8.5	366
8.6 注重解决问题	367
8.6.1 问题一: 多个对象的汽油泵仿真	367
8.6.2 应用一: 重载插入(<<)和提取(>>)运算符	372
练习 8.6	375
8.7 常见编程错误	376
8.8 章节回顾	377
8.8.1 关键术语	377
8.8.2 摘要	377
8.8.3 练习	378
第 9 章 继承和动态内存分配	381