


# C/C++程序设计

(第三版)

主编 夏宝岚

编著 杨富琴 夏 耘 文欣秀

 华东理工大学出版社

## 内 容 简 介

本书由 13 章正文和 7 个附录组成。第 1 章对高级语言特别是 C 语言的形成与发展作了具体的叙述；第 2、第 3 章介绍了 C 程序基础；第 4、第 5 章介绍了 C 程序的控制结构；第 6 章介绍了组合数据结构“数组”；第 7 章介绍了 C 语言的模块化程序设计“函数”；第 8、第 9 章详细、全面地介绍了 C 语言的精华——“指针”与“动态数据结构”；第 10 章介绍了 C 语言的文件系统；第 11 章介绍了 C 语言的“图形处理”，通过形象生动的绘图实例为学生全面、系统了解 C 语言并提高知识的综合运用能力锦上添花；第 12 章介绍了 C++ 的基础知识；第 13 章通过实例介绍了如何运用 C++ 知识点进行面向对象的程序设计。

本书可用作高等院校各专业的 C 程序设计课程教材、计算机应用能力考试(二级 C)的参考用书,亦可用作计算机爱好者的自学读本。

# 前 言

随着计算机技术的飞速发展以及计算机应用的不断深入,如今程序设计的工具与方法也不断发展与更新,这就对大学生程序设计能力的培养提出了更高的标准,要求当代大学生不仅能够编写开发面向过程的结构化程序,而且能够熟练地利用面向对象的程序设计技术,在可视化的编程环境中开发 C 和 Windows 应用程序,以适应当今信息时代用人单位的需求。基于该背景,我们修改了教材与课程结构体系,将传统的面向过程的程序设计 Turbo C 与现代的面向对象的程序设计 C++以及可视化编程环境 Visual C++进行有机的结合与重建。

C++近年来在国内外得到了广泛的应用,部分高校将它选作本科生的选修课或必修课。但是 C++内容多、概念深、语法复杂是众所周知的,尤其对那些没有程序设计基础的初学者来说学习难度是非常大的。相比之下,C 语言以其小巧、灵活、高效等诸多优点成为当今软件开发的主要编程语言,早已在国内外得到了广泛的应用。我国的大专院校在多年计算机教学改革、实践与探索中,对于将“C 程序设计”课程作为大学本科生的“第一计算机语言”,作为他们获取程序设计能力的首选课程的观点达成共识。事实证明“C 程序设计”这门课程是进一步学习数据结构、C++、Visual C++、Java 等第三代编程语言的先修课程,是不可或缺的基础。

本书重新改编,力求从初学者的认知规律出发,紧扣信息社会的发展脉搏又不偏废本地区的实际情况(学生学完本课程后需要参加上海市高校计算机能力 C 二级的考试),对教材内容作了重新构思与编排:对于那些在实际应用中用得很少甚至不用的内容大胆地作了删减,增加了 C++和面向对象程序的章节,章节的顺序安排遵循由浅入深、循序渐进的原则,先介绍 C 程序设计,以此作起点,在学生理解并掌握了程序设计的思想与方法后再学习教材的第二部分内容,即 C++与面向对象的程序设计。尤其值得一提的是本书的配套教材《C/C++程序实验教程》不仅给出了与教材相一致的 C++实验内容,而且对面向对象程序设计的方法和实现以及可视化编程平台 Visual C++作了详细介绍,包括教学生如何在 Visual C++平台上编辑、调试 C++程序,如何在 Visual C++环境下编辑、调试、运行 C 程序等内容。学习本课程的过程中阅读本实验教程学生会有如鱼得水的感觉。

本书以美国国家标准化协会颁布的 C 语言的最新版本 87 ANSI C 为基础,全书共安排了 13 章内容,前 10 章涵盖了 C 程序设计必须具备的语言、算法、数据结构、程序设计方法等方面的全部知识点;为了提高读者的学习兴趣,充分显示 C 语言的特色和精华,使读者更全面地掌握 C 语言及其综合知识运用能力,第 11 章“图形处理初步”,帮助读者掌握计算机绘图的方法与技巧。该章通过形象生动的绘图实例,将前面所学的知识进行综合应用,所举实例都具有较强的趣味性和实用性(该章可根据自身学校的教学时数酌情安排,一般作为学生的



自学内容);第12章以较大篇幅介绍了C++,较详细地介绍了C++的基本知识点、程序设计方法以及一些实用算法;第13章介绍了面向对象程序设计,通过若干实例,将C++的知识点引入实际的应用之中。除此之外,本书还提供了详尽的习题参考答案和高校计算机等级考试模拟试题及解析,为学生自学、提高编程技巧与能力提供了方便。

本书在内容选取和结构编排上同样遵循“循序渐进、由浅入深”的原则,在文字叙述上力求条理清晰、简明扼要、通俗易懂。每一章都先从实际问题切入,引起读者思考,然后才引出概念,阐述语法和语义,最终使问题得以解决。这样做使读者对每一章学习都有明确的目的,有利于理解、消化和吸收所学的知识。另外,本书在编写风格上还特别注重实例化,即每一个知识点都列举了大量例子(全书的每一个例题均通过调试),而且举例实用、有针对性,只涉及本章和此前章节所介绍过的内容,以减少读者阅读程序和分析问题时的困难,使他们既能掌握知识要点又不感到抽象、枯燥,并在程序设计的能力方面得到训练。2004年本书十分荣幸地被上海市教育考试院指定为上海市高校计算机应用能力考试(二级C)复习参考用书。

本教材由多年从事计算机基础教学的资深教师编写,其中第1、6、12章由杨富琴编写,第2、3、10章由夏耘编写,第4、5、13章由文欣秀编写,第7、8、9、11章由夏宝岚编写。

本书在编写过程中得到了华东理工大学信息学院院长邵志清教授,计算中心主任顾春华、刘江,网络中心主任黄建华、华东师范大学汪燮华教授、上海水产大学张慕蓉教授、上海金融学院黄俊民教授、龚骏博士、范开元、米西峰等老师的悉心指导与帮助,在此一并表示衷心的感谢。

编者

2005年12月

# 目 录

<b>1 C 语言概述</b>	
1.1 C 语言的发展史	1
1.2 C/C++程序简介	2
1.2.1 简单的 C/C++程序实例	2
1.2.2 C/C++程序设计规则	4
1.2.3 C/C++程序的编辑、编译与运行	5
1.3 C 程序的基本语法成分	7
1.3.1 基本字符集	7
1.3.2 标识符	7
1.3.3 关键字	7
1.4 编译预处理	8
1.4.1 无参数宏定义	8
1.4.2 文件包含	9
习题 1	10
<b>2 数据类型及基本运算</b>	
2.1 C 数据类型	11
2.2 基本数据类型	12
2.3 数据的表示	12
2.3.1 常量	12
2.3.2 变量	13
2.4 常用的运算符和表达式	15
2.4.1 算术运算符和算术表达式	15
2.4.2 赋值运算符和赋值表达式	18
2.4.3 逗号运算符和逗号表达式	19
2.4.4 关系运算符和关系表达式	20
2.4.5 逻辑运算符和逻辑表达式	21
2.4.6 位运算符和位表达式	23



2.5	常用的库函数 .....	25
2.5.1	库函数的调用 .....	25
2.5.2	使用库函数注意事项 .....	25
	习题 2 .....	26
<b>3</b>	<b>顺序结构程序设计</b>	
3.1	用于顺序结构的基本语句 .....	29
3.1.1	表达式语句 .....	29
3.1.2	复合语句 .....	30
3.1.3	空语句 .....	30
3.2	输入与输出 .....	30
3.2.1	数据的输出 .....	31
3.2.2	数据的输入 .....	34
3.3	顺序结构程序设计举例 .....	37
	习题 3 .....	38
<b>4</b>	<b>选择结构程序设计</b>	
4.1	if 语句 .....	41
4.1.1	if 语句的三种基本形式 .....	41
4.1.2	if 语句的嵌套形式 .....	43
4.2	条件表达式 .....	45
4.3	switch 语句 .....	46
4.4	应用举例 .....	48
	习题 4 .....	51
<b>5</b>	<b>循环结构程序设计</b>	
5.1	while 语句 .....	53
5.2	do... while 语句 .....	54
5.3	for 语句 .....	55
5.4	循环结构的嵌套 .....	57
5.5	循环结构的中断与转移 .....	58
5.5.1	break 语句 .....	58
5.5.2	continue 语句 .....	59
5.5.3	goto 语句 .....	60
5.6	循环结构应用举例 .....	61



习题 5 .....	62
<b>6 数组</b>	
6.1 数组的概念 .....	64
6.2 数组的定义与引用 .....	64
6.2.1 数组的定义 .....	64
6.2.2 数组的引用 .....	65
6.3 数组的存储结构和初始化 .....	67
6.3.1 数组的存储结构 .....	67
6.3.2 数组的初始化 .....	69
6.4 数组应用举例 .....	70
6.5 数组与字符串 .....	74
6.5.1 用字符数组处理字符串 .....	75
6.5.2 字符串结束标志 .....	75
6.5.3 字符串的输入与输出 .....	76
6.5.4 常用的字符串处理函数 .....	79
6.5.5 字符串应用举例 .....	81
习题 6 .....	84
<b>7 函数</b>	
7.1 函数的概念 .....	86
7.1.1 函数实例 .....	86
7.1.2 实例分析 .....	88
7.2 函数的定义 .....	89
7.2.1 函数定义形式 .....	89
7.2.2 说明事项 .....	91
7.3 函数的调用 .....	92
7.3.1 函数调用形式 .....	92
7.3.2 函数调用方式 .....	93
7.3.3 参数传递 .....	94
7.3.4 函数声明 .....	97
7.4 函数的嵌套与递归 .....	99
7.4.1 嵌套函数 .....	99
7.4.2 递归函数 .....	101



C/C++程序设计(第三版) .....	
7.5 变量的作用域 .....	104
7.5.1 局部变量 .....	104
7.5.2 全局变量 .....	106
7.6 变量的存储类别 .....	107
7.6.1 动态存储 .....	108
7.6.2 静态存储 .....	108
7.6.3 寄存器型存储 .....	111
7.6.4 外部存储 .....	111
7.7 带参数宏与函数的区别 .....	114
7.7.1 有参数宏定义 .....	114
7.7.2 有参数宏与函数的区别 .....	114
习题7 .....	115
<b>8 指针</b>	
8.1 指针的概念 .....	119
8.1.1 地址与指针 .....	119
8.1.2 指针变量及其定义 .....	120
8.1.3 指针变量的引用 .....	120
8.2 指针与数组 .....	122
8.2.1 指针与一维数组 .....	122
8.2.2 指针与二维数组 .....	126
8.3 指针与函数 .....	130
8.3.1 指针变量作为函数的参数 .....	130
8.3.2 函数指针 .....	132
8.3.3 通过函数指针调用其他函数 .....	132
8.3.4 函数指针作为函数的参数 .....	134
8.4 指针与字符串 .....	135
8.4.1 单字符串处理 .....	135
8.4.2 多字符串处理 .....	138
8.5 多级指针 .....	141
8.5.1 二级指针的定义 .....	141
8.5.2 二级指针与指针数组的联系 .....	141
8.6 命令行参数 .....	143



8.7 指针类型小结 .....	148
习题 8 .....	148
<b>9 结构与其他自定义类型</b>	
9.1 结构类型的认识 .....	151
9.2 结构类型的定义 .....	152
9.3 结构变量、结构数组和结构指针 .....	153
9.3.1 结构变量 .....	153
9.3.2 结构数组 .....	155
9.3.3 结构指针 .....	159
9.4 动态数据结构“链表” .....	163
9.4.1 链表概述 .....	163
9.4.2 单链表结点的类型定义 .....	164
9.4.3 动态存储分配函数 .....	165
9.4.4 创建链表 .....	166
9.4.5 结点的删除与插入 .....	170
9.4.6 链表综合应用举例 .....	174
9.5 共用体类型 .....	177
9.5.1 共用体类型的定义 .....	177
9.5.2 共用体变量的引用 .....	178
9.6 枚举类型 .....	180
9.6.1 枚举类型的定义 .....	180
9.6.2 枚举类型的使用规则 .....	180
9.7 类型自定义 .....	182
9.7.1 typedef 语句的形式 .....	183
9.7.2 typedef 语句使用说明 .....	183
习题 9 .....	183
<b>10 文件</b>	
10.1 文件类型和文件指针 .....	186
10.1.1 文件类型概述 .....	186
10.1.2 文件类型和文件指针 .....	187
10.2 文件的打开与关闭 .....	188
10.2.1 文件的打开 .....	188



10.2.2	文件的关闭 .....	189
10.3	文件的读写 .....	189
10.3.1	fputc 和 fgetc 函数 .....	189
10.3.2	fputs 和 fgets 函数 .....	191
10.3.3	fwrite 和 fread 函数 .....	192
10.3.4	fscanf 和 fprintf 函数 .....	194
10.4	文件的定位与出错检测 .....	194
10.4.1	文件的定位 .....	194
10.4.2	文件的出错检测与处理 .....	197
习题 10	.....	199

## 11 图形处理初步

11.1	图形处理基础 .....	201
11.1.1	显示器的图形模式 .....	201
11.1.2	像素坐标 .....	201
11.1.3	图形模式的初始化 .....	202
11.1.4	色彩设置 .....	202
11.1.5	线型设置 .....	203
11.2	基本绘图函数 .....	204
11.2.1	点函数 .....	204
11.2.2	直线函数 .....	205
11.2.3	多边形函数 .....	206
11.2.4	圆弧函数 .....	207
11.3	图形填充 .....	209
11.3.1	设置填充模式 .....	209
11.3.2	填充 .....	210
11.3.3	具有自动填充功能的封闭图形函数 .....	211
11.4	图形方式下的文本输出 .....	211
11.4.1	文本输出函数 .....	211
11.4.2	设置文本的输出格式 .....	212
11.5	图形视口 .....	213
11.6	简单动画 .....	215
11.6.1	简单动画的制作方法 .....	215



11.6.2	动画处理函数	215
11.6.3	动画算法设计步骤	216
	习题 11	218
<b>12</b>	<b>C++基础</b>	
12.1	C++对 C 的扩充	219
12.1.1	C++数据类型、常量、变量、操作符与 C 的差别	219
12.1.2	C++函数使用上与 C 的差别	223
12.2	类和对象	226
12.2.1	类的概念及其定义	226
12.2.2	对象的概念及其定义	228
12.2.3	构造函数	230
12.2.4	析构函数	231
12.2.5	静态成员	232
12.2.6	友元函数和友元类	234
12.3	继承和派生	236
12.3.1	基类和派生类	236
12.3.2	派生类的继承关系	238
12.3.3	派生类的构造函数和析构函数	241
12.4	多态性	244
12.4.1	函数重载	244
12.4.2	运算符重载	245
12.4.3	虚函数和动态联编	248
12.4.4	纯虚函数	250
12.4.5	抽象类	252
12.5	C++的 I/O 流	252
12.5.1	屏幕输出	253
12.5.2	键盘输入	254
12.5.3	格式化输入/输出	256
12.5.4	磁盘文件的输入和输出	261
	习题 12	265



### 13 面向对象程序设计

13.1 面向对象程序设计概述 .....	266
13.1.1 面向过程设计与面向对象程序设计的区别 .....	266
13.1.2 面向对象程序设计的基本概念 .....	267
13.1.3 面向对象程序设计的特点 .....	268
13.2 面向对象程序设计应用举例 .....	269
习题 13 .....	272

### 附录

附录 A C 语言主要关键字及其用途 .....	274
附录 B 运算符优先级和结合性 .....	276
附录 C 常用字符与 ASCII 编码对照表 .....	277
附录 D 常用库函数 .....	278
附录 E 常用关键字和术语的中英文对照表 .....	285
附录 F 习题参考解答 .....	288
附录 G 高校计算机等级考试(二级 C 语言)模拟试题及解析 .....	324

图书在版编目(CIP)数据

C/C++程序设计/夏宝岚主编.—3版.—上海:华东理工大学出版社,2006.1

ISBN 7-5628-1825-8

I. C... II. 夏... III. C语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 153954 号

C/C++程序设计(第三版)

.....

主 编 / 夏宝岚

编 著 / 杨富琴 夏 耘 文欣秀

责任编辑 / 张 波

封面设计 / 王晓迪

责任校对 / 徐 群

出版发行 / 华东理工大学出版社

地 址:上海市梅陇路 130 号,200237

电 话:(021)64250306(营销部)

传 真:(021)64252707

网 址:www.hdlgpress.com.cn

印 刷 / 上海崇明裕安印刷厂

开 本 / 787×1092 1/16

印 张 / 21.75

字 数 / 554 千字

版 次 / 2002 年 1 月第 1 版

2006 年 1 月第 3 版

印 次 / 2006 年 1 月第 1 次

印 数 / 29711—37760 册

书 号 / ISBN 7-5628-1825-8/TP·143

定 价 / 32.00 元

(本书如有印装质量问题,请到出版社储运部调换)



# 1 C 语言概述

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言。它既具有一般高级语言特性,又具有低级语言特性。因此它既可以用来编写系统软件,也可以用来编写应用程序。本章将简要介绍 C 语言的发展史,并通过 C/C++简单程序实例说明 C 程序的基本结构和编译运行等基础知识,使初学者对 C/C++ 程序设计有一个初步的了解。

## 1.1 C 语言的发展史

C 语言是 1972 年由美国学者 Dennis Ritchie 设计发明的,并首次在 Unix 操作系统的 DEC PDP - 11 计算机上使用。它由早期的编程语言 BCPL(Basic Combind Programming Language)发展演变而来。1970 年,AT&T 贝尔实验室的 Ken Thompson 又依据 BCPL 语言设计出了较为先进的、名字为 B 的语言。B 语言精练、接近硬件,但过于简单,数据无类型。据此,在 B 语言基础上增加数据类型的想法便诞生了 C 语言(取 BCPL 的第二个字母)。C 语言良好的可移植性很快使 Unix 从 PDP 计算机移植到其他计算机平台,随着 Unix 的广泛应用,C 语言也得到推广。从此 C 语言和 Unix 像一对孪生兄弟,在发展中相辅相成,Unix 和 C 语言很快风靡全球。

随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准,这也成了现行的 C 语言标准版本。

1980 年,贝尔实验室的 Bjarne Stroustrup 博士及其同事,开始对 C 语言的内核进行必要的修改和扩充,使其能满足面向对象模型的要求,最初被称为“带类的 C”,1983 年取名为 C++,以后又经过不断的完善和发展,成为目前的 C++。

C++与 C 语言完全兼容,它是 C 语言的超集,它保存了 C 语言的所有组成部分,继承了 C 语言高效率的特性,同时具有更完美的体系结构和更强大的功能。现在 C++已被 ANSI、BSI、DIN、ISO 和其他若干国家标准机构定为标准。ISO 标准于 1997 年 11 月 4 日经投票正式通过。

C/C++常用的编译软件有 Turbo C, Quick C, Borland C, Microsoft Visual C++, Borland C++, Watcom C++等等。使用时,C 源程序扩展名为 .c, C++源程序扩展名为 .cpp。

C++是一种大型语言,其功能、概念和语法都比较复杂,要深入掌握它,需要花较多时间。对大多数同学来说,应先掌握好 C 语言,有了 C 语言的基础再学习 C++会相对容易些,本教材修订后增加了第 12 章和第 13 章,介绍 C++的初步知识,为学生进一步学习 C++打下良好基础。



## 1.2 C/C++ 程序简介

### 1.2.1 简单的 C/C++ 程序实例

下面通过两个简单的程序实例(分别用 C 及 C++ 编写)的讲解,使大家对 C 程序和 C++ 程序有一个初步的感性认识,并分析程序的特点,找出 C 及 C++ 两者的基本区别,以培养 C 程序设计的学习兴趣。

〔例 1.1〕 由键盘输入两个整数,计算它们的商和余数,并在屏幕上输出。

程序设计如下:

```
main( )
{ int a,b,c,d;                /* 变量说明 */
  printf("\n input a,b=?\n");  /* 屏幕输出提示语 */
  scanf("%d %d",&a,&b);       /* 由键盘输入 a,b 的值 */
  c=a/b;                      /* 求 a 除以 b 的商,赋予 c */
  d=a%b;                      /* 求 a 除以 b 的余数,赋予 d */
  printf("\n %d / %d = %d \n %d %% %d=%d\n",a,b,c,a,b,d); /* 输出 */
}
```

本程序中仅由一个函数模块组成。第一行 `main( )` 表示这是主函数,一对大括号 `{}` 括起来的是函数体。函数体所描述的内容是:定义整型变量 `a`、`b`、`c` 和 `d`,然后屏幕上印出“input a,b=?”提示语,意思是提醒你输入 `a`、`b` 两个数,接着调用 `scanf` 函数,从键盘输入两个整数,存入变量 `a` 和 `b` 的存储单元中,再计算 `a` 与 `b` 相除的商(存于 `c` 中)和余数(存于 `d` 中),最后调用 `printf` 语句输出两个算术运算式。

程序中的 `/*.....*/` 代表注释,是给编写程序和阅读程序者看的,它不是程序的组成部分,不影响程序编译和运行,仅起注释作用。至于程序中的某些细节,例如“`%d`”、“`\n`”等内容,我们暂且不去讨论它,第 2 章开始就会学习的。

程序运行后屏幕输出:

```
input a,b=? 11 4 ↵(键盘输入两数为 11 和 4)
11 / 4 = 2
11 % 4 = 3
```

下面用 C++ 改写该程序:

```
#include <iostream.h>          //I/O 流库头文件
void main( )
{ int a,b;
  cout<<" \n input a,b=?\n" ;  //输出提示语
  cin>>a>>b;                  //对 a,b 变量输入
  int c;                      //变量可随用随定义
```



```

c=a/b;
int d=a%b;
cout<<" \n a/b="<<c<<endl;           //输出 a 除以 b 的商
cout<<" \n a%b="<<d<<endl;         //输出 a 除以 b 的余数
}

```

将上面的两个程序作比较,显然C++程序与C程序有三点明显的不同之处:其一,程序的注释改变为简化的标记形式“//”(针对单行注释);其二,数据的输入输出改用 cin 和 cout 两个标准流设备,而且不用考虑数据类型,其中 cin 用于键盘输入,cout 用于显示器输出;其三,输入、输出流 cin 和 cout 未被默认,因此程序首部必须写上编译预处理命令 #include <iostream.h>。

〔例 1.2〕 由键盘输入圆柱底面的半径和高,计算圆柱体的体积。

程序设计如下:

```

float volm(r,h)           /* 定义 volm 函数,用于计算圆柱体的体积 */
float r,h;
{ float x;
  x=3.14159*r*r*h;
  return x;
}                          /* volm 函数结束 */
main( )                   /* 定义主函数 */
{ float radius,height,volume;
  scanf("%f%f",&radius,&height);
  volume=volm(radius,height);
  printf(" volume of cylinder is:%f\n",volume);
}

```

本例包含两个函数,一个是 main(主)函数,另一个是名字为 volm 的辅函数。volm 函数有两个参数 r(底面半径)和 h(柱体高),函数的功能是对依据 r 和 h 计算圆柱体的体积,再通过 return 语句将结果返回给主调函数。main 函数首先对实型变量 radius 和 height 输入半径和高,然后调用 volm 函数计算圆柱体的体积,并在屏幕上输出计算结果。

下面用C++改写该例程序:

```

#include <iostream.h>           //I/O 流库头文件
/* 类的声明部分 */
class Cylinder                 //圆柱类
{ public:                      //定义公有成员函数或数据成员
  Cylinder(float r,float h);   //带两个参数的构造函数
  float GetVolume ( ) const;   //计算圆柱体积的成员函数
private:                       //定义私有成员函数或数据成员
  float rd;                    //圆柱底面半径
  float hd;                    //圆柱高
};

```



```
/* 类的实现部分 */
Cylinder::Cylinder(float r, float h) //定义带两个参数的构造函数并完成初始化
{ rd=r; hd=h;
}
float Cylinder::GetVolume ( ) const //定义成员函数,计算圆柱体体积
{ return 3.14159*rd*rd*hd;
}
int main( ) //定义测试程序(即主函数)
{ Cylinder myCylinder(2,3); //定义对象并初始化
  cout<<"The volume is:"<<myCylinder.GetVolume( );//调用函数并输出结果
}
```

本程序运行后,屏幕输出的内容为:The volume is:37.6991。

与上面的C程序相比,本程序也仅有一个主函数,并也是通过在主函数中调用其他函数实现圆柱体积的计算。但两者的区别是:C程序中,对于数据 $r$ 、 $h$ 以及计算体积的操作没有进行封装,是分散的;而C++程序把与圆柱相关的数据( $r$ 和 $h$ )以及操作(计算并返回体积值)集中起来形成一个类,使用户通过创建对象来控制这个圆柱,这样只有对象中的函数可以改变数据,避免给对象内部的信息实施不恰当的操作。第12、13章将对类和对象进行详细介绍。

### 1.2.2 C/C++程序设计规则

(1) 数据是程序加工的对象,因此C/C++程序中使用到的一切数据都必须对其类型和存储属性加以定义和说明。例如,〔例1.1〕程序中出现的`int a,b,c,d;`和`int d=a%b;`等语句都是说明语句,但C规定,变量的说明语句必须出现在可执行语句之前,而在C++中则可以随用随定义,出现在程序的任意位置。

(2) 函数是C/C++程序的基本模块,它应该包括函数名、参数说明表和函数体三个部分。函数体必须以“{”开始,以“}”结束。任何一个C/C++程序都必须含有一个名为`main`的函数(即主函数),程序运行时总是从`main`函数中的第一个语句开始。其他函数称为辅函数,程序中可以有,也可以有一个或多个。但C++的函数被定义成为类的成员函数。

(3) 语句是函数的基本单位,在C/C++程序中,一行上允许写多个语句,或一个语句可分写在多行上,但每个语句结束必须加分号。

(4) 为了增加程序的可读性,C/C++允许在程序中插入注释。注释行必须以规定的符号“/\*”开头,以符号“\*/”结束。注释可插在程序中需要作解释的任意位置上,注释仅起到提示语的作用,它不是程序的组成部分,在程序的编译中不产生代码,因此对程序的执行不产生任何影响。适当加入注释,对于程序的阅读和维护是十分必要的。C++中还常用“//”符号加在注释信息前面,作为单行注释说明。

(5) C/C++程序都是自由格式书写的程序,因此程序的正文排列与程序功能没有必然联系。但是为了提高程序的可读性,通常在书写C语言程序时,应该按结构化程序设计原则,采用“缩进”的方式将程序设计成“层次型”,即从外层到内层逐层缩进。例如: