



职业教育规划教材(高职高专适用)

C++程序设计



贾振华 主 编

赵 辉 李新荣 李永学 副主编



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>





清华大学出版社

C++程序设计

清华大学出版社

图

21 世纪职业教育规划教材(高职高专适用)

C++ 程序设计

贾振华 主 编
赵 辉 李新荣 李永学 副主编

清华大学出版社
北京交通大学出版社
· 北京 ·

内 容 简 介

本书全面、系统地讲述了 C++ 语言的基本概念、基本语法、程序控制流程、数据表示、程序模块设计等内容,还详尽地讲述了 C++ 语言面向对象的重要特征:对象、类、继承、多态性和虚函数等内容。本书结合了结构化程序设计和面向对象程序设计的编程方法。本书包含丰富的典型性、实用性、趣味性的例题。每章后面配备适量的习题。

本书内容丰富、结构合理、重点突出。讲解力求通俗易懂、由浅入深。本书适合作为高等职业教育教材,也适合作为广大 C++ 语言编程爱好者的参考书。

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

C++ 程序设计 / 贾振华主编. —北京:清华大学出版社;北京交通大学出版社,2005.9
(21 世纪职业教育规划教材)

ISBN 7-81082-627-1

I. C… II. 贾… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据字(2005)第 108869 号

责任编辑:周益丹 特邀编辑:逢积仁

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京鑫海金澳胶印有限公司

发行者:新华书店总店北京发行所

开 本:185×260 印张:17.5 字数:423 千字

版 次:2005 年 9 月第 1 版 2005 年 9 月第 1 次印刷

书 号:ISBN 7-81082-627-1/TP·236

印 张:1~4 000 册 定价:23.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。
投诉电话:010-51686043,51686008;传真:010-62225406;E-mail:press@center.bjtu.edu.cn。

前 言

C++ 是当前最流行和最实用的一种计算机程序设计语言,它的功能非常强大。C++ 具有语言简洁、高效、灵活、易移植、可重用性好、扩展性强等优点。C++ 语言既支持面向过程的程序设计,也支持面向对象的程序设计。目前已经成为大多数高校和计算机爱好者学习面向对象的首选语言。读者在学习过程中要深刻理解和充分掌握面向对象程序设计的特点和方法。

作者根据多年的计算机程序设计的教学经验和科研经验,针对学生的学习规律及在学习过程中容易出现的问题,构建了新的教材体系,改进了内容的叙述方式。本书采用“任务驱动”的模式,便于激发学生的学习兴趣。在每章的开始部分给出了学习目标,有助于学生对每章重点的把握。在每章后面,提供了精心设计、难易适度、数量适当的实训题目,供学生练习使用。

本书在语言上通俗易懂,简洁、明确,在内容上由浅入深,循序渐进。本书重点在于培养学生分析问题和解决问题的能力。在全书编写过程中,注重理论知识与实际应用相结合,尽量避免枯燥的理论讲授。通过一定的具有代表性、实用性和趣味性的程序实例来帮助读者掌握理论知识,使读者能够在尽量短的时间内掌握 C++ 语言的理论和程序开发的技术与方法。

本书共分 13 章,第 1 章为 C++ 语言概述,第 2 章为基本数据类型和表达式,第 3 章为程序控制结构,第 4 章为函数,第 5 章为数组,第 6 章为结构体和共用体,第 7 章为指针和引用,第 8 章为类与对象,第 9 章为运算符重载,第 10 章为继承性,第 11 章为虚函数和多态性,第 12 章为 C++ 的输入/输出及 I/O 流类库,第 13 章为模板和异常处理。

本书所给实例均在 Visual C++ 6.0 集成环境中调试通过。

本书适合作为高职高专计算机及其相关专业学习“C++ 程序设计”课程的教材,也可作为广大计算机应用人员和面向对象程序设计爱好者的自学参考书。

本书由贾振华担任主编,赵辉、李新荣、李永学任副主编。各章编写分工如下:第 1、2、3、4 章由贾振华、李永学共同编写,第 5 章由赵丽艳编写,第 6、7、13 章由赵辉编写,第 8、9 章由贾振华编写,第 10、11 章由李新荣编写,第 12 章由王静编写,庄连英、李杰、崔玉宝、李瑛、陈征峰等也参加了部分内容的编写及程序调试工作。全书由贾振华负责统稿。

由于时间仓促、作者水平有限,书中难免有一些疏漏和不足之处,恳请读者批评指正。

编 者

2005 年 9 月

目 录

第1章 C++语言概述	(1)
本章学习目标	(1)
1.1 计算机程序设计	(1)
1.1.1 程序设计和程序设计语言	(1)
1.1.2 计算机的解题过程	(2)
1.2 C++语言的发展和特点	(3)
1.2.1 C++的发展	(3)
1.2.2 C++的特点	(3)
1.3 C++的语法及词法规则	(4)
1.4 C++程序结构	(5)
1.5 C++程序的上机实现	(8)
1.5.1 C++程序的实现	(8)
1.5.2 创建和运行C++程序	(9)
1.5.3 Visual C++ 6.0集成开发环境	(13)
小结	(16)
习题1	(16)
第2章 基本数据类型和表达式	(17)
本章学习目标	(17)
2.1 基本数据类型	(17)
2.2 常量和变量	(18)
2.2.1 常量	(18)
2.2.2 变量	(21)
2.3 运算符	(22)
2.3.1 算术运算符	(23)
2.3.2 关系运算符	(24)
2.3.3 逻辑运算符	(25)
2.3.4 位操作运算符	(26)
2.3.5 赋值运算符	(28)
2.3.6 其他运算符	(29)
2.3.7 运算符的优先级和结合性	(31)
2.4 表达式	(31)
2.5 类型定义	(34)

小结	(34)
习题2	(34)
第3章 程序控制结构	(36)
本章学习目标	(36)
3.1 顺序结构	(36)
3.1.1 数据的输入和输出	(36)
3.1.2 表达式语句	(38)
3.1.3 空语句	(39)
3.1.4 复合语句	(39)
3.2 选择结构	(39)
3.2.1 if 语句	(40)
3.2.2 switch 语句	(43)
3.3 循环结构	(45)
3.3.1 while 语句	(45)
3.3.2 do-while 循环语句	(46)
3.3.3 for 循环语句	(46)
3.3.4 循环嵌套	(48)
3.4 转向语句	(50)
3.4.1 break 语句	(50)
3.4.2 continue 语句	(51)
3.4.3 goto 语句	(52)
小结	(52)
习题3	(53)
第4章 函数	(54)
本章学习目标	(54)
4.1 函数的定义与调用	(54)
4.1.1 函数的定义	(54)
4.1.2 函数的调用	(55)
4.1.3 函数原型	(57)
4.2 函数的参数	(57)
4.2.1 函数的参数传递	(57)
4.2.2 函数参数的默认值	(60)
4.2.3 函数的嵌套调用和递归调用	(61)
4.2.4 main() 函数的参数	(65)
4.3 内联函数	(66)
4.4 函数重载	(67)
4.5 标识符作用域与变量的存储特性	(68)
4.5.1 标识符作用域	(68)

4.5.2 变量的存储特性	(69)
4.6 C++ 的系统函数	(72)
小结	(73)
习题4	(74)
第5章 数组	(77)
本章学习目标	(77)
5.1 一维数组	(77)
5.1.1 一维数组的定义与初始化	(77)
5.1.2 一维数组元素的引用	(78)
5.2 二维数组	(81)
5.2.1 二维数组的定义与初始化	(81)
5.2.2 二维数组的引用	(82)
5.3 数组作为函数参数	(86)
5.3.1 数组元素作为函数参数	(86)
5.3.2 数组名作为函数参数	(87)
5.4 数组与字符串	(88)
5.4.1 字符串的存储	(88)
5.4.2 字符串的访问	(90)
小结	(94)
习题5	(95)
第6章 结构体和共用体	(96)
本章学习目标	(96)
6.1 结构体	(96)
6.1.1 结构体的定义与初始化	(96)
6.1.2 结构体的引用	(101)
6.2 共用体	(103)
6.2.1 共用体的定义与初始化	(104)
6.2.2 共用体的引用	(105)
小结	(107)
习题6	(107)
第7章 指针和引用	(110)
本章学习目标	(110)
7.1 指针	(110)
7.1.1 指针的概念	(110)
7.1.2 指针变量的定义与使用	(111)
7.1.3 指针与数组	(113)
7.1.4 指针与字符串	(117)
7.1.5 指针与函数	(117)

7.1.6	指针与结构体和共用体	(121)
7.1.7	结构体指针与链表	(124)
7.2	引用	(132)
7.2.1	引用的定义和使用	(132)
7.2.2	引用与函数	(133)
小结	(134)
习题7	(135)
第8章	类与对象	(138)
本章学习目标	(138)
8.1	定义类和对象	(138)
8.1.1	类和对象	(138)
8.1.2	访问对象成员	(141)
8.1.3	this 指针	(143)
8.2	构造函数和析构函数	(144)
8.2.1	构造函数	(144)
8.2.2	带参数的构造函数	(146)
8.2.3	复制构造函数	(150)
8.2.4	析构函数	(152)
8.3	静态成员	(154)
8.3.1	静态数据成员	(155)
8.3.2	静态成员函数	(157)
8.4	友元	(161)
8.4.1	友元函数	(161)
8.4.2	友元类	(164)
小结	(166)
习题8	(166)
第9章	运算符重载	(169)
本章学习目标	(169)
9.1	概述	(169)
9.2	运算符重载的一般规则	(170)
9.3	用成员函数或友元函数重载运算符	(171)
9.3.1	用成员函数重载运算符	(171)
9.3.2	用友元函数重载运算符	(174)
9.4	一些特殊操作符的重载	(176)
9.4.1	重载 ++ 与 - 运算符	(176)
9.4.2	重载赋值运算符	(179)
9.4.3	重载运算符 [] 和 ()	(180)
9.4.4	重载插入 (<<) 和提取 (>>) 运算符	(183)

9.5 类型转换	(185)
9.5.1 构造函数用于类型转换	(185)
9.5.2 类型转换函数	(186)
小结	(188)
习题9	(188)
第10章 继承性	(190)
本章学习目标	(190)
10.1 基类和派生类	(190)
10.1.1 派生类的定义	(191)
10.1.2 继承方式	(192)
10.2 单继承	(196)
10.2.1 单继承的定义	(196)
10.2.2 成员访问权限在继承中的控制	(199)
10.2.3 构造函数和析构函数的调用次序	(201)
10.2.4 赋值兼容规则	(203)
10.3 多继承	(204)
10.3.1 多继承的概念	(204)
10.3.2 多继承的构造函数	(205)
10.3.3 二义性问题	(207)
10.4 虚基类	(208)
10.4.1 虚基类的引入和说明	(208)
10.4.2 虚基类的构造函数与析构函数	(210)
小结	(210)
习题10	(211)
第11章 虚函数和多态性	(215)
本章学习目标	(215)
11.1 虚函数	(215)
11.1.1 虚函数的定义	(215)
11.1.2 纯虚函数	(218)
11.2 抽象类	(219)
11.3 虚析构函数	(219)
11.4 多态性	(221)
11.4.1 多态性的含义	(221)
11.4.2 多态的应用	(221)
小结	(224)
习题11	(225)
第12章 C++ 的输入/输出及 I/O 流类库	(228)
本章学习目标	(228)

12.1 C++ 中 I/O 流类库概述	(228)
12.2 输出/输入流基本操作	(229)
12.2.1 基本输出操作	(229)
12.2.2 基本输入操作	(231)
12.3 输出格式控制	(233)
12.3.1 设置流的格式化标志	(233)
12.3.2 格式输出函数	(235)
12.3.3 操作算子	(237)
12.4 文件的输入/输出	(239)
12.4.1 文件概述	(239)
12.4.2 文件的打开和关闭	(239)
12.4.3 文件的读写	(241)
12.4.4 随机访问文件	(245)
小结	(247)
习题 12	(248)
第 13 章 模板和异常处理	(250)
本章学习目标	(250)
13.1 模板	(250)
13.1.1 模板的概念	(250)
13.1.2 函数模板	(251)
13.1.3 类模板	(254)
13.2 异常处理	(260)
13.2.1 异常处理的基本概念	(260)
13.2.2 异常处理的实现	(261)
小结	(266)
习题 13	(267)
参考文献	(269)

第 1 章 C++ 语言概述

本章学习目标

本章主要介绍 C++ 语言的发展和特点及 C++ 程序基本结构等内容。通过本章的学习,读者应该掌握以下内容:

- 程序设计的概念;
- C++ 语言的发展和特点;
- C++ 程序基本结构;
- C++ 程序的开发过程。

1.1 计算机程序设计

1.1.1 程序设计和程序设计语言

程序是控制计算机运行的一系列有序指令的集合。程序设计是使用某种计算机语言,根据要解决的问题,设计出能够完成这一任务的程序。

程序设计语言是人与计算机进行交流的一种形式语言,是人们利用计算机解决问题的一个基本工具,如同在社会生活中,自然语言是人们进行思想交流的工具一样。自然语言是由字、词、句法和语法等构成的一个完整系统;计算机程序设计语言是由字、词和语法等构成的指令系统。

自然语言随着社会的发展在不断地改变,计算机程序设计语言也随着计算机技术的快速发展而不断地更新。最初,程序开发人员使用的程序设计语言是原始的计算机指令,即能够被计算机直接识别的一系列二进制数,称为机器语言。虽然机器语言便于计算机识别,但对于人类来说却是晦涩难懂,更不便于记忆。用机器语言开发程序,难度大、周期长、功能有限。

在机器语言的基础上,人们设计出汇编语言,它可以将机器语言的指令用一些易于人们记忆和阅读的助记符来表示,如 ADD、SUB、MOV 等。计算机要运行汇编程序,首先将汇编语言的源程序转换成计算机能够识别的机器指令,然后识别和执行机器指令,得到所要的结果。虽然汇编语言相对于机器语言更容易被人们接受,但仍需考虑大量的计算机细节。

使用机器语言或汇编语言编写程序,只有极少数计算机专家才能熟练掌握。随着电子技术的发展,为了加速编程过程,人们开发出像 FORTRAN、BASIC、Pascal、COBOL、C、C++ 等多种计算机高级语言,用一条语句可以完成大量任务。高级语言程序中可以使用人们容易理解的执行语句,像英语式的指令容易被人接受和记忆,还可以包含具有一定意义的数据命名、常用的数学符号。高级语言屏蔽了计算机的细节,提高了语言的抽象层次。从程序员的角度来看,高级语言比机器语言和汇编语言都要强得多。其中 C 和 C++ 是使用最为广泛的高级语言。

C 语言是面向过程的,它是结构化和模块化的程序设计语言。在处理较小规模的程序时,使用 C 语言还比较得心应手,但处理较大规模、较复杂的问题时,结构化程序设计方法就显示出它的不足。面向对象程序设计,是针对较大规模、较复杂程序而提出来的,目的是提高软件开发的效率。面向对象和面向过程是统一的,各有用途、互为补充。在面向对象程序设计中仍然要用到结构化程序设计的内容和知识,对于简单的问题可以直接使用面向过程的方法来解决。C++ 语言既可以进行面向过程结构化程序设计,也可以进行面向对象程序设计。本书既介绍 C++ 在面向过程程序设计中的应用,也介绍 C++ 在面向对象程序设计中的应用。

在使用计算机程序设计语言设计程序时,要达到的目标是:在保证程序正确的前提下,力求程序可读性强、容易维护、移植性好。程序的可读性是指程序要有良好的书写风格和用简单易懂的语句编写程序,书写风格包括语句的对齐、规范的注释等。容易维护是指当业务规则发生变化时,要求以最小的开销就可以对程序功能进行更改或增加。移植性是指编好的程序可以在不同的计算机和操作系統上都能运行,并且运行的结果一样。

1.1.2 计算机的解题过程

利用计算机解决实际问题时,一般要经过三个过程:建立数学模型、确定计算方法、编写程序实现。

1. 建立数学模型

当要解决一个实际问题时,首先要建立其数学模型,即将它用数学形式表示出来。如天气预报中天气变化可以用数学中的微分方程近似表示。非数值计算,如文件管理也有相应的描述方法。大部分工作常由本行业专业人员来解决。

2. 确定计算方法

给出数学模型并不意味着计算机就可以计算了,因为有些数学模型非常复杂,如积分方程等,利用计算机程序中给出的基本运算不可以完成,所以需要进一步对数学模型进行处理,将其表示为适合计算机运算的近似公式。例如计算机采用如下公式计算 e^x :

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

3. 编写程序实现

根据近似公式,用计算机程序设计语言编写出具体的操作解题过程,执行程序,输出结果。即在程序设计过程中,程序设计主要完成两方面工作:

(1) 数据描述

数据描述把被处理的实际问题中的信息写成计算机可以接受的数据形式,如整型数据、实型数据、字符型数据等。

(2) 数据处理

根据计算方法对数据进行输入、计算、存储、输出等活动,数据处理目的是为了获得需要的数据。

1.2 C++ 语言的发展和特点

1.2.1 C++ 的发展

C++ 语言是从 C 语言发展演变而来的, C++ 完全支持面向对象程序设计。在介绍 C++ 语言之前, 首先介绍一下 C 语言。

1967 年英国剑桥大学的 Martin Richards 设计了 BCPL (Basic Combined Programming Language) 语言, 用于编写操作系统软件和编译器。BCPL 语言深刻地影响了美国 AT&T 公司贝尔实验室的 Ken Thompson, 在他设计的 B 语言中大量采用 BCPL 特性, 并用 B 语言在 DEC PDP-7 计算机上生成了 UNIX 操作系统的早期版本。

C 语言是在 B 语言的基础之上发展起来的, 由贝尔实验室 Dennis Ritchie 设计, 并在 DEC PDP-11 计算机上实现。C 语言使用了 BCPL 语言和 B 语言的许多概念, 同时增加了数据类型和其他特性。1973 年 Ken Thompson 和 D. M. Ritchie 合作把 UNIX 操作系统 90% 以上的代码从由汇编语言编写改为由 C 语言来编写。C 语言最初作为 UNIX 操作系统的开发语言而闻名于世。

随着 C 语言的发展, 出现了各种 C 语言版本, 它们之间存在差异, 且互不兼容。为了改变这种状况, 1983 年美国国家标准协会 (ANSI) 制定了新的标准, 称为 ANSI C。1987 年对 ANSI C 又作了新的修订, 称做 87 ANSI C。

C++ 语言是 C 语言的扩充, C 语言在 C++ 中作为子集保留下来。1980 年, 美国贝尔实验室的 Bjarne Stroustrup 博士为了仿真课题研究, 编写了称为“带类的 C”语言版本。1983 年 7 月用 C++ 将该语言名字定下来, 并向研究小组之外发表。C++ 的名字强调了从 C 语言而来的演化特性, “++”是 C 的增量运算符, 之所以没有叫 D 语言, 是因为 C++ 是 C 语言的扩充, 它继承了 C 语言的优点, 又极大地扩充了 C 语言的功能, 是在 C 语言的基础上增加了面向对象程序设计的特征。

C++ 已经在众多应用领域中迅速成为程序员首选的程序设计语言, 尤其适用于开发大、中型项目, 从软件开发时间、费用到软件的可重用性、可扩充性、可维护性及可靠性等方面都显示出 C++ 的优越性。

1.2.2 C++ 的特点

C++ 语言的特点主要表现在两个方面:

1. C++ 全面兼容 C 语言, 并有新的改进

C++ 全面兼容 C 语言, 许多 C 语言代码不经修改就可以在 C++ 中使用。它既保持了 C 语言的简洁、高效的特点, 同时又对 C 语言作了改进和补充。用 C++ 编写的程序可读性、可移植性更好, 代码的结构更为合理, 质量也更高。C++ 中增加了一些新的运算符, 如::、new、delete、-> 等, 使 C++ 语言的功能更为强大。C++ 中改进了 C 语言类型系统, 增加了安全性, C 语言中的类型转换很不严格, C++ 规定类型转换多采取强制转换。C++ 中函数的说明必须采用原型, 对默认类型作了一些限制, 增加了编译系统检查类型的能力。在 C++ 中使用“引用”作为函数参数, 给编程人员带来了更大的灵活和方便。C++ 中允许函数和运

算符重载,允许设置缺省参数,提高了程序的灵活性,减少了冗余度。C++ 中引进了内联函数的概念,提高了程序的效率。在 C++ 中对变量说明更加灵活,C 语言只允许在函数体内部进行说明,而且必须先声明后使用,否则编译出错,C++ 则打破了这一限制,可以根据需要随时对变量进行说明。

2. 支持面向对象程序设计

C 语言是面向过程的程序设计语言,C++ 与 C 语言兼容,是在 C 语言的基础上增加了面向对象程序设计的新内容,所以 C++ 语言既支持面向过程的程序设计,也支持面向对象程序设计。可以说 C++ 不仅仅是对 C 语言进行了一些改进,更重要的是进行了一次变革,使得 C++ 成为学习面向对象程序设计的首选。在学习 C++ 程序设计时,不能用面向过程的思想去学习面向对象程序设计,要转变以往程序设计的观念,采用面向对象的思维方法,从全新的面向对象的角度来学习 C++ 程序设计。

1.3 C++ 的语法及词法规则

计算机语言是人与计算进行交流的工具,它也有严格的字符集和严密的语法规则。C++ 语言的基本要素是字符集,由字符集中的若干字符按照一定的语法规则组成 C++ 程序中各种成分。下面介绍字符集、标识符、关键字、分隔符和空白符。

1. 字符集

C++ 语言的字符集包括以下几种。

- (1) 大小写英文字母:A~Z,a~z。
- (2) 数字:0~9。
- (3) 特殊字符:空格 ! # % ^ & * _(下划线) + - =
~ < > / \ ' " ; . , () [] { } :

2. 标识符

标识符是用户用来标识程序中函数名、类型名、常量名、变量名、对象名等实体而定义的单词。C++ 语言中标识符的命名规则如下:

- (1) 标识符是由英文字母(包括大写和小写)、数字和下划线组成,并且以字母和下划线开始,其后跟零个或多个字母、数字或下划线。

注意:标识符不可以数字开始。

例如:Accd、X1、_x1、desk、Int 等都是合法的标识符,而 3A、A#B 等是非法的。

- (2) 标识符区分大小写字母。例如,Aa 和 aa 是两个不同的标识符。
- (3) 标识符的长度是任意的,但有的编译系统仅能识别前 32 个字符。
- (4) 标识符不能和 C++ 语言的关键字同名。

3. 关键字

关键字是 C++ 系统用于特殊用途的单词。它们在程序中有不同的使用场合。下面列举出 C++ 中常用的关键字:

auto	break	bool	case	char	class	const	continue
default	delete	do	double	else	enum	explicit	extern
float	for	friend	goto	if	inline	int	long

mutable	new	operator	private	protected	public	register	return
short	signed	sizeof	static	static_cast	struct	switch	this
true	typedef	union	unsigned	using	virtual	void	while

4. 分隔符

分隔符用于分隔各个词法记号或程序正文,C++的分隔符如下:

```
( ) { } , : ;
```

5. 空白符

空白符用于分隔程序中的单词,指示词法记号的开始和结束,包括空格、制表符、换行符和注释。空白符在处理时按一个空格来计算,其余空白将被忽略。

1.4 C++ 程序结构

通过简单的C++程序实例,介绍C++程序的最基本结构。

【例1.1】计算圆柱体的体积。

```
/*
   Example1_1.cpp
   给出圆柱体的半径和高,计算圆柱体体积
*/
#include <iostream.h>           //文件包含
#define Pi 3.14159             //定义符号常量
double calVolume(double,double); //函数声明
void main( )                   //主函数
{
    double radius,height,volume;
    cout << "请输入圆柱体的半径和高:"; //提示输入信息
    cin >> radius >> height;           //输入两个数据
    volume = calVolume (radius,height); //调用函数,计算圆柱体积
    cout << "圆柱体:半径 =" << radius << " 高 =" << height << endl;
                                         //输出数据
    cout << "体积 =" << volume << endl;
}
double calVolume (double r,double h) //计算圆柱体积
{
    double v; //声明 v 为整型变量
    v = Pi * r * r * h; //通过计算为 v 赋值
    return v;
}
```

程序运行后,屏幕显示:

请输入圆柱体的半径和高:3.2 10(用户输入)


```
圆柱体:半径=3.2 高=10  
体积=321.699
```

下面来分析例 1.1 的程序结构。

(1) “/ * … */”是程序的注释部分,注释内容是为了增加程序的可读性,系统不编译注释内容,自动忽略从“/ * ”到“ */”之间的内容。

C++ 中以“//”开头直到本行结束的部分也是注释部分。与“/ * … */”的区别在于只能注释一行,不能跨行,这种注释也称为行注释,“/ * … */”注释可以跨行,称为块注释。

(2) #include <iostream. h > 是一条编译预处理命令,声明该程序要使用 iostream. h 文件中的内容,iostream. h 文件中包含了输入 cin 和输出 cout 的定义。编译时系统将头文件 iostream. h 中的内容嵌入到程序中该命令的位置。C++ 中编译预处理命令都以#开头。C++ 提供 3 类编译预处理命令:宏定义命令、文件包含命令和条件编译命令,例 1.1 中出现的#include <iostream. h > 是文件包含命令,其中尖括号内是被包含的文件名。

(3) #define Pi 3. 14159 是宏定义命令,用于定义符号常量,编译时程序中的所有 Pi 都用 3. 14159 字符串替代。定义符号常量的一般形式为

```
#define 标识符 常量
```

注意,符号常量一般用大写表示,以区别变量名;一行只能定义一个常量符号,行末一般不加分号。

(4) double calVolume(double, double); 是用户定义函数的声明,用于通知系统将要用到函数的类型、参数类型和个数及函数名称。

(5) 程序中定义了一个主函数 main(), 其中 main 是函数名,void 表示该函数返回值类型。程序执行从主函数开始。一个 C++ 的程序可以包含多个文件,每个文件又可以包含多个函数。函数之间地位是相互平行的、相互独立的。一个 C++ 程序,必须有一个且只能有一个主函数 main()。执行程序时,系统先从主函数开始运行,其他函数只能被主函数调用或通过主函数调用的函数所调用,函数可以嵌套调用,即在一个函数中可以调用另外一个函数。主函数可以带参数,也可以不带参数。函数在调用之前,必须先定义好,定义函数要按照系统规定的格式进行,后面将详细介绍。

由 { } 括起来的内容是主函数 main() 的函数体,其中左大括号“{”表示函数的开始,右大括号“}”表示函数的结束。函数体部分由许多 C++ 语句组成,这些语句描述了函数的功能实现。

(6) 函数体中的“double radius, height, volume;”语句是变量声明,通知系统为变量分配存储空间。

(7) 函数体中的其他语句用于完成相应功能,是函数体的重要组成部分。包括三条输出语句和一条输入语句,其中 cout 是标准输出流对象,指定显示器为标准输出设备;“<<”是 cout 中的插入运算符,表示把它后面的数据在输出设备上输出显示;双引号中的内容要原样输出,变量要用显示具体的值;endl 表示回车换行,分号“;”表示语句结束,C++ 规定语句必须要用分号“;”结尾。cin 是标准输入设备,一般指键盘,“>>”是 cin 中的提取运算符,从标准输入设备读取数据并存储到变量中。在使用 cin 和 cout 时,需要在程序的前面加上#include <iostream. h > 或#include " iostream. h "。