

高等学校计算机专业规划教材

C++程序设计基础

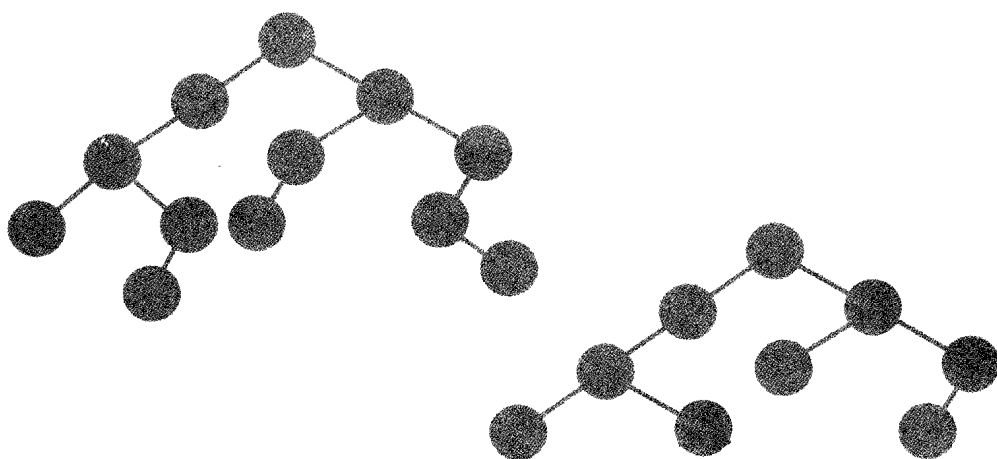
管建和 编著

清华大学出版社

高等学校计算机专业规划教材

C++程序设计基础

管建和 编著



清华大学出版社
北 京

内 容 简 介

本书是作者结合了多少年的C语言和C++语言的教学实践基础上编写而成的教学用教程。本书假定读者没有程序设计语言基础,是完整学习和掌握C++语言的教学用书。本书按照由浅入深、循序渐进的原则,逐步地介绍C++语言的基本概念、语法规则和编程方法,确保在学习完本书内容之后,掌握C++语言的基础知识和编程技能。有C语言基础的读者可以通过该书的进一步地学习,掌握C++语言的编程技术。

本书可作为高等院校计算机专业或非计算机专业的第一门程序设计语言教学用书,也可作为程序设计人员的自学参考用书。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C++ 程序设计基础/管建和编著. --北京:清华大学出版社,2013.3

高等学校计算机专业规划教材

ISBN 978-7-302-31214-7

I. ①C… II. ①管… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第002181号

责任编辑:龙启铭

封面设计:何凤霞

责任校对:李建庄

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京市人民文学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:21.5

字 数:540千字

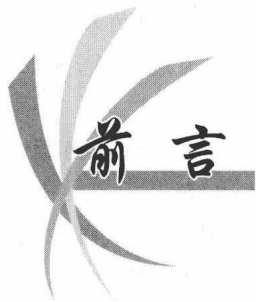
版 次:2013年3月第1版

印 次:2013年3月第1次印刷

印 数:1~3000

定 价:36.00元

产品编号:048254-01



C++ 语言是一种使用广泛和深受程序员喜爱的程序设计语言之一,只有全面深入地了解 and 掌握 C++ 语言的基础知识、基本编程技巧以及面向对象的编程特征,才能高效率地开发出一个面向对象的软件系统。C++ 语言是在 C 语言基础上发展起来的编程语言,它是带类的 C 语言。C 语言既具有高级语言的表达能力丰富、可移植性好等特点,又具有低级语言的能够很方便地实现汇编语言级的操作、目标程序效率高的一些特点。因此,发明 C++ 程序设计语言的主要目标就是既要保留 C 语言的原有精华,又要提供全面的面向对象的编程支持,使得编写的程序结构更加明晰、更容易维护和扩充,同时又不丧失其高效性。C++ 语言对 C 语言作了大量的扩充,如增加了常值(const)数据、显式类型转换、语句中变量说明、inline 函数、引用、函数参数默认、函数及操作符重载等新的表现手段以及更严格的类型机制,同时还完全支持面向对象的概念,如对象、类、属性、方法、封装、派生与继承、多态等。C++ 和 C 一样,现在已被广泛地应用。

本书全面而又系统地讲述了 C++ 语言的基本概念、数据类型、指针、语句、函数、自定义数据类型,并对 C++ 面向对象语言的抽象性、封装性、继承性与多态性进行了全面介绍。本书共分为 13 章,下面分别简单介绍各章的内容。

第 1 章主要介绍 C++ 语言的发展简况及其特点,C++ 语言的程序组成及其结构,C++ 语言程序的格式特点,关键字和标识符,三个特殊的 C++ 语句即空语句、复合语句和注释语句,简单输入输出和数制与编码等方面的知识。

第 2 章主要介绍 C++ 语言的基本数据类型,整型常量、实型常量、字符型常量、字符串型常量和换码序列常量等六种常量类型,变量的说明和使用,数组的数据结构,各种运算符及其表达式,运算符的优先级等内容。

第 3 章主要介绍 C++ 语言的基本程序结构,各种分支语句,for 语句、while 语句和 do-while 语句的循环控制过程,以及 break 语句、continue 语句、return 语句和 goto 语句辅助控制编程等内容。

第 4 章主要介绍 C++ 语言的指针的概念,指针的说明与初始化,指针的几种运算,指针与数组的关系,字符指针与字符串的应用,指针数组与多级指针,以及 new 与 delete 运算符,引用,const 型指针和 void 型指针等内容。



第5章主要介绍C++语言中函数的概念,函数的定义及其说明,函数之间的参数传递,数组在函数间的传递,指针函数和函数指针,函数指针数组,函数重载,内联函数,递归函数,标准函数,宏和其他预编译语句等内容。

第6章主要介绍结构体的定义及其说明,结构体数组与结构体指针,结构体与函数,结构体嵌套,位字段结构体简介,联合体,枚举类型以及typedef语句等内容。

第7章主要介绍面向对象技术与C++类类型,类的定义,已有类的对象的说明和使用,对象数组和对象指针,构造函数和析构函数,对象在函数之间的传递,this指针,以及常对象与常成员函数等内容。

第8章主要介绍类的程序结构,基类与派生类,单继承、多继承与继承链,派生类与基类之间的内在关系,二义性与虚基类,类层次中的访问规则等内容。

第9章主要介绍友元的特性、编程方法和友元使用的局限性,静态成员的特性以及如何使用静态成员函数等内容。

第10章主要介绍静态绑定和动态绑定,实现多态性的虚拟函数,虚拟函数的技术内幕,纯虚拟函数及抽象类等内容。

第11章主要介绍重载和模板实现多态性的基本概念,运算符重载的实现,函数模板的定义及其用法,类模板的定义及其用法,模板的有关特性等方面内容。

第12章主要介绍异常的概念,异常处理编程方法,异常处理类等有关内容。

第13章主要介绍C++语言运行库中提供的流输入输出,磁盘文件的输入输出等有关内容。

本书是根据作者多年来的教学实践讲稿,并查阅了国内外相关书籍和资料写成的,在此向书后列举的参考文献的作者表示衷心的感谢。另外,在本书编写过程中,吴志宇、付国楠、张帆、程汤培和李求实等对本书的部分章节进行了整理等工作,在此一便致以感谢。

由于作者水平有限,书中难免会有缺点和错误之处,敬请广大读者批评指正,并给以宝贵的修改意见。

编 者

2013年1月



第 1 章 概述 /1

§ 1.1 C++ 语言的发展简况及其特点	1
1.1.1 C++ 语言的发展简况	1
1.1.2 C++ 语言的特点	2
§ 1.2 C++ 语言程序组成及其结构	3
§ 1.3 关键字和标识符	5
1.3.1 关键字	6
1.3.2 标识符	6
§ 1.4 几个特殊的 C++ 语句	6
1.4.1 空语句	6
1.4.2 复合语句	7
1.4.3 注释语句	7
§ 1.5 简单的输入输出	8
1.5.1 格式化输入输出函数	8
1.5.2 I/O 流	13
§ 1.6 数制与编码	15
1.6.1 二进制、八进制和十六进制	16
1.6.2 二进制、八进制、十六进制和十进制之间的转换	16
1.6.3 原码、反码和补码	20
1.6.4 编码	21
思考与练习题	22

第 2 章 数据与运算 /25

§ 2.1 基本数据类型	25
2.1.1 类型 bool	25
2.1.2 类型 char	26
2.1.3 类型 short	26
2.1.4 类型 int	27
2.1.5 类型 long	27
2.1.6 浮点类型	27



§ 2.2 常量	28
2.2.1 整型常量	28
2.2.2 浮点常量	28
2.2.3 字符常量	30
2.2.4 字符串常量	30
2.2.5 标识常量	32
2.2.6 换码序列	32
§ 2.3 变量	34
2.3.1 变量和变量的地址	34
2.3.2 C++ 存储类型	35
2.3.3 变量使用范围	37
2.3.4 变量的初始化和赋值	38
2.3.5 强制数据类型的转换	38
2.3.6 常量化的变量	40
§ 2.4 数组	41
2.4.1 一维数组	41
2.4.2 字符数组	44
2.4.3 多维数组	46
§ 2.5 运算符及其运算	48
2.5.1 算术运算符	48
2.5.2 关系运算符和逻辑运算符	50
2.5.3 位运算符	52
2.5.4 赋值运算符与赋值复合运算符	58
2.5.5 C++ 语言的特殊运算符	59
2.5.6 C++ 运算符的优先级	61
思考与练习题	62

第3章 控制语句 /64

§ 3.1 程序设计方法与程序结构	64
3.1.1 结构化程序设计	65
3.1.2 面向对象的程序设计	67
§ 3.2 分支语句	67
3.2.1 单分支语句——if 语句	68
3.2.2 双分支语句——if-else 语句	69
3.2.3 多分支语句——else if 语句	70
3.2.4 嵌套结构及其分析	73
3.2.5 switch 语句	76
§ 3.3 循环语句	79



3.3.1	for 循环语句	79
3.3.2	while 循环语句	82
3.3.3	do-while 循环语句	84
3.3.4	循环语句使用的注意事项	86
3.3.5	循环语句的嵌套	87
§ 3.4	辅助控制语句	88
3.4.1	break 语句——中途退出语句	88
3.4.2	continue 语句——截流语句	92
3.4.3	return 语句	93
3.4.4	goto 语句	95
	思考与练习题	96

第 4 章 指针 /101

§ 4.1	指针变量	101
§ 4.2	指针变量的声明与初始化	104
4.2.1	指针变量的声明	104
4.2.2	指针的初始化	105
§ 4.3	指针运算	107
4.3.1	指针的算术运算	107
4.3.2	指针的关系运算	111
4.3.3	指针的赋值运算	112
§ 4.4	指针与数组	113
§ 4.5	字符指针与字符串	115
§ 4.6	指针数组	117
§ 4.7	多级指针	119
§ 4.8	new 和 delete 运算符	121
4.8.1	new 运算符	121
4.8.2	delete 运算符	123
§ 4.9	引用	124
4.9.1	普通变量的引用	124
4.9.2	数组的引用	124
§ 4.10	const 型指针和 void 型指针	125
4.10.1	常量化的指针变量	126
4.10.2	void 型指针变量	128
	思考与练习题	128

第 5 章 函数与宏 /130

§ 5.1	自定义函数的定义、声明和使用	130
-------	----------------	-----



5.1.1	函数的定义	130
5.1.2	函数的声明	132
5.1.3	函数的使用	133
§ 5.2	函数之间的参数传递	135
5.2.1	复制传递方式	135
5.2.2	地址传递方式	136
5.2.3	引用方式	137
5.2.4	全局变量	138
5.2.5	默认参数值的函数	139
§ 5.3	函数与数组	140
5.3.1	复制传递方式	140
5.3.2	地址传递方式	141
5.3.3	引用方式	142
5.3.4	多维数组与函数	143
§ 5.4	指针函数	146
§ 5.5	函数指针与函数指针数组	147
5.5.1	函数指针	147
5.5.2	函数指针数组	149
§ 5.6	函数重载	150
§ 5.7	内联函数	152
§ 5.8	递归函数	152
§ 5.9	标准函数	153
§ 5.10	宏和其他预编译语句	156
5.10.1	宏定义语句 #define	156
5.10.2	包括语句	158
5.10.3	条件编译	158
思考与练习题		159

第 6 章 自定义数据类型 /167

§ 6.1	结构体的定义及其声明	167
6.1.1	结构体的定义	167
6.1.2	结构体的声明	168
6.1.3	结构体的使用	169
6.1.4	结构体的初始化	170
6.1.5	匿名结构体	171
§ 6.2	结构体数组与结构体指针	171
6.2.1	结构体数组	171
6.2.2	结构体指针	172



§ 6.3	结构体与函数	174
6.3.1	结构体在函数体间的传递	174
6.3.2	结构体函数	177
6.3.3	结构体指针函数	179
§ 6.4	结构体嵌套	180
§ 6.5	位字段结构体简介	182
§ 6.6	联合体	184
§ 6.7	枚举类型	188
6.7.1	枚举类型的定义	188
6.7.2	枚举变量的声明与使用	188
§ 6.8	typedef 语句	190
	思考与练习题	192

第 7 章 类与对象 /195

§ 7.1	面向对象技术与 C++ 类类型	195
7.1.1	传统开发方法存在的问题	195
7.1.2	面向对象技术	196
7.1.3	C++ 语言的三种类类型	197
7.1.4	class 类类型	198
§ 7.2	类的定义	199
§ 7.3	对象的声明和使用	202
§ 7.4	对象数组和对象指针	204
7.4.1	对象数组	204
7.4.2	对象指针	205
§ 7.5	构造函数和析构函数	207
7.5.1	构造函数	207
7.5.2	析构函数	208
7.5.3	默认构造函数和默认析构函数	209
7.5.4	复制构造函数	209
7.5.5	浅复制和深复制	211
§ 7.6	对象在函数间的传递	214
7.6.1	对象的复制传递方式	214
7.6.2	对象的地址传递方式	215
7.6.3	对象的引用方式	217
§ 7.7	this 指针	217
§ 7.8	常对象与常成员函数	218
7.8.1	常对象	218
7.8.2	常成员函数	219



思考与练习题	220
--------	-----

第 8 章 派生与继承 /222

§ 8.1 类的程序结构	222
§ 8.2 基类与派生类	223
8.2.1 派生类的定义	223
8.2.2 派生类的继承方式	226
8.2.3 基类与派生类之间的关系	228
8.2.4 基类指针与派生类指针	228
§ 8.3 单继承、多继承与继承链	228
§ 8.4 派生类与基类之间的内在关系	233
8.4.1 派生类的构造函数和析构函数	234
8.4.2 构造函数和析构函数的执行顺序	235
8.4.3 子类型	238
8.4.4 类型适应	239
8.4.5 对象的内存占用情况	239
§ 8.5 二义性与虚基类	241
8.5.1 二义性问题	241
8.5.2 虚基类	244
8.5.3 虚基类的构造函数	246
§ 8.6 类层次中的访问规则	246
思考与练习题	247

第 9 章 友元与静态成员 /249

§ 9.1 友元	249
9.1.1 友元函数	249
9.1.2 友元类	251
§ 9.2 静态成员	252
9.2.1 静态数据成员	252
9.2.2 静态成员函数	255
思考与练习题	257

第 10 章 虚拟函数与多态性 /259

§ 10.1 静态绑定和动态绑定	259
10.1.1 静态绑定	259
10.1.2 动态绑定	261
§ 10.2 虚拟函数	261
10.2.1 虚拟函数的技术内幕	264



10.2.2 虚拟函数的应用举例	265
10.2.3 虚析构函数	269
§ 10.3 纯虚拟函数与抽象类	270
10.3.1 纯虚拟函数	270
10.3.2 抽象类	271
思考与练习题	276

第 11 章 重载与模板 /279

§ 11.1 重载	279
11.1.1 运算符重载	279
11.1.2 类型转换	284
11.1.3 特殊运算符的重载	285
§ 11.2 模板	290
11.2.1 函数模板	291
11.2.2 类模板	293
11.2.3 模板与友元	298
11.2.4 模板与 static 成员	300
11.2.5 类模板的派生与继承	300
思考与练习题	300

第 12 章 异常处理 /303

§ 12.1 异常的概念	303
12.1.1 异常的定义	303
12.1.2 异常处理的基本思想	303
§ 12.2 异常处理的编程方法	304
12.2.1 异常处理的三个语句	304
12.2.2 异常处理的执行过程	305
12.2.3 异常处理的基本步骤	305
12.2.4 多类型的异常处理	306
12.2.5 再抛出异常	308
12.2.6 构造函数、析构函数与异常处理	309
§ 12.3 异常处理类	309
思考与练习题	310

第 13 章 C++输入和输出 /312

§ 13.1 流输入输出	312
13.1.1 提取运算符和插入运算符	312
13.1.2 标准出错输出(cerr/clog)	315



13.1.3	输入流类·····	315
13.1.4	输出流类·····	316
13.1.5	格式化输入输出·····	317
§ 13.2	磁盘文件的输入输出·····	324
13.2.1	文本文件的输入输出·····	324
13.2.2	二进制数据文件·····	328
13.2.3	随机数据文件·····	329
	思考与练习题·····	331

程序设计语言 (Programming Language) 是一组用来定义计算机程序的语法规则。它是一种标准化的交流语言, 用来书写计算机程序, 向计算机发出指令。计算机语言能让程序员准确地定义计算机所需要使用的数据, 并精确地定义在不同情况下所应当采取的动作。

C++ 语言是一种使用非常广泛的计算机编程语言, 本章介绍 C++ 语言的发展简况、C++ 语言的特点、C++ 语言程序组成及其结构、关键字和标识符、C++ 的几个特殊语句、简单的输入输出以及数制的基本知识等内容。

§ 1.1 C++ 语言的发展简况及其特点

C++ 语言是在 C 语言的基础之上发展而来的, 是带类的 C 语言。由于它增加了一种抽象数据类型——class 类类型而改变了 C 语言的局限性, 从而支持面向对象的程序设计, 它是当前计算机科学的主流语言之一, 深受程序员的欢迎。

1.1.1 C++ 语言的发展简况

由于 C++ 语言是从 C 语言发展而来的, 介绍 C++ 语言的发展简况当然要先说清楚 C 语言的发展历史。实际上, C 语言是一种编译型的面向过程的程序设计语言, 它最早是从英国剑桥大学一个名叫 Martin Richards 的学者在 20 世纪 60 年代开发的 BCPL (即 Basic Combined Programming Language) 语言的基础上发展而来的。

BCPL 语言是 Martin Richards 在开发系统软件时, 作为描述性语言而使用的一种程序设计语言。该语言具有能够进行结构化的程序设计、能够直接处理与机器本身数据相近的数据类型、能够处理与内存地址相对应的指针计算方式等特点。

1970 年, 美国 Bell 实验室的 Ken. Thompson 在软件开发工作中, 继承和发展了 BCPL 语言的许多特点, 进一步提出了一种叫“B 语言”的语言。并且使用 B 语言描述和开发了当时美国 DEC 公司最新型的 PDP-7 型小型机上的 UNIX 操作系统。

在美国 Bell 实验室实现的更新型小型机 PDP-11 的 UNIX 操作系统的研发工作中, Dennis. M. Ritchie 和 Brian. W. Kernighan 对 B 语言做了进一步的充实和完善, 于 1972 年推出了一种新型的程序语言——C 语言。该语言一经推出就在国际上广泛流行。

20 世纪 80 年代, 由于软件工程的需要, 面向对象程序设计 (Object-Oriented Programming, OOP) 方法在软件设计领域引起了普遍的重视 (实际上, 关于面向对象的程

程序设计的基本概念已经提出了 20 多年)。AT&T Bell 实验室的计算机科学家 Bjarne Stroustrup 结合流行的 C 语言的所有成分,开发出了面向对象的程序设计语言 C++,因此许多人都认为 C++ 是由对 C 的改进或扩充而来的。然而,C++ 本身确实是一种完备的程序设计语言。

图 1.1 说明了 C++ 面向对象程序设计语言的发展简况。

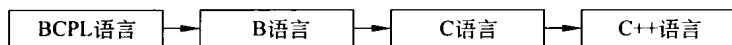


图 1.1 C++ 面向对象程序设计语言发展简况

1.1.2 C++ 语言的特点

C++ 语言发展迅速,已经成为目前最受欢迎的程序设计语言之一,这主要是由于 C++ 程序设计语言具有强大的编程功能以及该语言本身具有诸多特点。目前,许多操作系统平台及其工具软件和应用软件系统均是由 C++ 程序设计语言编写的。

C++ 语言具有以下主要特点:

(1) C++ 语言是一种中级的程序设计语言。

C++ 语言继承了 C 语言的诸多语言特性,它能把高级语言的基本结构和语句与低级语言的实用性结合起来。和 C 语言一样,C++ 语言能够处理汇编语言中的位、字节和地址等数据,而上述这三种数据是计算机系统运行所涉及的最基本的工作单元数据。

(2) C++ 是一种结构化的程序设计语言。

结构化语言的显著特点就是代码和数据能够分离,即程序的各个部分除了必要的信息交流外彼此独立。C++ 语言结构化方式可使编写的程序层次清晰,便于调试、使用以及维护。C++ 语言的每个功能模块都是以函数形式提供给的,这些函数可方便地通过类对象来调用;该语言具有多种条件分支语句、循环语句以控制程序流向,从而使功能模块程序完全结构化。

(3) C++ 是一种面向对象的程序设计语言。

C++ 语言是带类的 C 语言,C++ 语言中的类类型是集抽象性、封装性、继承性和多态性于一体的数据类型,可以帮助人们开发出具有模块化、数据抽象程度高、体现信息隐蔽、可复用、易修改、易扩充等特性的应用程序。

(4) C++ 语言是一种功能齐全的编程语言。

C++ 语言具有各种各样的数据类型,并继承了 C 语言中的指针概念,可使编程人员编程效率更高。C++ 语言也具有访问操作系统功能和强大的 I/O 功能。另外,该语言计算功能和逻辑判断功能也比较强大,完全可以实现智能决策的目的。

(5) C++ 语言是一种移植性很好的编程语言。

C++ 语言适合于多种类型的操作系统,如 Windows 98、Windows NT、UNIX 等,因此,C++ 语言也适用于多种机型。一种系统平台上的 C++ 程序代码可以不加修改或稍加修改就能运行在另一种系统平台上。

当然,C++ 语言还有其他许多特点,这里就不再一一列举。



§ 1.2 C++ 语言程序组成及其结构

C++ 语言像其他语言一样有其基本的程序组成元素和结构,用户可按其规定的格式和提供的语句来编写应用程序。

下面给出一个简单的 C++ 语言程序例子:

【例 1.1】 编写一个求圆柱体体积的应用程序。

```
//计算表达式:  $V=s * l=PI * r * r * l$ 
#include "iostream.h"
#define PI 3.1415926
double Volume(double,double); //函数声明
void main(void)
{
    double radius, high; /* 声明存放半径和高度的两个变量 */
    cout<<"Please enter radius and high: "; //输入提示
    cin>>radius>>high; //输入语句
    cout<<"\nThe volume of cylinder is:"<<Volume(radius, high)<<endl;
}
double Volume(double r, double h) //求体积函数
{
    double v;
    v=PI * r * r * h;
    return v;
}
```

运行结果:

```
Please enter radius and high: 3 5 ✓
The volume of cylinder is:141.372 ✓
```

该例是一个完整的 C++ 语言的程序。

实际上,C++ 语言的程序主要由以下几个基本组成部分构成。

(1) 预处理命令: C++ 语言提供了三类预处理命令: 宏定义命令(define)、文件包含命令(include)、条件编译命令(if-else-endif),预处理命令为我们编写复杂应用程序带来了极大方便。

(2) 输入输出语句(Input/Output 语句,简称 I/O 语句): C++ 语言的程序中总是少不了输入和输出的语句,以实现与程序内部的信息交流。特别是屏幕输出的功能,几乎每个程序都要用到,我们使用它可以把计算机的结果显示在屏幕上。

(3) 函数(Function): C++ 的程序是由若干个文件组成的,每个文件又是由若干个函数组成,因此,可以认为 C++ 的程序就是函数串,即由若干个函数组成,函数与函数之间是相对的,并且是并行的,函数之间可以调用。在组成一个应用程序的若干个函数中,必须有一个也只能有一个函数名称为 main()的函数。



(4) 语句(Statement): 语句是组成程序的基本单元。每个函数都是由若干条语句组成的。但是,空函数是没有语句的函数。语句是由单词(Keyword)组成,单词间用空格分隔,C++ 程序中的语句是以分号结束。分号既是语句的结束符,也是语句之间的分割符。语句除了有声明语句、表达式语句和空语句之外,还有复合语句、分支语句、循环语句和转向语句等。

(5) 变量(Variable): 变量是通过声明语句来定义的,多数程序都需要声明变量和使用变量。在面向对象程序设计中,变量也称为对象(Object)。将对象看做是某类类型的实例,而实例就是某个类的对象。C++ 语言本身对变量和对象不加区分。

(6) 其他: 除了以上讲述的 5 个部分以外,C++ 还有其他组成部分。例如,符号常量和注释信息也是程序的一部分。C++ 程序中尽量把有些常量定义为符号常量,在程序中使用符号常量时,该符号常量就代表着某个确定的常量值。

另外,C++ 程序的书写格式通常需要遵从以下的基本原则:编写程序时,一般一行只写一条语句;短语句可以一行写多个;长语句可以将一条语句分为多行来写;分行原则是不能将一个单词分开;用双引号引用的一个字符串最好也不要分开,如果一定要分开,有的编译系统要求在行尾加续行符("\"),有的编译系统则不要求。

从例 1.1 程序可以看出, C++ 语言编写的程序在格式上与许多语言不一样。

实际上,C++ 语言编写的源代码程序(简称源程序)的格式具有以下特点:

- (1) C++ 语言的关键字是小写字母构成的,习惯上也使用小写字母书写程序;
- (2) 大多数语句结尾必须用 ";" 作为终止符,否则 C++ 不认为该语句的结束;
- (3) 每个程序必须有一个且只能有一个主函数,其名称为 main() 的函数;
- (4) 每个函数的函数体(包括主函数和每个子函数,如例 1.1 中的 main() 函数和 Volume() 函数)必须用一对花括号{和}括起来;
- (5) 一个较完整的程序文件大致含有:包括文件程序段(一组 #include <*.h> 语句)、函数声明部分、全局变量声明、主函数和若干个子函数。在主函数和子函数中又包括局部变量定义、库函数调用、流程控制语句、用户函数的调用语句等;
- (6) 注释部分包含在 /* 和 */ 之间或 // 之后,在编译时它将被 C++ 编译器忽略;
- (7) 像其他一些语言一样,C++ 的变量在使用之前必须先声明其数据类型,未经声明的变量不能使用。声明变量类型应在可执行语句前面,如例 1.1 main() 函数中的第一条语句就是变量声明语句,它必须放在所用的执行语句前面。
- (8) 在 C++ 语言中,大小写字母是有区别的,相同字母的大小写代表不同的变量;
- (9) 在 C++ 语言中,程序的书写格式非常灵活,没有严格限制。例 1.1 的程序可写成下面等价的格式:

```
//计算表达式: V=s* l=PI*r*r* l
#include "iostream.h"
#define PI 3.1415926
double Volume(double,double);           //函数声明
void main() {double radius, high; cout<<"Please enter radius and high: ";cin>>
radius>>high; cout<<"\nThe volume of cylinder is:"<<Volume(radius, high)<<
```