

21世纪高校计算机系列规划教材

Visual C++ 程序设计

潘克勤 华 伟 主编 刘永良 张晓如 王 芳 副主编 张再跃 主审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

清华大学出版社 中国计算机教育出版

Visual C++ 程序设计

清华大学出版社 中国计算机教育出版

清华大学出版社

21 世纪高校计算机系列规划教材

Visual C++ 程序设计

潘克勤 华 伟 主 编

刘永良 张晓如 王 芳 副主编

张再跃 主 审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书共 10 章,分为两个部分:前 6 章是传统的面向过程程序设计,包括基本概念和基本知识、结构化程序设计、数组、函数、链表等,这是程序设计的基础;后 4 章介绍面向对象程序设计基础知识,内容包括类和对象、继承和派生、多态性和文件操作等内容。

本书在编写过程中,以创新教学模式、实行“授人以渔”、强化学习与实践为原则,以易于教、善于教、易于学、乐于学、教学有成效为目标,对传统教材的体系结构进行了调整,分散难点,突出重点。本书可作为高等学校非计算机专业学生学习 Visual C++程序设计用书,也可供编程爱好者参考使用。

图书在版编目(CIP)数据

Visual C++程序设计 / 潘克勤, 华伟主编. —北京: 中国铁道出版社, 2008. 1

(21 世纪高校计算机系列规划教材)

ISBN 978-7-113-08551-3

I. V… II. ①潘…②华… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 009864 号

书 名: Visual C++程序设计

作 者: 潘克勤 华 伟 等

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 张围伟

责任编辑: 周 欢

封面设计: 付 巍

封面制作: 白 雪

特邀编辑: 薛秋沛

责任校对: 张国成

印 刷: 河北省遵化市胶印厂

开 本: 787×1092 1/16 印张: 14.75 字数: 337 千

版 本: 2008 年 1 月第 1 版 2008 年 1 月第 1 次印刷

印 数: 1~3 000 册

书 号: ISBN 978-7-113-08551-3/TP·2685

定 价: 25.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

本教材是根据教育部《关于进一步加强高校计算机基础教学的意见》的教学基本要求和江苏省教育厅对江苏省高等学校非计算机专业学生《计算机基础知识和应用能力等级考试大纲(二级 Visual C++)》的考核内容编写而成的。

随着科学技术的发展和计算机应用的不断深入,社会各行业对大学生计算机应用能力特别是程序设计能力的要求不断提高。

为了更加有效地开展计算机基础教育,不断提高学生对 Visual C++程序设计的认识、实践与应用能力,作者在长期的 Visual C++课程教学过程中努力探索、大胆实践,在注重理论知识教学的同时,不断强化实验教学环节,形成了一整套行之有效的教学方法,并确立了具有自身特色的教学思想,作为长期教学与实践经验的总结,编写并出版《Visual C++程序设计》一书。

本书共 10 章,分为两个部分:前 6 章是传统的面向过程的程序设计,包括基本概念和基础知识、结构化程序设计、数组、函数、链表等,这是程序设计的基础;后 4 章介绍面向对象的程序设计,内容包括类和对象、继承和派生、多态性和文件操作等内容。

本书在编写过程中,以创新教学模式、实行“授人以渔”、强化学习与实践为原则,以易于教、善于教、易于学、乐于学、教学有成效为目标,对传统教材的体系结构进行了调整,分散难点,突出重点。

作为江苏科技大学 Visual C++课程教学改革的重要内容,在本教材成书之前,我们就编写并印刷了讲义,针对 18 个专业的学生进行了为期多年的试教,取得了良好的教学效果。除本教材编写人员外,我校从事计算机基础教学的其他教师在前期讲义试讲、教材编写研讨、文本校对等过程中做了大量工作,提出了许多富有建设性的意见与建议,他们是祁云嵩、邹晓华、於跃成、宋晓宁、段旭、常本勤、邹全、张绛丽、范燕、石亮、潘舒、王勇、黄霞、束鑫、肖寒、严熙,对此我们深表谢意。

另外,教材的出版不仅得到江苏科技大学教材委员会的资助,而且也得到了学校教务处、高教研究所等许多部门的积极支持,尤其是中国铁道出版社为本教材的出版和发行工作给予了大力的支持与帮助,在此一并表示感谢!

由于编者水平所限,书中难免有不当之处,恳请广大读者批评指正。

编 者

2007 年 11 月

目 录

第 1 章 Visual C++程序设计入门	1
1.1 程序设计语言与程序设计	1
1.2 Visual C++字符集与标识符	2
1.3 Visual C++函数的概念	3
1.4 基本数据类型	4
1.5 常量	5
1.5.1 值常量	5
1.5.2 符号常量	7
1.6 变量	8
1.7 文件包含与输入输出流	9
1.8 指针类型变量	11
1.9 引用类型变量	13
1.10 枚举类型	14
1.11 运算符和表达式	15
1.11.1 运算符	15
1.11.2 表达式	18
1.11.3 typedef 语句	19
1.12 练习题	19
第 2 章 流程控制语句	21
2.1 操作运算语句	21
2.2 流程控制语句	22
2.2.1 顺序结构	22
2.2.2 选择结构	23
2.2.3 循环结构	29
2.2.4 循环嵌套	34
2.2.5 break 语句与 continue 语句	37
2.2.6 abort()函数与 exit()函数	38
2.3 练习题	39
第 3 章 数组	40
3.1 一维数组	40
3.1.1 一维数组的定义	40
3.1.2 一维数组的初始化	41

3.1.3 一维数组的基本操作.....	41
3.2 二维数组	44
3.2.1 二维数组的定义	44
3.2.2 二维数组的初始化	45
3.2.3 二维数组的基本操作.....	45
3.3 字符数组	47
3.3.1 字符数组的定义与初始化.....	47
3.3.2 字符数组的基本操作.....	48
3.3.3 字符串处理函数	49
3.4 数组与指针	52
3.4.1 指针变量与一维数组.....	52
3.4.2 指针变量与二维数组.....	55
3.4.3 通过指针变量操作字符数组	59
3.5 指针数组	61
3.5.1 指针数组的定义	61
3.5.2 通过指针数组操作一维数组和二维数组.....	61
3.5.3 通过指针数组和指向指针的指针变量操作系列字符串.....	62
3.6 练习题	64
第 4 章 函数.....	65
4.1 概述.....	65
4.2 函数的定义与调用.....	66
4.3 函数嵌套调用和递归调用	68
4.3.1 函数的嵌套式调用	68
4.3.2 函数的递归调用	70
4.4 函数的参数传递.....	73
4.4.1 函数参数为数值变量的传递形式	73
4.4.2 函数参数为地址的传递形式	74
4.4.3 函数参数为引用变量的传递形式	80
4.5 函数与指针	82
4.5.1 指针类型的函数	82
4.5.2 指向函数的指针变量.....	83
4.5.3 带参数的 main()函数	87
4.6 函数的其他特性.....	87
4.6.1 内联函数.....	87
4.6.2 函数的重载.....	88
4.6.3 函数参数的默认值	88

4.7 练习题	89
第 5 章 作用域和编译预处理	91
5.1 作用域	91
5.2 存储类型	96
5.3 编译预处理	101
5.3.1 文件包含	102
5.3.2 宏定义	102
5.3.3 条件编译	106
5.4 练习题	107
第 6 章 结构体与简单链表	109
6.1 结构体	109
6.1.1 结构体类型的定义	109
6.1.2 结构体类型变量的说明	110
6.1.3 结构体类型变量的引用	111
6.1.4 指向结构体变量的指针变量	113
6.1.5 结构体数组	115
6.1.6 用结构体变量作函数参数	115
6.2 链表	118
6.2.1 new 和 delete 运算符	118
6.2.2 链表	119
6.3 链表的基本操作	120
6.3.1 链表结点的创建	120
6.3.2 链表的建立	121
6.3.3 链表的输出	122
6.3.4 释放链表的结点空间	123
6.4 链表的综合操作	124
6.4.1 结点的删除	125
6.4.2 结点的插入	126
6.4.3 查找结点	128
6.4.4 链表的综合举例	129
6.5 练习题	132
第 7 章 类和对象	134
7.1 面向对象程序设计概述	134
7.2 类和对象	135
7.2.1 类和对象的概念	135
7.2.2 类的定义	136

7.2.3	成员函数.....	137
7.2.4	对象的定义和使用.....	139
7.2.5	类的作用域.....	141
7.3	构造函数.....	142
7.3.1	对象赋值的方法.....	142
7.3.2	构造函数.....	143
7.3.3	类型转换构造函数.....	149
7.3.4	复制构造函数.....	150
7.3.5	this 指针.....	151
7.4	析构函数.....	152
7.4.1	析构函数的定义.....	152
7.4.2	析构函数的使用.....	153
7.4.3	不同存储类型的对象调用构造函数及析构函数.....	155
7.5	练习题.....	157
第 8 章	继承和派生.....	159
8.1	基类和派生类.....	159
8.2	继承.....	160
8.2.1	单一继承.....	160
8.2.2	多重继承.....	164
8.2.3	派生类的构造函数.....	165
8.2.4	派生类的析构函数.....	169
8.3	冲突、支配规则和赋值兼容性.....	169
8.3.1	冲突.....	169
8.3.2	支配规则.....	170
8.3.3	赋值兼容性.....	171
8.4	静态成员.....	172
8.4.1	静态数据成员.....	172
8.4.2	静态成员函数.....	173
8.5	友元函数与友元类.....	174
8.5.1	友元函数.....	174
8.5.2	友元类.....	176
8.6	虚基类.....	177
8.7	练习题.....	179
第 9 章	多态性.....	181
9.1	多态性的概念.....	181
9.1.1	多态性的含义.....	181

9.1.2 多态性的种类	181
9.2 虚函数与运行的多态性	182
9.2.1 虚函数	182
9.2.2 通过虚函数实现运行的多态性	183
9.2.3 虚函数与构造函数、析构函数	184
9.2.4 纯虚函数	185
9.3 运算符重载与编译的多态性	187
9.3.1 运算符重载的概念	187
9.3.2 运算符重载的方法	188
9.3.3 几种典型的运算符重载	191
9.3.4 字符串类的运算符重载	196
9.4 抽象类	198
9.5 练习题	198
第 10 章 流类和文件	200
10.1 Visual C++ 的基本流类体系	200
10.1.1 基本流类体系	200
10.1.2 流的格式控制	201
10.1.3 使用 I/O 成员函数	204
10.1.4 运算符“<<”和“>>”的重载	205
10.2 文件的概述	207
10.3 文件流	208
10.4 文件的打开与关闭	209
10.4.1 打开文件	209
10.4.2 关闭文件	210
10.5 文件的读写操作	210
10.5.1 文本文件的读写操作	211
10.5.2 二进制文件的读写操作	213
10.5.3 随机访问文件的函数	215
10.6 练习题	215
参考文献	217
附录 A ASCII 码表	218
附录 B 常用库函数	220

第 1 章

Visual C++程序设计入门

1.1 程序设计语言与程序设计

1. 程序设计语言

在日常生活中，人们用自然语言进行相互之间的交流。然而，当人们运用计算机解决实际问题时，就必须使用计算机能够“理解”的语言，这类语言通常称为计算机程序设计语言。程序设计语言的发展经历了机器语言、汇编语言、高级语言三大阶段。其中，机器语言是以二进制代码表示的语言；汇编语言是采用助记符表示的语言；而高级语言则是与自然语言较为接近的语言，因此得到普遍使用。高级语言的种类很多，如 Basic、Pascal、FORTRAN、Visual FoxPro、C、C++、Visual C++等，虽然它们有着各自的特点和应用范围，但是它们在表达和功能体现等方面所产生的共性多半是相通的。因此，努力学好并掌握一门程序设计语言，往往可触类旁通，不仅可以更好地驾驭计算机，而且也会为今后充分挖掘自身的潜力和提高自身的能力打下坚实的基础。

2. 程序设计语言的基本要素和基本成分

语言是用符号来表示的，如汉语中的字、英语中的字母等，一种语言使用的符号全体称为该语言的“符号集”。符号按照一定规则组合起来形成词，词与词按照一定规则组合起来形成语句，一些语句按照一定规则排列起来便形成文章，这其中的规则称为“语法”。词和语句都有一定的含义，一些语句通常表达一种思想，称为“语义”。计算机程序设计语言也不例外，“符号集”、“语法”和“语义”是计算机程序设计语言的三大基本要素，其中“符号集”通常由计算机键盘上的一些符号组成；“语法”是语句形成与相互间关系的约定，学习过程中需要通过记忆加以理解；“语义”是语句和语句组合所能实现的功能，要通过反复的练习和实践才能得以掌握。

高级语言的种类千差万别，但是其基本成分都可归纳以为以下 4 种。

(1) 数据成分：关于数据的描述，如整型、实型、指针、引用、数组、结构体、类等。

- (2) 控制成分: 关于控制结构的描述, 如顺序、条件、循环等。
- (3) 运算成分: 关于数据运算的描述, 如算术运算、关系运算、逻辑运算等。
- (4) 传输成分: 关于信息传输的描述, 如输入操作、输出操作等。

3. 程序设计

程序设计 (Programming) 是指设计、编制、调试程序的方法和过程, 是目标明确的智力活动。程序并非软件, 但它却是软件的本体, 软件的质量主要通过程序的质量来体现。因此, 在软件研究与开发中, 程序设计的工作非常重要。

程序设计涉及的内容很多, 如有关的基本概念、工具、方法等。一般而言, 在程序设计过程中, 首先要搞清楚“做什么”, 即所谓的“需求分析”; 接下来是“如何做”, 包括数据结构的选择、算法设计和结构设计等; 最后是编程实现。程序设计往往不会“一蹴而就”, 需要经过反复的修改、运行和调试, 才能最终达到目的。

4. 程序设计的基本要求

- (1) 程序的正确性: 根据任务设计的程序要求正确无误, 包括语法正确、语义正确和算法正确等。
- (2) 程序的可读性和可理解性: 设计的程序被他人阅读时, 做到容易读懂、容易理解, 通常包括程序的书写格式规范、程序的结构性好、程序中注解详细等。
- (3) 程序的可维护性: 程序易于修改, 易于增加新的功能。

1.2 Visual C++字符集与标识符

1. Visual C++字符集

一般而言, 计算机键盘上除功能键以外的大多数键符都可以用做 Visual C++语言的字符。字符集中的字符可以组成语句, 形成程序代码, 实现一定功能。Visual C++字符集由下列字符组成。

- (1) 大小写英文字母: a~z 和 A~Z。
- (2) 数字: 0~9。
- (3) 下画线: _。
- (4) 其他符号¹。

2. Visual C++语言的关键字

Visual C++程序设计语言中事先规定的、有特定含义与用途的词汇称为关键字 (又称保留字), 如 void、int、char 等。用户不能用关键字来命名变量名、函数名等。

3. Visual C++语言的标识符

标识符是程序员根据编程需要, 自行定义的函数名、变量名、数组名等。用户定义 Visual C++标识符的规则是: 第一个字符只能是英文字母或下画线, 后面可跟英文字母、数字、下画线。

特别提示: 用户自定义的标识符不能是 Visual C++语言的关键字。

¹ 详见教材《Visual C++程序设计解析与实训》第1章。

合法，非法标识符定义实例如表 1-1 所示。

表 1-1 合法、非法标识符定义实例

合法标识符	print	_good	MD300	Void
非法标识符	123	-good	A+B	void
非法原因	数字开头	非法字符开头	中间使用非法字符	使用关键字

1.3 Visual C++函数的概念

程序在运行过程中通常有输入和输出，如果 F 表示某个程序，输入 $X_1、…、X_n$ ，输出为 Y ，那么 Y 可以看成输入 $X_1、…、X_n$ 关于程序 F 函数，形式上可表示成 $Y=F(X_1,X_2,…X_n)$ 。换句话说就是程序可视为函数，因此，在 Visual C++程序设计语言中所有程序均为函数的形式，函数是 Visual C++程序编译²的基本单位。Visual C++函数通常分为自定义函数与库函数。

1. Visual C++主函数

主函数是用户自定义的函数，Visual C++主函数基本结构的严格定义将在第 4 章给出。下面是一个主函数的例子：

```
void main(void) { }
```

其中 void 是关键字，main 是函数名，表示该函数是主函数，第一个 void 表示函数类型，第二个 void 表示参数类型，可以省略，“{”和“}”之间的内容为函数体，实现程序的具体功能。

特别提示：Visual C++任何程序的执行都从主函数开始，并且与它在源程序中的位置无关。

上面定义的主函数可以通过编译，运行也没有错，但由于函数体为空，因而没有任何实际功能。

【例 1.1】在计算机显示器上显示一行字符“北京 2008 奥林匹克运动会”。

编程实现：

```
/* 程序名: cl_1.cpp                                //程序的第一部分
编制时间: 2007 年 8 月 1 日
主要功能: 输出字符串 */
#include <iostream.h>                                //程序的第二部分
void main(void)                                     //程序的第三部分
{
    cout << "北京 2008 奥林匹克运动会";           //函数体
}
```

调试与运行：

北京 2008 奥林匹克运动会

程序解读：

下面以该题为例，简单介绍程序的基本结构和各部分的作用。

(1) 程序的第一部分为多行注释，从标志“/*”起至“*/”结束；在程序中还包含单行注释，

² 用户用高级程序设计语言编写的程序称为源程序，计算机必须将源程序转化成目标程序（机器代码程序）后才能运行。转化的方式有解释和编译两种，其中解释是逐条指令转化并执行；编译则是整个程序转化后再执行。

从标志“//”起至该行结束。注释起到对程序说明的作用,以增强程序的可读性和可理解性。删除这些注释对程序的运行结果没有影响。

(2) 程序的第二部分为编译预处理指令,该例中的作用是将输入、输出流对象包含进来。

(3) 程序的第三部分为程序主体,由主函数 main 实现程序输出字符串的功能。

(4) Visual C++编译器严格区分大、小写字母。

(5) 程序的书写规则:对 Visual C++的编译器而言,一个语句可以写成若干行,一行内也可以写若干个语句。为了便于程序的阅读与理解,程序员编写源程序代码最好符合以下要求。

- 对齐规则:同一层次的语句最好从同一列开始,同一层次的花括号最好在同一列上。
- 缩进规则:属于内一层次的语句必须缩进几个字符。

学习与实践:

(1) 在#include <iostream.h>部分行前加注释符“//”,程序调试将会怎样?

(2) 改变主函数名分别为 mian、Main,程序调试将会怎样?

(3) 省略 main (void)中的 void,考查运行结果。

2. Visual C++库函数

Visual C++编译系统定义了许多常用的函数,为用户编写程序提供了很大的便利。只要在程序中包含其相应的头文件,在程序中就可以直接使用库函数。常用数学公式库函数如表 1-2 所示,其包含的编译预处理指令为:include <math.h>。

表 1-2 常用数学函数

函 数 名	含 义	实 例	结 果
log(x)	求以 e 为底的自然对数	log(10)	2.3
fabs(x)	求实数绝对值	fabx(-5)	5
sqrt(x)	求平方根	sqrt(9)	3.0
sin(x)	求正弦函数	sin(3.14*30/180)	0.5

【例 1.2】编写程序,利用库函数求-13 绝对值的平方根。

编程实现:

```
#include <iostream.h>
#include <math.h>    //使用数学公式库函数必须包含此头文件
void main(void)
{
    //先通过 fabs(-13) 求出-13 的绝对值,再通过 sqrt(13) 求其平方根
    cout <<“-13 绝对值的平方根为: ”<<sqrt( fabs(-13));
}
```

调试与运行:

-13 绝对值的平方根为: 3.60555

1.4 基本数据类型

在 Visual C++中数据类型可以分成两大类:一类是基本类型,由系统提供,用户可直接使用;另一类为构造类型,是在基本类型的基础上,由系统或用户自行定义的。

特别提示：程序中用到的数据都必须指明其数据类型。

指明数据类型的作用有两个：

(1) 指明数据占用存储空间的大小（占用字节数）。

(2) 规定了数据允许执行的操作或运算。

Visual C++ 中常用的基本数据类型如表 1-3 所示。

表 1-3 Visual C++ 中常用的基本数据类型

名 称		类 型	占用字节数	取 值 范 围
字符型		char	1	-128~127
整型		int	4	$-2^{31} \sim (2^{31}-1)$
短整型		short int	2	-32 768~32 767
长整型		long int	4	$-2^{31} \sim (2^{31}-1)$
实型	单精度	float	4	$-10^{38} \sim 10^{38}$
	双精度	double	8	$-10^{308} \sim 10^{308}$
无值型		void	0	无值

在 Visual C++ 程序设计中经常要使用变量，在使用前通常要说明变量的数据类型，用以给出变量的变化范围。

1.5 常 量

常量是在程序运行过程中值保持不变的量。常量有两种形式：一是以字面值的形式直接出现在程序中，如 2.16、'A' 等，称为值常量；二是以符号的形式表示，称为符号常量。

1.5.1 值常量

值常量有整型常量、实型常量、字符型常量和字符串常量。

1. 整型常量

整型常量通常由十进制整数表示，也可以用八进制和十六进制表示，如：123、-256 等。

2. 实型常量

实型常量包含小数点和 10 的方幂表示的数，其表示方式有十进制小数形式和指数形式。

十进制表示形式如：3.5、-2.0、78、+32.2、0.5 等。

指数表示时 1.25×10^{-9} 可表示为：1.25E-9（或 1.25e-9）。

特别提示：用指数形式表示时，在 E 或 e 的前面必须有数字，且在其后必须是整数。

例如：e12、E9、.e4、1.0e3.0 都是不合法的实数。

3. 字符型常量

字符型常量是用单引号引起来的单个字符。如 'a'、'5'、'+' 等。在计算机内部是以它们的 ASCII 码表示的，每个字节存储一个字符。

值得注意的是, Visual C++程序设计中, 有些控制功能是通过一些特殊的字符来实现的, 为了区分这些字符在其功能表示和一般字符表示方面的不同, Visual C++提供了一种称为转义字符的表示方法, 就是用反斜杠加在其字符前来表示该字符是一个特殊功能符号。常用的转义字符如表 1-4 所示。

表 1-4 常用转义字符³

字符形式	功能或用途
'\n'	换行
'\a'	响铃
'\t'	水平制表符 (Tab 键)
'\\'	输出反斜杠符
'\''	输出单引号符
'\"'	输出双引号符
'\ddd'	输出 1~3 位八进制数所代表的字符
'\xdd'	输出 1~2 位十六进制数所代表的字符

【例 1.3】下列程序是一个关于转义字符运算的, 分析程序运行结果。

```
#include "iostream.h"
void main(void)
{
    cout <<'m'<<'\\t'<<'n'<<'\\n'<<'\\''<<endl;    //A 行
    cout <<'\\101'<<'\\x61'<<endl;
}
```

调试与运行:

```
m      n
,
Aa
```

程序解读:

(1) 要注意常量的变化: 101 为十进制数, 而 '\\101'则表示一个 ASCII 码字符'A'。

(2) 同样, '\\x61'是用十六进制数表示一个 ASCII 码字符'a'。

不管是用八进制数还是用十六进制数表示一个 ASCII 码字符, 取值范围对应的十进制数都是在 0~255 之间。

学习与实践:

(1) 将程序中 A 行换为 “cout <<6<<'6'<<'\\6'<<6+'6'<<endl;”, 输出如何?

(2) 将程序中 A 行换为 “cout <<'a'<<'\\a'<<'b'<<'\\b'<<endl;”, 输出如何?

4. 字符串常量

用双引号引起来的若干个字符称为字符串常量。字符串常量在计算机中是按字符的 ASCII 码顺序逐个存放的, 每个字节存储一个字符。在字符串常量存储过程中, 系统在其最后自动存放一

³详见教材《Visual C++程序设计解析与实训》第 1 章。

⁴详见附录 A。

个字符'\0'，它是字符串结束的标志，其 ASCII 码值是 0。

例如，字符串"china"、"a"和字符'a'的存储方式如图 1-1 所示。



图 1-1 字符串与字符的不同存储方式

学习与实践：

(1) 将例 1.3 中 A 行换为“cout<<"m\tn\n\'x"<<endl;”，输出如何？

(2) 将例 1.3 中 A 行换为“cout<<"abc\101\0x61"<<endl;”，输出如何？

1.5.2 符号常量

如果在程序中经常用到某些常数，则可以用一些符号来标识这些常数。经过定义后，这些标识符便成了符号常量。这样做往往有利于程序的编写、阅读和理解。Visual C++中可以用 const 或 #define 定义符号常量。

1. 用常量说明语句 const 定义符号常量

格式如下：

const 数据类型 符号常量=值常量；

其中，const 为关键字，数据类型用来定义符号常量的类型，它们之间的空格必不可少，最后以分号结束。

例如：

```
const int MAX=666;
```

该语句定义了一个整型符号常量 MAX，MAX 的值为 666。

2. 用预处理命令#define 说明一个符号常量

格式如下：

```
#define 符号常量 值常量
```

注意：该命令是一条预处理命令，不允许带数据类型。

例如：

```
#define PI 3.1415926
```

【例 1.4】利用符号常量，求出半径为 5 的圆的周长。

分析：

根据圆的周长公式求值。

编程实现：

```
# include "iostream.h"
```

```
# define PI 3.1415926
```

```
void main (void)
```

```
{
```

```
const int R=5;
```

```
cout<<"半径为 5 的圆的周长为："<<2*PI*R<<endl;
```

```
}
```