

21世纪高等学校精品教材

主 编 王永国

Visual C++ 程序设计



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等学校精品教材

Visual C++程序设计

主 编 王永国

参 编 王永国 丁亚涛 张其亮 翟素兰



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书以少课时、没有 C/C++ 程序设计基础、直接学习 Visual C++ 6.0 语言的读者为对象, 兼顾相关考试, 由浅入深地介绍了 C++ 语言的概念、语法、结构化程序设计的思想、面向对象的程序设计方法及 Visual C++ 可视化编程基础。

全书共分 7 章, 其中前 5 章讲述 Visual C++ 6.0 语言的基础内容, 第 6 章介绍面向对象的程序设计的知识, 最后一章介绍 Visual C++ 6.0 可视化编程的基本方法。本书以案例驱动教学, 内容精炼、结构紧凑、通俗易懂、重点突出, 注重实用与能力的培养, 它克服了一般教科书学习 C++ 枯燥、学了以后却不会用 Visual C++ 在开发环境中解决问题的难点, 有助于提高学生的学习兴趣。

书中精选的大量例题、习题都经过测试, 并可在与本书配套的《Visual C++ 程序设计实训与考试指导》一书中找到参考答案及其源代码, 也可通过实训教材的配套光盘在机器上练习评分。光盘中的“Visual C++ 程序设计”课程教学包可以说是一本学习 Visual C++ 的百科全书与立体化教材, 从 Visual C++ 的学习方法、电子课件、上机实训、综合案例, 到试卷生成系统、等级考试模拟训练系统等, 信息量大, 完全能够满足教师教学、学生自学测试的需要, 对培养学生的实战能力与创新精神有重要指导作用。

本书除供高校非计算机专业及信息与计算科学专业使用外, 也可作为各类技术人员学习 Visual C++ 语言或参加全国计算机等级考试 C++ 的学习参考书。

本书相关教学资源读者可以从中国水利水电出版社网站上免费下载, 网址为:
<http://www.waterpub.com.cn/softdown/>。

图书在版编目 (CIP) 数据

Visual C++ 程序设计 / 王永国主编. —北京: 中国水利水电出版社, 2008

21 世纪高等学校精品教材

ISBN 978-7-5084-6002-4

I. V… II. 王… 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 166475 号

书 名	21 世纪高等学校精品教材 Visual C++ 程序设计
作 者	主 编 王永国 参 编 王永国 丁亚涛 张其亮 翟素兰
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 18 印张 450 千字
版 次	2008 年 11 月第 1 版 2008 年 11 月第 1 次印刷
印 数	0001—4000 册
定 价	28.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

前 言

为了更好地服务于教学并兼顾相关考试,根据教育部《关于进一步加强高校计算机基础教学的意见》中有关“计算机程序设计基础课程教学基础要求”的精神,考虑到高等学校教学计划要求压缩课堂教学时数的发展趋势与加强学生创新能力的培养方向,结合我们近年来从事 C/C++/Visual C++ 程序设计的教学体会与开发经验,我们编写了《Visual C++ 程序设计》一书。

Visual C++ 是微软公司开发的基于 Windows 平台的 C++ 可视化开发环境,学习和掌握标准 C++ 内容是使用 Visual C++ 开发必不可少的环节,C++ 内容既系统化又与 Visual C++ 相互联系。因此该书在编写过程中注重将传统的面向过程程序设计、流行的面向对象程序设计及 Visual C++ 可视化编程有机地结合起来。全书共分 7 章,其中前 5 章主要介绍了 Visual C++ 环境下 C++ 的基础内容,包含 Visual C++ 概述、数据类型、运算符与表达式、结构化程序设计、数组和字符串、指针、结构与链表、函数等基础知识;第 6 章主要是流行的面向对象程序设计的基本内容,包括类与对象、继承和派生、多态性和虚函数、I/O 流等;第 7 章是 Visual C++ 可视化编程部分,介绍了 API 与 MFC 应用程序框架、常用控件及其成员函数的使用,菜单、工具栏与状态栏的设计,SDI 与 MDI 程序开发基础等,对提高学生学习的积极性与创新能力的培养有重要作用。

本书具有以下特点:

(1) 以少课时、没有 C/C++ 程序设计基础、直接学习 Visual C++ 6.0 语言的读者为对象,考虑到知识层次渐进的特点,由浅入深地介绍了 C++ 语言的概念、语法、结构化程序设计思想、面向对象的程序设计方法及 Visual C++ 可视化编程基础。

(2) 以 Microsoft Visual C++ 6.0 简体中文版为依据,通过案例驱动教学,内容精炼、结构紧凑、通俗易懂、重点突出,注重实用与能力的培养,从而克服了一般教科书学习 C++ 枯燥、学了以后不会用 Visual C++ 在开发环境中解决问题的难点,有助于提高学生的学习兴趣。

(3) 每一章均精选了大量的例题、习题,且都经过测试,并可在与本书配套的《Visual C++ 程序设计实训与考试指导》一书中找到参考答案及其源代码,也可通过实训教材的配套光盘在机器上练习评分。该书由实验、习题及解答、《Visual C++ 程序设计》课程教学包及其使用与考试指导 4 部分组成。为了适应新的教学需求、提高教学质量,配套光盘中含有我们开发的“Visual C++ 程序设计”课程教学包,该教学包可以说是一本学习 Visual C++ 的百科全书与立体化教材,从 Visual C++ 的学习方法、电子课件、上机实训、综合案例,到试卷生成系统、C++ 等级考试模拟训练系统等,信息量大,完全能够满足教师教学、学生自学测试的需要,对培养学生的实战能力与创新精神有重要指导作用。使用本书并需要后台管理功能的老师可以通过出版社与作者联系。

本书由王永国主编,参与编写的人员有王永国、丁亚涛、张其亮、翟素兰。另外参与本书编写、习题测试与校对的还有王多超、程中林等。本书在出版过程中,得到了许多同仁的关心和帮助,中国水利水电出版社计算机编辑室的领导和编辑也为本书的编辑和出版给予了大力支持和悉心指导,在此一并致谢。

由于作者水平有限,书中难免会有一些疏漏和错误,希望广大读者批评指正,以便我们再版时修正,如果你有好的建议或要求欢迎与我们联系,我们的 E-mail: ygwang21@163.com。

作 者

2008 年 8 月

目 录

前言

第 1 章 Visual C++概述.....	1
1.1 C++语言的发展与特点	1
1.1.1 C++语言的发展	1
1.1.2 C++语言的特点	2
1.2 简单的 C++程序	3
1.3 C++基本语法成分	5
1.3.1 C++字符集	5
1.3.2 C++语言的词汇	6
1.4 基本输入输出	7
1.5 Visual C++ 6.0 简介	11
1.5.1 Visual C++集成开发环境	11
1.5.2 简单 C++程序的编写、运行过程	14
习题一	18
第 2 章 数据类型、运算符与表达式	20
2.1 数据类型	20
2.1.1 数据类型分类	20
2.1.2 基本数据类型	21
2.2 常量、变量和函数	21
2.2.1 常量	21
2.2.2 变量	24
2.2.3 标准函数	25
2.3 运算符和表达式	26
2.3.1 运算符	27
2.3.2 表达式	31
2.4 数据类型转换	32
2.4.1 自动类型转换	32
2.4.2 强制类型转换	33
习题二	35
第 3 章 结构化程序设计	38
3.1 程序设计概述	38
3.2 C++语句	39
3.3 顺序结构	40

3.4 选择结构	43
3.4.1 if 语句	43
3.4.2 switch 语句	48
3.5 循环结构	50
3.5.1 循环的基本概念	50
3.5.2 while 语句	51
3.5.3 do...while 语句	52
3.5.4 for 语句	53
3.5.5 循环的嵌套	54
3.6 其他控制语句	55
3.6.1 goto 语句	55
3.6.2 break 语句	56
3.6.3 continue 语句	57
3.7 常用算法	58
习题三	63
第4章 数组、指针与字符串	71
4.1 数组的基本概念	71
4.2 一维数组	72
4.2.1 一维数组的定义、初始化和引用	72
4.2.2 一维数组的应用	73
4.3 二维数组	78
4.3.1 二维数组的定义、引用与初始化	78
4.3.2 二维数组的应用	79
4.4 字符数组	81
4.4.1 字符数组的定义	82
4.4.2 字符数组的初始化	82
4.4.3 字符数组的引用	82
4.4.4 字符串处理函数	82
4.5 指针与数组	84
4.5.1 指针	84
4.5.2 动态存储空间	87
4.5.3 指针与一维数组	88
4.5.4 指针与二维数组	89
4.5.5 指针与字符串	91
4.6 结构与链表	93
4.6.1 结构	93
4.6.2 链表基础	96
4.7 常用算法	99
习题四	104

第5章 函数	112
5.1 函数概述	112
5.2 函数的定义、调用和说明	113
5.2.1 函数的定义	113
5.2.2 函数的调用	114
5.2.3 函数说明	117
5.3 函数间的参数传递	118
5.3.1 参数的传递方式	118
5.3.2 函数参数默认	124
5.4 内联函数	125
5.5 递归函数	126
5.6 函数重载	128
5.7 函数模板	129
5.7.1 函数模板的定义	130
5.7.2 模板实参的省略	132
5.7.3 函数模板的定制	134
5.8 作用域与存储类别	135
5.8.1 变量的存储属性	135
5.8.2 变量的存储类型	136
5.9 应用举例	140
习题五	144
第6章 面向对象程序设计基础	154
6.1 面向对象程序设计概述	154
6.1.1 面向对象程序设计的产生	154
6.1.2 面向对象的基本概念	156
6.1.3 面向对象程序设计的特点	157
6.2 类和对象	158
6.2.1 类的定义	158
6.2.2 对象的定义	160
6.2.3 对象的初始化	162
6.2.4 常对象和常成员	168
6.2.5 静态成员	171
6.2.6 友元	174
6.3 继承和派生	178
6.3.1 基类和派生类	178
6.3.2 单继承	178
6.4 多态性和虚函数	186
6.4.1 函数重载与运算符重载	186
6.4.2 虚函数	189

6.4.3 抽象类	191
6.5 C++输入/输出流	192
6.5.1 I/O 流类	193
6.5.2 标准输入输出流的成员函数	193
6.5.3 格式化 I/O 流	195
6.5.4 磁盘文件	197
习题六	204
第 7 章 Visual C++可视化编程基础	220
7.1 Windows 程序设计基本概念	220
7.1.1 基于 Visual C++的 Windows 应用程序设计方法	220
7.1.2 Windows 程序设计的基本概念	221
7.1.3 Windows 应用程序中的常见消息	223
7.2 API 及应用程序基本框架	223
7.2.1 Windows 应用程序的组成	223
7.2.2 Windows API 应用程序实例	224
7.2.3 Windows API 应用程序结构分析	226
7.3 MFC 及应用程序框架	227
7.3.1 MFC 概述	227
7.3.2 宏和全局函数	228
7.3.3 应用程序框架	229
7.4 对话框与常用控件	230
7.4.1 创建对话框程序的步骤	230
7.4.2 对话框的调用	232
7.4.3 通用消息对话框	234
7.4.4 控件基本操作	235
7.4.5 CString 类	237
7.4.6 常用控件与功能函数	239
7.4.7 综合应用	256
7.5 菜单、工具栏和状态栏	258
7.5.1 菜单	258
7.5.2 工具栏	263
7.5.3 状态栏	265
7.6 文档/视图结构应用程序	268
习题七	271
附录 1 ASCII 码表	274
附录 2 C++常用标准库函数	276
参考文献	280

第 1 章 Visual C++概述

【本章导读】本章主要介绍 C++语言的发展历史、基本特点，简单的 C++语言程序结构、语法成分，基本输入输出和 Visual C++集成开发环境，目的是为以后各章的学习奠定基础。

要求通过本章的学习，使读者了解 C++语言的发展历史和基本特点；掌握 C++语言程序的基本结构和语法组成；掌握基本输入输出语句的使用及利用 Visual C++集成开发环境编辑、编译、连接、运行程序的过程，能够发现并排除程序中的常见错误。

1.1 C++语言的发展与特点

1.1.1 C++语言的发展

早在 20 世纪 60 年代, Martin Richards 为便于软件人员开发系统软件设计出了 BCPL 语言。1970 年, Ken Thompson 在吸收 BCPL 语言优点的基础上设计了 B(Basic Combined Programming Language) 语言, 但 B 语言功能有限。1972 年, 美国贝尔实验室的 Dennis Ritchie 和 Brian Kernighan 根据 B 语言开发设计出了 C 语言。最初, 发明 C 语言的目的是用它来代替汇编语言为小型机 DEC-11 编写 UNIX 操作系统。后来, 随着 UNIX 操作系统的推广, C 语言被越来越多的程序设计人员所了解和使用。到 20 世纪 70 年代末, C 语言已经凭借其如下独有的优势风靡了全世界的程序设计领域:

- (1) 语言结构化、简洁紧凑, 使用灵活方便。
- (2) 具有丰富的运算符和数据类型。
- (3) 兼有汇编语言与高级语言的特点, 接口开放, 适合开发系统软件。
- (4) 程序的运行效率高, 代码的可移植性好。

然而, 随着 C 语言的广泛应用, 它的一些不足也受到了人们的关注, 如:

- (1) 类型检查机制相对较弱, 这使得程序中的一些错误不能及时被发现。
- (2) 缺少支持代码重用的语言结构, 因此为一个程序设计的模块很难再用于其他程序。
- (3) 没有面向对象技术的支持, 不适合开发大型软件, 当程序的规模大到一定程度时, 维护工作会变得相当复杂。

为了满足开发大规模程序的需要, 1980 年贝尔实验室的 Bjarne Stroustrup 和他的同事们开始对 C 语言进行改进和扩充, 把 Simula67 (一种早期的面向对象语言) 中类的概念引入到 C 语言, 并将改进后的 C 语言称为“带类的 C”(C with class), 1983 年正式命名为 C++ (C Plus Plus)。1985 年由 Bjarne Stroustrup 编写的《C++程序设计语言》一书出版, 标志着 C++ 1.0 版本的诞生。此后, 贝尔实验室于 1989 年和 1993 年分别推出了 C++ 2.0 版本和 C++ 3.0 版本。表 1.1 列出了各版本中 C++语言所添加的一些新特性。

表 1.1 C++支持的新特性

版本	年份	在 C 语言基础上添加的新特性
带类的 C	1980	类和派生类、公有成员和私有成员、构造函数和析构函数、友元、内联函数、赋值运算符的重载
C++ 1.0	1985	虚函数、函数运算符的重载、引用、const 常量
C++ 2.0	1989	类的保护成员、多重继承、赋值和初始化的递归定义、抽象类、静态成员函数、const 成员函数
C++ 3.0	1993	模板、异常、类的嵌套、名字空间

经过对 C++语言的 3 次修订后,美国国家标准委员会(ANSI)于 1994 年制定了 ANSI C++ 标准草案,这个草案最终于 1998 年被国际标准化组织(ISO)批准为国际标准(ISO/IEC 14882)。

由于上述原因, C++语言受到软件厂商的极大支持,纷纷推出各自的商业化 C++编译系统,从早期的 Turbo C++、Borland C++、Watcom C++、Quick C++到流行的 Visual C++、C++ Builder 等。

1.1.2 C++语言的特点

在众多的高级程序设计语言中, C++能够取得成功的原因在于它有着许多与众不同的优点,这些突出的特点主要体现在以下几个方面:

(1) C++是 C 语言的超集。

所谓“C++是 C 语言的超集”是指 C++中包含 C 语言的全部语法特征。因此,每一个用 C 语言编写的程序都是一个 C++程序。C++语言的设计宗旨就是在不改变 C 语言语法规则的基础上扩充新的特性。C++与 C 语言的关系可以用图 1.1 表示。

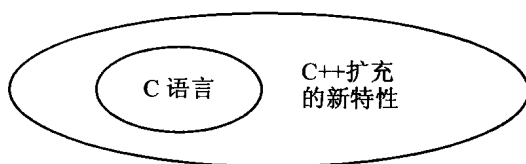


图 1.1 C++与 C 语言的关系

实际上,能够很好地兼容 C 语言正是 C++取得成功的原因之一,这是因为:

- 1) C++继承了 C 语言简明、高效、灵活等众多优点。
- 2) 以前使用 C 语言编写的大批软件可以不加任何修改直接在 C++开发环境下运行。
- 3) C 语言程序员只需要学习 C++扩充的新特性就可以很快地使用 C++编写程序。

(2) C++是一种面向对象的程序设计语言。

C++语言支持几乎所有的面向对象程序设计特征。可以说, C++语言集中体现了近 20 年来在程序设计和软件开发领域出现的新思想和新技术,这主要包括:

- 1) 抽象数据类型。
- 2) 封装和信息隐藏。
- 3) 以继承和派生方式实现程序的重用。
- 4) 以运算符重载和虚函数来实现多态性。

5) 以模板来实现类型的参数化。

(3) C++具有很好的通用性和可移植性。

C++语言是一种标准化的、与硬件基本无关的、广泛使用的程序设计语言，继承了C语言灵活、高效的优点，具有很好的通用性和可移植性。

(4) C++具有丰富的数据类型和运算符，并提供了功能强大的函数库。

由于具有上述特点，C++已经开始取代C语言，被广泛地应用于各种领域的程序设计工作中。实践表明，对于中型和大型程序的开发工作，使用C++的效果比C语言好得多。C++正在从软件的可靠性、可重用性、可扩充性、可维护性等方面体现出它的优越性。

1.2 简单的C++程序

【例 1.1】一个最简单的 C++ 程序。

```
//Ex1_1.cpp
#include <iostream.h>
void main()
{
    cout << "This is a simple C++ program." << endl;
}
```

该程序经编译、连接、运行后，结果如图 1.2 所示。

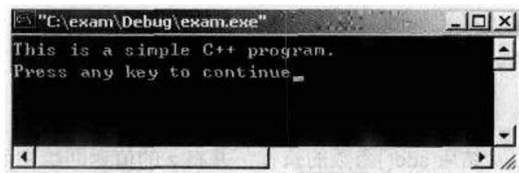


图 1.2 运行结果

说明：本程序虽然只有 6 行，但它却包含了每一个 C++ 程序都要具备的几个基本组成部分。现对它逐行进行解释。

第 1 行：// Ex1_1.cpp

这是一个注释行。注释是程序编写者为读者作的一种说明，有助于提高程序的可读性，编译时，注释将被忽略。本行注释中的内容为源程序文件的名称——Ex1_1.cpp，通常 C++ 源程序文件以 .cpp 为扩展名。

第 2 行：# include <iostream.h>

这是一条预处理命令。使用预处理命令可以更好地进行程序的组织。本行预处理命令 # include <iostream.h> 的作用是将文件 iostream.h 的内容加入到程序中。iostream.h 是一个 C++ 标准头文件，其中定义了一些输入输出流对象。

第 3 行：void main()

本行是主函数的声明。主函数是所有 C++ 程序开始执行的入口。无论主函数处于程序中的什么位置，其中的代码总是最先被执行。按照 C++ 语言的规定，每个程序都必须有且仅有一个主函数，主函数的名称必须为 main。main() 的一般形式为：

```

void main()           //函数头, void 表示 main()函数没有返回值
{
    语句组             } 函数体
}

```

main 前面的 void 表示主函数 main 没有返回值。void 代表空，它是 C++ 中的一个基本数据类型。

跟在 main 后面的圆括号 “()” 说明它是一个函数。在 C++ 中所有的函数名称后面都紧跟着一对圆括号，其中可以没有内容（即没有函数的参数，例如本行），也可以包含函数的参数。

第 4 行和第 6 行：在主函数 main 的声明之后用花括号 “{}” 括起来的是函数主体部分。

第 5 行：cout << “This is a simple C++ program” << endl;

本行是一条 C++ 语句，它完成了此程序的主要功能，即向屏幕上输出一行字符串。cout 是 C++ 中的标准输出流对象，它通常代表计算机的屏幕。cout 在标准头文件 iostream.h 中被声明，因此要使用它就先包含此文件（见程序的第 2 行）。“<<” 是输出操作符，功能是将其右边的内容输出到它左边的指定设备上。这里，要输出的内容是用双引号括起来的字符串 “This is a simple C++ program.”，endl 代表换行符。本行末尾的分号 “;” 表示这条语句的结束。在 C++ 中所有的语句都要以分号结束。

【例 1.2】求两个数之和。

```

//Ex1_2.cpp
#include "iostream.h"
int add(int x, int y)           //定义 add() 函数，求两个数之和
{
    int z;                     //说明 z 为 int 类型的变量
    z=x+y;                     //将 x+y 的值赋值给 z
    return z;                  //结束 add()函数的执行，并将 z 的值返回给主函数
}

void main()                     //定义主函数
{
    int a, b, c;                //说明 a、b、c 为 int 类型的变量
    cin>>a>>b;                 //从键盘输入两个数据送到 a 和 b 中
    c=add(a,b);                 //调用 add()函数，计算 a+b，并把结果赋给 c
    cout<<c<<endl;            //先输出 c 的值，再输出换行符
}

```

程序运行结果如图 1.3 所示。其中 5 和 7 是用户输入的数据，中间用空格分隔。

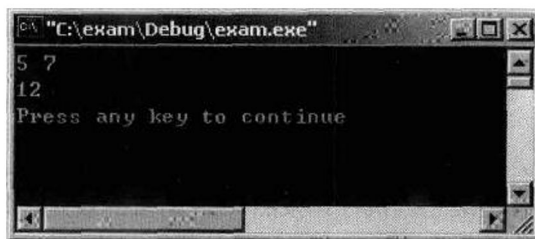


图 1.3 运行结果

说明:

- (1) 该程序由 `main()`和 `add()`两个函数组成。
- (2) 尽管 `add()`函数在 `main()`函数的前面, 但仍然是从 `main()`函数开始执行的。
- (3) 语句 `c=add(a,b);` 的作用是调用 `add()`函数计算 `a+b`。
- (4) `add()`函数是一个用户定义的函数, 其功能是求两个数的和。调用时, 先将实在参数 `a`、`b` 的值分别传递给对应的形式参数 `x`、`y`, 然后依顺序执行 `add()`函数中的各个语句, 遇到语句 `return z` 后将 `z` 的值作为函数的返回值返回给主函数并结束 `add()`函数的执行, 继续执行 `main()`函数中的语句, 变量 `c` 将获得调用 `add()`函数所返回的 `a+b` 的值。

从上面的两个例子可以看出, 一个 C++程序往往是由预处理命令、语句、函数、变量对象、输入和输出以及注释等几个基本部分组成, 具体组成及书写规则如下:

- (1) C++程序是由一个或多个函数组成的, 其中必须要有一个 `main()`函数。无论这个函数的位置在哪里, 程序总是从它开始执行。
- (2) 函数又包括函数首部与用“{”和“}”括起来的函数体两个部分。
- (3) 函数体中各语句结尾要用“;”, 且各语句是从上到下执行的。
- (4) 注释是从“//”开始到本行结束的内容, 作用是增加程序的可读性。C++中提供了“行”和“块”两种注释方法。行注释的内容从双斜杠“//”开始到本行末尾结束, 例如, `// Ex1_1.cpp` 就是行注释; 块注释的内容从符号“/*”开始到符号“*/”结束, 例如:

```
/* only one line */
```

和

```
/* this is the first line  
   this is the second line  
   this is the third line */
```

都是块注释。

- (5) C++程序书写形式自由。一行可以写多条语句, 一条语句也可以分写在不同行上。但 C++语言对大小写敏感, 即大小写字母是有区别的, 初学者要特别注意。

1.3 C++基本语法成分

C++作为一种程序设计语言有一个严格的字符集和一套严密的语法规则。程序中各种成分(如常量、变量、表达式、语句等)是根据语法规则由字符集中的字符构成的。程序中不能使用这个字符集以外的字符, 也不能违反语法规则。

1.3.1 C++字符集

字符集是组成语言的最小元素, 作为一种计算机程序设计语言, C++同汉语、英语一样也是由基本符号构成的。C++中的字符可以分为:

- (1) 字母: 大小写英文字母各 26 个: `a~z` 和 `A~Z`。
- (2) 数字: `0~9` 共 10 个数字。
- (3) 特殊符号: `+`、`-`、`*`、`/`、`=`、`,`、`.`、`_`、`:`、`;`、`?`、`\`、`“`、`’`、`~`、`|`、`!`、`#`、`%`、`&`、`(`、`)`、`[`、`]`、`{`、`}`、`^`、`<`、`>`和“空格”共 30 个符号。

要特别注意的是，在 C++ 中大小写字母是敏感的，即大小写字母是不等价的。

1.3.2 C++ 语言的词汇

字符集本身并没有什么含义，而由它们按照一定规则组合成的单词就能表达出某种语义。

1. 关键字

有特定含义的、专用的单词称为关键字，也称保留字，如 `int`、`char`、`break`、`for`、`define` 等。用户不能用关键字来命名变量名、符号常量名、函数名、类名等。

2. 标识符

标识符是以字母或下划线开始的，由字母、数字字符和下划线组成的字符串，用来标识变量名、符号常量名、函数名、类名等。如 `var`、`myBook`、`a3` 都是合法的标识符，而 `2A`、`A-B`、`$sys`、`M.D`、`int` 为不合法标识符。

说明：

(1) 标识符必须由字母或下划线开头，其他部分可由字母、下划线或数字组成，一般只有前 32 个字符有效。

(2) 标识符中的字母区分大小写。例如，`ABC` 与 `abc` 是两个不同的标识符。

(3) 标识符不能使用 C++ 的关键字，尽量不使用系统预定义标识符。例如，`int`、`break` 等关键字不能作为标识符。

(4) 定义标识符时最好按“见名知义”的原则，以提高程序的可读性。如 `min` 表示最小值，`average` 表示平均值，`day` 表示日期等。

(5) 尽量不要使用 `l(L)`、`I(i)`、`o(O)` 等易混淆字母作为标识符。

(6) 在 Visual C++ 中，标识符中不能有汉字，但是字符串和注释中可以有汉字。

3. 运算符

运算符是对程序中的数据进行操作的符号，C++ 语言中运算符的种类非常丰富，如 `+`、`-`、`*`、`/`、`++`、`<<` 等，有关具体内容及其使用方法将在第 2 章中做进一步介绍。

4. 标点符号

标点符号是在程序中起分割内容和界定范围作用的一类单词，有关 C++ 语言中的标点符号如表 1.2 所示。

表 1.2 C++ 语言中的标点符号

标点符号	描述
空格	语句中各成分间的分隔符
;	语句的结束符
'	字符常量的起止标记符
"	字符串常量的起止标记符
#	预处理命令的开始标记符
{	复合语句的开始标记符
}	复合语句的结束标记符
//	行注释的开始标记符
/*	块注释的开始标记符
*/	块注释的结束标记符

1.4 基本输入输出

数据的输入输出是一个应用程序不可缺少的部分。没有输出，运算的结果不知道，程序也就没有意义；有时一些程序还会要求用户输入一些信息。C++程序对数据的输入输出进行了扩充，引入了标准设备 `cin`（代表键盘）和 `cout`（代表显示器），把数据的输入输出处理为从一个对象到另一个对象的流动。所谓“流”是从数据的传输（流动）抽象而来的，可以把它理解为“特殊的文件”；从操作系统的角度来说，每一个与主机相连的输入、输出设备都可以看做一个文件（或输出流）。在输入操作中，信息从输入设备流向内存；在输出操作中，信息从内存流向输出设备。`cout` 是由 `c` 和 `out` 两个单词组成的，代表 C++ 的输出流，`cin` 是由 `c` 和 `in` 两个单词组成的，代表 C++ 的输入流。它们是在头文件 `iostream.h` 中定义的。要使用 `cin` 和 `cout`，需要在程序前面加上语句：

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

或

```
#include <iostream.h>
```

1. 文件包含

头文件：`.h`，如 `cstdio.h`、`iostream.h`，它们包含了大量的函数说明、常量定义。

作用：将头文件的内容插入到源文件（`.cpp`）中。

形式：（1）`#include <文件名>`

从系统指定的文件夹中找。

（2）`#include "文件名"`

从当前文件夹中找，再到系统指定的文件夹中找。

注意：一条文件包含命令只能包含一个文件。

说明：

（1）在 1998 年通过的 ISO C++ 标准中对 C++ 语言的标准库进行了部分修订，其中包括重新命名定义输入输出功能的相关头文件，命名方法是将原来头文件名中的扩展名 `.h` 去掉，例如将 `iostream.h` 改为 `iostream` 就形成了新的头文件名。此外，新标准中还引入了命名空间（`namespace`）的概念，其目的是为了避免一个程序不同模块中相同名称所引起的命名冲突，减少用户程序中的标识符与标准库中的标识符产生命名冲突。用户程序可以使用两种方法来引用 `std` 命名空间中的标识符。

方法 1：

使用关键字 `using` 将 `std` 名字空间中的标识符全部引入到全局命名空间（即用户程序中），具体用法是在包含标准头文件的 `include` 预处理命令后面添加语句行：

```
using namespace std;
```

这样，在用户程序中就可以像原来一样继续使用 `cin` 和 `cout` 进行输入输出操作了。

【例 1.3】

```
//Ex1_3.cpp
```

```
#include "iostream" //新标准中的头文件名
```

```
using namespace std; //引入 std 命名空间中的标识符
```

```

void main()
{
    int a;
    float b;
    cout<<"input a,b:";
    cin>>a>>b;
    cout<<a<<'+ '<<b<< '='<<a+b<<endl;
}

```

方法 2:

使用范围限定符::（即两个相邻的冒号）在用户程序中显式地指明属于 std 命名空间的标识符。具体用法是由前缀 std::来引导 std 命名空间中的标识符。例如，将 cin 改为 std::cin，将 cout 改为 std::cout。

【例 1.4】

```

//Ex1_4.cpp
#include "iostream"           //新标准中的头文件名
void main()
{
    int a;
    float b;
    std::cout<<"input a,b:"; //使用前缀 std::
    std::cin>>a>>b;
    std::cout<<"a+b="<<a+b<<std::endl; //endl 也是定义在 std 命名空间中的标识符
}

```

（2）使用早期的函数库。

由于 C++是从 C 语言发展而来，在 C 语言的发展过程中建立了功能丰富的函数库。因此，对于熟悉 C 语言的读者还可以在 C++程序中使用 C 语言的函数库。在 C++中使用这些头文件有两种方法：

1) 用 C 语言的传统方法。

头文件名包括后缀.h，如 stdio.h、math.h 等。由于 C 语言没有命名空间，头文件并不存放在命名空间中，因此在 C++程序文件中如果用到带后缀.h 的头文件时，不必用命名空间，只需在文件中包含所用的头文件即可，如#include <math.h>。

2) 用 C++的新方法。

C++标准要求系统提供的头文件不包括后缀.h，例如 iostream、string。为了表示与 C 语言的头文件有联系又有区别，C++所用的头文件名是在 C 语言的相应的头文件名（但不包括后缀.h）之前加一字母 c。例如，C 语言中有关输入与输出的头文件名为 stdio.h，在 C++中相应的头文件名为 cstdio。C 语言中的头文件 math.h，在 C++中相应的头文件名为 cmath，C 语言中的头文件 string.h 在 C++中相应的头文件名为 cstring。注意在 C++中，头文件 cstring 和头文件 string 不是同一个文件。前者提供 C 语言中对字符串处理的有关函数（如 strcmp、strcpy）的声明，后者提供 C++中对字符串处理的新功能。

此外，由于这些函数都是在命名空间 std 中声明的，因此在程序中要对命名空间 std 作声明。如：

```
#include <cstdio>
```

```
#include <cmath>
using namespace std;
```

鉴于目前所用的大多数 C++ 编译系统既保留了 C 的用法，又提供了 C++ 的新方法。下面两种用法等价，可以任选。

C 传统方法	C++ 新方法
#include <stdio.h>	#include <cstdio>
#include <math.h>	#include <cmath>
#include <string.h>	#include <cstring>
	using namespace std;

可以使用传统的 C 方法，但应当提倡使用 C++ 的新方法。

2. 标准输出设备 cout

使用格式：cout<<表达式 1<<表达式 2<<……<<表达式 n;

功能：将各表达式的值按顺序输出到显示器上，数据的输出格式由系统自动决定。

说明：

(1) <<称为输出运算符，也称为插入运算符。

(2) 各表达式可以是任意类型的。

(3) 可以在一个输出语句中使用多个运算符<<将多个输出项插入到输出流 cout 中，<<运算符的结合方向为自左向右，但要注意每输出一项要用一个<<符号，不能写成 cout<<a,b,c,"A"; 形式。

(4) 可以使用格式控制符控制数据的输出格式，但应使用 #include "iomanip.h" 文件包含命令。常用格式控制符如表 1.3 所示。

表 1.3 常用格式控制符

格式控制符	说明
endl	插入换行符（即'\n'）
ends	插入字符串结束符（即'\0'）
dec	十进制表示
hex	十六进制表示
oct	八进制表示
left	在设定的宽度内左对齐，设置的对齐方式一直有效，直到再次设置
right	在设定的宽度内右对齐
fixed	按定点格式显示浮点数，如 123.4 就输出为 123.4
setw(int n)	设置输入输出宽度，只对一次输入或输出有效
setfill(int n)	设置填充字符，设置的填充字符一直有效，直到再次设置
setprecision(int n)	设置浮点数输出的小数位，默认为 6，设置的精度值一直有效，直到再次设置

【例 1.5】cout 应用示例。

```
//Ex1_5.cpp
#include "iostream"
using namespace std;
```