普通高等教育电子信息类“十三五”课改规划教材

C/C++程序设计

主　编　朱智林
副主编　原燕东　高　文

西安电子科技大学出版社
http://www.xduph.com

普通高等教育电子信息类“十三五”课改规划教材

C/C++程序设计

主　编　朱智林

副主编　原燕东　高　文

参　编　王永玉　马加庆　杨　莉　杨福刚　张树粹

西安电子科技大学出版社

内 容 简 介

本书主要讲述 C/C++ 程序设计的基本原理和基本思想方法，在 C 语言的基础上扩充了 C++ 的运算符重载、函数重载、类和对象的封装性等内容，目的是使读者具备面向对象程序设计的能力。全书共 9 章，包括概述、基本数据类型及运算符、程序控制结构、数组、函数、指针、构造数据类型、文件、编译预处理等内容。各章均精选了各类计算机水平考试试题作为例题和习题。

本书适合作为普通高等院校、高职高专、各类成人教育院校程序设计基础课程的教材，也可作为编程人员和参加计算机考试(C/C++ 模块)人员的参考书。

图书在版编目(CIP)数据

C/C++ 程序设计 / 朱智林主编. —西安：西安电子科技大学出版社，2019.3(2019.5 重印)
ISBN 978-7-5606-5281-8

Ⅰ. ① C… Ⅱ. ① 朱… Ⅲ. ① C 语言—程序设计—高等学校—教材 Ⅳ. ① TP312.8

中国版本图书馆 CIP 数据核字(2019)第 049288 号

策划编辑　万晶晶
责任编辑　王　静
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西日报社
版　　次　2019 年 3 月第 1 版　2019 年 5 月第 2 次印刷
开　　本　787 毫米×1092 毫米　1/16　印　张　18.5
字　　数　438 千字
印　　数　501～2500 册
定　　价　42.00 元
ISBN 978-7-5606-5281-8 / TP

XDUP 5583001-2

前　言

C/C++ 语言是国际上广泛应用的计算机程序设计语言。它以功能强大、表达灵活、代码效率高和可移植性好而著称，广泛用于编写各种系统软件和应用软件。

本书是作者根据多年程序设计语言的教学经验编写而成的，力求以通俗、简练的语言叙述 C/C++ 程序设计中的概念、语法和设计方法。本书选用了 VC++ 6.0 编译系统，以使 C/C++ 程序的调试更加直观、方便。

本书主要讲述 C/C++ 程序设计的基本原理和基本思想方法，在 C 语言的基础上扩充了 C++的相关内容，使 C 语言和 C++ 有机结合在一起，以便读者在编写面向过程程序的同时，能够运用 C++的新增功能简化程序、提高效率。

本书主要特点概括如下：

(1) 定位准确，取舍合理。本书是针对高等教育本科及高职高专学校计算机及其相关专业、非计算机专业等的程序设计基础课程编写的。根据不同层次的教学要求，本书内容可灵活取舍，而不失其教材内容的科学性与系统性。

(2) 精选例题，通俗易懂。为使 C/C++ 程序设计的基本概念、基本理论叙述更加通俗易懂，本书精心选编了采用 Visual C++ 6.0 编译系统调试成功的示例。

(3) 合理设计，综合实例。程序设计是一门实践性很强的课程，不仅要讲授程序设计的基本概念和基本理论，而且更要着力培养学生的设计和编程能力。为此，每一章后面都选编了与其教学内容紧密相关的实验题目，方便了教与学。本书结合数组、函数和自定义类型等章节内容，设计了一个综合实例，以利于循序渐进地培养学生的综合应用能力。

(4) 循序渐进，为面向对象程序设计打下基础。本书以面向过程程序设计为主，介绍了 C++对 C 的改进，引进了 C++ 的运算符、函数重载，同时，对类和对象的封装性进行了叙述，为面向对象编程打下基础。

(5) 本书配备了学习指导书(由西安电子科技大学出版社同时出版)，书中精心设计了各知识点的实验题目，对初学者编程和程序调试能力的提高会有极大帮助。学习指导中对本书的许多习题提供了习题解析，同时又为有提高需求的学生设计了“补充提高习题”。

(6) 本书配有供教师教学使用的电子教案、例题源代码和习题参考答案等教学资源。

书中标有“*”号的内容可根据教学要求进行取舍。根据作者学校使用情况，建议教学计划学时数为 78 学时(计算机专业)，其中讲授 48 学时，实验 30 学时。

本书编写分工如下：第 1～3 章由朱智林编写，第 4 章由原燕东编写，第 5 章由高文编写，第 6 章由杨莉和王永玉编写，第 7 章由马加庆编写，第 8 章由杨福刚编写，第 9 章由

张树粹编写。全书由朱智林统稿。

在本书的编写过程中，编者参考了大量有关 C/C++ 程序设计的书籍和资料，在此对这些参考文献的作者表示最诚挚的谢意!

由于水平有限，书中疏漏之处在所难免，请各位读者不吝指正。

编　者

2018 年 12 月

目　录

第1章 概 述

本章主要介绍程序与程序设计的基本概念、算法与程序基本结构、C 语言的基本词法和基本语句、C 语言标准输入/输出函数，以及使用 Visual C++ 6.0 调试程序的方法和步骤。

1.1 程序设计与高级语言

1.1.1 程序与程序设计

程序是使用计算机语言解决问题的方法和步骤的描述。计算机程序设计是指在某一程序语言环境下，编写出能够使计算机理解并执行的程序代码。程序的特点是有始有终，每个步骤都能操作，所有步骤执行完后则对应问题得到了解决。

例如：求两个整数和的方法和步骤如下：

第一步，获取两个整数 a 和 b；

第二步，计算 c = a + b；

第三步，输出 c；

第四步，结束。

如果将上述问题转化成用计算机语言编写的程序，则如例 1.1 所示。

【例 1.1】 求两数和。

程序清单如下：

```
#include <iostream>
using namespace std;
void main()                          //声明主函数
{                                    //主函数开始
  int a, b, c;                       //定义 a、b、c 为整数
  cin>>a>>b;                         //给整型变量 a 和 b 赋值
  c=a+b;                             //计算 c = a + b
  cout<<"sum="<<c<<endl;             //输出计算结果 c
}                                    //程序结束
```

该程序包括了三个部分，一是对变量 a，b，c 进行数据类型说明，二是确定了求两数和的计算方法，三是将计算结果打印输出。

从上述例子可以看到一个 C++ 程序的基本形式：

```
void main()
{
    //一组程序语句
}
```

其中，main 是主函数名，一对大括号是函数体，包括一组程序语句，每条语句以分号结束，程序中类似“//输出计算结果 c”的字符串是注释语句。注释语句不是程序可执行语句，它的作用是增强程序的可读性。在 C 语言中也可使用形如“/*……*/”的形式进行注释。

1.1.2 程序设计语言

目前，通用的计算机还不能识别自然语言，只能识别特定的计算机语言。

计算机语言一般分为低级语言和高级语言。低级语言直接依赖于计算机硬件，不同的机型所使用的低级语言是完全不一样的。高级语言则不依赖计算机硬件，用高级语言编写的程序可以方便地、几乎不加修改地运行于不同类型的计算机上。

需要强调的是，无论采用何种计算机语言来编写程序，程序在计算机上的执行都是由 CPU 所提供的机器指令来完成的。机器指令是用二进制表示的指令集。每种类型的 CPU 都有与之对应的指令集。

1. 低级语言

低级语言包括机器语言和汇编语言。

直接使用二进制表示的指令来编写程序的语言就是机器语言。使用机器语言编写程序时，必须准确无误地牢记每一条指令的二进制编码。

机器语言的优点是执行速度快，并且可以直接对硬件进行操作，例如主板上的一些设备驱动程序等都是由机器语言编写的。机器语言所编写的程序不易读懂(例如编码“1011 1000 1110 1000 0000 0011”根本看不出表示什么命令)，也就难以维护。由于依赖于计算机硬件指令集，而不同类型计算机的指令集之间不兼容，因而用机器语言所编写的程序的可移植性差，另外，用机器语言编写程序效率低下，且不能保证程序有好的质量。

为了方便地编写程序，用一些符号和简单的语法来表示机器指令，这就是汇编语言。例如编码“1011 1000 1110 1000 0000 0011”用汇编语言表示就是“mov AX, 1000”，其功能就是“将 1000 送入 AX 寄存器中”。但是 CPU 不能识别汇编语言，因此需要一个“翻译”将汇编语言翻译成机器语言，这个“翻译”我们称之为“汇编器”。汇编语言和机器语言的指令是一一对应的，其可读性有所提高，但并未改变机器语言的特点，也就是说，汇编语言是面向机器语言的。当然，汇编语言也仍然具备机器语言的优点。许多大型系统的核心部分都是用汇编语言编写的，它们直接和硬件打交道，效率需求高。

有没有办法真正提高程序的可读性、可维护性和可移植性呢？

2. 高级语言

高级语言是一种比较接近自然语言和数学语言的程序设计语言。高级语言的出现大大提高了程序员的工作效率，降低了程序设计的难度，并改善了程序质量。用高级语言编写程序可使程序具备良好的可读性和可维护性，更易于人们掌握程序设计方法，从而使计算机技术得到迅速的应用和普及。

例如，语句段：

```
if (a>b)
    c=a;
else
    c=b;
```

表示的是“如果 a 大于 b，则 c=a，否则 c=b”。这与自然语言(英语)非常接近，容易理解。当然，这里的“=”和数学语言中的等号是有本质区别的，这在后面内容中会详细介绍。

每一种高级语言，都有自己人为规定的专用符号、英文单词、语法规则和语句结构。高级语言与自然语言(英语)更接近，而与硬件功能相分离。高级语言的通用性强、兼容性好、便于移植。高级语言的发展经历了从无序化的程序设计到面向过程的程序设计、面向对象的程序设计的发展过程。面向过程的语言的特点是，程序员要告诉计算机“做什么”和“怎么做”。FORTRAN、PASCAL 和 C 语言等都是面向过程的高级语言。面向对象的语言的特点是面向具体的应用功能，即面向“对象”，其方法就是软件的集成化，将数据和处理数据的过程作为一个整体来处理，采用类的封装、数据隐藏、继承性和多态性等方法，便于将实际问题分解和抽象，使代码易于维护并保持较高的复用性。典型的面向对象的高级语言如 C++、Java 等。

使用计算机语言编写的程序叫源程序。因为计算机只能接收 0 和 1 组成的二进制程序(又称二进制机器指令)，所以高级语言源程序必须通过编译系统将其翻译成二进制程序后才能执行。翻译程序有两种执行方式：一种是通过“解释程序”将源程序翻译一句执行一句，这种执行方式称为“解释执行”方式；另一种是通过“编译程序” 将源程序全部翻译成二进制程序后再执行，此种执行方式称为“编译执行”方式。大多数高级语言采用“编译执行”方式，C 语言就是其中之一。从源程序到在计算机上得到运行结果，其操作过程(以 C 语言为例)如图 1.1 所示。

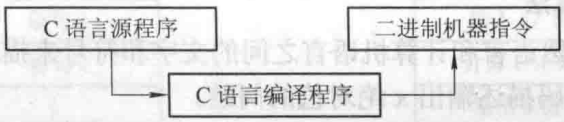

图 1.1 C 语言程序编译过程

1.2 算 法

学习计算机程序设计语言的目的，是用语言作为工具，设计出计算机能够运行的程序。设计一个程序首先需要做两方面的工作：一是组织合理结构的数据，二是设计解决问题的算法。这样，就可用任何一种计算机语言编写程序，也就是常说的计算机源程序。因此，可以用以下公式来表示程序：

程序=算法+数据结构

1.2.1 算法的特性

算法是指为了解决某个特定问题而采用的确定且有效的步骤。计算机算法可分为两大

类：数值运算和非数值运算。数值运算的目的是求解，例如，求方程的根、求圆的面积、求 n 的阶乘等，都属于数值运算。非数值运算包括的面十分广泛，主要用于事务管理，例如人事管理、图书管理、学籍管理等。一个算法应当有以下 5 个特性。

1．有穷性

一个算法应当包含有限个操作步骤。也就是说，在执行若干个操作之后，算法将结束，而且每一步都在合理的时间内完成。

2．确定性

算法中的每一条指令必须有确切的含义，不能有二义性，对于相同的输入必须能得出相同的结果。

3．可行性

算法中的每一步都应当有效执行，通过基本运算后能够实现目标。

4．有零个或多个输入

在计算机上实现算法，所需的处理数据大多数情况下要在程序执行时通过输入得到，但也有些程序不需要输入数据。

5．有一个或多个输出

算法的目的是求解(得到结果)，结果要通过输出得到。

1.2.2　算法表示

算法可以用各种描述方法来进行描述，常用的有自然语言、伪代码、传统流程图和 N-S 流程图。使用流程图(又称框图)等将算法描述出来，然后根据流程图编写程序代码是计算机程序设计经常采用的方法。

1．用伪代码表示算法

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法的。

【例 1.2】 用伪代码描述输出 x 绝对值的问题。

程序清单如下：

```
IF x is positive THEN
   print   x
ELSE
   print -x
```

也可以中英混用，将上例改写成：

```
若 x 为正，则
   打印 x
否则
   打印 -x
```

2．用传统流程图表示算法

传统流程图也是描述算法的很好的工具，传统流程图中使用的符号如图 1.2 所示。

图 1.2 传统流程图符号

其中，流程线包括 4 个方向线，即：→、←、↓、↑。

【例 1.3】 求两数之和的算法的流程图，如图 1.3 所示。

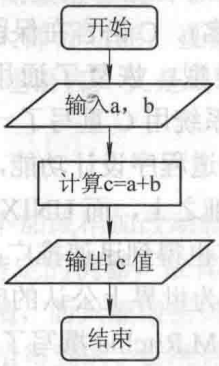

图 1.3 求两数和的传统流程图

3. 用 N-S 流程图表示算法

随着结构化程序设计方法的出现，1973 年美国学者 I. Nassi 和 B. Shneiderman 提出了一种新的流程图形式。这种流程图去掉了流程线，算法的每一步都用一个矩形框来描述，一个完整的算法就是按用户设计的执行顺序连接起来的一个大矩形。人们把这种流程图称为 N-S 流程图，如图 1.4 所示。

例如，求两数之和的算法的 N-S 流程图如图 1.5 所示。

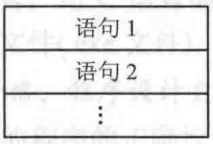

图 1.4 N-S 流程图

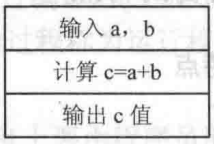

图 1.5 求两数和的 N-S 流程图

1.3 C/C++ 的发展史与特点

1.3.1 C/C++ 的发展史

C 语言是目前国际上广泛流行的一种结构化的程序设计语言，计算机专业人员使用它来开发系统软件，软件开发人员使用它来编写应用软件。特别是近些年来，不仅是计算机专业人员和软件开发人员，广大计算机爱好者也越来越青睐 C 语言。

C 语言的前身是 ALGOL 语言。ALGOL 语言是 1960 年开发出的一种面向问题的高级语言，用 ALGOL 来描述算法很方便，但它的缺点是不具有与硬件打交道的底层处理能力，

不宜用来编写系统程序。1963 年英国剑桥大学在 ALGOL 语言基础上增添了处理硬件的能力，推出了 CPL(Combined Programming Language)语言。CPL 语言比 ALGOL 语言接近硬件一些，但由于规模较大，学习和掌握困难，没有流行开来。1967 年英国剑桥大学的 Dennis M.Richards 对 CPL 语言进行了简化，推出了 BCPL 语言。1970 年美国贝尔实验室(Bell Laboratories)的 Ken Thompson 对 BCPL 语言又进行了进一步简化，设计出了既简单又接近硬件系统的 B 语言(以 BCPL 的第一个字母命名)，同时，使用 B 语言编写出 UNIX 操作系统，在 PDP-7 上实现。1972 年美国贝尔实验室的 Dennis.M.Ritchie 在 B 语言的基础上设计出 C 语言(以 BCPL 的第二个字母命名)。C 语言在保留 BCPL 语言和 B 语言的强大的硬件处理功能的基础上，扩充了数据类型，恢复了通用性。1973 年 Dennis M.Ritchie 和 K.Thompson 两人合作将 UNIX 操作系统用 C 重写了一遍(即 UNIX V5)。系统的代码量比以前的版本增加了三分之一，加进了多道程序设计功能，特别是整个 UNIX 操作系统(包括 C 编译程序本身)都建立在 C 语言的基础之上，而 UNIX V5 奠定了 UNIX 操作系统的基础。随着 UNIX 的使用日益广泛，C 语言也得到迅速推广，可以说，C 语言和 UNIX 在发展过程中相辅相成，目前，C 语言早已成为世界上公认的应用最广泛的计算机语言。

1977 年，K.Thompson 和 Dennis M.Ritchie 撰写了《C 程序设计语言》一书，对 C 语言进行了规范化的描述，成为当时的标准，称为 K&R 标准。随着微型机的普及，出现了不同的 C 语言标准版本，为了统一标准，美国标准化协会(ANSI)于 1983 年制定了一套 ANSI C 标准，此后又相继推出 87 ANSI、C99 标准。

C++ 是由 C 发展而来的，与 C 兼容，C++ 是 C 的超集。C++ 语言既可用于面向过程的结构化程序设计，又可用于面向对象的程序设计，是一种功能强大的程序设计语言。C++ 保留了 C 的风格和特点，同时对 C 某些不足做了大量的改进，并增加了面向对象的机制。改进后的 C++ 与 C 相比，在数据类型方面更加严格，使用更加方便了。

1.3.2 C/C++ 语言的特点

1. C 语言的特点

1) 结构化语言

C 语言是结构化程序设计语言，面向过程编程。每一类语言都有它的特点。结构化语言的一个显著特点是代码和数据的分离化，即程序的各部分除了必要的信息交流外，彼此互不影响，相互隔离。体现 C 语言主要特点的是函数。

C 语言的程序是由函数构成的，一个函数为一个“程序模块”。一个 C 源程序至少包含一个函数，就是 main()函数(主函数)，也可以包含一个 main()函数和若干个其他函数(子函数)。所以说，函数是 C 程序的基本单位。同时，C 语言系统也提供了丰富的库函数(又称系统函数)，用户可以在程序中直接引用相应的库函数，根据需要编制和设计用户自己的函数。所以说，一个 C 程序由用户自己设计的函数(以下简称用户函数)和库函数两部分构成。

2) 简洁、紧凑、灵活

C 语言中只有 32 个保留字(关键字)，9 种控制语句，程序书写自由，主要是小写字母，C 语言编译程序的体积很小。另外 C 语言是一种自由格式的语言，没有像 FORTRAN 语言那样的书写格式的限制，故用 C 语言编写程序自由方便。

3) 运算符丰富

C 语言的运算符种类很多，共有 15 类运算符(见附录 B)。C 语言可以进行字符、数字、地址、位等多种运算，并可完成由硬件实现的普通算术运算、逻辑运算。灵活使用各种运算符可以完成许多在其他高级语言中难以实现的运算或操作。

4) 中级语言

我们通常称面向问题的语言为“高级语言”，而面向机器的语言为“低级语言”，C 语言既具有高级语言的功能，又具有低级语言的许多功能。C 语言能够对内存单元中的二进制位(bit)操作，实现汇编语言的大部分功能，直接对硬件进行操作。由于 C 语言的这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言，所以称它为中级语言。

5) 可移植性好

可移植性是指程序从一个环境下不加或稍加改动就可移到另一个完全不同的环境下运行。对汇编语言而言，由于它只面向特定的机器，故其根本不可移植。而一些高级语言(比如 FORTRAN)，其编译程序也不可移植，而只能根据国际标准重新实现。但 C 语言在许多机器上的实现是通过将 C 编译程序移植得到的。据统计，不同机器上的 C 编译程序 80%的代码是共同的。

6) 功能强大

高级语言不适用于编写系统软件，除了语言表达能力之外还有一个很大的因素是该语言的代码质量。如果代码质量低，则系统开销就会增大。一般来说，语言越低级其代码质量就越高。由于 C 语言具有低级语言的功能，所以现在许多系统软件都用 C 语言来描述，从而大大提高了编程效率。

7) 编译语言

C 语言是编译语言，用 C 语言编写的源程序必须经过编译后(生成 .obj 文件)，再与库文件连接生成可执行文件(.exe 文件)，执行可执行文件的过程称为运行程序。

8) 语法限制不严格，程序设计自由度大

用 C 语言所编写的程序的正确性和合法性在很大程度上要由程序员而不是 C 语言编译程序来保证。如 C 语言编译程序对数组下标不做越界检查、数据类型检验功能较弱等。故对 C 语言不熟悉的人员，编写一个正确的 C 语言程序可能会比编写其他高级语言程序难一些。所以用 C 语言编写程序，要对程序进行认真的检查，而不要过分依赖 C 语言编译程序的查错功能。正是因为 C 语言放宽了语法的限制，所以换来了程序设计的较大自由度和灵活性。

2. C++ 的特点

C++ 程序中出现了类和对象，因此 C++ 语言与 C 语言的本质区别是增加了面向对象的内容，如支持数据封装，支持基类、派生类的继承性，支持重载、多态性等。

下面简单介绍几点 C++ 的改进内容：

(1) C++ 规定函数说明必须使用原型说明，不可以简单说明。

(2) C++ 规定凡是从高类型向低类型的转换都要进行强制转换。

(3) C++ 中符号常量建议使用 const 关键字来定义(常变量)。

(4) C++ 中引进了内联函数，可以取代 C 中的带参的宏。

(5) C++ 允许设置函数参数的默认值，提高程序运行的效率。

(6) C++ 引进函数重载和运算符重载，编程更加方便。

(7) C++ 可以使用变量的引用进行数据传递。

(8) C++ 提供了 I/O 流类库，使输入/输出更加方便快捷。

本书将采用 VC++ 6.0 为编译程序讲解 C/C++ 的语法和编程方法，以结构化程序设计方法为主，简单介绍 C++ 相应方面的内容。

1.4　C 程序结构及书写规则

1.4.1　C 程序的基本结构

C 程序是由一个主函数和若干个(或 0 个)用户函数组成的，主函数和用户函数的位置是任意的。但它们的调用关系是一定的。即主函数可以调用任何用户函数，用户函数间可以互相调用，但不能调用主函数。用户函数甚至可以调用自己，这种调用称为递归调用。

C 程序总是从 main()函数开始执行，而不论 main()函数在整个程序中的位置如何。从主函数的第一条语句开始执行，直到主函数的最后一条语句结束。非主函数必须通过“函数调用”才能执行。

一个函数由两部分组成，即函数说明部分和函数体部分。

1. 函数说明部分(又称函数头)

函数说明部分包括函数值类型、函数存储类型、函数标识符、函数的参数和参数的类型；其格式是函数名后紧接着一对圆括号(详细说明参见第 5 章)。

2. 函数体部分

函数体部分即用大括号括起来的部分。函数体内有若干条语句，它们能完成各种操作，具体函数的功能都写在函数体中。C 语言允许函数体内为空。主函数 main()的常见表示形式有：

(1)
```
void main()
{
    ¦       //函数代码写在这儿
}
```

这种格式，主函数类型为 void，函数没有返回值。

(2)
```
int main()
{
    ¦       //函数代码写在这儿
  return 0;
}
```

主函数类型为 int，执行完毕向系统返回数值 0。C 语言标准推荐使用这种格式。

1.4.2 程序的书写规则

C 程序书写格式随意，除了保留字外，任何地方都可以插入空格、回车换行符。

为了便于阅读程序，建议采用格式化的书写格式，采用缩进纵向对齐方式。可以在程序的任何一处插入“注释”。注释语句是非执行语句，不参加编译，也不会出现在目标文件中，只起到帮助阅读程序的作用，如同读文章加上注释一样。

1.5 C 语言的基本词法

学习自然语言要学习词汇和语法规则，根据语法规则使用词汇书写句子或文章。C 语言是一种计算机语言，用计算机语言编写程序就要使用其基本字符、基本词类，根据它的语法规则，按照算法描述出解决问题的方法和步骤。

1.5.1 C 语言使用的字符集

C 语言程序允许出现的所有基本字符的组合称为 C 语言的字符集，C 语言的字符集就是 ASCII 字符集，主要分为下列几类：

(1) 大小写英文字母。A，B，C，…，Z，a，b，c，…，z。

(2) 数字。0，1，2，3，4，5，6，7，8，9。

(3) 键盘符号如表 1.1 所示。

表 1.1 键 盘 符 号

符号	含 义	符号	含 义	符号	含 义
~	波浪号	)	右圆括号	:	冒号
`	重音号	_	下划线	;	分号
!	叹号	-	减号	“	双引号
@	a 圈号	+	加号	‘	单引号
#	井号	=	等号	<	小于号
$	美元号	|	或符号	>	大于号
%	百分号	\	反斜杠	,	逗号
^	异或号	{	左花括号	.	小数点
&	与符号	}	右花括号	?	问号
*	星号	[	左方括号	/	(正)斜杠
(	左圆括号	]	右方括号		空格符号

注意：有些运算符是由两个字符共同构成的。如 &&，||，<=，>=，==，<<，>>，!=，++，-- 等，在 C 程序中应将它们看成一个整体，而不要当成两个字符来对待(见附录 B)。