

全国高等教育自学考试指定教材

计算机及应用专业(专科)

高级语言程序设计

(附高级语言程序设计自学考试大纲)

全国高等教育自学考试指导委员会组编

迟成文 主 编

汪小琼 副主编

经济科学出版社

图书在版编目 (CIP) 数据

高级语言程序设计 / 迟成文主编 . - 北京 : 经济科学出版社 , 2000.3

全国高等教育自学考试指定教材 . 计算机及应用专业

ISBN 7-5058-2016-8

I . 高… II . 迟… III . C 语言 - 程序设计 - 高等教育 - 自学考试 - 教材 IV . TP312

中国版本图书馆 CIP 数据核字 (1999) 第 54588 号

高级语言程序设计

(附高级语言程序设计自学考试大纲)

全国高等教育自学考试指导委员会组编

迟成文 主编

经济科学出版社出版

社址 : 北京海淀区万泉河路 66 号 邮编 : 100086

网址 : www.esp.com.cn

电子邮件 : esp@public2.east.net.cn

涿州市星河印刷厂印刷

787×1092 16 开 15.25 印张 330000 字

2000 年 3 月第一版 2000 年 3 月第一次印刷

印数 : 001-10 100 册

ISBN 7-5058-2016-8/G·428 定价 : 20.40 元

(图书出现印装问题 , 请与当地教材供应部门调换)

(版权所有 翻印必究)

内 容 提 要

本书是受国家教育部考试中心委托编写的全国计算机及应用专业（专科）自学考试指定教材，包括自学考试大纲和教材的所有内容。

本书全面介绍了高级语言 C 的基本语法和程序设计方法。内容循序渐进，语言通俗易懂，便于自学；例题丰富，习题数量和难易程度适中，便于练习和自我考查。

本书既是自学考试指定教材，也可作为各类院校计算机及应用专业学生的教材或参考书，对从事程序设计的技术人员也具有很好的参考价值。

组编前言

当您开始阅读本书时，人类已经迈入了二十一世纪。

这是一个变幻难测的世纪，这是一个催人奋进的时代。科学技术飞速发展，知识更替日新月异。希望、困惑、机遇、挑战，随时随地都有可能出现在每一个社会成员的生活之中。抓住机遇，寻求发展，迎接挑战，适应变化的制胜法宝就是学习——依靠自己学习、终生学习。

作为我国高等教育组成部分的自学考试，其职责就是在高等教育这个水平上倡导自学、鼓励自学、帮助自学、推动自学，为每一个自学者铺就成才之路。组织编写供读者学习的教材就是履行这个职责的重要环节。毫无疑问，这种教材应当适合自学，应当有利于学习者掌握、了解新知识、新信息，有利于学习者增强创新意识、培养实践能力、形成自学能力，也有利于学习者学以致用、解决实际工作中所遇到的问题。具有如此特点的书，我们虽然沿用了“教材”这个概念，但它与那种仅供教师讲、学生听，教师不讲、学生不懂，以“教”为中心的教科书相比，已经在内容安排形式体例、行文风格等方面都大不相同了。希望读者对此有所了解，以便从一开始就树立起依靠自己学习的坚定信念，不断探索适合自己的学习方法，充分利用已有的知识基础和实际工作经验，最大限度地发挥自己的潜能，以达到学习的目标。

欢迎读者提出意见和建议。

祝每一位读者自学成功。

全国高等教育自学考试指导委员会

1999.5

编者的话

C语言是目前广泛用于软件开发的一种编译型程序设计语言。和其它高级语言相比，C语言具有丰富的数据类型和运算符，语句极其简单，源程序清单简洁清晰；具有汇编语言的能力，可以直接处理硬件系统和外围设备接口的控制；是一种结构化程序设计语言，支持自顶向下的结构化程序设计技术；具有完善的模块化结构，为中大型软件设计中采用模块化程序设计方法提供了基础。

本书以目前微型机上流行的ANSI C为版本，兼顾集成化环境Turbo C编译程序，系统地、循序渐进地介绍了C语言的数据类型和运算符、语句格式和功能、结构化程序设计思想和方法。全书共分为九章。第一章介绍C语言的基本字符集、基本词类、基本句类和基本的程序结构。第二章介绍基本数据类型（整型、实型、字符型、字符串型）及运算符和表达式。第三章介绍三种基本程序结构（顺序、选择、循环）的相关的语句和程序设计方法，以及Turbo C编译程序的基本使用方法。第四章介绍数组的使用及程序设计方法。第五章介绍指针的使用及程序设计方法。第六章介绍函数的定义、调用及程序设计方法。第七章介绍结构型、共用型、枚举型数据的特点、定义和使用及程序设计方法。第八章介绍文件处理及程序设计方法。第九章介绍宏、包含文件、条件编译等编译预处理方法和带参数的主函数程序设计方法。

本书是全国计算机及应用专业（专科）的自学考试指定教材，在本书后面配有本课程的自学考试大纲供读者参考。考虑到本书是以自学为主的教材，所以在撰写时，力求取材得当、通俗易懂、结构清晰、层次分明，通过精选的大量例题验证语法和说明程序设计方法，章末配有大量与自学考试题型相同的习题，作为自我测试之用。

由于篇幅所限，本书的章末习题解答将收录在配套的“高级语言程序设计（C语言）自学考试辅导用书”中，该辅导书将对考试大纲中的重点和难点进行分析和总结，并对学习方法提供指导意见，对典型题目进行分析和解答，并配有少量模拟试卷供参加自学考试的读者参考。

本书的第一、三、四、九章由汪小琼同志编写，第二、五、六、七、八章由迟成文同志编写，各章例题和习题由迟琳同志和夏毅飞同志在计算机上调试，全书由迟成文同志统稿。

在本书的编写过程中，参考了大量有关C语言程序设计的书籍和资料，编者在此对这些参考文献的作者表示感谢。中国科学技术大学高性能计算中心主任陈国良教授、安徽大学程锦松教授、合肥工业大学张佑生教授仔细审读了“自学考试大纲”和本书的原稿，提出了宝贵的修改意见，在此表示衷心的感谢。

由于编者水平有限，书中难免存在错误和不当之处，敬请广大读者不吝赐教，以便再版时修改。

迟成文

1999.8

目 录

高级语言程序设计

第 1 章 概述	(1)
1.1 程序与程序设计	(1)
1.2 高级语言与 C 语言	(2)
1.3 C 语言的基本词法	(4)
1.3.1 字符集	(4)
1.3.2 保留字	(5)
1.3.3 标识符	(5)
1.3.4 C 语言的词类	(5)
1.4 C 语言的基本语句	(6)
1.5 C 程序的基本结构	(7)
习题	(9)
第 2 章 基本数据类型和运算	(10)
2.1 C 语言的数据类型	(10)
2.2 常量	(11)
2.2.1 整型常量	(11)
2.2.2 实型常量	(12)
2.2.3 字符常量	(12)
2.2.4 字符串常量	(13)
2.2.5 符号常量	(13)
2.3 变量	(14)
2.3.1 变量的数据类型及其定义	(14)
2.3.2 变量的存储类型及其定义	(15)
2.3.3 变量的初始化	(16)
2.3.4 变量的定义语句	(17)
2.3.5 变量的生存期和作用域	(17)
2.4 运算符	(18)
2.4.1 算术运算符	(20)
2.4.2 关系运算符	(22)

2.4.3	逻辑运算符	(23)
2.4.4	赋值运算符	(24)
2.4.5	逗号运算符	(25)
2.4.6	条件运算符	(26)
2.4.7	长度运算符	(27)
2.4.8	位运算符	(27)
2.5	表达式	(30)
2.5.1	算术表达式	(30)
2.5.2	关系表达式	(31)
2.5.3	逻辑表达式	(31)
2.5.4	赋值表达式	(32)
2.5.5	逗号表达式	(32)
2.5.6	条件表达式	(33)
2.5.7	变量赋值及表达式计算时的数据类型转换	(33)
	习题	(34)
第3章	顺序结构 选择结构和循环结构的程序设计	(37)
3.1	结构化程序的三种基本结构	(37)
3.1.1	顺序结构	(37)
3.1.2	选择结构	(37)
3.1.3	循环结构	(38)
3.2	顺序结构的程序设计	(39)
3.2.1	赋值语句	(39)
3.2.2	字符输入/输出函数	(39)
3.2.3	格式输入/输出函数	(40)
3.2.4	顺序结构程序设计例	(44)
3.3	C 程序的开发环境	(45)
3.3.1	Turbo C 的启动	(45)
3.3.2	Turbo C 的主菜单	(45)
3.3.3	在 Turbo C 环境下开发 C 程序的过程	(46)
3.4	选择结构的程序设计	(48)
3.4.1	单分支选择语句	(48)
3.4.2	双分支选择语句	(49)
3.4.3	多分支选择语句	(51)
3.4.4	选择结构程序设计例	(54)
3.5	循环结构的程序设计	(55)
3.5.1	while 语句	(55)
3.5.2	do-while 语句	(57)
3.5.3	for 语句	(58)
3.5.4	break 语句和 continue 语句	(59)
3.5.5	多重循环结构的实现	(62)
3.5.6	循环结构程序设计例	(63)
3.6	goto 语句	(65)
3.6.1	goto 语句	(65)

3.6.2 用 goto 语句组成循环	(66)
习题	(66)
第 4 章 数组	(70)
4.1 一维数组	(70)
4.1.1 一维数组的定义	(70)
4.1.2 一维数组的初始化	(71)
4.1.3 一维数组元素的引用	(72)
4.1.4 一维数组程序设计例	(72)
4.2 多维数组	(75)
4.2.1 多维数组的定义	(75)
4.2.2 多维数组的初始化	(76)
4.2.3 多维数组元素的引用	(77)
4.2.4 二维数组程序设计例	(77)
4.3 字符数组与字符串	(78)
4.3.1 字符数组	(78)
4.3.2 字符串与字符数组	(80)
4.3.3 字符串处理函数	(82)
4.3.4 字符数组程序设计例	(85)
习题	(86)
第 5 章 指针	(90)
5.1 指针和指针变量	(90)
5.1.1 指针	(90)
5.1.2 指针变量	(92)
5.2 指针变量的定义、初始化和引用	(93)
5.2.1 指针变量的定义和初始化	(93)
5.2.2 指针变量的引用方式	(94)
5.2.3 取地址运算符与指针运算符	(94)
5.3 指针变量的使用	(95)
5.3.1 指向变量的指针变量的使用	(95)
5.3.2 指向一维数组的指针变量的使用	(96)
5.3.3 指向字符串的指针变量的使用	(99)
5.3.4 指向二维数组的指针变量的使用	(101)
5.4 指针数组和多级指针	(105)
5.4.1 指针数组的定义	(105)
5.4.2 指针数组元素的引用	(106)
5.4.3 多级指针的定义和应用	(107)
5.5 指针应用程序设计例	(108)
习题	(110)
第 6 章 函数	(113)
6.1 函数的概念	(113)
6.1.1 函数的概念	(113)
6.1.2 函数的定义	(114)

6.1.3	函数的调用	(115)
6.2	函数调用中的数据传递方法	(117)
6.2.1	值传递方式	(117)
6.2.2	地址传递方式	(118)
6.2.3	值传递和地址传递方式的区分	(118)
6.2.4	返回值方式	(120)
6.2.5	全局外部变量的传递方式	(120)
6.3	函数的嵌套调用和递归调用	(121)
6.3.1	函数的嵌套调用	(121)
6.3.2	函数的递归调用	(122)
6.4	指针型函数及其调用	(125)
6.4.1	指针型函数的定义	(125)
6.4.2	指针型函数的调用	(126)
6.5	系统函数	(127)
6.5.1	数学函数	(127)
6.5.2	字符处理函数	(129)
6.5.3	类型转换函数	(130)
6.5.4	内存管理函数	(131)
6.5.5	其它函数	(132)
6.6	函数应用程序设计例	(134)
习题	(137)
第7章	结构型、共用型和枚举型	(141)
7.1	结构型的定义	(141)
7.1.1	结构型数据的特点	(141)
7.1.2	结构型的定义	(142)
7.2	结构型变量的定义和引用	(143)
7.2.1	结构型变量的定义和初始化	(143)
7.2.2	结构型变量成员的引用	(144)
7.3	结构型数组的定义和引用	(147)
7.3.1	结构型数组的定义和初始化	(147)
7.3.2	结构型数组元素成员的引用	(148)
7.4	指向结构型数据的指针变量的定义和引用	(150)
7.4.1	指向结构型变量的指针	(150)
7.4.2	指向结构型数组的指针	(151)
7.4.3	在函数间传递结构型数据	(153)
7.5	结构型应用程序设计例	(155)
7.6	共用型	(158)
7.6.1	共用型的定义	(159)
7.6.2	共用型变量的定义	(159)
7.6.3	共用型变量的引用	(160)
7.7	枚举型	(164)
7.7.1	枚举型的定义	(164)
7.7.2	枚举型变量的定义	(164)

7.7.3 枚举型变量的引用	(165)
7.8 用户自定义类型	(166)
习题	(168)
第8章 文件	(171)
8.1 文件概述	(171)
8.1.1 文件概述	(171)
8.1.2 文件型指针	(173)
8.2 文件的打开与关闭函数	(174)
8.2.1 打开文件函数	(174)
8.2.2 关闭文件函数	(175)
8.2.3 标准设备文件的打开与关闭	(176)
8.3 文件的读/写函数	(176)
8.3.1 文件尾测试函数	(176)
8.3.2 字符读/写函数	(176)
8.3.3 字符串读/写函数	(179)
8.3.4 数据读/写函数	(181)
8.3.5 格式读/写函数	(184)
8.4 文件处理的其它常用函数	(186)
8.4.1 文件头定位函数	(186)
8.4.2 文件随机定位函数	(187)
8.4.3 错误测试函数	(188)
8.5 文件应用程序设计例	(190)
习题	(191)
第9章 编译预处理与带参数的主函数	(194)
9.1 宏	(194)
9.2 文件包含处理	(198)
9.3 条件编译	(200)
9.4 带参数的主函数	(202)
习题	(204)
附录一 ASCII 代码表	(206)
附录二 运算符及其优先级汇总表	(207)

高级语言程序设计自学考试大纲

出版前言	(211)
一、课程性质与设置目的	(213)
二、课程内容与考核目标	(214)
第1章 概述	(214)
第2章 基本数据类型和运算	(214)
第3章 顺序结构、选择结构和循环结构的程序设计	(216)
第4章 数组	(218)
第5章 指针	(219)

第 6 章 函数	(220)
第 7 章 结构型、共用型和枚举型	(221)
第 8 章 文件	(222)
第 9 章 编译预处理与带参数的主函数	(223)
实践环节	(223)
三、有关说明和实施要求	(226)
附录 题型举例	(229)
后记	(231)

第 1 章 概 述

本章在介绍高级语言和程序设计等基本概念的基础上，介绍了 C 语言的字符集、基本词类、基本句类和程序的基本结构。这些都是掌握 C 语言程序设计的基础。

1.1 程序与程序设计

从自然语言角度来说，程序是对解决某个问题的方法步骤的描述；从计算机角度来说，程序是用某种计算机能理解并执行的计算机语言描述解决问题的方法步骤。

程序的特点是有始有终、每个步骤都能操作、所有步骤执行完对应问题就能得到解决。

例如，某个会议的安排如下：

第一项 宣布会议开始。

第二项 全体起立唱国歌。

第三项 宣读嘉奖令。

第四项 颁发奖励证书。

第五项 宣布会议结束。

上述的步骤就是一个解决嘉奖问题的程序。

又如，求一元二次方程 $ax^2 + bx + c = 0$ (设 $a \neq 0$) 实数根的步骤如下：

第一步 获得系数 a 、 b 、 c 。

第二步 计算 $d = b^2 - 4ac$ 。

第三步 若 $d > 0$

$$\text{计算: } x_1 = \frac{-b + \sqrt{d}}{2a}, x_2 = \frac{-b - \sqrt{d}}{2a}$$

输出：有两个实数根 分别为 x_1 和 x_2 转第六步；

否则转第四步。

第四步 若 $d < 0$

输出：没有实数根，转第六步；

否则转第五步。

第五步 计算： $x_1 = x_2 = \frac{-b}{2a}$

输出：有两个相同的实数根，为 x_1 转第六步。

第六步 结束。

上述步骤就是一个求一元二次方程实根的程序。

程序设计就是分析解决问题的方法步骤，并将其记录下来的过程。从自然语言角度来说，就是用自然语言记录；从计算机角度来说，必须用计算机语言记录下来。上面的例子都是用自然语言记录的程序，其中第二个程序可以用计算机语言来记录，例如用 C 语言，记录结果如下：

```
# include "math.h"                /* 程序中使用了数学类的系统函数 */
main()
{ float a,b,c,d,x1,x2;             /* 说明存放数据的变量 */
  scanf("%f,%f,%f",&a,&b,&c);      /* *输入 a、b、c三个数据 */
  d = b * b - 4 * a * c;           /* 计算  $d = b^2 - 4ac$  */
  if (d > 0)                       /* 若  $d > 0$  计算并输出两个实根 */
  { x1 = (- b + sqrt(d))/(2 * a);
    x2 = (- b - sqrt(d))/(2 * a);
    printf("x1 = %f  x2 = %f \n",x1,x2);
  }
  else
  if (d < 0) printf("Does not have real root!\n"); /* 若  $d < 0$ , 输出没有实根 */
  else { x1 = x2 = - b/(2 * a);      /* 若  $d = 0$ , 输出重根 */
        printf("x1 = x2 = %f \n",x1);
  }
}
```

1.2 高级语言与 C 语言

在程序设计的发展过程中，出现了各种各样的计算机语言。

最早期的语言是二进制语言，程序设计人员只能用计算机能直接识别和执行的二进制代码来编写程序。为了减轻程序设计人员的负担，很快又出现了汇编语言，这种语言是用符号来代表二进制代码的，所以称为符号语言。用这种语言编写的程序需要通过一种软件（称汇编程序）翻译后才能执行，所以又称汇编语言。不同的计算机上提供不同的二进制语言或汇编语言，所编写的程序一般只能在同类型的计算机上运行，所以这种语言又称为“面向机器的语

程序设计的关键是将问题及解决问题的算法过程描述出来，设计人员很快就提供了一种描述算法过程很方便，同时脱离了对机型的要求，能在任何计算机上运行的计算机语言。程序设计人员可以利用这种计算机语言直接写出各种表达式来描述简单的计算过程，利用这种语言提供的各种控制语句可以描述复杂的加工处理过程。专家们将这种语言称为“高级语言”。而将二进制语言和汇编语言统称为“低级语言”。由于高级语言是面向问题和算法过程描述的，所以又将高级语言称为“面向问题的语言”。

高级语言编写的程序称为“源程序”。源程序是不能在计算机上直接运行的，必须将其翻

译成二进制程序后才能执行。翻译过程有两种方式：一种是翻译一句执行一句，称为“解释执行”方式，完成翻译工作的程序就称为“解释程序”；另一种是全部翻译成二进制程序后再执行，承担翻译工作的程序就称为“编译程序”，编译后的二进制程序称为“目标程序”。

世界上第一个高级语言是“ALGOL(算法语言)”语言，第二个高级语言是“FORTRAN(公式翻译语言)”语言。以后陆续出现了很多种高级语言，使用面大的有 BASIC 语言、COBOL 语言、FOXBASE 语言、PASCAL 语言和 C 语言。

C 语言的前身是 ALGOL 语言。1960 年 ALGOL60 版本推出后，很受程序设计人员的欢迎。用 ALGOL60 来描述算法很方便，但是它离计算机硬件系统很远，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，并命名为“CPL(复合程序设计语言)”。CPL 由于规模大，学习和掌握困难，没有流行开来。1967 年剑桥大学的马丁·理查德对 CPL 语言进行了简化，推出“BCPL(基本复合程序设计语言)”语言。1970 年美国贝尔实验室的肯·汤普逊对 BCPL 进行了进一步的简化，突出了硬件处理能力，并取了“BCPL”的第一个字母“B”作为新语言的名称。同时用 B 语言编写了 UNIX 操作系统程序。1972 年贝尔实验室的布朗·W·卡尼汉和丹尼斯·M·利奇对 B 语言进行了完善和扩充，在保留 B 语言强大的硬件处理能力的基础上，扩充了数据类型，恢复了通用性，并取了“BCPL”的第二个字母作为新语言的名称。此后，两人合作，重写了 UNIX 操作系统。C 语言伴随着 UNIX 操作系统成为一种很受欢迎的计算机语言。

1977 年为了让 C 语言脱离 UNIX 操作系统，成为在任何计算机上都能运行的通用计算机语言，卡尼汉和利奇(K&R)撰写了《C 程序设计语言》一书对 C 语言的语法进行了规范化的描述，成为当时的标准。随着微型机的普及，出现了不同的 C 语言版本，为了统一标准，美国标准化协会(ANSI)于 1987 年制定了 C 语言的标准，称为“ANSI C”。通常将 K&R 的标准称为旧标准，将“ANSI C”称为新标准。

目前在微型机上使用的 C 编译程序有：Turbo C、Microsoft C、Quick C。本书将以“ANSI C”为标准，以“Turbo C2.0”为编译程序介绍 C 语言的内容、程序设计和调试方法。

C 语言的主要特点可以概括成如下几条：

(1) 比其它高级语言更接近硬件，比低级语言更接近算法，程序易编、易读、易查错、易修改。可以说兼有高级语言和低级语言的优点。

(2) 数据类型和运算符十分丰富，程序设计和算法描述更为简单和方便。

(3) 语法结构十分简单，语句数目少，简单易学。

(4) 它是一种结构化程序设计语言，提供了完整的程序控制语句（选择语句和循环语句），很适合结构化的程序设计方法。

(5) 它是一种模块化程序设计语言，适合大型软件的研制和调试。

(6) 它提供了大量的库函数供调用，简化了程序设计工作。

关于这些特点，可以先了解一下，等到本书内容全部学完后，就能深刻领会。

1.3 C语言的基本词法

中文是一种自然语言，学习中文要首先学习单字，然后学习单词，再学习语句构成规则，最后学习用语句构成文章。C语言是一种计算机语言，也要先学基本字符、基本词类，然后再学习语句的构成规则，最后学习用语句构成程序。

1.3.1 字符集

在C语言程序中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 (52个)。
- (2) 数字 (10个)。
- (3) 键盘符号 (33个)。

表 1-1 键盘符号表

~	波浪号)	右圆括号	:	冒号
`	重音号	_	下划线号	:	分号
!	惊叹号	-	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

- (4) 转义字符。

转义字符是由“反斜杠字符 (\)”开始后跟单个字符或若干个字符组成的，通常用来表示键盘上的控制代码或特殊符号，例如回车换行符、响铃符号等。

表 1-2 转义字符表

\n	回车换行符号	\a	响铃符号
\t	Tab 符号	\"	双引号
\v	垂直制表符号	\'	单引号
\b	左退一格符号	\\	反斜杠
\r	回车符号	\ddd	1~3 位 8 进制数 ddd 对应的符号
\f	换页符号	\xhh	1~2 位 16 进制数 hh 对应的符号

1.3.2 保留字

在 C 语言的程序中有特殊含义的英语单词称为“保留字”，主要用于构成语句，进行存储类型和数据类型定义。

C 语言的保留字如下：

auto	自动	extern	外部	sizeof	计算字节数
break	中止	float	浮点	static	静态
case	情况	for	对于	struct	结构
char	字符	goto	转向	switch	开关
continue	继续	int	整	typedef	类型定义
const	常量	if	如果	union	共用
default	缺省	long	长	unsigned	无符号
do	做	register	寄存器	void	空
double	双	return	返回	volatile	可变的
else	否则	short	短	while	当
enum	枚举	signed	带符号		

1.3.3 标识符

标识符是用户自定义的一种字符序列，通常用来表示程序中需要辨认的对象名称，比如符号常量、变量、数组、函数等对象的名字。

C 语言规定，标识符是由字母或下划线开头的字母、数字、下划线组成的一串符号，ANSI C 规定标识符长度不得大于 32 个字符，而 PC 机中通常是前 8 个字符有效。

下面给出一些正确和错误的标识符。

正确的标识符：`sum` `i` `a2` `a_2` `_a2` `__a_2`

错误的标识符：`2a` （非字母或下划线开头）

`a?` （含有非字母、数字、下划线的字符：?）

`c.g` （含有非字母、数字、下划线的字符：.）

`a-2` （含有非字母、数字、下划线的字符：-）

由于标识符主要用来命名，所以用户应选取有意义的标识符，以便在程序中能从标识符看出所标识的对象。

C 语言规定，用户选取的标识符不能是 C 语言规定的保留字。

1.3.4 C 语言的词类

C 语言的词类主要分为下列几种。

(1) 常量。

在程序运行中其值不发生变化的数据，例如 `-10`、`12.5` 等。

(2) 变量。

用来存放程序运行中变化的数据，例如输入的原始数据、中间结果、最终结果等。

(3) 运算符。

用来表示简单加工计算的符号，如 +(加)、-(减)、*(乘)、/(除)等。

(4) 函数调用。

形如“函数名(实际参数表)”的式子，它代表调用指定函数后获得的结果。

(5) 表达式。

用常量、变量、函数调用、运算符组成的式子，用来表示简单的加工计算。

(6) 保留字。

在程序或语句中，用来表示特定语法含义的英语单词。

上述词类的构成规则将在以后章节中介绍。

1.4 C 语言的基本语句

C 语言的语句主要分为下列几种。

(1) 数据定义语句。

用来定义程序中使用的各种能存放数据的对象的名称和特性。

(2) 赋值语句。

形如“变量 = 表达式”的语句，功能是计算表达式的值并赋予变量。

(3) 函数调用语句。

形如“函数名(实际参数表)”的语句，功能是调用指定函数。

(4) 表达式语句。

由任何表达式组成的语句。在 C 语言中，赋值和函数调用都是表达式，所以赋值语句和函数调用语句也是一种特殊的表达式语句。

(5) 流程控制语句。

用来控制程序执行过程的语句，如选择控制语句、循环控制语句、中止语句、继续循环语句、返回语句、无条件转移语句等。

(6) 复合语句。

用花括号括住的若干个任意语句。

(7) 空语句。

无任何操作的语句。

(8) 其它语句。

包括编译预处理命令、类型定义语句等。

上述语句的格式和功能将在以后的章节中陆续介绍。