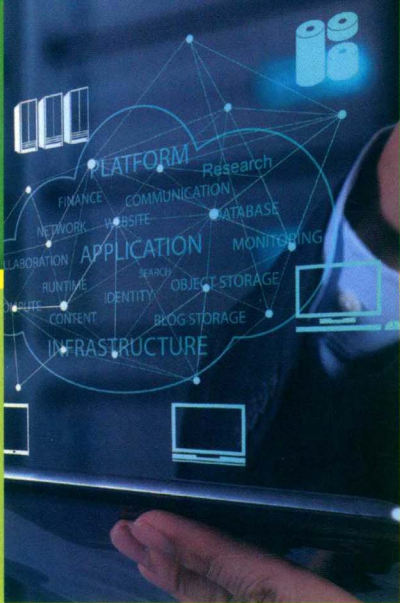


普通高等教育“十二五”规划教材



C语言

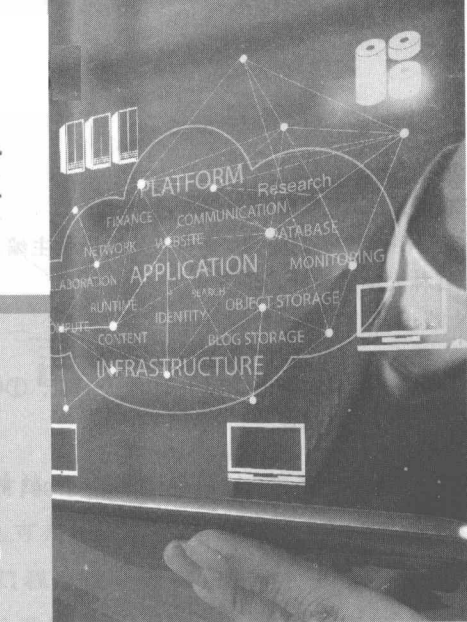
程序设计

C YUYAN CHENGXU SHEJI

高婕姝 胡祝兵 潘泽强 主编

吉林大学出版社

普通高等教育“十二五”规划教材



C 语言 程序设计

高婕姝 胡祝兵 潘泽强 主 编
薛 莹 任 娟 副主编

吉林大学出版社

图书在版编目(CIP)数据

C 语言程序设计 / 高婕姝, 胡祝兵, 潘泽强主编. —

长春: 吉林大学出版社, 2015. 4

ISBN 978-7-5677-3566-8

I. ①C… II. ①高… ②胡… ③潘… III. ①C 语言—
程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 087994 号

书 名: C 语言程序设计

作 者: 高婕姝 胡祝兵 潘泽强 主编

责任编辑: 矫正 责任校对: 郭湘怡

吉林大学出版社出版、发行

开本: 787×1092 毫米 1/16

印张: 14.5 字数: 332 千字

ISBN 978-7-5677-3566-8

封面设计: 孙雪丽

北京市彩虹印刷有限责任公司 印刷

2015 年 4 月 第 1 版

2015 年 4 月 第 1 次印刷

定价: 32.00 元

版权所有 翻印必究

社址: 长春市明德路 501 号 邮编: 130021

发行部电话: 0431-89580026/28/29

网址: <http://www.jlup.com.cn>

E-mail: jlup@mail.jlu.edu.cn



前言

C语言是一门出色的程序设计语言,它以精练、灵活、可移植性好、应用领域广泛、目标程序效率高而著称,既可用于开发系统软件,也可用于开发应用软件。由于C语言的重要性,当前高职院校都把C语言程序设计作为入门程序设计课程来对待。编者针对高职高专学生自身的特点与培养目标,从高职高专学生的思维方式、理解能力以及在后续课程中的应用等诸多方面的因素出发,对C语言程序设计进行了深入分析和探讨。

编者在多年从事高职院校“C语言程序设计”课程教学工作的基础上,结合教学实践中的经验,融入学生容易出现问题的典型例题,编写过程中力求体系结构安排合理、重点突出、难度适中,便于学生牢固掌握重点内容。在语言叙述上注重概念清晰、通俗易懂,以适应计算机教学的实际需要。

本书共分十章,具体内容如下:

第1章:C语言概述。通过简单的程序介绍了C语言的特点、C语言的书写规则以及C语言的标识符。

第2章:C语言程序设计初步。主要介绍了C语言的语句以及输入输出函数的用法;重点介绍了顺序结构程序设计、选择程序设计和循环程序设计中的语句用法及程序设计。

第3章:数据类型、运算符、表达式。本章主要强调了C语言的数据类型,常量与变量的基本用法,并详细介绍了C语言中的运算符和表达式的用法。

第4章:数组。本章主要介绍了数组、一维数组、二维数组的定义、引用以及数组在程序中的具体用法。

第5章:函数。本章主要对函数的分类、函数的定义、函数的参数进行了详细介绍,进一步诠释了C语言这种“函数式语言”的特点。同时通过函数的嵌套调用、递归调用讲解了函数调用的相关知识。

第6章:指针。本章重点介绍了指针的基本知识、数组指针变量的使用方法以及函数指针变量的用法。

第7章:结构与联合。本章详细介绍了结构体类型的定义与结构变量说明、结构体数组、结构体指针以及联合体的相关知识。

第8章:枚举和位运算。本章主要介绍了枚举类型的用法和位运算的实际应用。

第9章:预处理。本章主要介绍了宏定义、文件包含和条件编译。

第10章:文件。本章主要介绍了文件的概念及分类,文件的基本操作主要是对磁盘数



据文件的使用和操作。

本书主编为高婕姝(营口理工学院)编写第一至二章节与上机实战;胡祝兵(承德石油高等专科学校)编写第三至四章节;潘泽强(江西中医药大学)编写第五,六章节与附录B;副主编为薛莹(陕西警官职业学院警察技术系)编写第九至十章节与附录A;任娟(河南省工业科技学校)编写第七至八章节。

由于时间仓促,加上作者经验、水平有限,书中难免有遗漏或错误之处,真诚地希望得到广大读者的批评指正。

编者

2014年12月

目 录

第 1 章 C 语言概述

1.1 程序与程序设计语言简介	1
1.2 C 源程序的结构特点	2
1.3 C 语言的字符集	5
1.4 C 语言词汇	6
本章习题	8

第 2 章 C 语言程序设计初步

2.1 C 语言程序设计初步流程	9
2.2 C 程序的语句简介	9
2.3 输入输出函数	24
本章习题	32

第 3 章 数据类型、运算符和表达式

3.1 C 语言的数据类型	33
3.2 常量与变量	34
3.3 基本运算符和表达式	42
本章习题	52

第 4 章 数 组

4.1 数组概述	54
4.2 二维数组	58
4.3 字符数组	61
本章习题	68

第 5 章 函数

5.1 函数概述	70
5.2 函数定义的相关知识	72
5.3 函数的调用	80
本章习题	97

第 6 章 指针

6.1 指针基础知识	99
6.2 数组指针变量的说明和使用	104
6.3 函数指针变量	111



本章习题	118
第 7 章 结构与联合	
7.1 结构类型定义和结构变量说明	120
7.2 结构数组	124
7.3 结构指针变量	127
7.4 联合	139
本章习题	143
第 8 章 枚举和位运算	
8.1 枚举	146
8.2 位运算	148
本章习题	154
第 9 章 预处理命令	
9.1 宏定义	157
9.2 文件包含	163
9.3 条件编译	164
本章习题	166
第 10 章 文 件	
10.1 文件操作函数	170
本章习题	184
上机实战	185
附录 A 编译错误信息	216
附录 B C 语言的编程风格	223
参考文献	225



译程序先编译一个叫作“0 代码”的转换程序(representation),然后再使用虚拟器转换到可以运行于机器上的真实代码。这种成功的技巧之后又用于 Pascal 和 P-code,以及 Smalltalk 和二进制码,虽然在很多时候,中间过渡的代码往往是解译,而不是编译的。

如果所使用的翻译的机制是将所要翻译的程序代码作为一个整体翻译,之后运行内部格式,那么这个翻译过程就称为编译。因此,编译器是一个将可阅读的程序文本(叫作源代码)作为输入的数据,然后输出可执行文件(object code)。所输出的可执行文件可以是机器语言,由计算机的中央处理器直接运行,或者是某种模拟器的二进制代码。如果程序代码是在运行时才即时翻译,那么这种翻译机制就被称为解译。经解译的程序运行速度往往比编译的程序慢,但往往更具灵活性,因为它们能够与执行环境互相作用。

1.2 C 源程序的结构特点

本小节主要讲解 C 源程序的结构特点。

1.2.1 C 语言简介

C 语言是在 20 世纪 70 年代初问世的。1978 年美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的《THE C PROGRAMMING LANGUAGE》一书,通常简称为“K&R”,也有人称之为“K&R”标准。但是,在“K&R”中并没有定义一个完整的标准 C 语言。后来由美国国家标准学会在此基础上制定了一个 C 语言标准,于 1983 年发表,通常称之为 ANSI C。

早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们所认识,到了 20 世纪 80 年代,开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到广泛的使用。C 语言成为当代最优秀的程序设计语言之一。

C 语言是一种结构化语言。它层次清晰,便于按模块化方式组织程序,易于调试和维护。C 语言的表现能力和处理能力极强,它不仅具有丰富的运算符和数据类型,便于实现各类复杂的数据结构,还可以直接访问内存的物理地址,进行位(bit)一级的操作。由于 C 语言实现了对硬件的编程操作,因此它集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也适合于应用软件的开发。此外,C 语言还具有效率高、可移植性强等特点。因此,C 语言被广泛地移植到了各种类型的计算机上,从而形成了多种版本的 C 语言。

目前最流行的 C 语言有以下几种:

- Microsoft C 或称 MS C
- Borland Turbo C 或称 Turbo C
- AT&T C

这些 C 语言版本不仅实现了 ANSI C 标准,而且在此基础上各自做了一些扩充,使之更加方便、完美。



1.2.2 C源程序的结构特点

- 一个C语言源程序可以由一个或多个源文件组成。
- 每个源文件可由一个或多个函数组成。
- 一个源程序不论由多少个文件组成,都有且只有一个main函数,即主函数。
- 源程序中可以有预处理命令(include命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。
- 每一个说明、每一个语句都必须以分号结尾。但预处理命令、函数头和花括号“}”之后不能加分号。
- 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,也可不再加空格来间隔。

为了进一步说明C语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,表现了C语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中了解到一个C源程序的基本组成部分和书写格式。

例 1.1

```
main()
{
    printf("您好! 欢迎学习 C 语言\n");
}
```

main 是主函数的函数名,表示这是一个主函数。每一个C源程序都必须有,且只能有一个主函数(main函数)。函数调用语句,printf函数的功能是把要输出的内容送到显示器去显示。printf函数是一个由系统定义的标准函数,可在程序中直接调用。

例 1.2

```
#include <math.h> /* include 称为文件包含命令。它包含扩展名为.h的文件,也称为头文件或首部文件 */
#include <stdio.h>
main()
{
    double x,s; /* 定义两个实数变量,以被后面程序使用 */
    printf("input number:\n"); /* 显示提示信息 */
    scanf("%lf",&x); /* 从键盘获得一个实数 x */
    s = sin(x); /* 求 x 的正弦,并把它赋给变量 s */
    printf("sine of %lf is %lf\n",x,s); /* 显示程序运算结果 */
} /* main 函数结束 */
```

上述程序的功能是从键盘输入一个数x,求x的正弦值,然后输出结果。在main()之前的两行称为预处理命令(详见后面)。预处理命令还有其他几种,这里的include称为文件包含命令,其意义是把尖括号(<>)或引号(")内指定的文件包含到本程序来,成为本程序的



一部分。被包含的文件通常是由系统提供的,其扩展名为.h。这个文件也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此,凡是在程序中调用一个库函数时,都必须包含该函数原型所在的头文件。在本例中,使用了三个库函数:输入函数 scanf,正弦函数 sin,输出函数 printf。sin 函数是数学函数,其头文件为 math.h 文件,因此在程序的主函数前用 include 命令包含了 math.h。scanf 和 printf 是标准输入输出函数,其头文件为 stdio.h,因此主函数前也用 include 命令包含了 stdio.h 文件。

需要说明的是,C 语言规定对 scanf 和 printf 这两个函数可以省去对其头文件的包含命令。所以在本例中也可以删去第 2 行的包含命令 #include。同样,在例 1.1 中使用了 printf 函数,也省略了包含命令。

本例中的主函数体中又分为两部分,一部分为说明部分,另一部分为执行部分。说明是指变量的类型说明。例 1.2 中未使用任何变量,因此无说明部分。C 语言规定,源程序中所有用到的变量都必须先说明后使用,否则将会出错。这一点是编译型高级程序设计语言的一个特点,与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分。例 1.2 中使用了两个变量 x,s,用来表示输入的自变量和 sin 函数值。由于 sin 函数要求这两个量必须是双精度浮点型,故用类型说明符 double 来说明这两个变量。说明部分后的四行为执行部分或称为执行语句部分,用以完成程序的功能。执行部分的第 1 行是输出语句,调用 printf 函数在显示器上输出提示字符串,请操作人员输入自变量 x 的值。第 2 行为输入语句,调用 scanf 函数,接受键盘上输入的数并存入变量 x 中。第 3 行是调用 sin 函数并把函数值送到变量 s 中。第 4 行是用 printf 函数输出变量 s 的值,即 x 的正弦值。程序结束。

例 1.3

```
printf("input number:\n");
scanf("%lf",&x);
s = sin(x);
printf("sine of %lf is %lf\n",&x,s);
```

运行本程序段时,首先在显示器屏幕上给出提示串 input number:,这是由执行部分的第一行完成的。用户在提示下从键盘上键入某一数,如 5,按下回车键,接着在屏幕上给出计算结果。

1. 输入和输出函数

在例 1.1 和例 1.2 中用到的输入和输出函数 scanf 和 printf,将在第 3 章中作详细介绍。这里先简单介绍一下它们的格式,以便下面使用。scanf 和 printf 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入输出值。因此,这两个函数在括号中的参数表都由以下两部分组成:格式控制串及参数表。格式控制串是一个字符串,必须用双引号括起来,它表示了输入输出量的数据类型。各种类型的格式表示法可参阅第 3 章。在 printf 函数中,还可以在格式控制串内出现非格式控制字符,这时在屏幕上将原文显示。参数表中给出了输入或输出的量。当有多个量时,用逗号间隔。

例如:

```
printf("sine of %lf is %lf\n",x,s);
```



其中, %lf 为格式字符, 表示按双精度浮点数处理。它在格式串中两次出现, 对应了 x 和 s 两个变量。其余字符为非格式字符, 照原样输出在屏幕上。

例 1.4

```
int max(int a,int b);
main(){
    int x,y,z; /* 变量说明 */
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);
    z = max(x,y);
    printf("maxum = %d",z);
}
int max(int a,int b){
    if(a>b)return a;else return b;
```

/* 主函数 */
/* 输入 x,y 值 */
/* 调用 max 函数 */
/* 输出 */
/* 定义 max 函数 */
/* 把结果返回主调函数 */

例 1.4 中程序的功能是由用户输入两个整数, 程序执行后输出其中较大的数。本程序由两个函数组成, 主函数和 max 函数, 函数之间是并列关系, 可从主函数中调用其他函数。max 函数的功能是比较两个数, 然后把较大的数返回给主函数。max 函数是一个用户自定义函数, 因此在主函数中要给出说明(程序第 3 行)。可见, 在程序的说明部分中, 不仅可以有变量说明, 还可以有函数说明。关于函数的详细内容将在第 5 章介绍。在程序的每行后用“/*”和“*/”括起来的内容为注释部分, 程序不执行注释部分。

例 1.4 中程序的执行过程是, 首先在屏幕上显示提示串, 请用户输入两个数, 回车后由 scanf 函数语句接收这两个数送入变量 x, y 中, 然后调用 max 函数, 并把 x, y 的值传送给 max 函数的参数 a, b。在 max 函数中比较 a, b 的大小, 把大者返回给主函数的变量 z, 最后在屏幕上输出 z 的值。

2. 书写程序时应遵循的规则

从书写清晰, 便于阅读、理解、维护的角度出发, 在书写程序时应遵循以下规则:

(1) 一个说明或一个语句占一行。

(2) 用 {} 括起来的部分, 通常表示了程序的某一层结构。“{”和“}”一般与该结构语句的第一个字母对齐, 并单独占一行。

(3) 低一层次的语句或说明可比高一层次的缩进若干格后书写, 以便看起来更加清晰, 增加程序的可读性。

1.3 C 语言的字符集

字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空白符、标点和特殊字



符组成。在字符常量、字符串常量和注释中,可以使用汉字或其他可表示的图形符号。本小节将简要介绍这方面的内容。

1. 字母

小写字母 a~z 共 26 个,大写字母 A~Z 共 26 个。

2. 数字

0~9 共 10 个。

3. 空白符

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时,只起间隔作用,编译程序对它们忽略。因此在程序中使用空白符与否,对程序的编译不产生影响;但在程序中适当的地方使用空白符,将增加程序的清晰性和可读性。

4. 标点和特殊字符

包括常用的标点和一些特殊字符,和其他文档中的相同。

1.4 C 语言词汇

在 C 语言中使用的词汇分为 6 类:标识符、关键字、运算符、分隔符、常量、注释符等。本小节将分别进行介绍。

1.4.1 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外,其余都由用户自定义。C 语言规定,标识符只能是字母(A~Z, a~z)、数字(0~9)、下划线(-)组成的字符串,并且其第一个字符必须是字母或下划线。

以下标识符是合法的:

a, x, .3x, BOOK 1, sum5

以下标识符是非法:

3s 以数字开头

s * T 出现非法字符 *

-3x 以减号开头

bowy-1 出现非法字符-(减号)

在使用标识符时还必须注意以下几点:

(1)标准 C 语言不限制标识符的长度,但它受各种版本的 C 语言编译系统限制,同时也受到具体机器的限制。例如,在某版本 C 语言中规定标识符前 8 位有效,当两个标识符前 8 位相同时,则被认为是同一个标识符。

(2)在标识符中,大小写是有区别的。例如,BOOK 和 book 是两个不同的标识符。

(3)标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号。因此,命名



应尽量有相应的意义,以便阅读理解,做到“顾名思义”。

1.4.2 关键字

关键字是由C语言规定的具有特定意义的字符串,通常也称为保留字。用户定义的标识符不应与关键字相同。C语言的关键字分为以下几类。

1. 类型说明符

用于定义或说明变量、函数和其他数据结构的类型。如前面例题中用到的 `int`, `double` 等。

2. 语句定义符

用于表示一个语句的功能。如例 1.4 中用到的 `if-else` 就是条件语句的语句定义符。

3. 预处理命令字

用于表示一个预处理命令。如前面各例中用到的 `include`。

1.4.3 运算符

C语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式,表示各种运算功能。运算符由一个或多个字符组成。这部分将在第3章详细介绍。

1.4.4 分隔符

在C语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中,分隔各个变量。空格多用于语句各单词之间作间隔符。在关键字、标识符之间必须有一个以上的空格符作间隔,否则将会出现语法错误。例如,把 `int a;` 写成 `inta;`,C编译器会把 `inta` 当成一个标识符处理,其结果必然出错。

1.4.5 常量

C语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在第3章中将专门进行介绍。

1.4.6 注释符

C语言的注释符是以“`/*`”开头并以“`*/`”结尾的串。在“`/*`”和“`*/`”之间的即为注释。程序编译时,不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来,使翻译跳过不作处理,待调试结束后再去掉注释符。



本章习题

1. 填空题

(1) C 语言的字符集包括_____、_____、_____和_____。

(2) C 语言是通过_____来进行输入和输出的。

2. 选择题

(1) 以下不是 C 语言的特点的是()。

A. 语言简洁紧凑

B. 能够编制出功能复杂的程序

C. 可以直接对硬件操作

D. 移植性好

(2) 下列字符序列中,不可用作 C 语言标识符的是()。

A. abc123

B. no.1

C. _123_

D. _ok

(3) 下列字符列中,正确的 C 语言标识符是()。

A. _buy_2

B. 2_buy

C. ?_buy

D. buy?

(4) 下列符号中,不属于转义字符的是()。

A. \\

B. \0xAA

C. \t

D. \0

(5) 下列不属于 C 语言关键字的是()。

A. int

B. break

C. while

D. character

(6) 下列属于 C 语言提供的合法关键字的是()。

A. Float

B. signed

C. integer

D. Char

(7) 以下不能定义为用户标示符的是()。

A. scanf

B. Void

C. _3com_

D. int

3. 问答题

(1) 简述 C 语言的结构特点。

(2) 书写程序时需要注意什么?



第2章 C 语言程序设计初步

2.1 C 语言程序设计初步流程

本小节主要介绍 C 语言程序设计流程。

从程序流程的角度来看,程序可以分为三种基本结构,即顺序结构、分支结构、循环结构。这三种基本结构可以组成所有的各种复杂程序。

传统的流程图用流程线描述各框的执行顺序。但对流程线的使用并没有严格的限制,因此,若使用者不受限制地随意转移流程时,就变得很混乱。实际上,即便程序员努力控制流程的变化,但当算法复杂时,流程图往往不可避免地变得复杂和混乱。而当结构异常复杂时,“N-S 盒图”也很复杂。

那么,能不能只使用几种基本的结构,来组合表达出各种复杂的算法结构呢? 1966 年,Bohra 和 Jacopini 给出了肯定的答案。他们证明了:使用顺序、分支(也叫作“条件选择”)和循环这三种基本结构可以表示任何一个算法的基本单元。

从目前的编程实践看,结构化程序设计的思路已经被绝大多数程序员接受。人们普遍认为,必须采用结构化的程序设计方法。因为结构化程序具有结构清晰,便于阅读、修改和维护的优点。

结构化程序设计的基本思路是:把一个复杂的问题的求解过程分阶段进行,每个阶段处理的问题都控制在人们容易理解的范围之内。采取以下的方法可以保证得到结构化的程序:自顶向下;逐步细化(求精);模块化设计;结构化编码。

采用自顶向下、逐步细化的方法可以使程序的结构清晰、层次分明、容易改写。就像进行房屋设计所采用的施工步骤一样:先进行整体的规划,画出施工的图纸,再进行各个部分的设计,最后进行细节的设计(楼宇通信系统、安全设施的装备、办公系统、室内的装修)。

结构化程序具有如下的特征:

- 一个程序单元由顺序、分支和循环这三种基本结构组成;
- 一个大的程序由若干个不同功能的小模块组成;
- 每一个小模块只用一个入口和一个出口。

2.2 C 程序的语句简介

C 程序的执行部分是由语句组成的。程序的功能也是由执行语句实现的,C 程序主要包



含表达式语句、函数调用语句、控制语句、复合语句和空语句,其中控制语句包含多种情况,本小节将分别进行介绍。

2.2.1 表达式语句

表达式语句由表达式加上分号组成。其一般形式为:

表达式;

执行表达式语句就是计算表达式的值。

例如:

$x=y+z$; 赋值语句

$y+z$; 加法运算语句,但计算结果不能保留,无实际意义

$i++$; 自增 1 语句, i 值增 1

2.2.2 函数调用语句

函数调用语句由函数名、实际参数加上分号组成。

其一般形式为:

函数名(实际参数表);

执行函数语句就是调用函数体并把实际参数赋予函数定义中的形式参数,然后执行被调函数体中的语句,求取函数值。在第 5 章“函数”中将详细介绍。

例如, `printf("C Program");` 调用库函数,输出字符串。

2.2.3 控制语句

控制语句用于控制程序的流程,以实现程序的各种结构方式。

它们由特定的语句定义符组成。C 语言有 9 种控制语句,概括来说可分成以下三类:

1. 条件判断语句

`if` 语句, `switch` 语句。

2. 循环执行语句

`while` 语句, `do-while` 语句, `for` 语句。

3. 转向语句

`goto` 语句, `break` 语句, `continue` 语句, `return` 语句。

2.2.4 复合语句

把多个语句用 `{}` 括起来组成的一个语句称为复合语句,又称分程序。在程序中应把复合语句看成是单条语句,而不是多条语句。

例如:

```
{  
     $x=y+z$ ;  
     $a=b+c$ ;
```



```
printf("%d%d",x,a);  
}
```

是一条复合语句。复合语句内的各条语句都必须以“;”结尾,在“}”外不能加分号。

2.2.5 空语句

只有“;”组成的语句称为空语句。空语句是什么也不执行的语句。在程序中空语句可用来作空循环体。

例如:while(getchar() != '\n'); 本语句的功能是,只要从键盘输入的字符不是回车则重新输入。这里的循环体为空语句。

2.2.6 赋值语句

赋值语句是由赋值表达式再加上分号构成的表达式语句。其一般形式为:

变量=表达式;

赋值语句的功能和特点都与赋值表达式相同。它是程序中使用最多的语句之一。

在赋值语句的使用中需要注意以下几点:

(1)由于在赋值符“=”右边的表达式也可以是一个赋值表达式,因此,“变量=(变量=表达式);”是成立的,从而形成嵌套的情形。其展开之后的一般形式为:

变量=变量=...=表达式;

例如:

a=b=c=d=e=5;

按照赋值运算符的右接合性,它实际上等效于:

e=5;

d=e;

c=d;

b=c;

a=b;

(2)注意在变量说明中给变量赋初值和赋值语句的区别。给变量赋初值是变量说明的一部分,赋初值后的变量与其后的其他同类变量之间仍必须用逗号间隔,而赋值语句则必须用分号结尾。

(3)在变量说明中,不允许连续给多个变量赋初值。如下述说明是错误的: int a=b=c=5,必须写为 int a=5;b=5;c=5,而赋值语句允许连续赋值。

(4)注意赋值表达式和赋值语句的区别。赋值表达式是一种表达式,它可以出现在任何允许表达式出现的地方,而赋值语句则不能。

下述语句是合法的: if((x=y+5)>0) z=x;,语句的功能是,若表达式 x=y+5 大于 0 则 z=x。下述语句是非法的: if((x=y+5;)>0) z=x;,因为 x=y+5;是语句,不能出现在表达式中。