



21世纪全国职业院校计算机专业通用教材

C语言程序设计

主编 伍云辉 汤元斌



电子科技大学出版社

21 世纪全国职业院校计算机专业通用教材

C 语言程序设计

主 编 伍云辉 汤元斌

编 委 伍云辉 刘小平 汤元斌

杨 红 罗名兰 肖洪云

王彬华

电子科技大学出版社

图书在版编目(CIP)数据

C 语言程序设计/伍云辉, 汤元斌主编. —成都: 电子科技大学出版社, 2007.7

ISBN 978 - 7 - 81114 - 401 - 7

I. C… II. ①伍… ②汤… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 016965 号

内 容 提 要

C 语言是一种应用十分广泛的高级编程语言, 既有高级语言的特点, 又有汇编语言等低级语言的特点, 成为编制系统软件的首选语言。本书共分十二章, 分别从 C 语言基础、数据类型、表达式和运算符、C 语言程序设计初步、分支结构程序、循环结构程序、数组、函数、指针、结构与联合、文件处理、预处理等方面进行了由浅入深的介绍。

本书既可作为全国职业院校计算机专业通用教材和教学参考用书, 还也可作为全国计算机等级考试教材。

21 世纪全国职业院校计算机专业通用教材

C 语言程序设计

主编 伍云辉 汤元斌

出 版: 电子科技大学出版社(成都市一环路东一段 159 号电子信息产业大厦 邮编: 610051)

策划编辑: 张 俊

责任编辑: 周 岚

主 页: www.uestcp.com.cn 电子邮箱: uestcp@uestcp.com.cn

发 行: 新华书店经销

印 刷: 四川墨池印务有限公司

成品尺寸: 185 mm × 260 mm 印张 13.25 字数 340 千字

版 次: 2007 年 7 月第一版

印 次: 2007 年 7 月第一次印刷

书 号: ISBN 978 - 7 - 81114 - 401 - 7

定 价: 18.00 元

■ 版权所有 侵权必究 ■

- ◆ 邮购本书请与本社发行部联系。电话: (028) 83202323, 83256027
- ◆ 本书如有缺页、破损、装订错误, 请寄回印刷厂调换。
- ◆ 课件下载在我社主页“下载专区”。

前 言

近年来,面向对象的计算机程序设计语言大量出现,受到了广大程序员的青睐,使用面向对象程序设计方法可更快速、更高效地开发应用程序。面向对象的程序设计基础是以面向过程程序设计为基础的,所以必须首先学好面向过程的程序设计方法。

C 语言不仅具有速度快、占用资源少、目标程序效率高等汇编语言的优点,还具有可移植性强、便于理解、掌握、使用灵活方便等高级语言的优点,特别适用于开发系统软件。从 C 语言诞生开始就得到了快速的推广和应用。

在我国,C 语言是大中专院校计算机专业的必修课程,而且很多非计算机专业也开设了 C 语言课程。尤其是全国计算机等级考试二、三、四级都将考核 C 语言的相关知识,这进一步加快了 C 语言在我国的推广应用。学习 C 语言已经成为计算机专业技术人员和计算机应用人员的迫切要求。

C 语言具有强大的功能,因其语法限制不太严格,语言描述简单、明晰,没有复杂的控制结构,程序设计的自由度比较大,无论是设计系统软件,还是开发图形处理软件都可以用到。但由于 C 语言除了具有高级程序设计语言的特点外,还可直接访问物理地址,尤其是 C 语言提供的指针功能,涉及的概念比较复杂,规则繁多,虽然使用灵活,但容易出错,让不少初学者难以理解和应用。

本书从职业院校学生的特点和认知规律出发,精选例题,从容易理解的例题入手,分散难点,通过例题的运行效果来介绍复杂的概念,力求做到通俗易懂。在编写这本教材时我们突出了重点,兼顾了内容的深度和广度,增加了教材的灵活性,可适应不同学校、不同学制、不同专业的教学需要,又便于学生自学。

本书以使用最为广泛的 Turbo C 作为编程环境,所有的代码都使用该编译器调试通过。书中每章都提供了丰富的典型例题,使读者能更加深刻地理解相关概念,掌握各章知识要点,解读知识难点。

在本书的编写过程中,我们参考了所有能找到的有关方面的文献和资料,包括互联网上的一些信息,在此向提供信息资源的作者一并表示感谢!由于时间仓促,加上作者水平有限,书中错误在所难免,恳请专家和广大读者不吝赐教!

编 者

2007 年 7 月

目 录

第 1 章 C 语言概述.....	1
1.1 C 语言的发展及特点.....	1
1.1.1 C 语言发展.....	1
1.1.2 C 语言的特点.....	1
1.2 C 语言程序结构及书写规则.....	3
1.2.1 C 语言程序的总体结构.....	3
1.2.2 输入和输出函数.....	6
1.2.3 C 语言的字符集.....	6
1.2.4 C 语言词汇.....	6
1.2.5 C 语言源程序的结构特点.....	8
1.2.6 书写程序时应遵循的规则.....	8
1.3 Turbo C 概述.....	8
1.3.1 Turbo C 的发展.....	8
1.3.2 Turbo C 的安装.....	9
1.3.3 进入 Turbo C.....	9
1.3.4 主菜单.....	9
1.3.5 编辑窗口.....	14
1.3.6 信息窗口.....	14
1.3.7 功能键提示.....	14
1.3.8 Turbo C 的配置文件.....	15
1.3.9 C 语言程序的运行过程.....	15
【习题】.....	15
第 2 章 数据类型.....	17
2.1 数据类型概述.....	17
2.1.1 数据类型分类.....	17
2.1.2 常量和变量.....	18
2.1.3 变量初始化.....	19
2.2 整型数据.....	20
2.2.1 整型常量.....	20
2.2.2 整型变量.....	21
2.3 实型数据.....	22
2.3.1 实型常量.....	22

2.3.2 实型变量.....	22
2.4 字符型	23
2.4.1 字符常量.....	23
2.4.2 字符变量.....	24
2.4.3 字符串常量.....	24
2.5 数据的存储和类型转换.....	25
2.5.1 数据的存储.....	25
2.5.2 数据类型转换.....	26
【习题】	28
第3章 运算符和表达式	30
3.1 基本运算符和表达式.....	30
3.1.1 运算符的种类.....	30
3.1.2 优先级和结合性.....	31
3.1.3 表达式	32
3.2 算术运算符和算术表达式.....	32
3.2.1 算术运算符.....	32
3.2.2 算术表达式.....	34
3.2.3 表达式的注意事项.....	34
3.3 赋值运算符和赋值表达式.....	35
3.3.1 简单赋值运算符和表达式.....	35
3.3.2 复合赋值运算符及表达式.....	36
3.3.3 表达式中的类型转换.....	36
3.4 关系运算符和表达式.....	37
3.4.1 关系运算符.....	37
3.4.2 关系表达式.....	37
3.5 逻辑表达式	38
3.5.1 逻辑运算符.....	38
3.5.2 逻辑表达式.....	39
3.6 逗号运算符和表达式.....	40
【习题】	41
第4章 C 语言程序设计初步	43
4.1 算法的描述	43
4.1.1 算法的概念.....	43
4.1.2 算法的特性.....	43
4.1.3 算法的表示.....	44
4.2 C程序的语句.....	46
4.2.1 表达式语句.....	46
4.2.2 函数调用语句.....	46

4.2.3 控制语句.....	46
4.2.4 复合语句.....	47
4.2.5 空语句	47
4.3 赋值语句	47
4.4 数据输出语句.....	48
4.4.1 格式输出函数 (printf 函数)	48
4.4.2 字符输出函数 (putchar 函数)	51
4.5 数据输入语句.....	51
4.5.1 格式输入函数 (scanf 函数)	51
4.5.2 键盘输入函数 (getchar 函数)	55
4.6 简单程序设计实例.....	56
【习题】	57
第 5 章 分支结构程序.....	58
5.1 if 语句.....	58
5.1.1 if 语句的形式.....	58
5.1.2 if 语句的嵌套.....	62
5.1.3 条件运算符和条件表达式.....	63
5.2 switch 语句.....	64
5.3 分支结构程序设计实例.....	67
【习题】	69
第 6 章 循环结构程序.....	70
6.1 while 语句.....	70
6.2 do...while 语句.....	72
6.3 for 语句.....	73
6.4 转移语句	75
6.4.1 goto 语句	75
6.4.2 break 语句.....	76
6.4.3 continue 语句	76
6.5 循环的嵌套	77
6.6 循环程序设计实例.....	79
【习题】	82
第 7 章 数组.....	83
7.1 一维数组	83
7.1.1 一维数组的定义.....	83
7.1.2 数组元素的表示方法.....	84
7.1.3 一维数组的赋值.....	85
7.1.4 一维数组实例.....	86

7.2 二维数组	87
7.2.1 二维数组的定义	87
7.2.2 二维数组元素的表示方法	87
7.2.3 二维数组的初始化	87
7.2.4 二维数组程序举例	89
7.3 字符数组和字符串	90
7.3.1 字符数组	90
7.3.2 字符串	91
7.3.3 字符串的输入输出	92
7.3.4 字符串处理函数	93
7.4 数组实例	95
【习题】	99
第 8 章 函数	101
8.1 函数的概述	101
8.1.1 函数分类	101
8.1.2 函数的一般形式	102
8.2 函数的调用	104
8.2.1 函数调用的一般形式	104
8.2.2 被调函数的说明	105
8.3 函数的参数和返回值	107
8.3.1 函数的参数	107
8.3.2 函数的返回值	108
8.3.3 数组作为函数参数	109
8.4 函数的嵌套和递归	113
8.4.1 函数的嵌套	113
8.4.2 函数的递归调用	115
8.5 变量的作用域	118
8.5.1 局部变量	118
8.5.2 全局变量	120
8.6 变量的存储类型	122
8.6.1 自动变量	123
8.6.2 外部变量	125
8.6.3 静态变量	125
8.6.4 寄存器变量	127
8.7 结构化程序设计	128
8.7.1 文件包含	128
8.7.2 内部函数和外部函数	129
8.7.3 变量、函数与程序文件模块的关系	130
【习题】	131

第9章 指针	133
9.1 指针的基本概念	133
9.1.1 基本概念	133
9.1.2 指针变量的类型说明	134
9.1.3 指针变量的赋值	135
9.1.4 指针变量的运算	135
9.2 数组指针变量	138
9.2.1 数组和指针的关系	138
9.2.2 数组指针变量说明	138
9.2.3 用数组名和数组指针变量作函数参数	140
9.2.4 指向多维数组的指针变量	141
9.2.5 字符串指针变量	142
9.3 函数指针变量	145
9.3.1 使用函数指针变量调用函数	145
9.3.2 指针型函数	146
9.4 指针数组和指向指针的指针	148
9.4.1 指针数组的概念	148
9.4.2 指针数组作函数参数	149
9.4.3 main 函数的参数	151
9.4.4 指向指针的指针	152
【习题】	153
第10章 结构与联合	155
10.1 结构体	155
10.1.1 结构的定义	155
10.1.2 结构类型变量的说明	156
10.1.3 结构变量成员的表示方法	157
10.1.4 结构变量的赋值	158
10.1.5 结构变量的初始化	158
10.1.6 结构数组	159
10.1.7 结构指针变量	160
10.2 链表	164
10.2.1 链表的基本概念	164
10.2.2 分配存储空间	165
10.2.3 链表的基本操作	166
10.3 联合体	170
10.3.1 联合的定义	171
10.3.2 联合变量的说明	171
10.3.3 联合变量的赋值和使用	172
10.4 枚举类型	174

10.4.1 枚举类型的定义.....	174
10.4.2 枚举变量的说明.....	174
10.4.3 枚举类型变量的赋值和使用	175
10.5 位运算	176
10.5.1 位运算.....	176
10.5.2 位域	178
10.6 类型定义符 typedef.....	180
【习题】	181
第 11 章 文件	182
11.1 文件的基本概念.....	182
11.1.1 文件的分类.....	182
11.1.2 文件指针.....	182
11.2 文件的打开与关闭.....	183
11.2.1 文件打开函数 fopen.....	183
11.2.2 文件关闭函数 fclose	184
11.3 文件的读写.....	184
11.3.1 字符读写函数 fgetc 和 fputc.....	185
11.3.2 字符串读写函数 fgets 和 fputs	187
11.3.3 数据块读写函数 fread 和 fwrite	189
11.3.4 格式化读写函数 fscanf 和 fprintf.....	190
11.4 文件的随机读写.....	190
11.4.1 文件的定位.....	190
11.4.2 文件检测函数.....	192
【习题】	192
第 12 章 预处理	193
12.1 宏定义	193
12.1.1 无参宏定义.....	193
12.1.2 带参宏定义.....	195
12.2 条件编译	199
12.2.1 #ifdef.....	199
12.2.2 #ifndef.....	200
12.2.3 #if.....	200
【习题】	201
参考文献	202

第1章 C语言概述

C语言是一种比较流行的程序设计语言。它是一种高级程序设计语言，程序不依赖于特定的计算机硬件，具有良好的可读性和可移植性，接近于自然语言和数学语言；同时它又具有低级语言的特性，可以内嵌汇编指令，将汇编指令作为其指令，可以直接对计算机硬件进行操作（如对内存地址的操作、位操作、I/O操作等）。C语言集高级语言和低级语言的优点于一身，适合作为系统描述语言，用于编写大型的操作系统、编译系统，也适用于编写应用软件。

1.1 C语言的发展及特点

1.1.1 C语言发展

C语言由早期的编程语言BCPL(Basic Combined Programming Language)发展演变而来。并于1972年首次在UNIX操作系统的DEC PDP-11计算机上使用。

1970年，美国电话电报公司(AT&T)贝尔实验室的Ken Thompson以BCPL语言为基础，设计出了很简单而且很接近硬件的B语言，并用B语言编写了第一个UNIX操作系统。但B语言过于简单，功能有限。

1972年，贝尔实验室的D.M.Ritchie在B语言的基础上设计出了C语言，C语言既保持了BCPL和B语言的优点，又克服了它们的缺点。最初的C语言主要用来描述和实现UNIX操作系统。随着UNIX的日益广泛使用，C语言也迅速得到推广。

1983年，由贝尔实验室的Bjarne Stroustrup在C语言的基础上推出了面向对象的语言。它的出现进一步扩充和完善了C语言，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。同时，C++语言又提出了一些更为深入的概念，它所支持的一些面向对象的概念容易将问题空间直接地映射到程序空间，因而增加了整个语言的复杂性，掌握起来具有一定的难度。

随着微型计算机日益普及，C语言出现了许多不同的版本。由于这些版本没有统一的标准，使得各种版本C语言之间出现了许多不一致的地方。为了统一不同版本，美国国家标准研究所(ANSI)专为C语言制定了ANSI标准。目前广泛流行的各种C语言编译系统有Microsoft C、Turbo C、Borland C、Quick C等，它们的基本部分都是相同的，都符合ANSI C标准，但不同版本又对ANSI C进行了不同的扩充，使用上有一些差异。

C语言是C++的基础，C++语言和C语言在很多方面是兼容的。因此，掌握了C语言，再进一步学习C++就能以一种熟悉的语法来学习面向对象的语言，从而达到事半功倍的目的。

1.1.2 C语言的特点

C语言具有强大的功能，无论是设计系统软件，还是开发图形处理软件都可以用到。如

早期的数据库系统软件 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。C 语言具有如此强大的生命力，正是源于它具有下列不可替代的特点：

1. 语言简洁、紧凑，使用灵活、方便

C 语言的语法限制不太严格，语言描述简单、明晰，没有复杂的控制结构，程序设计的自由度比较大。C 语言一共有 32 个保留字（关键字），语言的许多成分都是通过显式的函数调用来完成，这样就使得 C 语言编译和硬件的相关性很小，故 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，没有像其他高级程序设计语言那样的书写格式限制，用 C 语言书写程序可以随心所欲，自由方便。

2. 运算符丰富表达能力强

C 语言包含 34 种运算符，并把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 语言的运算类型极其丰富，表达式类型也多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据结构丰富，具有现代化语言的各种数据结构

C 语言具有结构化程序的典型特征，程序代码与数据分离，具有现代语言的各种数据结构。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构类型、共用体类型等，能实现链表、树、栈等复杂数据结构的运算，具有结构化的流程控制语句，编制的程序具有模块化和结构化的特点。

4. 具有结构化的控制语句

C 语言的结构化控制语句有 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句等。C 语言程序还用函数作为程序的模块单位，便于实现程序的模块化，编制的程序具有模块化和结构化的特点。

5. C 语言允许直接访问物理地址

C 语言能实现汇编语言的大部分功能，如进行位、字节、地址等操作，也可以直接对硬件进行操作。因此，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，可用来编写系统软件。

6. 生成目标代码质量高，程序执行效率高

许多实验表明，针对同一个问题，用 C 语言描述，其代码效率只比汇编语言低 10%~20%。高级语言除了要求语言表达能力强之外，还要求该语言的代码质量高，这样才能适用于编写系统软件，如果代码质量低，则系统开销就会增大。一般说来，语言越低级其代码质量就越高，但编程的工作量也越大，故用汇编语言编程的代码质量最高。

7. C 语言写的程序可移植性好（与汇编语言相比）

C 语言程序基本上不做修改就能用于各种型号的计算机或各种操作系统。

8. 模块化的程序结构

C 语言的程序是由函数构成，一个函数即为一个“程序模块”。一个 C 语言源程序至少包括一个 main 函数，也可以包含一个 main 函数和若干其他函数。无论 main 函数在整个程序中的何处，C 程序总是从 main 函数开始执行的。因此，函数是 C 程序的基本单位。同时，系统也提供了丰富的库函数，用户可以在程序中直接引用相应的库函数，也可根据需要编制和设计自己的函数。

上面简单地介绍了 C 语言的主要特点，至于具体细节将在以后的各章节内容中介绍。对初学者来说，C 语言的这些特点现在可能体会不到，等学完本书后，相信定会有一个比较深

刻的认识。

1.2 C语言程序结构及书写规则

1.2.1 C语言程序的总体结构

下面先介绍几个简单的C程序，从中来分析C程序的基本结构。

【例 1.1】

```
main()      /*这里不包括可执行代码 */
{

}

```

说明：

该程序是最简单的一个C程序，它没有任何可执行语句。程序中各部分的作用如下：

(1) 主函数，main 表示“主函数”，每一个C程序都有且仅有一个main函数。函数体由一对大花括弧{ }括起来。C程序的执行就是从main函数的函数体中第一条可执行语句开始逐条进行的。

(2) 注释，在“/*”和“*/”之间的内容作为C程序的注释部分，它只起到对程序的注释或说明作用，并不影响C程序的执行。

本例不会产生任何执行结果，因为它一条可执行语句也没有。这在别的程序设计语言中可能不允许，但在C程序中则是完全合法的。

在主函数中增加一条输出语句，具体代码如下：

```
main ()      /* 主函数 */
{            /* 函数体开始 */
    printf("This is a C program. \n"); /* 用标准输出函数 printf 输出字符串 */
}            /* 函数体结束 */

```

此程序的作用是输出以下一行信息：

this is a C program.

说明：

程序的作用是输出“This is a C program.”，其中main是“主函数”，函数体由大花括弧{ }括起来。本例中主函数内只有一个输出语句，“printf”指的是C语言中的输出函数。双引号（半角）内的字符串按原样输出，“\n”是换行符，即在输出“This is a C program.”后自动回车换行，在语句最后面一定要有一个分号，表示语句结束。

【例 1.2】

```
main ()      /* 求两数之和 */
{
    int x,y,sum;      /* 这是定义变量 */
    x=83; y=124;      /* 以下3行为C语句 */
    sum=x+y;
    printf("sum is %d\n", sum);
}

```

```
}

```

此程序结果为：

sum is 207

说明：

本程序的作用是求两个整数 x 和 y 之和，并将结果保存到变量 sum 中。其中各行语句的作用如下：

(1) `/* .....*/` 表示注释部分。注释可以加在程序中任何位置。

(2) 第 3 行是声明部分，定义变量 x 和 y ，指定 x 和 y 为整型 (`int`) 变量。

(3) 第 4 行是两个赋值语句，使 x 和 y 的值分别为 83 和 124。

(4) 第 5 行进行求和运算，使变量 sum 的值为 $x+y$ 。

(5) 第 6 行中 `"%d"` 是输入输出的“格式字符串”，用来指定输入输出时的数据类型和格式，“`%d`”表示“十进制整数类型”。在执行输出时，此位置上代以一个十进制整数值。`printf` 函数中括弧内最右端 sum 是要输出的变量，现在它的值为 207 (即 $83+124$ 之值)。因此输出一行信息为“sum is 207”。

【例 1.3】

```
main() /*主函数*/
{
    int a,b,c; /*声明部分，定义变量*/
    scanf ("%d,%d",&a,&b); /*输入变量 a 和 b 的值*/
    c=max(a,b); /*调用 max 函数，将得到的值赋给 c*/
    printf ("max=%d",c); /*输出 c 的值*/
}

int max(int x,int y) /*定义 max 函数，函数值为整型，形式参数 x, y 为整型*/
{
    int z; /* max 函数中的声明部分，定义本函数中用到的变量 z 为整型*/
    if (x>y)z=x;
    else z=y;
    return(z); /*将 z 的值返回，通过 max 带回调用处*/
}
```

程序运行情况如下：

10,9↵ (将数值 10 和 9 分别赋给变量 a 和 b)

max=10 (输出 c 的值)

说明：

(1) 本程序包括两个函数：主函数 `main` 和被调用的函数 `max`。

(2) `max` 函数的作用是将 x 和 y 中较大的值赋给变量 z 。最后面的 `return` 语句是将 z 的值返回给主调函数 `main`。它是通过函数名 `max` 带回到 `main` 函数的调用处。

(3) `main` 函数中的 `scanf` 是“输入函数”的名字 (`scanf` 和 `printf` 分别是 C 语言提供的标准输入和输出函数)。程序中 `scanf` 函数的作用是输入 a 和 b 的值。`&a` 和 `&b` 中的“`&`”的含义是取 a 和 b 的地址，`scanf` 函数的作用是输入两个数值并将它们送到变量 a 和 b 的地址所

标志的单元中，也就是将数值输入给变量 *a* 和 *b*。这种形式是与其他语言不同的。它相当于 BASIC 语言中的 `Input a, b` 或 PASCAL 语言中的 `Read (a, b)`。`&a` 和 `&b` 前面的 “%d, %d” 只是用于“输入”，它指定输入的两个数据按十进制整数形式输入。

(4) 在 `main` 函数中调用 `max` 函数，在调用时将实际参数 *a* 和 *b* 的值分别传送给 `max` 函数中的形式参数 *x* 和 *y*。执行 `max` 函数后得到一个返回值（即 `max` 函数中变量 *z* 的值），把这个值赋给变量 *c*。然后输出较大的数值 *c* 的值。`printf` 函数中双引号内的 “`max=%d`”，在输出时，“%d” 将由 *c* 的值取代，“`max=`” 原样输出。

通过以上几个简单的 C 语言程序，可以看出 C 程序的基本构架如下：

1. C 语言程序的构成

一个 C 语言的源程序至少包含一个 `main` 函数，也可以包含一个 `main` 函数和若干个其他函数。因此，C 语言程序是由函数构成的，函数是 C 语言程序的基本单位。被调用的函数可以是系统提供的库函数（如 `printf` 和 `scanf` 函数），也可以是用户根据需要自己编制设计的函数（如 `max` 函数）。C 语言程序中的函数相当于其他语言中的子程序，用函数来实现特定的功能（如例 1.3 中求最大数的 `max` 函数）。可以说 C 语言是函数式的语言，程序中的全部工作都是由各个函数分别来完成的。

2. 函数的组成

一个函数由两部分组成：函数头部和函数体。

(1) 函数的头部，包括函数名、函数类型、函数属性、参数类型和函数参数（形参）。

例如：

<code>int</code>	<code>max</code>	<code>(int</code>	<code>a</code>	<code>int</code>	<code>b)</code>
↓	↓	↓	↓	↓	↓
函数类型	函数名	函数参数类型	函数参数名	函数参数类型	函数参数名

需要注意的是，一个函数名后面必须跟一对圆括弧，函数可以没有参数，如 “`main()`”。

(2) 函数体，即函数头部下面的大花括弧 {……} 中的内容。如果一个函数内有多个大花括弧，则最外层的一对为函数体的范围。

函数体一般包括声明部分和执行部分，一般形式为：

函数名（参数）

```
{
    声明部分;
    执行部分;
}
```

如果在一个 C 语言程序中没有声明部分和执行部分，也是合法的，如例 1.1 所示。

(3) `main` 函数是 C 语言程序的一个入口，每个 C 语言程序总是从 `main` 函数开始执行的，而不论 `main` 函数在整个程序中的哪个位置（`main` 函数可以放在程序最前面，也可以放在程序最后面，或在一些函数之前或在另一些函数之后）。

(4) C 语言程序书写格式自由，语句用分号 “;” 作结束符。一行内可以写几个语句，一个语句也可以分写在多行上。C 语言的书写支持缩进格式，将同一层次的语句的开头对齐在同一列，而将下一个层次的语句开头通过加空格缩进若干列，从而形成层层缩进对齐的书写格式。

程序中可加必要的注释，使用一对符号 “`/**/`” 作程序中注释的定界符，书写在 “`/**`”

和“*/”之间的内容是注释内容。注释不影响程序的编译和执行，可以增加程序的可读性。适量添加注释是一种良好的程序设计行为。

1.2.2 输入和输出函数

C 语言本身并没有提供输入输出语句，输入输出的操作是由库函数来实现的。

1. printf 函数

printf 函数用于向显示器输出多个各种类型和格式的数据，其基本格式如下：

printf(<格式控制字符串>, <输出表达式列表>);

“格式控制字符串”是一个字符串，必须用双引号括起来，用于定义输出的内容和输出内容的类型、格式以及相互之间的位置关系。由普通字符和格式说明符组成，格式说明符由“%”和一些数字和特定的字母组成，确定输出数据的类型和格式。

“输出表达式列表”包含需要输出的常量、变量、表达式，每个表达式用逗号隔开。表达式的数量应与“格式控制字符串”中的“%”的个数一致。

2. scanf 函数

scanf 函数用于从终端输入多个各种类型和格式的数据，其基本格式如下：

scanf(<格式控制字符串>, <地址列表>);

“格式控制字符串”的使用基本与 printf 函数相同。

“地址列表”由多个地址组成，用逗号隔开，可以是变量的地址，使用取地址运算符“&”，也可以是指针变量，还可以是字符数组的首地址（相关内容参阅后面章节）。地址的作用是指出由终端输入的数据将会保存到地址所对应的存储单元。

1.2.3 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、空格、标点和特殊字符组成。在字符常量，字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

(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，对程序的编译不发生影响，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。

1.2.4 C 语言词汇

在 C 语言中使用的词汇分为六类：标识符、关键字、运算符、分隔符、常量、注释符。

1. 标识符

标识符指在程序中使用的变量名、函数名、标号等。除库函数的函数名由系统定义外，其余函数名都由用户自定义。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:

abc, x1, student1, sum5

以下标识符是非法:

5q 以数字开头

x+y 出现非法字符+

-8x 以减号开头

bey-8 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 不限制标识符的长度, 但它受各种版本 C 语言的编译系统限制, 同时也受到具体机器的限制。如在某版本 C 中规定标识符前八位有效, 当两个标识符前八位相同时, 则会被认为是同一个标识符。

(2) 标识符要区分大小写, 大写字母与小写字母的含义是不相同的。如 NAME 和 name 是两个不同的标识符。

(3) 标识符是用来标识某个量的符号, 因此它的命名应尽量有相应的意义, 以便阅读理解, 起到“顾名思义”的作用。

2. 关键字

关键字又称为保留字, 是由 C 语言规定的具有特定意义的字符串。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、说明变量、函数或其他数据结构的类型。

auto	char	const	double	enum	extern	float
int	long	register	short	signed	static	struct
typedef	union	unsigned	void			

(2) 语句定义符

用于表示一个语句的功能。

break	case	continue	do	else	for
goto	If	return	switch	while	volatile

3. 运算符

运算符由一个或多个字符组成, 它与变量, 函数一起组成表达式, 表示各种运算功能, C 语言中包含了相当丰富的运算符。

4. 分隔符

分隔符有逗号和空格两种。逗号用于分隔各个变量, 主要用在类型说明和函数参数表中。空格多用于语句各单词之间作间隔符。在关键字和标识符之间必须要有一个以上的空格符作间隔, 否则将会出现语法错误。如若把“int a;”写成“inta;”则 C 编译器会把 inta 当成一个标识符处理, 其结果必然出错。

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。在第二章中将详细介绍。

6. 注释符

前面介绍过 C 程序的注释是“/*”和“*/”之间的内容, “/* ... */”为 C 语言的注释符。