

C 程序设计

孙凤美  主编



科学技术文献出版社
SCIENTIFIC AND TECHNICAL DOCUMENTATION PRESS

C 程序设计

主 编：孙凤美

副主编：李利萍 姜 伟 李明仑



科学技术文献出版社
SCIENTIFIC AND TECHNICAL DOCUMENTATION PRESS

· 北 京 ·

图书在版编目 (CIP) 数据

C程序设计 / 孙凤美主编. — 北京: 科学技术文献出版社, 2015. 9
ISBN 978-7-5189-0602-4

I. ① C… II. ① 孙… III. ① C语言—程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2015) 第 189446 号

C程序设计

策划编辑: 崔灵菲 责任编辑: 王瑞瑞 责任校对: 赵 媛 责任出版: 张志平

出 版 者 科学技术文献出版社
地 址 北京市复兴路15号 邮编 100038
编 务 部 (010) 58882938, 58882087 (传真)
发 行 部 (010) 58882868, 58882874 (传真)
邮 购 部 (010) 58882873
官 方 网 址 www.stdp.com.cn
发 行 者 科学技术文献出版社发行 全国各地新华书店经销
印 刷 者 北京九州迅驰传媒文化有限公司
版 次 2015 年 9 月第 1 版 2015 年 9 月第 1 次印刷
开 本 787×1092 1/16
字 数 348千
印 张 18.25
书 号 ISBN 978-7-5189-0602-4
定 价 46.00元



版权所有 违法必究

购买本社图书, 凡字迹不清、缺页、倒页、脱页者, 本社发行部负责调换

前 言

C语言是一种结构化语言，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；它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，其实现了将高级语言和低级语言的功能集于一体，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语言从诞生之日起就被广泛地移植到了各种类型的计算机上，从而受到了广大编程人员的喜爱。

现在许多大中专院校都开设了C语言课程，越来越多的计算机程序设计人员也把C语言作为入门语言，教育部考试中心在全国计算机等级考试的考试大纲中，也把C语言作为程序设计二级考试的可选语言之一、三级考试必考语言。

本书在编写过程中努力做到概念准确、叙述流畅、重点突出、例题实用性强、通俗易懂。

本书主要有以下特点：

1. 结构清晰、紧凑；
2. 精选理论，强化实践，突出技能；
3. 本着循序渐进的原则，先提出项目任务，引入涉及的知识和语法规则，然后通过实例分析加深理解，前导后续，最后对重点、难点进行总结；
4. 例题丰富，趣味性较强，全部上机调试通过；
5. 例题程序书写规范，读者通过学习和模仿，有利于养成良好的编程习惯；
6. 习题量大，针对性强。

不少人在学习C语言时，感到入门难，对很多问题的理解支离破碎，编者建议初学者一定要多读，反复研读，勤于思考，然后试着去写，举一反三，多读多写多上机调试，只有这样才能尽快掌握和运用C语言去解决实际问题。

本书项目一、项目二、项目三由孙凤美编写，项目四、项目五、项目六由李利萍编写，项目七、项目八由姜伟编写，其中项目四至项目八中的知识链接部分由孙



凤美编写，附录由李明仑编写，在编写过程中得到了领导和同仁们的大力支持和帮助，全书由孙凤美统稿定稿。在本书编写过程中邀请了潍坊北大青鸟华光照排有限公司软件技术总监殷建民、潍坊歌尔电子有限公司开发部经理郎卫东等一线软件编程人员参加编写，并担当主审，使本书的基础知识与实际工作要求紧密结合，例题全部能独立运行。

本书的编写和出版在各方面都得到了许多友人的支持和帮助，在此一并表示感谢。

由于作者水平有限，书中难免存在纰漏之处，敬请批评指正。

编 者
2015 年 7 月

目录 Contents

项目一	学生信息输入输出	1
任务 1	学生信息输入输出	1
1.1.1	任务描述	2
1.1.2	任务实现	2
1.1.3	任务分析	2
1.1.4	知识链接	3
任务 2	计算总分及平均分	20
1.2.1	任务描述	21
1.2.2	任务实现	21
1.2.3	任务分析	21
1.2.4	知识链接	22
任务 3	C 程序的编辑与运行	26
习题一		29
项目二	将学生成绩转换为等级	38
任务 1	输入学生成绩判断合法性	38
2.1.1	任务描述	39
2.1.2	任务实现	39
2.1.3	任务分析	39
2.1.4	知识链接	40
任务 2	将学生成绩转化为等级	50
2.2.1	任务描述	51
2.2.2	任务实现	51
2.2.3	任务分析	51
2.2.4	知识链接	52



选择结构程序举例	55
习题二	57
项目三 基于循环结构实现学生成绩统计	61
任务 1 统计全班一门课的总成绩及平均成绩	61
3.1.1 任务描述	62
3.1.2 任务实现	62
3.1.3 任务分析	63
3.1.4 知识链接	63
任务 2 统计全班四门课程的总成绩及平均成绩	73
3.2.1 任务描述	73
3.2.2 任务实现	73
3.2.3 任务分析	74
3.2.4 知识链接	74
循环结构程序举例	82
习题三	86
项目四 基于数组实现学生成绩管理	92
任务 1 输出一门课学生成绩	92
4.1.1 任务描述	93
4.1.2 任务实现	93
4.1.3 任务分析	94
4.1.4 知识链接	94
任务 2 学生名单的输入输出	100
4.2.1 任务描述	100
4.2.2 任务实现	100
4.2.3 任务分析	101
4.2.4 知识链接	101
任务 3 统计学生期末成绩总分及平均分	107
4.3.1 任务描述	107
4.3.2 任务实现	108
4.3.3 任务分析	110
4.3.4 知识链接	110
习题四	116

项目五 基于函数实现学生成绩汇总	120
任务 1 统计小组一门课程的总分及平均分	120
5.1.1 任务描述	121
5.1.2 任务实现	121
5.1.3 任务分析	122
5.1.4 知识链接	122
任务 2 统计小组若干门课程的总分及平均分	131
5.2.1 任务描述	131
5.2.2 任务实现	131
5.2.3 任务分析	133
5.2.4 知识链接	133
任务 3 输出排序后小组三门课成绩单	138
5.3.1 任务描述	139
5.3.2 任务实现	139
5.3.3 任务分析	141
5.3.4 知识链接	142
习题五	163
项目六 基于指针实现学生成绩输入输出	167
任务 1 使用指针输出学生成绩	167
6.1.1 任务描述	168
6.1.2 任务实现	168
6.1.3 任务分析	168
6.1.4 知识链接	169
任务 2 使用指针输出学生一门课的成绩	176
6.2.1 任务描述	176
6.2.2 任务实现	176
6.2.3 任务分析	177
6.2.4 知识链接	177
任务 3 用指针输出班级三门课成绩	185
6.3.1 任务描述	185
6.3.2 任务实现	185
6.3.3 任务分析	187
6.3.4 知识链接	187



习题六	194
项目七 基于结构体开发学生成绩管理系统	198
任务 1 利用结构体数组输入输出学生信息	198
7.1.1 任务描述	199
7.1.2 任务实现	199
7.1.3 任务分析	200
7.1.4 知识链接	200
任务 2 输出最高分学生信息	210
7.2.1 任务描述	210
7.2.2 任务实现	211
7.2.3 任务分析	212
7.2.4 知识链接	213
习题七	236
项目八 基于文件实现学生信息存储	240
任务 1 存储学生信息	240
8.1.1 任务描述	241
8.1.2 任务实现	241
8.1.3 任务分析	243
8.1.4 知识链接	243
任务 2 从文件存取学生成绩	249
8.2.1 任务描述	250
8.2.2 任务实现	250
8.2.3 任务分析	253
8.2.4 知识链接	254
习题八	267
附录 A 常用 ASCII 码字符对照表	270
附录 B 运算符的优先级、含义及结合性	272
附录 C C 语言中的关键字	274
附录 D C 语言中的头文件	275
附录 E C 语言中的库函数	277
参考文献	284

项目一

学生信息输入输出

学习情境

计算机应用技术班进行了一次考试，需要设计一个程序，实现下列功能：

1. 按指定格式输入输出学生成绩；
2. 计算学生的总成绩及平均成绩。

学习目标

了解 C 语言的基本结构，熟悉 Win-TC 开发程序的流程；

掌握 C 程序的基础知识；

掌握 C 语句及赋值语句的概念；

掌握各种数据类型的输入输出方法。

► 任务 1 学生信息输入输出

知识目标	熟练区分数据类型 学会顺序结构程序执行过程 学会不同数据类型的输入输出方法
能力目标	学会 Win-TC 软件的安装 调试运行简单 C 程序
素质目标	培养学生对新事物的接受能力 培养学生自我学习的能力
重点内容	标识符、常量、变量 数据类型 输入输出语句
难点内容	数据类型划分，输入输出格式



1.1.1 任务描述

计算机应用技术班进行了一次考试，要求设计一个程序，实现下列功能：

- (1) 新建一个文件 p1_1.c;
- (2) 按格式要求输入 2 个学生成绩并输出。

1.1.2 任务实现

```
#include <stdio.h>
main ()
{
    int x, y; /* 定义 2 个整型变量 x 和 y*/
    printf("please input 2 student achievement:"); /* 输出提示字符串 */
    scanf("%d%d", &x, &y); /* 从键盘接受 2 个值送给变量 x 和 y*/
    printf("2 student achievement is:");
    printf("x=%d, y=%d\n", x, y); /* 输出变量 x 和 y 的值 */
    getch();
}
```

程序执行结果如图 1.1 所示。



图 1.1 任务 1 执行结果

1.1.3 任务分析

(1) 这是一个最简单的 C 程序，学习 C 语言就由此开始吧！

(2) C 程序是由函数组成的，函数就是相对独立的可以完成特定功能的程序段。本程序就是由一个称为 main 的函数构成的。其中 main 是函数的名字，函数名后都要有一对圆括号，用来写函数参数的，本程序的 main 函数没有参数所以不写，但

圆括号不能省略。

(3) 一个完整的程序必须有一个 `main` 函数，称为主函数，程序总是从 `main` 函数开始执行的，也就是说，`main` 函数是程序的入口。

(4) `main()` 后面花括号内的部分称为函数体。本程序中的函数体只由一个语句组成。一般情况下函数体由“说明部分”和“执行部分”组成。说明部分的作用是定义数据类型；执行部分给出操作命令。

(5) C 程序的每一条基本语句都以“;”结束。

(6) 可以用 `/* ... */` 对 C 程序的任何部分作注释，用来解释该条语句或该段程序的含义或作用，只是为了帮助程序阅读者更好地理解程序中有关部分内容而写的。

(7) 首先应该考虑程序中要用到的数据的个数，显然本例中要用到两个数，即 `x` 和 `y`，所以先定义两个变量。语句 `int x, y;` 的作用就是定义两个变量，名字分别为 `x`、`y`，类型都是整型。`int` 表示整型，是 C 语言的关键字（见知识链接）。

(8) 库函数调用 `scanf("%d%d", &x, &y);` 用来实现从键盘接收用户输入的数据，并送到变量 `x` 和 `y` 所对应的地址单元中。这也意味着，变量 `x` 和 `y` 的值就是输入的值。例如从键盘输入 95 和 88，那么变量 `x` 的值就是 95，变量 `y` 的值就是 88。`&` 表示取地址（也称为取地址运算符），`%d` 用来限制输入数据的格式是整数（见知识链接）。

(9) 库函数调用 `printf` 用于输出结果到屏幕上（见知识链接）。

(10) `#include` 语句是编译预处理语句，其作用是将有双引号或尖括号括起来的文件内容代替这行 `#include` 命令，也就是使该文件的内容被整个地调到 `main` 函数的前面。“`.h`”是“头文件”的后缀，输入输出函数一般需要使用 `#include` 语句将包含输入输出函数说明信息的头文件“`stdio.h`”包含到源文件中。

1.1.4 知识链接

(1) 简单 C 语言程序的构成

通过任务 1 的程序可以看出，简单 C 程序结构如下：

```
#include <stdio.h> /* 头文件 */
main () /* 函数名 */
{
    /* 函数体 */
    变量声明部分；
    执行语句部分；
}
```

C 语言程序严格区分大、小写，相同字母的大、小写 ASCII 码不同，代表不同



的标识符，C 语言的关键字、库函数和基本语句都是用小写字母表示的。

(2) 标识符

在 C 语言中，标识符是用来标识变量名、符号常量名、函数名、数组名、文件名等的有效字符序列。简言之，标识符就是以字母、下划线开头的字母、数字和下划线的组合。

C 语言中的标识符可以分为以下三大类。

1) 关键字

关键字又称保留字，是一种系统预先定义的、具有特殊意义的标识符。用户不能重新定义关键字，也不能把关键字定义为一般的标识符，如关键字不能作变量名、函数名等，所有的关键字均用小写字母。

C 语言的关键字有类型标识符、控制流标识符、其他标识符等。

①类型标识符

为了对数据进行存储和处理，C 语言把数据分成了多种类型，每一种类型的数据各有其特点，在数据的表示、存储、运算等方面各自具有一些不同的特性，对于一个具体问题，要设计它的实现算法，必然要考虑数据的类型。

C 语言的数据类型分为基本类型和构造类型，下列关键字用于定义数据变量的类型：

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

②控制流标识符

C 语言语句中专用的标识符：

```
goto  return  continue  break  if  else  for  do  while  switch  case  default
```

③其他标识符

sizeof 用于计算数据类型所占的字节数，也称为求字节数运算符。

需要注意的是 C 语言中的关键字都是小写字母。

2) 预定义标识符

这些标识符在 C 语言中都有特定的含义，如库函数的名字和预编译处理命令等。C 语言语法允许这类标识符另作他用，但这将使这些标识符失去系统规定的原意，鉴于目前各种计算机系统的 C 语言都一致把这类标识符作为固定的库函数名和预编译处理中的专门命令使用，因此为了避免误解，建议用户不要把这些预定义标识符另作他用。

①预编译处理命令

这些编译指令通知编译器在编译工作开始之前对源程序进行某些处理，称为编译预处理。编译指令都是用“#”引导。C 语言编译预处理主要包括宏定义、文件包

含和条件编译：

```
#define #include #undef #ifdef #endif
```

②库函数

系统定义的函数，如输入输出函数、数学函数、字符处理函数等，例如 printf、scanf、fabs、sin、strcat、strlen 等。

3) 用户标识符

由用户根据需要定义的标识符称为用户标识符。一般用来给变量、函数、数组或文件等命名。程序中使用的标识符除要遵循命名规则外，还应注意做到“见名知义”，即选用具有相同含义的英文单词或汉语拼音，以增加程序的可读性，如 name、week、number、area 等。

如果用户标识符与关键字相同，程序在编译时将给出出错信息。如果与预定义标识符（预处理标识符或库函数名）相同，系统并不报错，只是该预定义标识符将失去原来含义，代之以用户确认的含义，或引发一些运行时的错误，所以尽量避免使用预定义标识符作为用户标识符。

C 语言规定，所有的标识符必须满足以下规则：

- ①标识符中打头的字符必须是字母（a～z，A～Z）或下划线（_）；
- ②标识符中的其他字符只能是字母、数字（0～9）或下划线；
- ③大小写字母代表不同的标识符，不能混用替代；
- ④用户标识符不能和 C 语言中的关键字相同；

⑤在 Turbo C 中系统能识别的标识符的最大长度是 32，所以一般情况下标识符的长度不宜超过 32 个字符。

下面列举几个合法和非法的用户标识符的例子：

合法标识符

sum

_a123

LONG

S1_no

Abs

非法标识符

5sum（以数字开头）

a+3（含有特殊字符+）

long（long 是 C 语言中的关键字）

s1 no（标识符中不能含空格）

ab\$（标识符中不能含字符\$）

(3) 常量

常量是指在程序运行过程中，其值保持不变的量。

常量在 C 语言中出现的形式一般有两种：直接常量和符号常量。从其字面形式即可判断出的常量称为字面常量或直接常量；用一个标识符来代表一个常量，则称



之为符号常量。

1) 直接常量

C 语言中的直接常量有以下四种。

①整型常量

C 语言中整型常量可以用十进制、八进制和十六进制来表示。

十进制整型常量由数字 0~9 和正 (+) 负 (-) 号组成, 如 68、0、-19、+10。

八进制整型常量由数字 0~7 组成, 在常量前加数字 0 (注意不是字母 o), 如 010、012、017, 它们分别代表十进制的 8、10、15。

十六进制整型常量由数字 0~9 和字母 a~f (或 A~F) 组成, 在常量前加 0x (或 0X), 如 0x10、0X12、0x1f、0XAB, 它们分别代表十进制的 16、18、31、171。

整型常量后可以用 u 或 U 说明为无符号整型数; 用 l 或 L 说明为长整型数。

②实型常量

实型常量分小数形式和指数形式两种表示形式。

小数形式由小数点 (.)、数字 (0~9) 和正 (+) 负 (-) 号组成, 如 31.6、-7.48。

指数形式则以 “e” 或 “E” 后跟一个整数来表示以 10 为底的幂数, 如数学中的 1.12×10^6 用 C 语言表示为 1.12e6, 12×10^4 用 C 语言表示为 12e4;

C 语言语法规定: 指数形式表示实型常量字母 e (或 E) 前面必须有数字, 且 e (或 E) 后面的指数必须为整数 (可加正负号)。注意, e (或 E) 的前后及数字之间均不得插入空格。

如 E4、.E6、12E3.6、1.2 e2、1.1e 6、2.1 e 4 均不合法。

指数形式的科学计数法要求: e (或 E) 前面的数字小数点前只能保留一位非 0 数字, 如 1.12e6 是科学计数法的实型常量, 而 12e4 不是科学计数法的实型常量, 但是指数形式的实型常量。

③字符常量

单引号括起来的一个字符, 有且仅有一个字符, 如 'a'、'c'、'8'、'\$'、' ' (表示空格, 以下同)。

字符常量还包括转义字符常量, 转义字符常量以一个反斜线 “\” 开头, 后跟一个特定的字符, 用来表示某一个特定的 ASCII 字符, 这些字符常量也必须括在一对单引号内。如:

'\n'	换行
'\r'	回车
'\b'	退格
'\t'	制表 (横向跳格)
'\''	单引号 (单撇号)

"'"	双引号（双撇号）
"ddd"	1~3 位八进制数所代表的 ASCII 码字符
"xhh"	1~2 位十六进制数所代表的 ASCII 码字符
"f"	走纸换页
"\""	反斜杠字符

④字符串常量

双引号括起来的一串（0~n 个）字符，如 "china"、"ligong"、"123ong"、"1234"、"\$%*&"、""（空串）、" "（空格串）。

思考：空串与空格串的区别。

2) 符号常量

用一个符号名（用户标识符）来代表一个常量，这个符号必须在程序中预先用 define “指定”。看下面的例子：

例 1.1 计算圆的面积。

程序分析：圆的面积 $s = \pi r^2$

程序代码：

```
#define PI 3.14
main ()
{
    float r, s; /* 定义圆的半径变量 r，圆的面积变量 s */
    scanf ("%f", &r); /* 输入圆的半径 */
    s=PI*r*r; /* 计算圆的面积 */
    printf ("s=%f\n", s); /* 输出圆的面积 */
    getch ();
}
输入：10
输出：s=314.000000
```

在这个程序中，用 #define 命令设置 PI 这个标识符为符号常量，即标识符形式的常量，它的值在程序运行期间不能改变，这里的 PI 在随后的程序中代表常量 3.14。

习惯上，符号常量名用大写，变量名用小写，以示区别。使用符号常量可以提高程序的可读性，便于修改，具有以下优点：

①望文生义

定义符号常量名应尽量考虑“见名知义”。

②修改方便，一改全改

在需要改变一个常量的值时，能做到“一改全改”。例如，上面的程序做如下修改：

```
#define PI 3.1415926
```

则在程序中出现的所有 PI 都代表 3.1415926。

注意：符号常量是常量，不同于变量，它的值在其作用域内不能改变。如果在程序中再用赋值语句对 PI 进行赋值是错误的，如 PI=10。

(4) 变量

在程序运行过程中，其值可以改变的量称为变量。一个变量应该有一个名字作为标识，变量名的命名必须遵循用户标识符命名规则并应考虑“见名知义”的原则。如用“sum”代表“总和”、用“name”代表“姓名”等。

实际上变量在其作用域内，在内存中占据一定的存储单元，在该存储单元中存放变量的值。在程序中使用一个变量之前，先要对它进行定义，即先定义后使用。

在 C 语言中所用到的变量都必须指定其数据类型。指定了数据类型，也就定义了变量在计算机中内存中所占的空间字节数。

C 语言中常用的数据类型如图 1.2 所示。

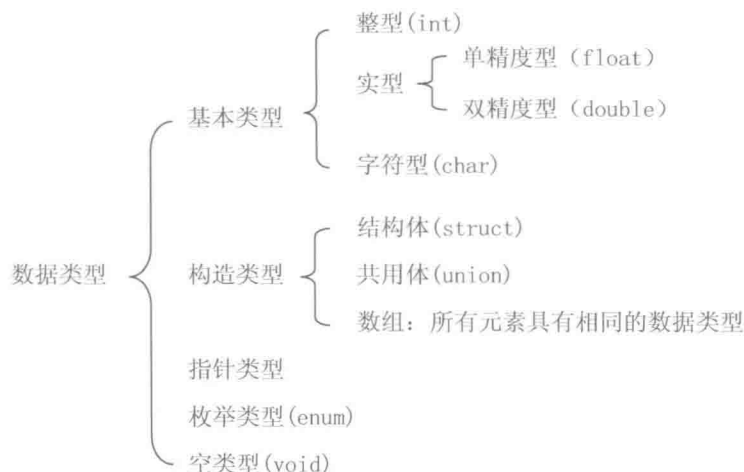


图 1.2 C 语言数据类型

变量定义的一般形式是：

类型名 变量名列表；

如 int i, j, k；

float r, s；

char ch；