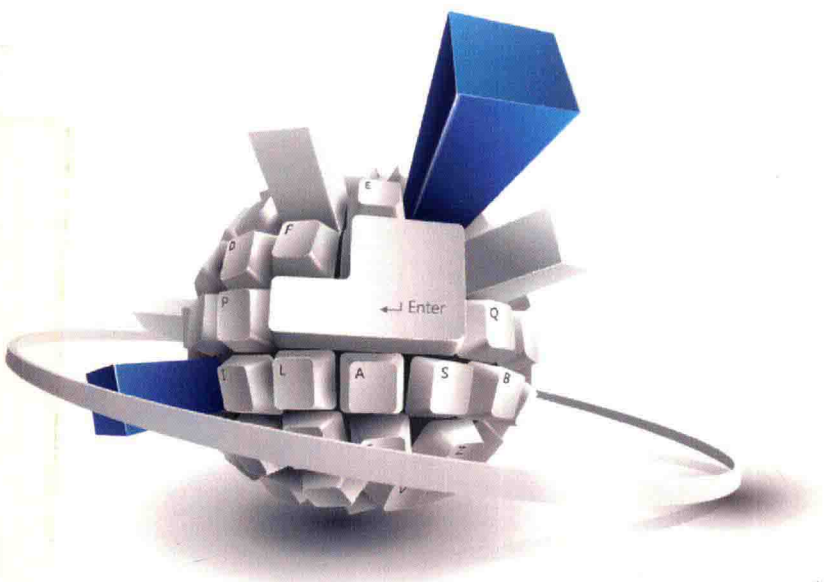
计算机类专业人才培养内涵建设项目系列教材

C语言程序设计与实训教程

下册

主编 胡 悦



WUHAN UNIVERSITY PRESS
武汉大学出版社

计算机类专业人才培养内涵建设项目系列教材

C语言程序设计与实训教程（下册）

主　编　胡　悦
副主编　严　岚
参　编　周　蒨

WUHAN UNIVERSITY PRESS
武汉大学出版社

图书在版编目(CIP)数据

C语言程序设计与实训教程. 下册/胡悦主编. —武汉:武汉大学出版社,2016.8

计算机类专业人才培养内涵建设项目系列教材

ISBN 978-7-307-18567-8

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

中国版本图书馆CIP数据核字(2016)第203417号

责任编辑:张 欣 曲生伟 责任校对:谢丽娟 装帧设计:吴 极

出版发行:**武汉大学出版社** (430072 武昌 珞珈山)

(电子邮件:whu_publish@163.com 网址:www.stmpress.cn)

印刷:虎彩印艺股份有限公司

开本:787×1092 1/16 印张:11.5 字数:294千字

版次:2016年8月第1版 2016年8月第1次印刷

ISBN 978-7-307-18567-8 定价:29.00元

计算机类专业人才培养内涵建设项目系列教材
编写委员会

前　言

C语言具有功能丰富、表达能力强、使用灵活方便、目标程序效率高、可移植性好等特点，其广泛应用于事务处理、科学计算、工业控制及数据库等领域。

根据高职高专的课程设置体系，C语言被推荐为计算机各专业学生的必修课程。因此，编者所在的学院在中高职贯通中设置了本课程。然而，与其他程序设计语言相比，C语言学习难度比较大，尤其对于中高职贯通计算机类专业中职阶段学生，学习C语言是一件难度非常大的事情。因此，提供一本适合中高职教学的教材是关键。为此，编者根据中高职贯通计算机类专业学生的具体情况，依据多年C语言的教学经验，编写了本书。

本书具有以下几个方面的特色：

1. 门槛较低，不要求具备程序设计语言的基础知识。

本书从程序设计的最基础知识入手，例题由浅入深设计，经典算法的来龙去脉讲解清楚，读者可以不具备程序设计语言的入门知识。

2. 以场景实例为先导，尽可能清晰易懂，并带有趣味性。

充分考虑中高职院校学生的学习特点、学习基础和接受能力，以工作场景实例为先导，突出职业技能培养。结合大量例题、课堂实践题、有趣的能力拓展题，激发学生的学习兴趣。

3. 深入浅出，通俗易懂，启发思维，教学互动。

结构清晰、衔接自然。实训题结合前导所述的知识点，深入浅出。总体根据知识、能力、结构要求，按提出问题、分析问题、解决问题的步骤完成程序设计。

4. 强化实践，重视应用，例题丰富。

重视实践教学环节，结合实际应用，每项任务训练中都列举了丰富的实例。通过例题，学生不仅仅可以掌握C语言的语法、语句，更重要的是可以具备编程解决实际应用问题的能力。

5. 强调良好的编程习惯。

给出合理的编程风格，指出要求采用统一的编程风格的重要性。

本书由胡悦担任主编，严岚担任副主编，周蒨担任参编。其中，任务6、任务7、任务9、任务10和任务11由胡悦编写，任务8由严岚编写。周蒨作为企业工程师为整本书的设计及内容安排提供了很多有益的意见和建议。全书由胡悦统稿、审核。

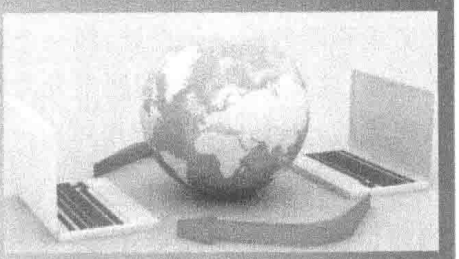
由于编者水平有限，书中疏漏及错误之处在所难免，敬请读者提出意见和批评指正，以便做进一步的完善。

编　者

2016年6月

目　录

任务6
函数设计能力训练



学习目标

在本任务中，将学习如下内容：

- 模块化程序设计；
- 创建函数、调用函数；
- 函数之间信息的传递。

重点难点

- 使用函数对程序进行模块化构建；
- 编写和调用自己的函数(即函数的递归调用)。

提纲

通过上册任务5,学习了一维数组、二维数组及排序算法设计,尤其在排序算法训练中,由于程序比较复杂,降低了程序的可读性,不易于程序的修改和维护。本任务主要是学习利用函数实现程序的模块化设计,即一个较大的程序分为若干个程序模块,每一个模块实现一个特定的功能,这样将复杂的问题划分为较为简单的多个子问题。在C语言中,模块的功能是由函数完成的,一个C语言程序可由一个主函数和若干个函数构成。下面就开始学习使用函数设计程序。

6.1 函数调用训练

◆ 工作场景一导入

(1)问题描述:某商场为了使顾客能够更加方便地进行消费,采用以下结账方式:使用现金和使用银联卡。同时,使用现金或银联卡的用户又分为会员用户和普通用户。现已知使用现金付款买单时可享有9.8折优惠,使用银联卡不享有打折优惠;会员用户付款买单时可享有9折优惠,普通用户不享有打折优惠。请你为商场编写一个程序,模拟该商场收银场景。

(2)问题分析:可以使用函数分别实现不同的功能,然后调用相应函数来完成相关任务。对程序进行结构化设计。

(3)演示执行:

```
请输入购物金额: 1000
请输入买单方式(1: 现金, 2: 银联卡) 1
请输入客户类型(1: 普通用户, 2: 会员用户) 2
请出纳现金: 882元
```

6.1.1 结构化程序设计

结构化程序设计方法的基本思路:对一个复杂问题进行分阶段求解,每个阶段处理的问题都控制在容易理解和处理的范围内。

具体采用以下方法得到结构化的程序:(1)自顶向下;(2)逐步细化;(3)模块化设计;(4)结构化编码。

这种设计方法的过程是从问题的总体目标开始,抽象低层的细节,先专心构造高层的结构,然后一层一层地分解和细化。这样,设计者能把握主题,高屋建瓴,避免一开始就陷入复杂的细节中,使复杂的设计过程变得简单、明了,结果也容易做到正确、可靠。结构化程序设计,

强调程序设计风格和程序结构的规范化，提倡结构的清晰，便于验证算法的正确性，减少程序出错的机会，提高程序的可靠性，保证程序的质量。

上册所学的顺序结构、选择结构和循环结构是结构化程序设计的三种基本结构。

6.1.2　函数的概念

结构化程序设计以模块化设计为中心，把一个较大的程序分为若干个程序模块，每个模块分别用来实现一个特定任务的功能。在不同的程序设计语言中，模块的实现方式不同。在C语言中，程序模块的实现方式是函数。因此，函数本质上是一段程序，该段程序可以被其他函数调用，以完成规定的功能。

一个C语言程序可由一个主函数main()和若干个函数构成，如图6-1所示。main()函数可以调用其他函数，其他函数也可以调用另外的函数，同一函数也可以被一个或多个函数调用一次或多次，但不能调用main()函数。

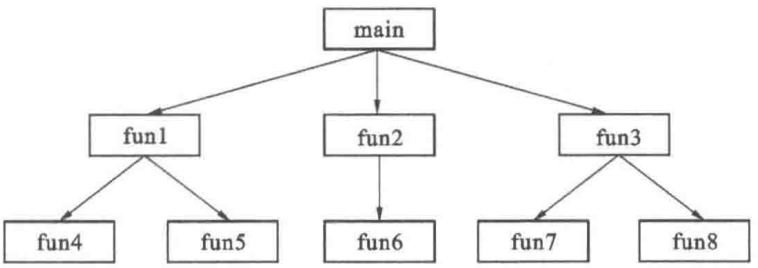


图6-1　函数调用关系图

如果从用户使用的角度来分，C语言中的函数可以分为标准函数和用户函数两种。标准函数即库函数，是C语言自身定义好的函数，用户可以直接调用；用户函数是程序员在程序中自己定义的函数，可以用来解决某些常用问题。如果从函数的形式上来分，函数可以分为无参数函数和有参数函数两种。无参数函数是指定义函数时没有参数，函数调用时无数据传递；有参数函数是指定义函数时有参数，函数调用时有数据传递。

使用函数的优点如下：

(1)将程序解决的大问题分解成若干个小问题，便于处理。

(2)将每个小问题编写成独立的函数存放于函数库中，随时供程序员调用，避免重复劳动。

(3)以函数为基本单位构成的程序，简洁、清晰，便于调试与维护。

(4)每个函数之间具有相对独立性，不同模块之间可以由不同的人来编写，缩短了编写程序的时间。

6.1.3　函数的定义

函数定义的一般形式为：

```
函数类型 函数名(形式参数列表)
{
    声明部分
    执行部分
}
```

通常一个函数由首部和函数体两部分组成。函数的首部，即函数的第一行，包括函数类型、函数名、形式参数列表，其中形式参数(简称形参)列表可以包含若干个函数参数名、参数类型，也可以没有形式参数。例如，例 6-1 中的 max 函数的首部为：

```
int     max   (  int    x,   int    y  )
 ↓       ↓        ↓       ↓       ↓       ↓
函数类型  函数名   形参类型  形参名  形参类型  形参名
```

程序 函数的定义使用示例

【例 6-1】 求 2 个数 x 和 y 中大者的函数。

程序分析：定义函数 max，形参分别为 x、y，在函数体中求解 x 与 y 的大者，并将大者的值返回。因此，函数类型与 x、y 的数据类型一致。

maxxy.c：

```
1  #include <stdio.h>
2  //由于要返回x、y中大者的值，函数类型与x、y的类型一致.
3  //函数体最后必须有return语句
4  int max(int x,int y)
5  {
6      int z;
7      if(x>y)
8          z=x;
9      else
10         z=y;
11     return (z);
12 }
```

函数定义用法小贴士

(1)函数体由一对花括号{}括起来，一般包括声明部分和执行部分两部分。

● 声明部分：定义本函数中使用的变量、数组、指针变量等，如例 6-1 中的 int z；另外，要对所调用的函数进行声明。

● 执行部分：由解决问题的若干条语句组成，是程序的核心部分。

(2)在某些情况下也可以没有声明部分，或者同时没有声明部分和执行部分，即空函数。如

```
fempty()
{    }
```

它是一个空函数，不做任何操作，但语法是合理的。

(3)如果函数没有返回值，函数类型为 void。

【例 6-2】 计算某个数 m 阶乘的函数。

程序分析：计算 m 的阶乘，m 作为形参用于传递参数，阶乘的结果可能比较大，定义为 long 型，由于函数需要返回阶乘结果，函数类型也为 long 型。

factor.c:

```
1   //求m的阶乘，m作为形参用于传递参数
2   //阶乘的结果可能比较大，定义为long型
3   long factor(int m)
4   {
5       int i;
6       long f=1;
7       for(i=1;i<=m;i++)
8           f=f*i;
9       return (f);
10  }
```

6.1.4　函数的调用

一个 C 语言程序总是从 main 函数开始执行，而不论 main 函数在整个程序中的位置如何。函数则需要被调用后才可以执行，函数可以由 main 函数调用，也可以由一般函数调用，甚至可以调用自己。

函数调用的一般格式为：

函数名(实际参数列表)；

函数调用用法小贴士

实际参数(简称实参)列表必须与定义函数时的形参列表个数相等，类型一致。函数调用时，要把实参数据按照一一对应的顺序传递给形参。如果实参列表包含多个实参，则各参数之间用逗号隔开；如果是调用无参函数，则没有实参列表，但是圆括号不能省略。

按照函数出现在程序中的位置，函数调用可以分为以下三种形式。

1. 函数语句

把函数作为一条语句出现在程序中进行调用，如果函数无返回值，则函数定义时，一般将函数定义为 void 类型。

函数语句使用示例

【例 6-3】 用主函数调用例 6-1 中的函数。

程序分析：在 main 函数中需要有两个数据传递给 max 函数。

callmaxxy.c:

```
#include <stdio.h>
int max(int x,int y)
{
    int z;
    if(x>y)
        z=x;
    else
        z=y;
    printf("\n  max=%d\n",z);
}
main()
{
    int a,b,c,m;
    printf("\n  请输入要比较的两个数: ");
    scanf("%d%d",&a,&b);
    //调用max函数，将a,b的值分别传递给x,y
    max(a,b);
}
```

程序运行结果为：

```
请输入要比较的两个数: 89 76

max=89
```

编程提示

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，在另一些函数之后。

2. 函数表达式

函数出现在一个表达式中，则这个表达式称为函数表达式，此时函数必须返回一个确定的值参与表达式的计算。

函数表达式使用示例

【例 6-4】 以函数表达式形式调用例 6-1 中的函数。

程序分析：将函数返回的值赋给变量。

callmaxxy2.c:

```
#include <stdio.h>
int max(int x,int y)
{
    int z;
    if(x>y)
        z=x;
    else
        z=y;
    return (z);
}
main()
{
    int a,b,c,m;
    printf("\n请输入要比较的两个数：");
    scanf("%d%d",&a,&b);
    //调用max函数，将a,b的值分别传递给x,y
    //将max返回的值赋给变量m
    m=max(a,b);
    printf("max=%d\n",m);
}
```

程序运行结果同例 6-3。

3. 函数参数

函数调用作为一个函数的实参，也就是函数的返回值作为函数再次调用的实际参数，实际上是函数表达式形式调用的一种特殊情况。

函数参数使用示例

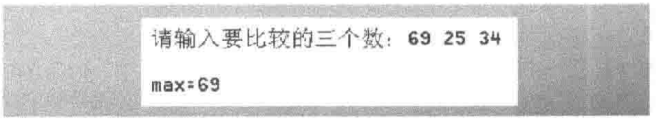
【例 6-5】 输入 3 个整型数据，求出其中的最大数。

程序分析：设计求 2 个数中大者的函数 max，则 max 函数有 2 个形参，调用 max 函数先求出前 2 个数中的大者，再次调用 max 函数，求出第 3 个数与前面求出的前 2 个数中的大者的最大值。

callmax3.c:

```
#include <stdio.h>
int max(int x,int y)
{
    return(x>=y ? x:y);
}
main()
{
    int a,b,c,nmax;
    printf("\n请输入要比较的三个数：");
    scanf("%d%d%d",&a,&b,&c);
    //max(b,c)返回b,c中大者的值，该值继而与a比较
    nmax=max(a,max(b,c));    /*二次调用函数*/
    printf("max=%d\n",nmax);
}
```

程序运行结果为：



函数调用用法小贴士

在一个函数调用另外一个函数时，需要注意以下几个条件：

- 被调用的函数必须是已经存在的。
- 如果使用库函数，还应在文件的开头用#include命令标注。
- 如果被调用的函数的定义出现在主调函数之后，则应在主调函数之前或在主调函数的函数体声明部分，对被调用函数加以声明。对函数的声明也称为函数原型说明。

函数声明的一般形式为：

函数类型 函数名(参数1的类型，参数2的类型，…)；

如 int max(int ,int)；

由此可见，函数声明和函数定义虽然在形式上很相似，但二者在本质上存在差异：

- 函数的声明是对编译系统的一个说明，末尾以分号结束，不含具体操作；
- 在程序中函数的定义只能有一次，而函数的声明可以有多次。

6.1.5 函数的返回值

函数调用后的结果称为函数的返回值，通过返回语句返回到主调函数中。

函数返回语句的一般格式如下：

return(表达式)

或 return 表达式；

其功能是把表达式的值返回到主调函数中。

return语句用法小贴士

return语句使用说明：

(1)return语句返回值的类型必须与函数定义的类型一致。当不一致时，以函数类型为准。

(2)一个子函数中可以包含多个return语句，但是一个子函数的调用，只有一个return语句被执行。当执行return语句后，就退出子函数返回到主调函数中。

(3)如果被调用函数中无return语句，并不带回一个确定的函数值，而是带回一个不确定的、用户不需要的函数值。应尽量避免此种情况。

(4)为了明确表示不带回值，一般定义为void类型。例如，定义函数为：void max(int x,int y)。

【课堂实践题 6-1】

编写一个计算圆柱底面积和体积的程序，其中求圆柱底面积的函数命名为 area()，求圆柱体积的函数命名为 volume()，圆柱的底面半径 r 和高 h 由用户通过键盘输入，体会函数定义、调用、声明和注释的使用。

cylinder.c:

```c
/**********************************************
* 文件名: cylinder.c
* [illegible]
* [illegible]
* [illegible]
**********************************************/
#include <stdio.h>
//[illegible]
float area(float r);
//[illegible]
float volume(float r,float h);
main()
{
    float r,h,s,v;
    printf("\n请输入圆柱的底面半径r和高h: ");
    scanf("%f%f",&r,&h);
    s=area(r);
    v=volume(r,h);
    printf("\n该圆柱的底面积为: %f,体积为: %f\n",s,v);
}
/*[illegible]*/
float area(float r)
{
    return 3.14*r*r;
}
/*[illegible]*/
float volume(float r,float h)
{
    return area(r)*h;
}
```

文件头部注释

单行注释

多行注释

程序运行结果为：

```
请输入圆柱的底面半径r和高h: 3.0 4.0

该圆柱的底面积为: 28.260000,体积为: 113.040000
```

编程提示

注释的原则是有助于对程序的阅读理解，在该加的地方应加入，注释不宜太多也不能太少，注释语言必须准确、易懂、简洁。

函数声明与定义使用小贴士

函数声明就是函数定义的头部，比较简略。函数是用来调用的，如果函数定义写到调用的位置后面，执行到调用位置，后面根本没执行，就找不到该函数，程序会报错，这时需要在前面加声明，表示有这个函数。但是，如果先写函数体，后调用，就不需要加声明了。

返回工作场景一

问题分析:可以对场景一进行如下结构化设计。

(1)设计两个函数,分别处理现金支付和银联卡支付。注意:两个支付函数中需要考虑用户的身份,才能正确计算打折率。

(2)两个函数均需要传递两个参数,price 表示总的消费金额,type 表示用户类型。

cash.c:

```
#include <stdio.h>
int cashproc(int price, int type)
{   if (type==1)
        return price*0.98;
    return price*0.9*0.98;
}
int cardproc(int price,int type)
{   if (type==1)
        return price;
    return price*0.9;
}
main()
{   int manner,type,sum;
    printf("\n请输入购物金额: ");
    scanf("%d",&sum);
    printf("\n请输入买单方式(1: 现金, 2: 银联卡) ");
    scanf("%d",&manner);
    printf("\n请输入客户类型(1: 普通用户, 2: 会员用户) ");
    scanf("%d",&type);
    switch(manner)
    {   case 1:
            printf("\n请出纳现金: %d元", cashproc(sum,type));
            break;
        case 2:
            printf("\n银行卡扣除: %d元", cardproc(sum,type));
            break;
        default:
            printf("\nError!!!\n"); break;
     }
}
```

6.2 数组作为函数参数调用训练

工作场景二导入

(1)问题描述:一个班级有多名学生(假设不超过 30 名),请输入平时成绩和期末考试成绩在总评成绩中所占的百分比,然后按百分比计算每个学生的总评成绩,计算结果四舍五入后取整。将实现此功能的程序段封装成函数并调用。

(2)问题分析:不超过 30 名学生的成绩值(平时成绩、期末考试成绩、总评成绩)最多需要用一个 30 行 3 列的二维数组 cj[30][3]存放。每个学生的总评成绩都由平时成绩和期末考试成绩按照一定的比例组成,可以定义一个函数来对数组进行处理。请问数组怎样作函数的参数?

(3)演示执行: