普通高等教育计算机规划教材

C程序设计实践教程

刘燕君　刘振安　孙晓艳　编著

机械工业出版社
CHINA MACHINE PRESS

普通高等教育计算机规划教材

C 程序设计实践教程

刘燕君　刘振安　孙晓艳　编著

机械工业出版社

本书是《C 程序设计》一书的配套实践教材，但又自成体系，可以单独使用。本书重点介绍如何使用目前流行的 Borland C++ 3.1 和 Visual C++ 6.0 集成环境，编辑、编译、调试和运行 C 语言程序，以便读者尽快地熟练掌握集成环境的使用。通过使用集成环境加强编程实践，可帮助读者加深对理论知识的理解。

本书每章均首先介绍该章的重点和难点，然后对习题进行解答，最后给出教材中实践题的解题思路和参考方法。为了保证程序的结构化设计质量，尤其是理解大程序的设计方法，本书单列一章讨论程序的测试方法及测试用例的设计问题。

本书重在培养读者的应用技能，特别适合作为自学教材及工程技术人员的参考书。

图书在版编目（CIP）数据

C 程序设计实践教程 / 刘燕君，刘振安，孙晓艳编著. —北京：机械工业出版社，2008.10
（普通高等教育计算机规划教材）
ISBN 978-7-111-25053-1

Ⅰ. C… Ⅱ. ①刘…②刘…③孙… Ⅲ. C 语言—程序设计—高等学校—教材
Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2008）第 135203 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：唐德凯
责任印制：洪汉军

北京振兴源印务有限公司印刷厂印刷

2009 年 1 月第 1 版 • 第 1 次印刷
184mm×260mm • 11.5 印张 • 282 千字
0001—5000 册
标准书号：ISBN 978-7-111-25053-1
定价：21.00 元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：（010）68326294 68993821
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379753 88379739
封面无防伪标均为盗版

前　言

本书以培养读者的C语言编程能力为主线，并通过实践环节加强实际动手能力的训练。主要特点如下：

1）每章均首先介绍该章的重点和难点以及学习方法，有的还结合典型例题进行解析。

2）对每章的习题进行解答，有的习题给出多种解法以便给读者更多的启发。

3）详细给出教材中实践题的解题思路和参考编程实现方法，并给出扩充的实践练习题以便读者自己练习。

4）为了帮助读者熟悉使用 Visual C++ 6.0 来调试运行C程序，尤其是设置断点和跟踪程序的方法，在第2章简要介绍了基本的调试命令。

5）为了保证程序的结构化设计质量，尤其是理解大程序的设计方法，本书在第7章讨论了程序的测试方法及测试用例的设计问题，并结合实践题，介绍头文件的编制、多个C 语言文件及工程文件的编制等方法，以培养读者的实际应用能力。

本书由主教材《C 程序设计》的编者之一刘燕君主编，孙晓艳参加了本书的习题解答部分的编写工作，全书由刘振安统稿。定稿后由刘燕君和孙晓艳验证了全部程序，以保证程序的正确性。习题的解答方法并不惟一，虽然有时给出几种解法，但未必全面。

对于书中存在的错误和不妥之处再所难免，请读者予以指证。

刘振安

2008 年 5 月

目　录

第1章　C程序设计基础实践

本章要求学生熟悉如何编辑、编译、调试和运行程序，以便为以后的边学边练打下良好的基础。本章给出两种典型集成环境的使用方法，并结合一个实例，演示C语言编程的全过程。

1.1　重点和难点

本章的重点是熟悉如何编辑、编译、调试和运行一个实际的程序。

著名计算机科学家沃思提出“数据结构+算法=程序”的思想，大大促进了程序设计的发展。由于软件产业迅速发展，产生了“软件工程”学科，使人们进一步认识到“程序设计方法”的重要性。编程是算法的具体实现，所以程序离不开编程。过去的编译环境简单，都是命令行方式，很容易学习。目前的编译环境都朝集成环境和可视化编程方向发展，而且还实现了一定的自动化功能。由于功能强大，使其具有一定的复杂性。典型的是Visual C++ 6.0，提供了MFC类库，实现许多Windows自动编程功能。因此，用好这些语言工具和环境，是提高编程效率的因素之一。目前认为可以用下面的公式表示程序：

数据结构+算法+程序设计方法+编程工具=程序

由此可见，一个程序设计人员应该掌握以上4个方面的知识。其中容易被忽视的是第4个因素。有些人能写出正确的程序，却无法使用编程工具产生正确的可执行文件。为了消除这种现象，本书将结合具体实训题加强训练，使学生既学会“做什么”，也掌握“如何做”的全过程。从而避免可以在纸上编程序，却不知道如何得到程序的可执行文件的现象。

本章要求掌握如下只有主函数的简单C程序的基本结构模式：

```
文件包含部分              //包含头文件等
int main()                //主程序
{                         //主程序开始
    声明语句部分          //声明程序中所要使用的变量
    执行语句部分          //程序的可执行语句
    return 0;             //返回值
}                         //主程序结束
```

声明部分目前只涉及变量，不管以后章节增加何种声明，这些声明必需都放在可执行语句之前，即所有声明放在一起。只要牢牢掌握并熟练使用这种结构，就能解决很多程序设计问题。

本章的难点是初步认识不同编译器的区别，以便在编程中回避编译器区别。其实，只要避免使用可能产生歧义的语句，就可以编写出正确的程序。

因为学生可能是第一次接触编程语言，所以首先应要求他们养成良好的书写习惯和风格，尤其是变量命名的良好习惯。学生的模仿能力很强，可以适当给出一些范例及提示，让他们去举一反三，融会贯通。

【例 1-1】 给出下面程序的输出结果。

```
#include <stdio.h>
#include <string.h>
void  main()
{
  int a, b, c, d;

  a=sizeof(char);
  b=sizeof(int);
  c=sizeof(float);
  d=sizeof(double);

  printf("\n%d %d %d %d",a,b,c,d);
}
```

解答：从表面分析这好像是一个很简单的问题。其运行结果是：1 2 4 8。

由于计算机硬件发展很快，所以 C 编译器也随之更新。这主要是所谓的从 16 位到 32 位操作系统的变化。Borland C++ 3.1（本书以后简称 BC）是为 16 位操作系统配置的，所以即使在 Windows 98 下运行，其运行结果也是：1 2 4 8。

自从 32 位操作系统盛行以来，编译器也发生了变化。例如 Microsoft Visual C++ 6.0(本书以后简称 VC)，它的运行结果是：1 4 4 8。

这个例子提醒读者，对很多 C 语言书上的结果，自己要加以分析。下面再看一个与编译系统有关的例子。

【例 1-2】 给出下面程序的输出结果。

```
#include <stdio.h>
void main()
{
      int inum;
      inum=8;
      printf("\n%d,%d",inum,inum++);
}
```

解答：程序使用 inum 的方式产生歧义。因为在调用 printf 函数时，C 标准并没有规定实参数的求值顺序。inum 和 inum++分别是 printf 函数的两个输出参数。VC 是自左而右，先使用 inum 作为第一个和第二个参数，然后自增 1，所以输出是“8，8”。BC 是自右而左求值，第二个参数的 inum 为 8，但它使用之后变为 9，将 9 再作为第一个参数 inum，所以输出是“9，8”。

在编程时，应该避免使用可能产生歧义的语句，更不要写出别人看不懂，也不知道系统

如何执行的程序。尤其是使用“++”和“--”运算符时，更要小心。

【例 1-3】 分析下面程序的输出结果，如果不修改程序，能否通过编译？

```
#include <stdio.h>
main ( )
{
    int a=25, b=5, c=4,d=1;
    a+=b;
    c*=d+5;
    a%=c;
    printf ( "%d,%d", a,c);
    printf ( "%d",++a);
    printf ( "%d",a++);
    printf ( "%d",a);
}
```

解答：其实，BC 和 VC 都能通过编译并正确运行，只是 VC 没有警告信息，BC 给出 main 没有返回值的警告（增加 void 即可解决）。printf 语句没有换行符“\n”，输出为：6,24778。

【例 1-4】 转义字符实例。

```
#include   <stdio.h>
void main( )
{
    printf("a\tb\n");
}
```

假定系统设置制表符的一个“输出区”占八列，用□代表空格，则输出为：

a□□□□□□□b。

【例 1-5】 已知 a=15，b=155.2114，求 a+b 的值。

```
#include   <stdio.h>
void main ( )
{
    int a=15;
    float b,c;
    b=155.2114;
    c=(float)a + b;          // 把变量 a 浮点化
    printf ( "%f\n",c);
}
```

输出结果为：170.211395。

【例 1-6】 分析下面程序并给出运行的结果。

```
#include   <stdio.h>
void main ( )
```

```
{
    int a;
    float x=3.6;
    a=(int)x + 2;
    printf ( "x=%f,a=%d", x,a);
}
```

解答：程序中(int)x 只是强制将 x 转换成一个中间变量参加运算，原来的变量类型并没有改变，即变量 x 的值也没有变化。程序输出为：x=3.600000, a=5。

【例 1-7】 不用第三个变量，将 a 和 b 的值进行互换的操作是_____。

A．a+=b;b-=a;a-=b;　B．a+=b;b=a-b;a-=b;　C．b-=a;a-=b;a-=b;　D．b-=a;a=b-a;a+=b;

解答：应该在 A 中 B 寻找答案。b-=a 的含义是 b=b-a，是不对的，应选 B。

1.2 习题参考答案

1．单项选择题

（1）下列数据中属于“字符串常量”的是_____。

A．123　　B．" 123 "　　C．'123 '　　D．'3 '

解答：字符串常量应该使用双引号，正确答案是 B。

（2）" Aa " 在内存中占用的字节数是_____。

A．1　　B．2　　C．3　　D．4

解答：" Aa " 是两个字符的字符串常量。使用字符串常量只要将字符串用一对双引号括起来就可以了，编译程序会自动在每个字符串末尾加上“\0”，以此来标志字符串的终了。因此字符串在内部表示所占的空间要比实际字符数多一个字节，" Aa " 是两个字符的字符串常量，再加上结束标志，所以需要 3 个字节，即正确答案是 C。

（3）不是 C 语言提供的合法关键字是_____。

A．case　　B．default　　C．long　　D．cin

解答：cin 是 C++语言提供的合法关键字，不是 C 语言提供的合法关键字。正确答案是 C。

（4）正确的标识符是_____。

A．case　　B．de_fault　　C．1。ong　　D．a.b

解答：在 C 语言中构成标识符必须以字母或下划线 “_” 符号中任一字符打头，在第一个字符之后，可以是任意的字母、下划线或数字组成的字符序列，这个序列可以是空串。由此可见，只有 de_fault 是正确的标识符。

2．填空题

（1）转义字符是由_____符号开始的单个字符或若干个字符组成。

解答：\

（2）C 程序是由_____组成，其中只能有一个_____函数。

解答：函数　主

3．程序分析题

（1）给出下面程序的输出结果。

```
#include <stdio.h>
#include <string.h>
void   main()
{
   int a, b, c, d;

   a=sizeof(char);
   b=sizeof(int);
   c=sizeof(float);
   d=sizeof(double);

   printf("\n%d %d %d %d",a,b,c,d);
}
```

解答：见例 1.1 的解答。

（2）分析下面程序的输出结果，如果不修改程序，能否通过编译？

```
#include <stdio.h>
main ( )
{
      int a=25, b=5, c=4,d=1;
      a+=b;
      c*=d+5;
      a%=c;

      printf ( "%d,%d", a,c);
      printf ( "%d",++a);
      printf ( "%d",a++);
      printf ( "%d",a);
}
```

解答：见例 1.3 的解答。

4．程序设计题

（1）编写一个计算 25.30+43 的程序。

解答：直接使用 printf 语句输出计算结果即可。

```
#include<stdio.h>                          //包含头文件
int main()
{
    printf("25.30+43=%f\n",25.30+43);      //输出结果
    return 0;
}
```

（2）编写一个计算 y=2x+8 的程序，在主程序前加上注释并运行之。

解答：要求询问输入数据，所以需要使用 scanf 语句接受输入。

```
#include<stdio.h>                          //包含头文件
int main()
{
    float x,y;
    printf("请输入 x：");                  //请求输入
    scanf("%f",&x);                        //输入 x
    y=2*x+5;
    printf("y=2x+5=%f\n",y);               //输出结果
    return 0;
}
```

运行实例如下：

请输入 x：25.5

```
y=2x+5=56.000000
```

其实，也可以直接使用 printf 语句输出计算结果。

```
#include<stdio.h>                          //包含头文件
int main()
{
    float x;
    printf("请输入 x：");                  //请求输入
    scanf("%f",&x);                        //输入 x
    printf("y=2x+5=%f\n",2*x+5);           //输出结果
    return 0;
}
```

（3）编写一个输出“25+75=100”的程序。

解答：直接使用 printf 语句即可。

```
#include<stdio.h>                          //包含头文件
int main()
{
    printf("25+75=100\n");                 //输出结果
    return 0;
}
```

（4）编写一个对话程序，例如：

李明：HOW ARE YOU，王莹。

王莹：FING, THANK YOU AND YOU？

解答：因为还没有学过字符数组，显然只是要求输出这段对话。直接使用 printf 语句即可。

```
#include<stdio.h>
int main()
```

```
{
    printf("李明：HOW ARE YOU, 王莹。\n");
    printf("王莹：FING, THANK YOU AND YOU ?\n");
    return 0;
}
```

（5）编写不使用变量求 55×11 的程序。

解答：直接使用 printf 语句即可。

```
#include<stdio.h>                          //包含头文件
int main()
{
    printf("55*11=%d\n",55*11);            //输出结果
    return 0;
}
```

（6）编写要求输入变量 a 和 b 的值，输出它们的和及乘积的程序。

解答：要求询问输入数据，所以需要使用 scanf 语句接受输入。

```
#include<stdio.h>                          //包含头文件
int main()
{
    float a,b;
    printf("请输入 a：");                  //请求输入 a
    scanf("%f",&a);
    printf("请输入 b：");                  //请求输入 b
    scanf("%f",&b);
    printf("%f+%f=%f\n",a,b,a+b);          //输出其和
    printf("%f*%f=%f\n",a,b,a*b);          //输出乘积

    return 0;
}
```

运行实例如下：

```
请输入 a：25
请输入 b：25
25.000000+25.000000=50.000000
25.000000*25.000000=625.000000
```

（7）分别用字符和 ASCII 码形式输出整数值 65 和 66。

解答：直接使用 printf 语句即可。

```
#include<stdio.h>
int main()
{
  printf("字符：%c,ASCII 码:%d\n",65,65);
  printf("字符：%c,ASCII 码:%d\n",66,66);
  return 0;
```

```
}
```

输出结果如下：

字符：A,ASCII 码:65

字符：B,ASCII 码:66

（8）编写输入三个字符，但无视第二个字符的程序。如果输入 fine，给出输出结果。

解答：在语句中使用“*”来解决。

```
#include<stdio.h>                          //包含头文件
int main()
{
    char a,b;
    printf("请输入字符：");                 //请求输入
    scanf("%c%*c%c",&a,&b);
    printf("输出字符：%c%c\n",a,b);         //输出结果
    return 0;
}
```

输入 fine，应丢弃第二个字符 i。因为多输入一个字符，所以不读入字符 e。输出结果如下：

请输入字符：fine

输出字符：fn

1.3 使用 C 程序解题的简单过程

首先要对计算机解决问题的步骤有一个初步认识。下面就结合一个实际问题进一步说明使用 C 语言解题的步骤。

1）设计一个解题的方法。

2）使用一种方式把它描述出来。

3）把它们转化成程序形式。

4）在计算机上编辑成 C 的源文件。

5）编译 C 程序源文件的过程，同时也是查错的过程。如果不能正确编译，进行查错，直到产生正确的 obj 文件。

6）将它们连接成 exe 文件，如果连接出错，返回步骤 4，再次检查和编译源文件。

7）运行 exe 文件，得出初步结果。

8）验证结果是否正确。如果结果不正确，就要返回查找原因，直到运行结果正确为止。

表面上看，运行结果不正确要返回第 4 步检查程序，其实这是对步骤 3 的复查。如果算法设计不对，要转到步骤 1，也即重复 1～4 步。如果题目复杂，则要从头开始。因此，一定要重视前三个步骤。

下面举一个简单例子，具体说明前面的几个过程。

【例 1-8】 编写将输入的两个整数相加，然后输出它们的和的程序。

1）设计解题方法。显然，这是一个顺序执行的过程，要求按顺序输入两个整数，再把它

们相加，最后输出它们的和。

2）使用语言方式把它描述出来。例如：

程序开始

```
要求提示输入 2 个整数
赋值给变量 number1, number2
计算 sum=number1+number2
输出 sum
```

程序结束

如果使用英文及语句说明，可以像下面这样说明：

```
BEGIN
  Print   InputMessage
  Input number1, number2
  sum=number1+number2
  Print sum
END
```

3）下面是 C 语言的实现程序：

```
#include   <stdio.h>
void main( )
{
  int   sum,   number1, number2;

  printf("Please input two numbers :\n");
  scanf("%d%d",&number1,&number2);
  sum= number1+ number2;
  printf("sum=%d", sum);

}
```

其他几个过程均可以在 C 编译器的集成环境中完成。现简述如下：

4）选择一种 C 语言开发环境，建立这个 C 程序的源文件，例如命名为 add.c。在这个文件里输入 C 语言的源程序，这就是源文件的编辑过程。

5）使用编译命令对这个源文件进行编译，如果编译正确，产生 add.obj 文件。如果程序有问题，修改后再重复这个过程，直到产生正确的 add.obj 文件。

6）使用 Link 命令，将 add.obj 文件连接成可执行的 add.exe 文件。

7）运行这个 add.exe 文件，验证结果是否正确。

8）如果不正确，需要返回去检查原因，甚至返回到第一步。

由此可见，前 4 步的工作很重要，要提高效率，就要做好这几步的工作。但没有后面的工作，则永远停留在纸上谈兵。一个程序员，必须熟练使用开发环境，才能实现最终目的。

【例 1-9】 要求将从键盘上输入的一个字符按下面格式输出，下面是一个运行示范：

```
Input：5
```

```
Your input is:5
```

这是教材要求的实训题。下面简述前几步的过程，而且下面给出的C语言实现是一个含有错误的程序，留待编译时去查错。

【选择算法】

这也是一个顺序执行过程，输入有提示信息，输出增加了说明。

程序开始

```
输出提示信息
赋值给变量 ch
输出组合信息
```

程序结束

【算法描述】

```
BEGIN
  Print   InputMessage
  Input ch
  Print OutputMessage
END
```

【编程实现】

下面是含有错误的C语言程序：

```
main( )
{
  char ch;

  printf("input:");
  scanf("%d", ch);
  printf("Your input is:%d\n", ch);
}
```

1.4 BC上机指南

因为使用BC学习C编程最方便，所以本节主要介绍使用该集成环境的基本方法，以便读者能顺利解决边学边练的问题。

1.4.1 工作界面及基本操作

BC集成开发环境集内部编辑程序、调试程序及其他实用程序于一体，无需独立的编辑、调试、编译、连接程序就能建立并运行程序，并且只要通过一个简单清晰的主屏幕，就能使用这些功能。

在使用中常用到双键或三键操作，约定按照按键的先后顺序书写，例如：

〈Alt+F5〉　代表先按下〈Alt〉键，然后按〈F5〉键，再同时放开，〈 〉号不是输入符号，仅用来表示其内是功能键。

〈Ctrl+KB〉　代表先按下〈Ctrl〉键，然后按下〈K〉键，同时松手之后再按〈B〉键。

如果 Borland C++是装在 Borlandc 目录下，可在 DOS 提示符下键入 Borlandc\bin\bc 并按〈Enter〉键，进入 BC 集成开发环境。初次启动屏幕，屏幕除显示主菜单外，还有版本信息。按任意键，版本信息消失并留下如图 1-1 所示的主屏幕。主屏幕由主菜单、编辑窗、信息窗和观察窗及相应的快速参考行等四部分组成。

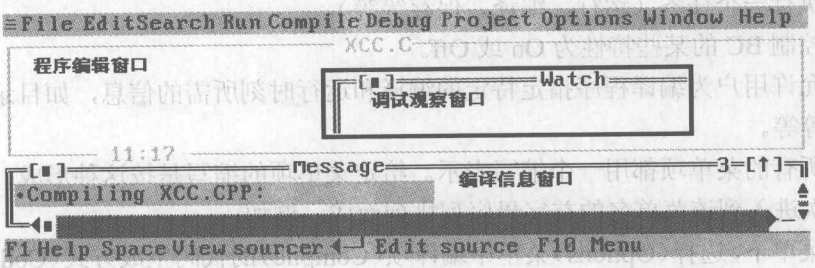

图 1-1　主屏幕信息

快速参考行的内容是变化的，它指示的内容随着所激活的相应窗口而变化，图中是激活信息窗的情况。观察窗与信息窗均处于下部，并且每次只能显示一个，可以使用菜单命令或〈F6〉键在这些窗口中切换。

1．调用帮助系统

1）按〈F1〉键即可调用该帮助系统，帮助窗口将解释用户当前所处位置的含义，在所有帮助屏幕关键字处按〈Enter〉键便可得到该选择项更详细的帮助信息。使用〈Home〉键或〈End〉键，可分别选择屏幕上第一个或最后一个关键字。按〈Alt+F1〉键可以回到前次帮助屏幕。在进入帮助系统后，再按〈F1〉键可以得到帮助索引。当使用 BC 编辑程序时，可能需要了解各种函数信息。将光标置于需要了解的函数名上，按〈Ctrl+F1〉键，将会得到有关该函数信息的帮助信息。要退出帮助系统返回原来的菜单选择，按〈Esc〉键或使用后面描述的“热键”即可。

2）按〈F5〉键可以放大/缩小活动窗口。

3）按〈F6〉键可切换活动窗口。

4）按〈F10〉键切换菜单和活动窗口。

按〈Alt〉键和主菜单命令的首字母（如 F，E，R，C，W，O 或 D 等）可直接调出该命令。例如按〈Alt+F〉键可选择文件菜单。

2．菜单内的操作

用红色的大写字母选择一个菜单项，或使用箭头键把绿色键移动到相应的选择项并按〈Enter〉键。按〈Esc〉键可以退出一层菜单。在主菜单或主菜单的某个下拉菜单中，按〈Esc〉键将直接退回活动窗口（当某窗口被激活时，其顶部为双线，窗口名呈高亮度显示）。按〈F10〉键从任何菜单层返回先前活动窗口。使用左、右箭头键可从一个下拉菜单进入另一个下拉菜单。

3．退出 BC 返回 DOS

在 File 菜单中选择 Quit（按〈Q〉键，或将光标移至〈Quit〉再按〈Enter〉键）将退出 BC 返回 DOS，但如果在选择 Quit 前未保存当前工作文件，则系统将询问是否需存盘（也可使用