“十二五”普通高等教育本科国家级规划教材

普通高等教育“十一五”国家级规划教材

高等学校计算机基础教育教材精选

C程序设计教程（第4版）

崔武子 主编
李青 李红豫 鞠慧敏 编著

清华大学出版社

第2版、第3版荣获“北京高等教育精品教材”称号
第2版荣获“第八届全国高等学校出版社优秀畅销书”

“十二五”普通高等教育本科国

普通高等教育“十一五”国家级规划教材

高等学校计算机基础教育教材精选

C程序设计教程

（第4版）

崔武子 主编
李红豫 鞠慧敏 编著

清华大学出版社
北京

内 容 简 介

本书是以C语言程序设计零起点的学习者为主要对象的程序设计教程，2012年8月出版了第3版，该书是普通高等教育“十一五”和“十二五”国家级规划教材，荣获“北京高等教育精品教材”称号，本次再版则在强调“教师方便教，学生容易学”的同时，按照C99标准规范了所有程序。

全书用例题组织所有的教学内容，并用两套实例贯穿整个教学过程，整体内容编排独特，组织形式新颖。全书共分9章，分别是C语言基础知识、顺序结构程序设计、分支结构程序设计、循环结构程序设计、数组、指针、函数、结构体和其他构造类型以及文件。

本书配备了动画丰富、内容生动的电子教案，所有程序的运行环境均为Visual C++ 6.0(但在1.6.4节中补充介绍了Visual C++ 2010环境)。

本书是高等院校C语言程序设计课程的教材，也可作为C语言自学者的参考书。

图书在版编目(CIP)数据

C程序设计教程/崔武子等编著. —4版. —北京：清华大学出版社，2015

高等学校计算机基础教育教材精选

ISBN 978-7-302-40032-5

Ⅰ. ①C… Ⅱ. ①崔… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2015)第086758号

责任编辑：谢　琛

封面设计：何凤霞

责任校对：白　蕾

责任印制：何　芊

出版发行：清华大学出版社

网　　址：http://www.tup.com.cn，http://www.wqbook.com

地　　址：北京清华大学学研大厦A座　　邮　　编：100084

社 总 机：010-62770175　　邮　　购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

课 件 下 载：http://www.tup.com.cn，010-62795954

印 装 者：北京国马印刷厂

经　　销：全国新华书店

开　　本：185mm×260mm　　**印　　张**：22.5　　**字　　数**：504千字

版　　次：2003年7月第1版　2015年6月第4版　　**印　　次**：2015年6月第1次印刷

印　　数：1～3000

定　　价：43.00元

产品编号：061982-01

第 4 版前言

本书是作者在围绕“教师方便教，学生容易学”的主题，开展一系列探索与实践活动后，以 C 语言程序设计零起点读者作为主要对象而编写的程序设计教程，本书于 2012 年 8 月出版了第 3 版，该书是普通高等教育“十一五”和“十二五”国家级规划教材，荣获“北京高等教育精品教材”称号，本次再版则进一步强化了应用能力，并按照 C99 标准规范了所有程序。本书的内容编排独特，组织形式新颖，能使读者在较短的时间内掌握 C 程序设计的精华。本书是高等院校 C 程序设计课程的教材，也可作为自学者的参考书。

1. 本书特点

(1) 每章内容分成基础部分和提高部分。考虑到 C 语言的语法现象众多，初学者往往难以接受，书中将每章的内容分成了基础和提高两个部分。将常识性的、基础类的、必须掌握的内容放在基础部分中；将具有扩展性的、提高性的内容安排在提高部分中。通过基础部分的学习，能够掌握最基本的语法，初步建立程序设计的思维方式和编写一般程序的能力，同时可培养学生的学习兴趣。即使因学时不足跳过提高部分，也不至于影响后续内容的学习。

(2) 所有教学内容用例题组织。在基本遵循 C 语言教学体系的情况下，用例题组织所有教学内容。即根据要介绍的内容精心编写相应的例题，将大量的、正确的、规范的程序介绍给学生，在讲解例题的过程中，使学生学习语法、了解概念、掌握算法。做到在解决实际问题中讲授语法，而不是为了教语法而举例。为了便于查找，在各章开头针对每道例题添加了知识要点。

(3) 涉及算法的例题均设有编程点拨。针对学生“读程序容易，编程序难”的情况，书中凡涉及算法的例题，在给出其完整程序之前，都增设了编程点拨，有些算法还提供了多种解法。

(4) 强调实践能力，注重个性化教育。在各章之后添加了上机训练内容，每个训练题均分为题目、目标、步骤、提示和扩展。为了培养学生调试程序、排除错误的能力，教材中分阶段通过具体例题介绍了调试程序的方法，程序的运行环境是 Visual C++ 6.0(在 1.6.4 节中补充介绍了 Visual C++ 2010 环境)。

(5) 讲授指针和函数时不涉及新算法。指针和函数是 C 语言中的重点和难点，为了使学生能够顺利接受新概念，将有关算法内容尽量安排在这两章之前，避免学生在接受指针和函数概念的同时，又要理解新算法。这样做不仅容易突破难点，而且有利于巩固已学过的知识。

(6) 用两套实例贯穿整个教学过程。为了使学生尽早接触应用程序的编写过程，提供了贯穿整个教学过程的两套实例，每套实例均随着讲授内容的增多，分八次逐步补充和完善其程序的功能。其中贯穿实例 A(较小实例)在基础部分中给出，供教师课堂教学；贯穿实例 B(较大实例)在提高部分中给出，供学生课后阅读。最后一章贯穿实例 B 中还提供模块图、主函数流程图和整个程序的源代码。

(7) 习题、讨论题和思考题齐全，提供单号习题答案。与教材内容相对应，各章习题也分为基础和提高两部分。书中单号习题提供参考答案，以方便学生自测和教师布置作业。为了促使互动教学，适当添加讨论题和思考题。

(8) 在附录 E 中提供关键字、运算符、库函数的索引。

(9) 配备含有电子教案、源程序代码等内容的课件。

该课件包括电子教案、本书所有例题和贯穿实例的源程序。为了减轻教师备课的负担，制作成生动的电子教案，通过演示可使读者尽早了解本课程的基本目标。

2. 使用建议

(1) 必学基础部分。基础部分是学生必须掌握的知识，但在教学过程中教师可将部分例题留给学生自学。

(2) 选学提高部分。书中的提高部分是为了帮助读者更上一层楼，教师可以根据实际情况，选择其中部分内容进行介绍(标有 * 的例题有一定难度)。

(3) 兼顾学时和学生编程能力的提高需求，建议课堂上介绍贯穿实例 A，安排学生课外阅读贯穿实例 B，这两套贯穿实例对学生学习有很大帮助。

(4) 单、双号习题成对做。单号习题提供参考答案，双号习题则在类型上与前一单号习题相同，侧重点也接近。基础部分中提供的习题都是最基本的，题量也不多，建议读者全部做完，提高部分中的习题可根据情况选做(标有 * 的习题有一定难度)。

(5) 选做上机训练题中的扩展题。在完成训练题的基础上可根据不同层次的学生情况，选做扩展题。

全书由崔武子主编并统稿，李青、李红豫、鞠慧敏、齐华山和孙力红参加部分内容的修订和编写。本书第 3 版得到北京联合大学规划教材建设项目资助。在本书第 3 版使用的过程中，教学团队的教师提出了许多宝贵意见，在此表示真挚的感谢。

限于作者水平，书中难免有错误和疏漏之处，恳请读者批评和指正。

作　者

2015 年 3 月

目　　录

第1章 C语言基础知识

本章将介绍的内容

基础部分：

- C程序的基本概念、上机步骤和Visual C++ 6.0集成环境。
- 整型、实型、字符型数据类型的常量和变量。
- 算术运算、赋值运算和逗号运算。

提高部分：

- 不带参数的主函数。
- 进一步学习赋值运算符。
- 进一步学习数据类型。
- Visual C++ 2010集成环境。

各例题的知识要点

例1.1　C程序形式和程序执行过程。

例1.2　主函数、函数体和输出控制。

例1.3　函数体包括多条语句；换行控制；C程序的书写格式。

例1.4　正确选用数据类型的重要性。

例1.5　常量和变量的概念；符号常量的概念。

例1.6　合法与非法的变量名。

例1.7　整型数据的输出格式说明符。

例1.8　整型变量的定义和数据的溢出现象。

例1.9　实型常量的不同输出形式。

例1.10　实型变量的定义；有效数字。

例1.11　常规字符不同的格式输出以及字符常量的算术运算。

例1.12　字符型变量的定义和赋值。

例1.13　将代数式转化为C语言表达式。

例1.14　强制类型转换。

例1.15　逗号表达式。

（以下为提高部分例题）

例 1.16　复合赋值运算符。

例 1.17　整型常量的不同进制表示法及相应输出说明符。

例 1.18　特殊字符的输出和转义字符的概念。

例 1.19　Visual C++ 2010 集合环境。

1.1　C 语言概述

1.1.1　C 语言与程序设计

人与人之间交换信息需要借助于语言工具，人与计算机交换信息也同样要用语言工具，这一工具就是计算机语言。用计算机语言编写的代码叫做程序。所谓程序，就是一系列的指令集合。计算机的一切操作都是由程序控制的，在运行程序时，程序中的指令集决定计算机如何对用户的输入进行处理。

随着计算机技术的发展，计算机语言逐步得到完善。最初使用的计算机语言是用二进制代码表达的语言——机器语言，后来采用与机器语言相对应的助记符表达的语言——汇编语言，人们称这两种计算机语言为低级语言。虽然用低级语言编写的程序执行效率高，但程序代码长，并且这些程序都依赖于具体的计算机，因此编码、调试、阅读程序很困难，通用性也差。现在使用最广的计算机语言是高级语言——用更接近于人们自然语言和数学语言的表达语言。用高级语言编写的程序独立于机器，编码相对短，可读性强，但必须通过编译和连接后，才能被计算机执行。用高级语言编写的程序叫做源程序。

由上可见，低级语言和高级语言各有利弊。C 语言是高级语言，它是一种用途广泛、功能强大、使用灵活的面向过程的语言，它不仅具有高级语言的功能，还具有低级语言的许多功能，因此是国际上广泛流行的计算机语言。Windows、Linux 和 UNIX 等操作系统都是用 C 语言编写的。

C 语言的主要特点是：语言简洁，使用方便，编程自由度大，具有结构化的控制语句，运算符和数据类型丰富，而且允许直接访问物理地址，能实现汇编语言的大部分功能，可以直接对硬件进行操作，用 C 语言编写的程序可移植性好，生成目标代码质量高，程序执行效率高。

要得到 C 语言程序的运行结果，首先将源程序输入计算机内（在计算机上输入或修改源程序的过程叫做编辑），然后将源程序翻译（叫做编译）成机器能识别的目标程序，最后还要把目标程序和系统提供的库函数等连接起来生成可执行文件，这时才可以运行程序，并看到运行结果。C 程序的编辑、编译、连接、运行过程可用图 1.1 表示（以文件名为 e1.c 的 C 程序为例）。

图 1.1　C 程序的编辑、编译、连接、运行过程

C 程序的编辑、编译、连接、运行过程可以在不同的环境中进行，本书中的所有例题都在 Visual C++ 6.0 集成环境下运行通过，在 1.6.4 节中还介绍了 Visual C++ 2010 集成环境。

程序设计是指从确定任务到得到结果、写出文档的全过程。程序设计的步骤大体上分为：

① 问题定义；
② 算法设计；
③ 流程图设计；
④ 编写程序代码；
⑤ 测试与调试；
⑥ 整理文档；
⑦ 系统维护。

限于篇幅，本书将重点放在前五项。

1.1.2 C 程序形式和程序执行过程

下面举一个 C 程序的完整例题，说明 C 程序的一般形式和程序的执行过程。程序中的具体语法规则和其他细节将在后续章节中陆续介绍。

【例 1.1】 编写一个完整的 C 语言程序示例。

【解】 程序如下：

```
#include <stdio.h>                          //包含文件
#include <math.h>                           //包含文件
int mysum(int m,int n);                     //函数原型说明
int main(void)                              //主函数首部
{    int a,b,x;                             //声明部分
     double c,y,z;                          //声明部分

     c=4.0;                                 //以下各行均为语句部分
     y=sqrt(c);
     a=10;
     b=20;
     x=mysum(a,b);
     z=x+y;
     printf("z=%lf\n",z);
     return 0;
}                                           //主函数到此结束

int mysum(int m,int n)                      //mysum 函数首部
{    int k=0,i;                             //声明部分
```

```
    for(i=m;i<=n;i++)k=k+i;              //以下 2 行均为语句部分
    return k;
}
```

运行结果：

```
z=167.000000
```

程序说明：

(1) 正如本例所示，C 语言程序是由若干函数构成的，函数中只能包含一个主函数，C 程序从主函数开始执行，主函数名必须是 main。例 1.1 中程序的执行过程如图 1.2 所示，程序按①～⑨的顺序执行。

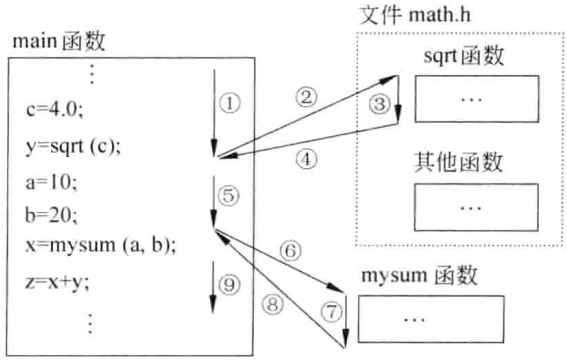

图 1.2　例 1.1 程序的执行过程

(2) 程序中各行从//到本行结束是注释部分，用此对该行代码进行说明，注释对程序的运行无任何作用，注释的目的是方便阅读程序。注释还可以用/ *　　* /标注，例如

```
#include <stdio.h>                    /* 包含文件 */
```

1.2　简单 C 程序及其上机步骤

本节将给出几个最简单的 C 程序，通过这些例题，介绍 C 语言的基本概念，以及在 Visual C++ 6.0 集成环境下的上机步骤。

1.2.1　简单 C 程序和编程风格

【例 1.2】 编写在屏幕上显示“Let's study the C language.”的程序。

【解】 程序如下：

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

```
    printf("Let's study the C language.");
    return 0;
}
```

运行结果：

```
Let's study the C language.
```

程序说明：

(1) 程序中 main 是主函数名，每一个 C 程序都必须包含而且只能包含一个主函数。本书按照 C99 标准，所有程序中的主函数框架均采用如下形式：

```
int main(void)
{
    ⋮
    return 0;
}
```

主函数框架中的 int，void，return 0 的含义和作用将在 1.6.1 节中介绍，初学者不必深究。

(2) 用一对花括号{ }括起来的部分是函数体。本例函数体中的语句“printf("Let's study the C language.");”是输出语句，其作用是按原样输出双引号内的字符串“Let's study the C language.”(详见 2.3.1 节)。语句最后的“;”不能丢。

(3) C 语言中区分大小写，即 main 不能写成 Main，printf 也不能写成 Printf。若程序中有此类错误，则很难发现。

(4) printf 是 C 语言标准库中提供的输出函数。需要在程序中使用输入、输出函数，程序的开头要加#include <stdio.h>命令行。实际上，每个程序中必定会有输出操作，因此编写程序时，在程序的第一行都写此命令行。

【例 1.3】 编写输出两行句子“Let's study the C language.”和“It's interesting.”的程序。

【解】 程序如下：

```
#include <stdio.h>
int main(void)
{
    printf("Let's study the C language.\n");          //输出字符串后换行
    printf("It's interesting.\n");
    return 0;
}
```

运行结果：

```
Let's study the C language.
It's interesting.
```

程序说明：

(1) 本程序的函数体包括两条输出语句，这两条语句可合并为一条语句，即“printf ("Let's study the C language.\nIt's interesting.\n");”。

(2) \n 是换行符，如果程序中去掉\n，输出形式则为：

```
Let's study the C language.It's interesting.
```

(3) C 程序的书写格式比较自由。例如，一行内可以包括多条语句、一条语句可以写在多行上、每行的内容可以从任何一列开始写等，但提倡学习者在编写程序时要养成良好的程序设计风格，良好的编程风格能提高程序的可读性、可维护性，也能促进技术交流，便于团队合作。在此介绍如下几种风格，其他风格在后面陆续介绍。

① 合理安排各成分的位置。一般 #include 命令行在程序的最前面，接着依次为 #define 命令行、类型声明(如结构体类型声明)、函数原型说明、各函数定义等。

② 适当加注释。一般在程序的开头加注释，解释本程序的功能和一些说明，在函数或程序段的开头加注释，解释其要实现的功能、算法、参数等，在变量的定义行后面加注释，解释该变量的用途等。

③ 在程序中适当加上空行。在命令行和类型声明之间、类型声明和函数原型之间、函数原型与函数定义之间、函数内部变量定义与其下执行语句之间均空一行，有些地方视情况可空两行。

④ 采用缩进格式。一般用 Tab 键将某些行向右缩进，这样可使程序的逻辑结构更加清晰，层次分明，显著提高程序的可读性。例如，

```
int main(void)
{    int i=0,n=0,s=0;

     for(i=1; i<10; i++)
     {    if(i%3==0)
               n++;
          s=s+i;
     }

     printf("s=%d,n=%d\n",s,n);
     return 0;
}
```

在多人共同完成一项任务时，如果不用 Tab 键而用空格键缩进，则可能对统一格式带来不便。

⑤ 标识符要见名知意。可用英文单词、拼音或缩写作为标识符的一部分，一般标识符的第一个字符用字母，其余字符用字母、数字或下划线。

⑥ 一行写一条语句。

⑦ 算法简单明了。尽量采用简单易懂的算法，不使用过分复杂的算法。

⑧ 用户界面友好。一般使用计算机解决问题时，采用人机对话形式。当要求用户输