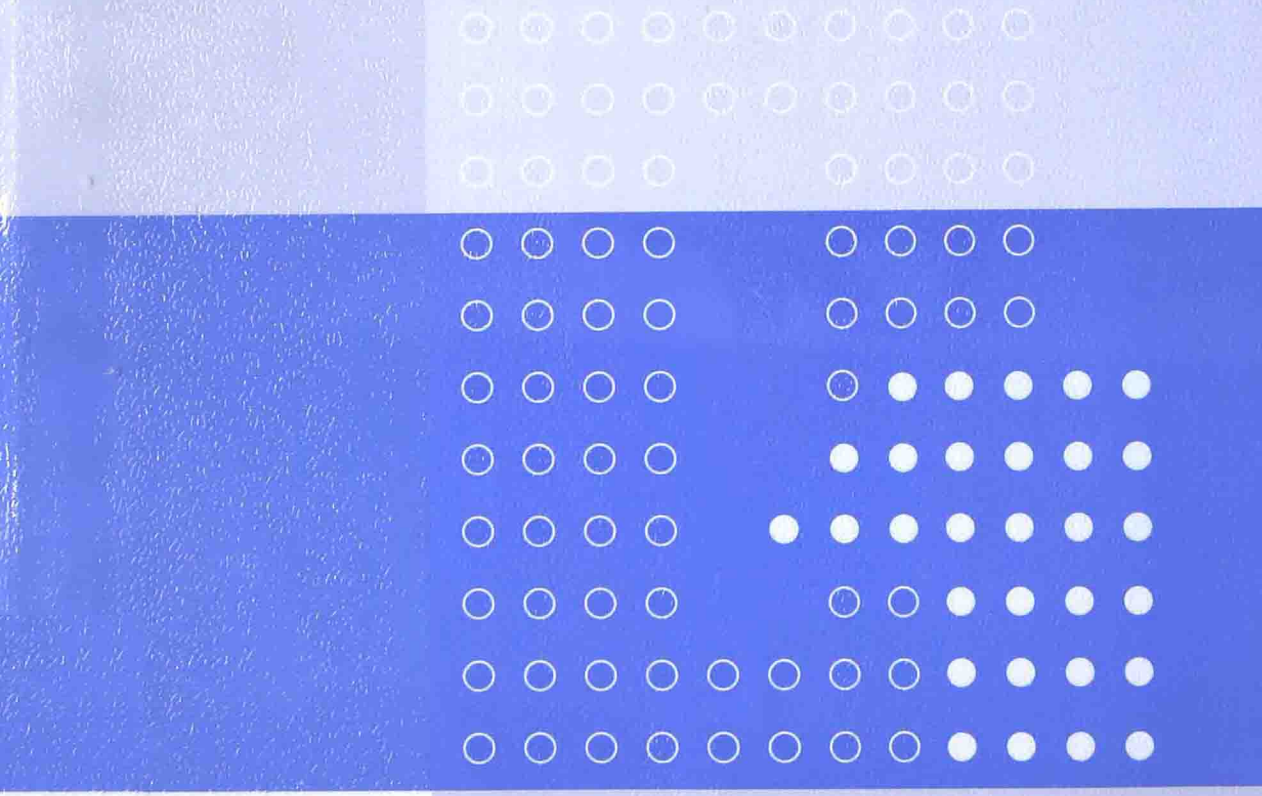
普通高等教育“十一五”国家级规划教材 计算机系列教材

C程序设计教程
（第2版）



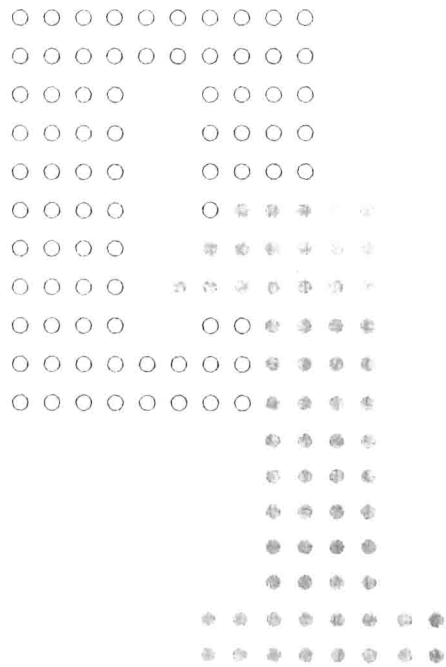
马瑞民 衣治安 主 编
刘华蓥 吴雅娟 副主编

清华大学出版社

普通高等教育“十一五”国家级规划教材　计

马瑞民　衣治安　主　编
刘华蓥　吴雅娟　副主编

C程序设计教程
(第2版)



清华大学出版社
北京

内 容 简 介

本书采用开门见山的编写思路，开篇即直奔主题，通过例题来介绍C语言的一些基本概念，让学生在做中学，在编程中体会，避免了基础知识的枯燥介绍过程。通过合理布局减少了一些臃肿的叙述，以循环、数组、函数和指针为重点，大幅度减少了数据类型、共用体、编译预处理和位运算的篇幅。

全书共分为10章，以概述开篇，然后是三种基本结构、数组、函数、指针、结构体与动态内存分配、文件和C语言涉及的其他知识。书中共有160道例题，同时引入了“通讯录管理系统”、“链表操作”等案例程序，除特别声明外，全部在Visual C++环境中调试运行。

本书是作者总结十几年C语言教学经验，参考众多国内外优秀教材的特点，综合分析学生的学习规律和接受能力后而精心组织编写的，适合作为高等学校的C语言教材，也适合作为广大编程爱好者的自学读物。

图书在版编目(CIP)数据

C程序设计教程/马瑞民，衣治安主编．--2版．--北京：清华大学出版社，2015
计算机系列教材
ISBN 978-7-302-38931-6

Ⅰ．①C…　Ⅱ．①马…　②衣…　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005666号

责任编辑：张瑞庆
封面设计：常雪影
责任校对：焦丽丽
责任印制：杨　艳

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 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：16.75　　**字　　数**：389千字
版　　次：2011年2月第1版　2015年2月第2版　　**印　　次**：2015年2月第1次印刷
印　　数：1～3000
定　　价：33.00元

产品编号：062129-01

本书第 1 版自 2011 年出版以来，已经过 4 轮的教学实践。一些高校将本书作为“C 程序设计”课程的教材或参考书，这些教师和学生感受到了“开门见山、循序渐进、叙述简洁”等特点对 C 语言教学的好处，对 C 语言的基本概念、基本理论、程序设计方法等有了更好的认识。当然，本书第 1 版的不足也在教学中有所体现。例如，重点内容的第 5 章、第 6 章的篇幅较少，学生在这部分的学习中感觉例题量不足、训练不充分。针对本书在教学实践中所反映出来的问题，第 2 版主要有以下变化：

第一，改写了第 5 章，增加了例题和算法等方面的介绍；充实了第 6 章；在第 8 章中丰富了动态分配内存方法，并且将原来非计算机专业学生较少涉及的链表部分移到第 10 章；修订了附录 D；增加了重点章节的习题量。

第二，进一步体现教材的逻辑性，对一些重要例题加强了“分析”、“说明”和“程序注释”部分内容，通过增加程序设计中抽象过程的篇幅，以达到强化编程前的分析和算法描述部分训练的目的，以提高读者的计算思维能力。

第三，对全书某些表述不准确或者不到位的地方予以修正，使得叙述更准确、更流畅、更简洁。

第 2 版的这些调整使得本书篇幅上有所增加，但这些调整是必要的，希望在教学实践中得到检验。

为了更好地满足学习的需要，本书的配套教材——《C 程序设计实验指导与习题集》也同时再版，希望对读者熟练地掌握程序设计的方法和技巧有更多的帮助。

感谢读者的信任和支持，希望使用此书的教师、学生多提宝贵意见。

作　者

2015 年 1 月

《C 程序设计教程（第 2 版）》第 1 版前言

C 语言具有功能丰富、应用灵活、执行效率高并能直接对计算机硬件进行操作等特点，既可以作为系统软件的描述语言，也可以用来开发应用软件，多年来一直作为高等学校计算机程序设计的入门课程之一。

但是，正由于它的优势十分明显，使得 C 语言的学习也有一定难度。比如，由于它的功能丰富，使得需要掌握的课程内容相对较多，如其运算符就有 15 个优先级别，数据类型种类繁多等；由于它的应用灵活，使得编程和调试都增加了难度，如其不进行数组下标和指针指向的越界检查，数组元素的地址可以作为实参传递给形参数组名等；由于它兼顾了系统软件和应用软件的开发，又必须完善许多特殊的功能，如其具有丰富的位运算、库函数、内存动态分配和复杂的指针类型等。如果不学习这些内容，很难反映 C 语言的特点，反之必将使得教学内容繁多、教材臃肿、学时增多，教师总觉得有讲不完的重点和难点，学生总有听不懂的内容，教学效果很难提高。

本书编写组成员二十多年来一直从事计算机基础教学，常年担任高级语言类课程的主讲和实验任务，主持了十多项省部级计算机基础教学改革课题的研究，出版了高等学校教材十余部。以本书编写组为骨干的教学团队是省级教学团队，该团队中有一人获得省级教学名师奖、两人获得校级教学名师称号，建设有“C 程序设计”等两门省级精品课，获国家级教学成果奖 1 项、省部级教学成果奖 11 项。近十几年来，该教学团队一直关注 C 程序设计课程的教学改革，也积累了编写此课程教材的实际经验。本书不像传统的 C 语言教材那样先从数据的类型、表达式讲起，而是通过例题来引入库函数、数据类型、语句、算法等程序设计要素的使用方法，对基本内容的介绍开门见山，等读者具备了一定的 C 程序设计基础以后再引入更深层次的内容。例题中先有程序设计思路的分析，书写程序之后还有相应的解释性的说明，能引导读者逐步掌握各种程序设计方法。书中正文篇幅不长，削减了对初学者用途较少的共用体、枚举类型、位运算的比例，既保证了 C 语言核心内容的篇幅，也更有利于组织教学。

本书的叙述以 89 ANSI C 为基础，同时兼顾 C99 的标准。除特别声明的之外，本书的例题在 Visual C++ 6.0 环境中调试运行，绝大多数例题也在 Turbo C 2.0 环境中调试运行。

本书由马瑞民、衣治安主编，刘华莹、吴雅娟担任副主编。其中第1章至第4章及附录A、B、C由刘华莹编写，第5章和第6章由吴雅娟编写，第7章和第8章由马瑞民编写，第9章和第10章及附录D由衣治安编写，全书由马瑞民和衣治安统稿。本教材建议的授课时间为70学时，其中理论占40学时，实验占30学时。本书有配套的实验指导与习题集，可以根据授课对象和教学需要调整授课学时和教学内容。

本书在编写过程中得到了东北石油大学计算机基础教育系老师们的指导与帮助，他们的教学资料和经验对本书的完善起到了很大的作用，在此致以诚挚的谢意。

由于水平有限，难免有不当之处，恳请批评指正。

作　者

2010年11月

FOREWORD

第1章 概 述

本章介绍C语言的发展简史、C程序的构成和程序设计算法，并通过几个简单的C程序使读者初步了解程序设计的基本知识。

1.1 C语言简介

C语言是一种应用非常广泛的程序设计语言，它既可以作为系统软件的描述语言，也可以用来开发应用软件。

C语言源于BCPL语言。1967年，英国剑桥大学的Martin Richards推出了**BCPL语言**。1970年，美国贝尔实验室的Ken Thompson对BCPL语言进行了简化，设计出了很简单的**B**(取BCPL的第一个字母)**语言**，并用B语言编写了第一个UNIX操作系统。1972年，贝尔实验室的Dennis M. Ritchie在B语言的基础上设计出了**C**(取BCPL的第二个字母)**语言**。

1978年，Brain W. Kernighan和Dennis M. Ritchie合著了著名的*The C Programming Language*，从而使C语言成为目前广泛流行的程序设计语言。此后，又有多种语言在C语言的基础上产生，如C++、Java和C#等。

1983年，美国国家标准协会(ANSI)对C语言进行了标准化，推出了第一个C语言标准草案(**83 ANSI C**)。1989年，ANSI发布了完整的C语言标准，通常称为**ANSI C**(又称**C89**)。1990年，国际标准化组织(ISO)采纳了ANSI C，并以国际标准发布。1999年，ISO对C标准做了全面修订，形成了C语言标准ISO/IEC 9899:1999，简称**C99**。目前，C语言最新标准是2011年发布的ISO/IEC 9899:2011，简称**C11**。

各主流厂家提供的C编译器有Microsoft Visual C++、Turbo C、Borland C等，都未实现C99以后标准所建议的全部功能，因此本书所采用的是ANSI C标准，程序的书写形式兼顾C99标准。

本书中的Visual C++指6.0版本，Turbo C指2.0版本。若无特别声明，书中的例题都能在Visual C++ 6.0和Turbo C 2.0环境下调试运行，仅适用于一种环境的例题基本上都有相应的提示或解释。

1.2 简单的C程序

本节通过几个简单的C程序来初步了解C源程序的构成。

1.2.1 printf函数

【例1-1】 在屏幕上输出一行信息。

```
#include "stdio.h"
main()
{
  printf("hello!");
}
```

运行结果为：

```
hello!
```

【说明】

（1）程序中

```
main()
{

}
```

是一个函数，这个函数的名字为 main，一般称为**主函数**。这个名字是专用的，每一个 C 程序必须有而且也只能有一个主函数。

（2）main()是**函数的首部**，函数首部下面一对花括号{ }括起来的部分称为**函数体**。

（3）本例中函数体内只有一条语句。C 语言规定**语句**必须以分号结束。printf 是格式输出函数，其双引号内的若干字符会原样输出。

（4）程序第 1 行的 #include "stdio. h"是编译预处理命令。编译预处理命令都是以 # 开头，后面没有分号，一般位于函数首部之前。#include 称为文件包含命令，其作用是将头文件 stdio. h 的内容包含进来。该命令也可写为 #include <stdio. h>。

（5）若将函数体中唯一的语句改为

```
printf("hello!\n");
```

其中\n 是换行符，则在输出 hello!后回车换行。

（6）程序改为

```
#include "stdio.h"
main()
{
  printf("hello!\n");
  printf("******\n");
}
```

则运行结果为：

```
hello!
******
```

（7）将(6)函数体中的两条语句改为下面的一条语句

```
printf("hello!\n******\n");
```

输出效果是一样的，但(6)读起来更清晰、易懂，建议采用(6)的方式。

1.2.2 基本整型与%d格式符

【例 1-2】 基本整型变量的定义和输出。

```
#include "stdio.h"
main()
{
   int a;                /*第 4 行,定义一个基本整型变量 a*/
   a=23;                 /*a 赋值为 23*/
   printf("%d\n",a);     /*用%d 格式符输出基本整型变量 a 的值*/
}
```

运行结果为：

```
23
```

【说明】

(1) 第4行是本程序的**声明部分**，**int** 是**基本整型**的类型标识符，该行定义 a 为基本整型变量。声明也由分号结束。在 Turbo C 中基本整型变量可以存放－32768～32767范围内的整数，在 Visual C++ 中可以存放－2147483648～2147483647 范围内的整数。在一个函数中，声明部分放在所有语句的前面。

(2) /*……*/是**注释**，可以根据需要加在程序中的任何位置。注释是给阅读程序的人看的，计算机并不编译也不执行。在 Visual C++ 中，还可以使用行注释//，注释范围从//起至换行符止。所以，第4行也可改为

```
int a;     //第 4 行,定义一个基本整型变量 a
```

(3) **%d** 是基本整型格式符，用于输入/输出基本整型数据。执行 printf 函数时将%d的位置换成双引号后对应变量的具体数值输出。

(4) 将最后一行语句改为

```
printf("%4d\n",a);
```

则运行结果为：

```
  23
```

%4d 指定输出数据占4列。a 的值是23，只有2列，故在其前补以两个空格，然后输出23，于是该数据的输出共占4列。若 a 的值是－2356，则运行结果为：

```
-2356
```

即数据的实际位数大于给定的列数时，按实际位数输出。

1.2.3 加、减、乘、除运算符和算术表达式

加、减、乘、除运算符分别是＋、－、*、/，它们都是基本算术运算符。用算术运算符和

括号将运算对象（也称操作数）连接起来的、符合语法规则的式子称为**算术表达式**。

【例 1-3】 加、减、乘、除运算示例。

```
#include "stdio.h"
main()
{  int a,b,s;                    /*定义 3 个整型变量*/
   a=5;b=3;                      /*a 赋值为 5,b 赋值为 3*/
   s=a+b;                        /*第 5 行,计算 a、b 之和*/
   printf("s=%d\n",s);           /*第 6 行,输出结果*/
}
```

运行结果为：

```
s=8
```

【说明】

（1）printf 函数格式说明（即双引号""括起来的部分）中的普通字符 s＝原样输出，使运行结果看起来更清晰。若将第 6 行语句改为：

```
printf("sum=%d\n",s);
```

则运行结果为：

```
sum=8
```

（2）将第 5 行改为

```
s=a-b;                           /*计算 a、b 之差*/
```

则运行结果为：

```
s=2
```

（3）将第 5 行改为

```
s=a*b;                           /*计算 a、b 之积*/
```

则运行结果为：

```
s=15
```

（4）将第 5 行改为

```
s=a/b;                           /*计算 a、b 之商*/
```

则运行结果为：

```
s=1
```

需要特别注意的是：类型相同的两个量做基本算术运算，结果值仍为原类型。所以 5/3 的结果值为整数 1，而不会是实数 1.666667。

C 语言规定了所有运算符的**优先级**和**结合方向**（见附录 C）。计算表达式的值时，按

运算符的优先级顺序执行。乘法和除法运算符的优先级相同，加法和减法运算符的优先级相同，而乘、除运算符的优先级高于加、减运算符，所以当它们同时出现在一个表达式中时，先做乘除，后做加减。

如果两个运算符的优先级相同，则根据运算符的结合方向处理。基本算术运算符的结合方向都是自左至右，即先左后右。如表达式为 a+b-3，则先做 a+b，然后再减 3。

自左至右的结合方向又称**左结合性**，简称**左结合**。类似地，自右至左的结合方向又称**右结合性**，简称**右结合**。

1.2.4 单精度浮点型与%f格式符

【例 1-4】 单精度浮点型变量的定义和输出。

```
#include "stdio.h"
main()
{  float a,b,s;                  /*定义 3 个单精度浮点型变量*/
   a=5.0;b=3.0;                  /*a 赋值为 5.0,b 赋值为 3.0*/
   s=a/b;                        /*计算 a、b 之商*/
   printf("s=%f\n",s);           /*用%f 格式符输出单精度浮点型变量 s 的值*/
}
```

运行结果为：

```
s=1.666667
```

【说明】

(1) **float** 是**单精度浮点型**的类型标识符，被定义为单精度浮点型的变量可以存放 -3.4×10^{38}～3.4×10^{38} 范围内的实数。

(2) **%f** 是单精度浮点型格式符，用于输入/输出单精度浮点型数据。输出时，整数部分全部输出，并输出 6 位小数。

(3) 将最后一条语句改为

```
printf("s=%7.2f\n",s);
```

则运行结果为：

```
s=   1.67
```

%7.2f 中的 7 表示该数据所占列数，2 表示保留两位小数，对小数点后第 3 位进行四舍五入。如果输出数据长度小于指定列数，则在其前补以空格。如该例中，在其前补以 3 个空格。

格式输出函数 printf 的一般形式是

```
printf("格式控制字符串",输出项 1,输出项 2, …,输出项 n)
```

作用是按照格式控制向终端显示器(系统隐含指定的输出设备)输出各输出项的值。其中