高等院校规划教材

# C语言程序设计

江宝钏　主　编

陈叶芳　贾晓雯　陈金飚　管博　副主编

高等院校规划教材

# C语言程序设计

江宝钏　主　编

陈叶芳　贾晓雯　陈金飚　管博　副主编

清华大学出版社

北　京

## 内 容 简 介

本书以程序实例驱动和知识点相结合为编排主线，通过实例程序引入知识点的讲解，侧重学生编程能力的提高。全书共分 10 章，分别介绍 C 语言程序设计的发展概况与简单 C 语言程序的运行；数据类型、表达式；C 语言基本语句的使用；选择结构、循环结构程序设计方法；函数和编译预处理；数组、指针和字符串的使用；结构体和链表的使用，共用体和枚举类型（仅作简单的介绍）；文件的类型与操作。本书每一章都有详细的程序范例和运行结果。

本书适合作为各类高等院校计算机专业及理工类非计算机专业的 C 语言程序设计课程的教材，也可作为程序设计爱好者的自学用书。



**图书在版编目（CIP）数据**

C 语言程序设计 / 江宝钏主编．--北京：清华大学出版社，2010.3
ISBN 978-7-302-21664-3

Ⅰ．①C…　Ⅱ．①江…　Ⅲ．①C 语言－程序设计　Ⅳ．①TP312

中国版本图书馆 CIP 数据核字(2010)第 020564 号

**责任编辑**：孟毅新
**责任校对**：李　梅
**责任印制**：何　芊

**出版发行**：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**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

**印 装 者**：三河市春园印刷有限公司
**经　　销**：全国新华书店
**开　　本**：185×260　　**印　　张**：15.25　　**字　　数**：370 千字
**版　　次**：2010 年 3 月第 1 版　　**印　　次**：2010 年 3 月第 1 次印刷
**印　　数**：1～4000
**定　　价**：25.00 元

# 前　言

C语言程序设计是高校计算机专业和理工类非计算机专业重要的计算机基础的必修课程，一般都是作为计算机程序设计的第一门语言课程。通过对本课程的学习，使学生学会提出问题，解决问题，并提高他们综合分析问题的能力；还要让学生掌握程序设计的思想和方法，提高他们的实际编程能力。

本书是把以程序实例为驱动方式和以知识点为主线的编排方式结合起来，用循序渐进的方法，以提高学生的实际编程能力为目的安排教材内容，从编程主线、知识点和应用面3个层次逐步提高学生的编程能力。

在内容的组织上按照“实例问题的提出与解决→引出相关语法知识→归纳总结→举例说明→修改与扩展→实践训练”来编排。重点讲解与分析程序设计的过程，辅以语言知识的介绍。

本书将淡化C语言的复杂语法知识、复杂的数值计算方面的应用，举例以通俗易懂、贴近生活和富有趣味性的内容为主，在讲解C语言语法知识的过程中，给出重要的知识点和经典的例题分析，并让学生对例题进行修改和扩展，指导学生如何分析问题、组织数据、设计算法和编写程序以及调试、运行与测试，强调对学生应用编程能力的培养。对于难度较大或复杂的语法叙述的内容做*标注，以便教师根据学生情况和教学时数做适当调整，具有较大的灵活性，为差异性教学做好教材方面的准备工作，以便学生能较自主地选择想学并且能够学的东西，能够为在一个更高的层次上运用计算机解决专业问题打下基础。

本书具有如下特色。

(1) 尽量淡化语法规则复杂的叙述。对于前面4章，语法规则和输入输出仅做较简单叙述，考虑首先不能让学生陷入啰唆的语法规则之中，如自增自减问题，不能让学生以为学习C语言程序设计就是为了学和记这些规则。

(2) 每章开头都设“内容要点提示”、“问题的提出与示例程序分析”来引入内容，让学生知道为什么要学习本章内容，通过实例程序分析对本章的内容大概有些了解，先把学生引进门。

(3) 让学生多模仿和修改程序。在编程的开始阶段有意识地编排一些简单程序让学生多模仿和修改原来的程序和数据，尽量让学生尽快进入编程入门状态，让学生有一定的成功感，从而获得进一步学下去的愿望和动力。准备组织编写的实验教材中还将加强实例程序的调试和分析、实例程序的模仿和修改。

(4) 设“程序运行结果的示意截屏图”。每个实例程序都将设置“运行结果的示意截屏图”，不同的输入参数，不同的输出结果，这样让学生感觉有真实的运行结果。大部分程序实例将提出相关的思考或问题，改变一些参数、条件或扩展一些解题思路，让学生试一试。本书中所有的例题都在VC++ 6.0中上机调试通过，可直接下载使用，学生在此基础上直接修改即可。

(5) 单独设“提高部分”。考虑到学生的差异性和教学时数的限制，在每章的最后一节

设“深入探讨”或注 *，把一些相对复杂的内容放入，前面几章主要放入一些复杂的、不太常用的语法规则，后面几章放入一些不太常用或较难的内容，学生开始学习阶段完全可以抛开这些细节，教师可以根据学生的情况做灵活的调整。

本书由江宝钏担任主编，其他编者均是具有丰富计算机程序设计教学经验的一线教师。第 1 章、第 3 章、第 7 章、第 9 章前半部分、附录由江宝钏编写，第 8 章和第 9 章后半部分由陈叶芳编写，第 2 章由管博编写，第 4 章由王爱冬编写，第 5 章由贾晓雯编写，第 6 章由裘姝平编写，第 10 章由陈金飚编写。

在本书的编写过程中得到了宁波大学王让定教授的帮助与支持，还得到了宁波大学计算机科学与技术国家特色专业的建设经费资助；李纲、邬延辉、沈明忻、董一鸿、石守东、王亚猛、杨任尔等对本书的大纲和内容提出了许多修改意见和建议，在此表示衷心的感谢。

本书经过多次修改讨论而定稿，限于编者的水平，书中难免存在疏漏与不足之处，恳请读者批评指正，以便进一步修订。

作者联系地址：浙江省宁波大学信息科学与工程学院，邮编：315211，E-mail：jiangbaochuan@nbu.edu.cn。

作　者

2009 年 11 月

# 目　录

# 第 1 章　C 程序设计语言概述

**内容要点提示：**

- 一个 C 语言程序的基本组成结构是什么？
- 运行 C 语言程序需要哪些步骤？

要让计算机为人们完成各种各样的工作，就必须让它执行相应的程序，这些程序都是人通过程序设计语言编写出来的。

所谓程序，实际上是用计算机语言描述的解决某一问题的步骤，是符合一定语法规则的有序的符号序列。人们借助程序设计语言告诉计算机要做什么以及如何做。通过在计算机上运行程序，向计算机发出一系列指令，让计算机按人们的要求解决问题。本章主要简述程序设计语言的发展概况、C 语言程序的基本结构、如何运行简单的 C 程序。

## 1.1　程序设计语言的发展概况

不同的问题可以用不同的程序设计语言来解决，即为了解决某一个问题所编写的程序并不是唯一的，不同的程序有不同的解决方法，有不同的效率。程序设计语言的发展经历了从机器语言、汇编语言到高级语言的历程。

（1）机器语言

直接使用二进制表示的指令来编程的语言即机器语言。由“0”和“1”组成的一个二进制编码，表示一条机器指令，使计算机完成一个简单的操作。用机器指令编写的程序，即为机器语言程序，是计算机唯一能够直接识别执行的程序。

用机器语言编写程序，编程人员要首先熟记所用计算机的全部指令代码和代码的含义，还得记住编程过程中每步所使用的工作单元处在何种状态。这是一件十分烦琐的工作，编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且，编出的程序全是些 0 和 1 的指令代码，直观性差，容易出错，且不同的机型有不同的机器指令。目前很少有人直接用机器语言编程。

（2）汇编语言

汇编语言是指用比较容易识别、记忆的助记符替代特定的二进制串。通过助记符，人们可以较容易地读懂程序，调试和维护也较为方便。但这些助记符号计算机无法识别，需要一个专门的程序将其翻译成机器语言，这种翻译程序被称为汇编程序。汇编语言与机器语言性质上是一样的，只是表示方式做了改进，可移植性与通用性仍然很差。

（3）高级语言

汇编语言和机器语言是面向机器的，同属于低级语言。高级语言是一种用能表达各种意义的“词”和“数学公式”按一定的“语法规则”编写程序的语言，也称为高级程序设计语言或算法语言，这类语言接近于人的自然语言。半个多世纪以来，有几百种高级语言问世，影

响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、PL/1、Pascal、C、PROLOG、Ada、C++ 、Visual C++ 、Visual Basic、Delphi、Java 等。高级语言的发展也经历了从早期语言到结构化程序设计语言，再到面向对象程序设计语言的过程。高级语言的使用，大大提高了程序编写的效率和程序的可读性。

对于用高级语言写成的源程序，通常每一条语句对应一组机器指令，它必须经过翻译程序，翻译成机器语言指令形式才能执行。翻译程序有编译程序和解释程序两种。每种高级语言都有自己的翻译程序，用 C 语言编写的源程序，需通过 C 编译程序，将整个源程序翻译成机器语言程序，通过连接程序生成可执行程序，在机器上运行。

C 语言是 20 世纪 70 年代初由美国贝尔实验室在 B 语言的基础上发展起来的，它保持了 B 语言精练、接近硬件的特点，又改进了 B 语言过于简单的缺点。早期的 C 语言主要是在贝尔实验室内部使用。1978 年以 C 语言编译程序为基础，B. W. Kernighan 和 D. M. Ritchie 合著了著名的“The C Programming Language”一书。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础。但是，在这本书中并没有定义一个完整的标准 C 语言，后来由美国国家标准协会(ANSI)在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称之为 ANSI C。

早期的 C 语言主要是应用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识，到了 20 世纪 80 年代，C 开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

在 C 的基础上，贝尔实验室在 1983 年推出了 C++ 。C++ 进一步扩展和完善了 C 语言，成为一种面向对象的程序设计语言。

## 1.2 简单的 C 语言程序

### 1.2.1 简单 C 语言程序实例

先看几个简单的 C 语言程序实例，然后从中分析 C 语言程序的特点。

**【例 1-1】** 在屏幕上输出：Hello,Word!。

```
/* 源程序: exp1_1.cpp */
#include<stdio.h>                    /* 编译预处理命令 */
void main()                          /* 定义主函数 main,void 表示没有函数返回值 */
{   printf("这是一个最简单的屏幕输出程序\n");      /* 输出双引号中的文字到屏幕 */
    printf("Hello, World!");                      /* 输出 Hello, World!到屏幕 */
}
```

运行结果如图 1-1 所示。

图 1-1 程序 exp1_1.cpp 的运行结果

这是一个简单的 C 程序。该程序的功能是在屏幕上输出“Hello,Word!”。下面简单分析一下程序的构成。

(1) “#include<stdio.h>”是编译预处理命令，其目的是使输入输出能正常执行。

(2) main 是主函数的函数名，每一个 C 源程序都必须有，且只能有一个主函数，函数后面必须有一对圆括号。main 前面的 void 表示函数返回值为“空类型”，即执行本函数后没

有函数返回值。

(3) 一对花括号内的部分称为函数体。

(4) 程序的执行总是从 main 函数的花括号“{”开始到 main 函数的花括号“}”结束。

(5) C 语言的关键字和特定字使用小写字母。如 main、include 是特定字，必须用小写字母。在 C 语言中是要区分大小写的。

**【例 1-2】** 一个简单的计算程序：已知圆的半径，求圆的面积。

```
/* 源程序: expl_2.cpp */
#include<stdio.h>
void main()
{
    float  r,s;                          /* 定义变量 */
    printf("请输入圆的半径: \n");          /* 在屏幕上显示提示信息 */
    scanf("%f", &r);                     /* 从键盘输入半径值给变量 r */
    s=3.1415*r*r;                        /* 计算面积 */
    printf("s=%f\n", s);                 /* 输出面积到屏幕 */
}
```

运行结果如图 1-2 所示。

程序由一个主函数 main 组成。函数体部分由 5 条语句组成。

(1) “float r,s;”是定义变量的语句。变量是内存中的存储单元，能够存储供程序使用的数据，变量必须先定义后使用。

(2) “scanf("%f", &r);”语句用于要求用户从键盘上输入圆的半径给变量 r。

(3) “s=3.1415*r*r;”用于在已知 r 的情况下计算圆面积，并把结果存放到变量 s 中，C 语言中“*”表示数学中的乘号。

(4) scanf、printf 是 C 语言中最常用的输入输出函数，用来输入输出数据。

(5) “/*文字……*/”是注释，不是程序部分，在程序执行中不起任何作用，只为增加程序的可读性。

图 1-2　程序 expl_2.cpp 的运行结果

图 1-3　程序 expl_3.cpp 的运行结果

**【例 1-3】** 输入两个整数，计算并在屏幕上输出它们的和值。

```
/* 源程序: expl_3.cpp */
//这是一个计算两数之和的程序
#include<stdio.h>
void main()
{
   int a,b,sum;                          //定义变量
   printf("请输入两个整数:\n");           //提示用户输入数据
   scanf("%d%d",&a,&b);                  //调用输入函数,要求用户从键盘输入两个整数
   sum=a+b;
   printf("sum=%d\n",sum);               //输出和值
}
```

运行结果如图 1-3 所示。

本程序的作用是求两个整数之和。“int a,b,sum”是声明部分,声明变量 a、b 和 sum 为整型(int)变量;接下来的 printf 语句是在屏幕上输出提示信息;scanf 语句是调用库函数让用户从键盘读入两个整型数据;“sum＝a＋b;”指的是把 a、b 两个变量的值求和赋给变量 sum;最后是在屏幕上输出和值。注释也可用“//”,程序中从“//”开始到行尾都为注释文字。注释对编译和运行不起作用,注释可以放在程序中的任何位置。

### 1.2.2 C 语言程序的组成结构

C 程序由函数、编译预处理命令及注释 3 部分组成。

一个完整的 C 程序可以由一个或多个函数组成,其中主函数 main 必不可少,且只能是唯一的。C 程序执行时,总是从主函数 main 开始,与 main 函数在整个程序中的位置无关。

C 语言程序的基本结构形式如下:

```
#include<stdio.h>                          /* 编译预处理命令 */
void main()
{
    定义部分                               /* 这是程序的定义部分 */
    语句序列                               //这是程序的语句部分
}
```

**1. 编译预处理命令**

程序中每一个以“#”号开头的命令行,是编译预处理命令,一般放在程序的最前面。不同的编译预处理命令完成不同的功能。如“#include ＜stdio.h＞”命令的作用是将特定目录下的“stdio.h”文件嵌入到源程序中。

**2. 函数**

(1) 函数首行,即函数的第一行,如 void main(),包括函数类型、函数名、函数参数、参数类型等。

(2) 函数体,是函数首行下面花括号对中的内容。函数体由各类语句组成,执行时按语句的先后次序依次执行,各语句间用分号“;”结束。

**3. 注释**

注释不是程序部分,在程序执行时不起任何作用,其作用是增加程序的可读性,方便别人的阅读或自己以后的回顾。C 语言有以下两种注释方法。

(1) “/* 注释内容 */”:适用于注释多行,“/*”和“*/”之间的内容即为注释。

(2) “//注释内容”:适用于注释单行,“//”后面的部分(行)即为注释。

其中,“注释内容”可以是汉字或西文字符。

### 1.2.3 C 语言的特点

C 语言是一种通用、灵活、结构化和使用普遍的计算机高级语言,既可作为系统软件的描述语言,也可用来开发应用程序,特别适合进行系统程序设计和对硬件进行操作的场合。

C 语言的主要特点如下。

(1) C 语言简洁紧凑,使用方便

标准 C 语言(ANSI C)只有 32 个关键字,9 种控制语句。书写形式自由,一行可以书写多条语句,一个语句也可写在不同行上。

(2) C 语言介于汇编语言与高级语言之间

C 语言既像汇编语言那样允许直接访问物理地址,能进行位运算,能实现汇编语言的大部分功能,直接对硬件访问;也有高级语言面向用户、容易记忆、容易学习及易于书写的特点。

(3) C 语言是一种结构化语言

C 语言的主要成分是函数。函数是 C 语言程序的基本结构模块,程序可以由不同功能的函数有机组装而成,从而可以达到结构化程序设计中模块的要求。另外,C 语言提供了 3 种基本结构(顺序、分支、循环),使程序流程具有良好的结构性。

(4) C 语言有丰富的数据类型

C 语言具有现代化语言的各种数据类型;用户能自己扩充数据类型,实现各种复杂的数据结构,完成用于具体问题的数据描述。尤其是指针类型,是 C 语言的一大特色,灵活的指针操作,能够高效处理各种数据。

(5) C 语言具有较高的移植性,目标代码质量高,运行效率高

在 C 语言中,没有专门与硬件有关的输入输出语句,程序的输入输出通过调用库函数实现,使 C 语言本身不依赖于硬件系统,这大大提高了程序的可移植性。用 C 语言编写的程序,其生成的目标代码质量高,程序的运行效率高。

但 C 语言也存在一些缺点,如运算优先级太复杂,语法定义不够严格,数值运算能力相对薄弱等。

## 1.3　运行 C 程序的步骤与方法

### 1.3.1　运行 C 语言程序的步骤

高级语言处理系统,主要由编译程序、连接程序和函数库组成。如果要使 C 程序在一台计算机上执行,必须经过编辑源程序、编译、连接及调试运行几个阶段,最后得到可执行程序。

**1. 编辑**

编辑是创建或修改 C 源程序文件的过程,C 源程序以文本的形式存储在磁盘上,文件名的扩展名为. c 或. cpp。

**2. 编译**

C 语言是计算机高级语言,其源程序必须经过编译程序对其进行编译,生成目标程序,目标程序文件的扩展名为. obj。

**3. 连接**

编译生成的目标程序机器可以识别,但不能直接执行,由于程序中使用到一些系统库函数,还需将目标程序与系统库文件进行连接,经过连接后,生成一个完整的可执行程序,可执行程序的扩展名为. exe。

**4. 运行**

C 源程序经过编译、连接后生成的可执行文件，可脱离编译系统直接执行，输入可执行文件名或在 Windows 资源管理器下双击可执行文件名即可。图 1-4 给出了实现 C 程序运行的过程。

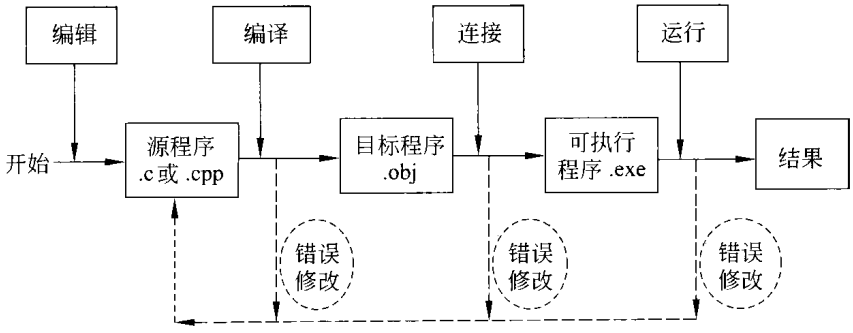


图 1-4 实现 C 程序运行的过程步骤

图 1-4 中，当编译或连接时出现错误，说明 C 程序编写时有语法、句法错误；若在运行时出现错误或结果不正确，说明程序设计上有逻辑错误，都需要返回编辑状态修改源程序并重新编译、连接和运行，直至将程序调试正确为止，最后得到. exe 可执行文件，该文件可以脱离 C 编译系统，直接在计算机上运行。目前大多数的 C 语言编译程序都将这 4 个功能集成在一个全屏幕的环境下，编辑、编译、修改错误，最后运行程序非常方便。

由于操作系统不同，或系统中安装了不同版本的 C 语言处理系统，所使用的 C 语言支持环境也会有所不同。常用的 C 语言编译环境有 Turbo C、Visual C++ 、C-Free 等。本教材中介绍 Visual C++ 6.0 运行环境。

## 1.3.2 在 Visual C++ 6.0 下运行 C 程序

Visual C++ 6.0 是一款功能强大的，面向对象的程序设计语言，是 C++ 的版本，但是 C++ 语言是在 C 语言的基础上扩展而成，所以 C 程序也能在该环境下运行 C 语言程序，但只是 Visual C++ 6.0 软件强大功能中的很小方面的应用。

Visual C++ 6.0 提供了全屏幕程序调试环境，编辑、编译、连接和运行都可以在该环境下完成，在这个环境下编辑的 C 语言源程序的后缀一般是. cpp。

**1. 启动 Visual C++ 6.0**

在 Windows 环境下选择“开始”→“程序”→Microsoft Visual Stdio 6.0→Microsoft Visual C++ 6.0 命令。启动 VC++，出现 VC++ 集成开发环境的操作窗口，如图 1-5 所示。

**2. 新建/打开调试 C 程序文件**

(1) 新建 C 程序文件

① 选择“文件”→“新建”命令。

② 打开“文件”选项卡，如图 1-6 所示。选择 C++ Source File 项，输入想保存的文件名，选中对应的文件夹，单击“确定”按钮，即可在编辑窗口输入程序。