



普通高等教育“十二五”规划教材

C语言 程序设计与实践

周传生 屈 巍 李 晖 赵娇洁 王晓丹 编著

清华大学出版社

普通高等教育“十二五”规划教材

C语言

程序设计与实践



周传生 屈巍 李晖 赵娇洁 王晓丹 编著

清华大学出版社
北京

内 容 简 介

C 语言作为一种通用程序设计语言,简捷、高效、使用广泛,既适于编写应用软件,又适于编写系统软件。对于计算机应用人员来说,掌握 C 编程知识与技能是至关重要的。本书较全面地讲述了 C 语言及其程序设计方法。全书包括三个部分,第一部分理论篇详细介绍了 C 语言程序设计,程序的简单算法设计,数据类型,运算符与表达式,顺序结构,选择结构,循环结构,数组,函数,指针,用户自定义类型,数据文件等内容;第二部分实验篇为上机指导内容,实验项目按照“由浅入深、循序渐进”的原则设置,用于对第一部分学习内容进行巩固与提高;第三部分实训篇,在第二部分的基础上着重培养学生 C 语言编程应用能力,通过实训项目的训练提高学生分析问题和实际动手解决问题的能力。

本书力求概念叙述准确、严谨,语言通俗易懂。适合作为高等学校相关课程的教材,也可以作为计算机等级考试的辅导用书,以及相关研究人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计与实践/周传生等编著. —北京:清华大学出版社,2014

(普通高等教育“十二五”规划教材)

ISBN 978-7-302-37331-5

I. ①C… II. ①周… 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(2014)第 159649 号

责任编辑:朱敏悦

封面设计:汉风唐韵

责任校对:王凤芝

责任印制:沈 露

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

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24 字 数:483 千字

版 次:2014 年 9 月第 1 版 印 次:2014 年 9 月第 1 次印刷

印 数:1~3000

定 价:49.00 元

前 言

C 语言是国内外广泛使用的主程序设计语言之一,其功能丰富、表达能力强、使用灵活方便、应用面广、可移植性好,既具有高级语言的特性,又具有直接操纵计算机硬件的能力,因此特别适用于编写系统软件。目前,C 语言被许多高校列为程序设计课程的首选语言。

C 语言程序设计是一门注重实践应用的课程,学生经过基础理论知识学习,虽然能够了解和掌握一些语句的语法知识和语义,但是要想真正应用 C 语言解决实际问题,必须通过大量的编程训练来实现,在实践中掌握基础知识,培养程序设计的基本能力,并逐步掌握程序设计的思想和方法。

编写本书是为了让学生在加强基础理论知识学习的基础上,提升实际编程、动手解决问题的能力。为实现该目标,我们采用如下策略:

(1) 在介绍基础语法知识的同时,阐述基本的编程技巧和注意事项,注重基础语法知识的实际应用。

(2) 对于基础理论知识,配有相应例题辅助阐述,使得知识点清晰、直观,便于理解。

(3) 每章设置应用实例,注重实际应用,让学生更快地将所学理论知识与实践应用相结合。

(4) 每章后设置相应的习题,方便学生检测自己的学习水平,加深对基础理论知识的理解与巩固。

(5) 结合理论知识与实际应用设置实验,使得学生学习的理论知识得以巩固与提高。

(6) 设置综合实训案例分析及实训项目练习,有利于提高学生的实际编程与解决实际问题的能力。

全书以程序设计为主线,以编程应用为驱动,重点介绍程序设计的基本思想和基本方法。本书分为理论篇、实验篇和实训篇三部分,其中理论篇分为 11 章,第 1~6 章侧重于介绍 C 语言的基础知识和基本编程技能,主要包括 C 语言程序设计、程序的简单算法设计、数据类型、运算符与表达式以及顺序、分支、循环 3 种基本的流程控制语句。第 7~11 章侧重于指针和数组、函数、用户自定义类型、文件等知识的综合运用,采用结构化程序设计的思想来实现复杂问题的编程和基本算法等。实验篇以训练学生的实际编程能力为目的,结合理论篇基础知识设置了 11 个实验。实训篇结合实际应用设置了 2



个具有代表性的综合实训。

本书编者全部来自从事 C 语言教学工作的一线教师,有着丰富的教学经验。在本书的编写过程中,我们的同事孙笑微、高源、张悦、王艳等老师提供了很多宝贵的意见和建议,在此表示衷心的感谢!

尽管编者们在选词造句、资料的规范化等方面都做了大量工作,但由于学识和见闻的局限,书中不足之处在所难免,恳请广大读者和同行批评指正,多提宝贵意见,编者将不胜感激!

编 者

2014 年 4 月

目 录

第一篇 理 论 篇

第 1 章 C 语言程序设计概述	(2)
1.1 C 语言发展简史	(2)
1.2 C 语言特点	(3)
1.3 C 语言程序的组成	(4)
1.4 C 语言程序的开发步骤	(7)
习题	(8)
第 2 章 程序的简单算法设计	(11)
2.1 结构化程序的算法设计	(11)
2.2 结构化算法的描述方法	(12)
2.3 程序设计典型算法	(16)
习题	(17)
第 3 章 数据类型、运算符与表达式	(18)
3.1 C 语言的标识符和关键字	(18)
3.2 C 语言的数据类型	(20)
3.3 常用运算符及表达式	(27)
3.4 应用实例	(38)
习题	(40)
第 4 章 顺序结构	(45)
4.1 C 语句	(45)
4.2 输入输出的实现	(51)
4.3 顺序结构程序设计举例	(59)
4.4 应用实例	(61)
习题	(64)



第 5 章 选择结构	(67)
5.1 选择结构的应用环境	(67)
5.2 选择语句	(68)
5.3 应用实例	(77)
习题	(81)
第 6 章 循环结构	(87)
6.1 循环结构的应用环境	(87)
6.2 while 语句	(89)
6.3 do... while 语句	(92)
6.4 for 语句	(94)
6.5 循环语句的比较	(101)
6.6 循环嵌套	(103)
6.7 转向语句	(106)
6.8 应用实例	(110)
习题	(113)
第 7 章 数组	(116)
7.1 数组的概念	(116)
7.2 一维数组	(117)
7.3 二维数组	(122)
7.4 字符数组	(128)
7.5 应用实例	(143)
习题	(145)
第 8 章 函数	(149)
8.1 函数的基本概念	(149)
8.2 数组作为函数参数	(164)
8.3 函数的嵌套调用和递归调用	(168)
8.4 变量的作用域和存储方法	(175)
8.5 内部函数和外部函数	(184)
8.6 编译预处理	(185)
8.7 应用实例	(191)
习题	(193)
第 9 章 指针	(197)
9.1 指针的概念及定义	(197)
9.2 指针运算	(199)

9.3 指针变量与数组	(204)
9.4 指针与字符串	(210)
9.5 指针与函数	(212)
9.6 指针数组和指向指针变量的指针	(218)
9.7 内存空间的动态分配	(224)
9.8 main 函数的参数	(229)
9.9 应用实例	(231)
习题	(234)
第 10 章 用户自定义类型	(239)
10.1 结构体类型和结构体变量	(239)
10.2 结构体数组	(245)
10.3 结构体和指针	(247)
10.4 共用体类型	(251)
10.5 枚举类型	(254)
10.6 用 typedef 声明新类型名	(256)
10.7 应用实例	(258)
习题	(261)
第 11 章 数据文件	(268)
11.1 文件的基础	(268)
11.2 文件的操作	(271)
11.3 文件读写的出错检测	(285)
11.4 应用实例	(286)
习题	(289)

第二篇 实 验 篇

实验 1 在 Visual C++ 6.0 环境下运行 C 程序的步骤	(294)
实验 2 数据类型、运算符与表达式	(301)
实验 3 输入输出函数	(304)
实验 4 顺序结构程序设计	(307)
实验 5 选择结构程序设计	(308)
实验 6 循环结构程序设计	(310)
实验 7 数组及应用	(312)
实验 8 函数	(314)



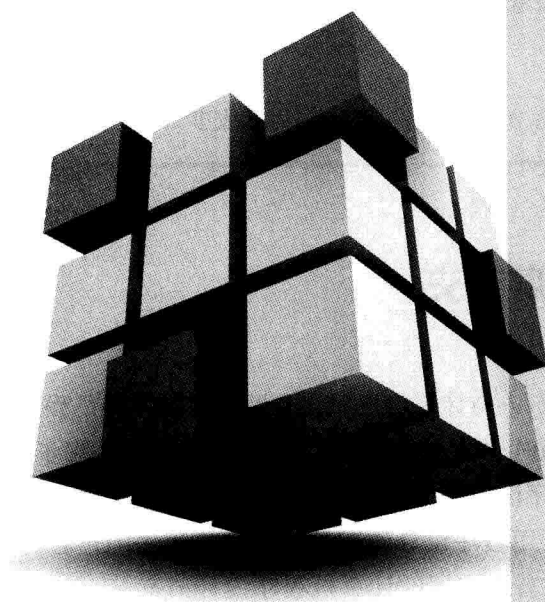
实验 9 指针及应用	(315)
实验 10 结构体与共用体	(317)
实验 11 读写文件	(320)

第三篇 实 训 篇

实训一 学生成绩管理系统	(324)
实训二 贪吃蛇游戏	(355)
附录 A C 语言的关键字	(370)
附录 B ASCII 字符表	(371)
附录 C 运算符的优先级与结合性	(375)
参考文献	(376)

第一篇

理论篇





第1章 C语言程序设计概述

C语言是20世纪70年代初由美国贝尔实验室的D. M. Ritchie在B语言基础上设计出的程序设计语言,具有功能丰富、使用灵活方便、应用面广、可移植性好等优点。C语言既具有高级语言的特性,又具有直接操纵计算机硬件的能力,因此特别适用于编写系统软件。



学习目标

- ◇ 理解C语言的发展过程及特点。
- ◇ 掌握C语言程序结构。
- ◇ 重点掌握C语言的开发步骤。



知识点

C语言的特点、C语言程序结构、C语言的开发步骤。

1.1 C语言发展简史

在计算机科学领域,人与计算机之间交流最通用的手段是程序设计语言。当人们想用计算机解决某个问题时,必须使用程序设计语言安排好处理步骤,并存入计算机内供计算机执行,这一用程序设计语言安排好的处理步骤称为计算机程序。计算机程序是计算机操作指令的集合。用程序设计语言编制计算机程序的过程称为程序设计。C语言是一种得到广泛重视并普遍应用的计算机程序设计语言,适合作为系统描述语言使用,既可以用来编写系统软件,也可以用来编写应用软件。

早期的操作系统等系统软件主要采用汇编语言编写,例如大家熟知的UNIX操作系统。由于汇编语言依赖于计算机硬件,程序的可读性和可移植性都较高级程序设计语言相差甚远,但是一般的高级语言又难以实现汇编语言的某些功能,例如直接对硬件进行操作。因此,人们设想能否找到一种既具有一般高级语言特性,又具有低级语言特性的

语言,C语言便在这种需求下应运而生。

C语言的前身是ALGOL 60,ALGOL 60是1960年出现的一种面向问题的高级语言,离硬件较远,不宜用来编写系统程序。1963年剑桥大学将ALGOL 60语言发展成为CPL(combined programming language)语言。CPL语言更接近硬件,但规模较大,实现较难。1967年剑桥大学的Martin Richards对CPL进行简化,形成了BCPL(basic combined programming language)语言。1970年美国贝尔实验室的Ken Thompson对BCPL进行修改,设计出了简单且接近硬件的B语言,并用B语言编写了第一个UNIX操作系统。

1973年美国贝尔实验室的D. M. Ritchie在B语言基础上设计出了C语言。C语言既保持了BCPL和B语言的精练、接近硬件的优点,又克服了它们过于简单、数据无类型的缺点。1977年Dennis M. Ritchie发表了不依赖于具体机器系统的C语言编译文本《可移植的C语言编译程序》。1978年Brian W. Kernighan和Dennis M. Ritchie合著了影响深远的《The C Programming Language》一书,1983年美国国家标准协会(ANSI)在此基础上制定了第一个C语言标准草案('83 ANSI C),从而使C语言成为目前世界上流行最广泛的高级程序设计语言。

1989年ANSI公布了一个完整的C语言标准ANSI X3. 159—1989(C89),1990年国际标准化组织(ISO)接受C89作为国际标准ISO/IEC9899:1990,它和ANSI的C89基本上是相同的。1995年,ISO对C90做了一些修订,即“1995基准增补1(ISO/IEC9899/AMD1:1995)”。1999年ISO又对C语言标准在基本保留原来C语言特征基础上,针对应用的需求,增加了一些功能,尤其是C++中的一些功能,命名为ISO/IEC 9899:1999。2001年和2004年先后进行了两次技术修正,即2001年的TC1和2004年的TC2。ISO/IEC 9899:1999及其技术修正被称为C99,C99是C89(及1995基准增补1)的扩充。本书的叙述以C99标准为依据。

目前由不同软件公司所提供的一些C语言编译系统并未完全实现C99建议的功能,它们多以C89为基础开发,本书所举的示例程序是在Visual C++ 6.0上编译和运行的。

1.2 C语言特点

C语言是一种结构化的程序设计语言,简明易懂,功能强大,适合于各种硬件平台。C语言区别于一般高级语言:C语言既具有高级语言的特点,又具有低级语言的功能。

C语言的主要特点如下:

(1) 程序设计结构化

用函数作为程序的模块化单位,每个模块具有不同的功能,程序将不同功能的模块有机结合,通过模块间相互协同,共同完成程序所要完成的任务。模块化的程序设计方



式使得 C 语言程序易于调试和维护。

(2) 运算符丰富

C 语言共有 34 种运算符,C 语言把括号、赋值、逗号等作为运算符处理,使得 C 语言运算符极其丰富,可以实现其他高级语言难以实现的一些运算。

(3) 数据类型丰富

C 语言除具有系统自身规定的数据类型外,还允许用户定义数据类型,以满足程序设计需要。

(4) 书写灵活

只要符合 C 语言的语法规则,程序书写格式限制不严格。

(5) 允许直接访问物理地址

C 语言能进行位操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。

(6) 适应性广泛

生成目标代码质量高,执行效率高,程序可移植性和硬件控制能力强。

1.3 C 语言程序的组成

1.3.1 简单 C 语言程序举例

为了说明 C 语言的结构特点,首先介绍几个简单的 C 语言程序。

【例 1-1】 向终端输出以下一行信息。

This is a C program!

```
#include<stdio.h>           //预编译处理命令
void main ()                 //主函数头部
{                             //主函数体, 由一对“{}”括起来
    printf("This is a C program!\n"); //输出语句, 向终端输出一串字符串
}
```

程序的运行结果:

This is a C program!

程序分析:本程序由两部分组成,编译预处理和 main() 函数。C 语言没有提供数据的输入\输出语句,但是给用户提供了用于输入\输出的函数,其中本函数中的 printf() 是用于终端输出数据的库函数,程序中使用库函数之前必须将库函数所属的头文件包含到源文件中来,包含头文件的预处理指令格式为 #include <库函数头文件名.h> 或

#include" 库函数头文件名.h"。main 是主函数的函数名,表示这是一个主函数。每一个 C 语言程序源程序有且仅有一个主函数。

【例 1-2】 计算任意两个数的和。

```
#include<stdio.h>           //预编译处理命令
void main()                 //主函数头部
{                           //主函数体,由一对“{}”括起来
    int x,y,sum;            //定义了3个整型变量
    x=100;y=200;           //给变量赋值
    sum=x+y;               //求和
    printf("sum is %d\n",sum); //输出结果
}
```

程序的运行结果:

sum is 300

程序分析:本程序由两部分组成,编译预处理和 main() 函数。主函数由数据说明部分和执行部分组成。实现了计算两个数 x 和 y 的和,并输出至终端的目的。

【例 1-3】 编写一个程序,输入两个数,求其中较大的数。

```
#include<stdio.h>           //预编译处理命令
int max(int x,int y)         //定义 max 函数
{
    int z;
    if(x>y)
        z=x;
    else z=y;
    return(z);              //将 z 的值返回调用处
}
main()                      //主函数
{
    int a,b,c;
    scanf("%d%d",&a,&b);    //输入两个数
    c=max(a,b);             //调用 max 函数
    printf("max=%d\n",c);
}
```

5□9 (键盘输入 5,9 给 a 和 b,□代表空格)

程序的运行结果:

max=9

程序分析:本程序包含两个函数,主函数 main() 和被调用函数 max()。max() 函数的作用是将 x、y 中较大的数赋值给 z,通过调用 return 语句将 z 的值返回到主函数 main



的调用处。 x 、 y 是两个形式参数,它们的值由主函数中对应的实际参数 a 和 b 的值传递过来。

1.3.2 C 语言程序的结构

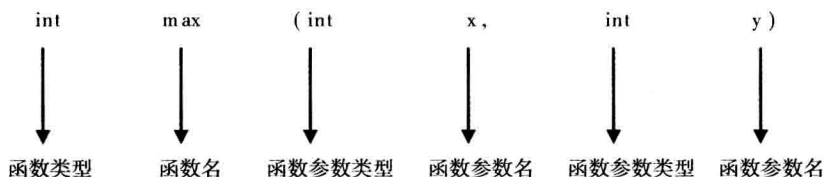
上述 3 个 C 语言程序例子虽然功能简单,但却包含了 C 语言程序的基本组成部分。通过以上例子可以看出,一个 C 语言程序的结构有如下特点:

(1) C 程序由函数构成。一个 C 源程序至少包含一个 `main` 函数,也可以包含一个 `main` 函数和若干其他函数。函数是 C 程序的基本单位。被调用的函数可以是系统提供的库函数(如 `printf` 和 `scanf` 函数),也可以是用户根据需要自定义的函数(如,例 1-3 中的 `max` 函数)。

(2) 一个函数由两部分组成。

① 函数首部,即函数的第一行,包括函数名、函数类型、函数属性、函数参数名、参数类型。

如,例 1-3 中 `max` 函数首部对应各部分为:



一个函数名后面必须跟一对圆括号,函数参数可以没有,如 `main( )`。

② 函数体

函数体一般包括两部分:

声明部分:定义所用到的变量,对所调用的函数进行声明。

执行部分:由若干语句组成,指定在函数中所进行的操作。

某些情况下可以没有声明部分(如,例 1-1),甚至可以既无声明部分,也无执行部分,例如:

```
void dump( )  
{ }
```

它是一个空函数,什么都不做,但这是合法的。

(3) 一个 C 程序总是从 `main` 函数开始执行,不论 `main` 函数在整个函数中的位置如何。

(4) C 程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上。

(5) 每个语句和数据声明的最后必须有一个分号,分号是 C 语言的必要组成部分,

例如:

```
c = a + b;
```

(6) C 语言本身没有输入\输出语句。输入\输出操作由库函数 `scanf` 和 `printf` 等函数完成。

(7) 可以采用“/* */”或者“//”对 C 程序进行注释。“/* */”内的文字是程序的注释部分,一般情况下,如果注释内容在程序中占用多行,习惯采用“/* */”。当采用“//”标注注释部分时,如果某行程序代码前面插入该符号,该符号后面的部分成为注释行,且本行有效,不能跨行,单行注释内容习惯采用“//”。一个好的、有使用价值的源程序都应当加上必要的注释,以增加程序的可读性。

1.4 C 语言程序的开发步骤

C 语言编写的程序是源程序,计算机不能直接识别和执行用高级语言编写的指令,必须使用编译程序(编译器)把 C 源程序翻译成二进制形式的目标程序,然后再将该目标程序与系统的函数库以及其他目标程序连接起来,形成可执行的目标程序。一个 C 程序的具体开发步骤如下:

(1) 编辑

选择一种 C 语言编译系统(TC、Visual C++ 6.0 等),输入编写好的程序代码,称之为源程序,它以文件的方式存在,在 TC 编译器下编辑源文件保存时,系统默认以“.c”为扩展名。在 Visual C++ 6.0 编辑器下,系统默认为以“.cpp”为源文件的扩展名。

(2) 编译

采用 C 编译系统将源程序翻译成二进制的机器目标程序,并生成扩展名为.obj 的目标程序文件。如果在编译过程中发现语法错误,编译系统会给出错误提示,包括错误的类型和源程序中出现语法错误的位置,以使用户根据提示对源程序进行修改,然后重新编译,直至排除所有语法错误为止。

(3) 连接

编译结束,得到一个或多个目标文件,利用 C 语言的连接程序将一个程序的所有目标文件、系统的库文件以及系统提供的其他信息连接起来,形成一个可执行的二进制文件,扩展名为“.exe”。

(4) 运行

运行最终得到的可执行文件,得到运行结果。

(5) 调试

分析程序结果,如果结果不正确,修改源代码并重复以上步骤,直至获得正确的结果。



(6) 优化

进一步提高程序的运行效率,主要通过改进算法,来缩短程序运行时间,以及合理分配使用内存,减少所用存储空间。

程序从编辑到运行通常需要经过 4 个阶段:编辑、编译、连接和执行,其开发过程如图 1-1 所示。

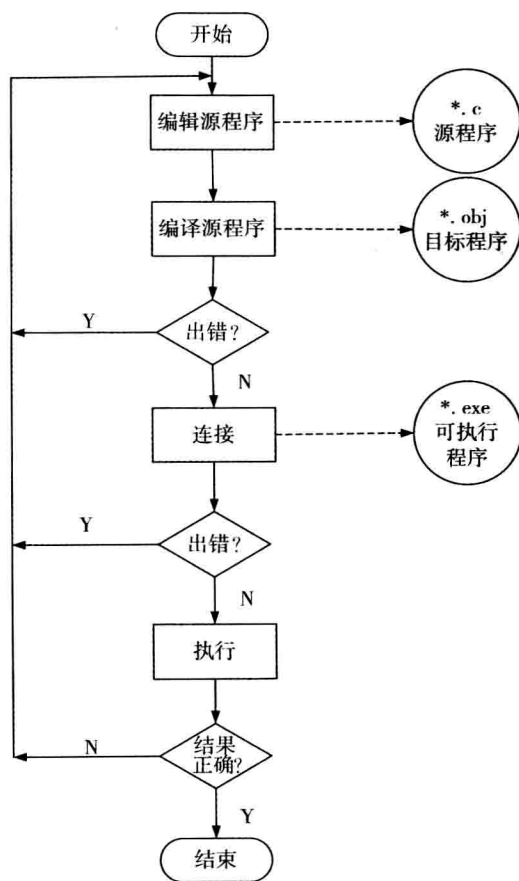


图 1-1 C 程序的开发过程

习 题

一、填空题

1. 一个完整的 C 程序有且仅有一个 _____ 函数。
2. C 程序以 _____ 为基本单位。
3. C 语言源程序的语句分隔符是 _____。