高职高专计算机实用教程系列规划教材

C语言程序设计与实验指导

李爱玲　姬秀荔　主编

钟家民　朱宗胜　罗　颖　姚玉钦　副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职高专计算机实用教程系列规划教材

C 语言程序设计与实验指导

李爱玲　姬秀荔　主　编
钟家民　朱宗胜　罗　颖　姚玉钦　副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书对知识点的讲解由浅入深，强调算法设计，突出编程思路，注重实例讲解和对学生动手能力的培养。

全书共分14章，主要内容包括：C语言概述，数据类型、运算符、表达式，顺序结构程序设计，选择结构程序设计，循环结构程序设计，数组，函数，预处理，指针，结构体、联合体与枚举类型，运算符，文件，综合实例程序设计，上机实验指导。

本书适合作为高职高专计算机程序设计的基础教材，也可作为中职及社会各类人士的自学参考书。

图书在版编目（CIP）数据

C语言程序设计与实验指导/李爱玲，姬秀荔主编.
北京：中国铁道出版社，2008.12
（高职高专计算机实用教程系列规划教材）
ISBN 978-7-113-09479-9

Ⅰ.C… Ⅱ.①李…②姬… Ⅲ.C语言－程序设计－高等学校：技术学校－教材 Ⅳ.TP312

中国版本图书馆CIP数据核字（2008）第190156号

书　　名：C语言程序设计与实验指导
作　　者：李爱玲　姬秀荔　主编

策划编辑：秦绪好　王春霞
责任编辑：翟玉峰　　**编辑部电话：（010）63583215**
编辑助理：周海燕　　**封面设计：付　巍**
封面制作：白　雪
责任印制：李　佳

出版发行：中国铁道出版社（北京市宣武区右安门西街8号　　邮政编码：100054）
印　　刷：河北省遵化市胶印厂
版　　次：2008年12月第1版　　2008年12月第1次印刷
开　　本：787mm×1092mm　1/16　　印张：18.75　　字数：436千字
印　　数：4 000册
书　　号：ISBN 978-7-113-09479-9/TP・3103
定　　价：28.00元

FOREWORD

前 言

“C 语言程序设计”是高职高专各专业的一门基础课程，也是程序设计的启蒙语言课程。本课程的学习，可以使学生更好地了解和应用计算机，培养学生应用计算机独立解决问题的能力，为今后进一步学习奠定良好、扎实的计算机语言基础。

本书是学习 C 语言程序设计的基础教材，由教学经验丰富的一线教师精心组织编写。书中对 C 语言的精华部分做了较为详细的介绍；对较难的题目给出了编程思路；还针对学生学完 C 语言后普遍感觉提高和综合应用难的问题，在第 13 章安排了综合实例设计与分析，方便学生对全书内容的综合理解和应用；考虑到 C 语言程序设计是一门实践性比较强的课程，故在书后安排了 C 语言实验，实验安排和章节内容合理对应，集教程和实验于一体，方便学生使用。

在教材的第 1、2 章，介绍了 C 语言的基本概念、各种数据类型；第 3～5 章介绍了 C 语言的基本程序设计技术、C 语言函数的特点；第 6、7 章介绍数组及函数的相互调用及变量的特性；第 8、9 章详细地介绍了指针的特点和灵活性及预处理的内容；第 10～12 章介绍了结构体、链表技术、位运算和文件的操作方法；第 13 章是综合实例程序设计；第 14 章给出了上机实验；附录给出了常用 ASCII 码字符对照表、常见编译错误信息等。书中的例题都是作者精心设计的，并全部在 Visual C++ 6.0 环境下调试通过。

本书的讲述深入浅出，配合典型例题，通俗易懂，实用性强，适合作为高职高专各专业学生的 C 语言教材，也可以作为自学者的参考用书。可免费给读者提供由本书作者开发的 C 语言 for Windows 集成实验与学习环境、书中源码和课件。

本书的第 1～5 章由罗颖、朱宗胜编写，第 6～9 章由姬秀荔、李爱玲编写，第 10～13 章由钟家民、姚玉钦编写，实验和附录部分由赵浩婕、张珊靓编写，张世聪等对本书代码进行了调试。感谢孙保平副校长一直以来的支持和帮助，感谢杨文龙教授和谢培泰教授给我们的指导。

本书的电子课件及课后习题答案均可从 http://edu.tqbooks.net 下载。

由于作者水平有限，书中难免会有不足和疏漏之处，希望广大读者提出宝贵意见，以帮助我们进一步完善此教材。

编 者

2008 年 10 月

CONTENTS

目录

第 1 章 C 语言概述

本章导读：本章主要介绍 C 语言的发展过程及 C 语言的简单应用，要求能够了解 C 语言的基本内容，掌握简单的 C 语言程序。

1.1 C 程序设计语言发展

C 语言是近年来国际上广泛推行的、很有发展前途的计算机高级语言。它适合作为系统描述语言，既可用来写系统软件，也可用来写应用软件。

以前的操作系统等系统软件主要是用汇编语言编写的（包括 UNIX 操作系统在内）。汇编语言依赖于计算机的硬件，程序的可读性和可移植性都比较差。为了提高可读性和可移植性，最好改用高级语言，但是一般的高级语言难以实现汇编语言的某些功能（汇编语言可以直接对硬件进行操作，例如对内存地址的操作、位操作等）。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，并集它们的优点于一身，于是 C 语言就应运而生。

C 语言的根源可以追溯到 ALGOL 60。ALGOL 60 是于 19 世纪 60 年代早期设计的，它是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序。CPL（Combined Programming Language，混合编程语言）由剑桥大学和伦敦大学于 1963 年开发而成，比 ALGOL 60 更加接近硬件一些，但是规模比较大，难以实现。BCPL（Basic Combined Programming Language，基础混合编程语言）由剑桥大学的 Martin Richards 于 1967 年发明。B 语言由贝尔实验室的 Ken Thompson 于 1970 年设计出来，第一个 UNIX 操作系统就是用 B 语言写成并在 PDP-7 上实现的。1971 年，Ken Thompson 在 PDP-11/20 上实现了 B 语言，并写了操作系统。但是 B 语言过于简单，且功能有限。C 语言由贝尔实验室的 Dennis Ritchie 于 1972 年设计出来。C 语言既保持了 BCPL 和 B 语言的优点（精练，接近硬件），又克服了它们的缺点（过于简单，数据无类型等）。1975 年 UNIX 第六版公布后，C 语言的突出优点才引起人们的普遍注意。1978 年以后，C 语言先后移植到大、中、小、微型机上，并逐渐独立于 UNIX 和 PDP。目前，C 语言以其独有的特点，已经成为世界上应用最广泛的语言之一，并且广泛地用在教学上。

20 世纪 80 年代初期，许多工程项目都令结构化方法到达了极限，为解决这个问题，诞生了一种称为面向对象编程的新编程方法。1979 年，当 Bjarne Stroustrup 在新泽西州的 Murray Hill 实验室工作时，发明了 C++。这种语言最初称为“带类的 C”，1983 年改名为 C++，它包括了 C 语言的所有特征属性和优点。C++的发明不是企图创造一种全新的编程语言，而是对 C 语言这个高度成功语言的改进。

20 世纪 90 年代初，Sun 公司在以 C 语言为核心的基础上开发了一种称为 Oak 的语言，1994 年 Oak 语言重新命名为 Java，这种语言能方便地用于许多平台。1999 年，Sun 公司在 Java 的基础上开发了 Java 2 Enterprise Edition。J2EE 作为服务器语言被开发并强力推荐，并于 2000 年开始得到广泛应用。与此同时，微软公布了.NET。它几乎包括了 J2EE 的所有特性，但没有 J2EE 昂贵的价格。.NET 包含了一个新的 C 家族语言——C#，读作 C-pound。2001 年，微软员工们认为应该将其称为 C Sharp。2002 年，C# 成为.NET 发行版本的一部分，它具有自动清理内存的特色。三种语言的特点比较如下：

C 语言是一种灵活性很强、应用非常广泛的面向过程的编程语言。C++是一种多范型程序设计语言，它不仅仅是一种面向对象的语言，同时也是一种面向过程的语言。C++语言在 C 语言的基础上发展起来，并引入了类的概念。VC 则是 Visual C++的简称，是可视化开发工具的一种。和 VB（Visual Basic）、VJ（Visual Java）、VF（Visual FoxPro）相似，VC 提供了 C++的集成化开发环境（IDE），可以说 VC 是一种开发工具，本质上是利用 C++语言编程的。

Java 是一种跨平台、适于分布式计算环境的面向对象编程语言。具体来说，它具有如下特性：简单性、面向对象、分布式、解释型、可靠、安全、平台无关、可移植、高性能、多线程、动态性等。

C#不允许直接对内存进行操作，它没有指针，关键字也更加明了；C#中的每种类型都可以当做对象，并只允许单继承，也没有全局变量和全局常数，所有的一切都必须封装在类中，这样可以使代码具有更好的可读性，减少了命名冲突的可能；网络服务看起来就像是 C#的本地对象，C#组件能够方便地为 Web 服务，并允许它们通过 Internet 运行在任务操作系统上的任何语言调用；C#中不能使用未初始化的变量，对象的成员变量由编译器负责置零；C#不支持不安全的指向，不能将整数指向引用类型；C#还提供了边界检查和溢出检查功能；C#在语言中内置了版本控制功能，保证复杂的软件可以方便地开发和升级。

在对操作系统和系统实用程序以及需要对硬件进行操作的场合，C 语言明显优于其他高级语言。C 语言是一种面向过程的结构化语言，且描述能力强，同样适应于教学和其他广泛的应用领域，因此很有生命力。

1.2 C 语言特点

C 语言之所以能存在和发展，并具有生命力，是因为它有不同于（或优于）其他语言的特点。C 语言具有强大的功能，许多著名的操作系统，如 UNIX、Linux 等都是由它编写的。C 语言的主要特点如下：

① C 语言的标识符要求区分大小写。C 语言以英文小写字母为基础，符合人们日常阅读和书写的习惯。同一个单词的大小写代表不同的标识符。

② 语言简洁、紧凑，使用方便、灵活。C 语言一共只有 32 个关键字，9 种控制语句，程序书写形式自由，压缩了一切不必要的成分，因此 C 程序比较简练，易写易懂，源程序短，输入程序时工作量少。

③ 程序由函数组合而成，因此程序功能结构比较清楚。而且，每个函数都是独立的，可以单独编译，对大型程序设计来说，有利于分工编译和调试。

④ 运算符丰富。C 语言的运算符包含的范围很广泛，共有 34 种运算符。C 语言把括号、赋

值、强制类型转换等都作为运算符处理，因此运算类型极其丰富，表达式类型灵活多样。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

⑤ 数据结构丰富，具有现代化语言的各种数据结构。C的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂的数据结构（如链表、树、栈等）的运算，尤其是指针类型数据，使用起来比Pascal更为灵活方便。

⑥ 具有结构化的控制语句（如if...else语句、while语句、do...while语句、switch语句、for语句）。用函数作为程序的模块单位，便于实现程序的模块化。C语言是理想的结构化语言，符合现代编程风格的要求。

⑦ 语法限制不太严格，程序设计自由度大。例如，对数组下标越界不做检查，由程序编写者保证程序的正确。对变量的类型使用比较灵活，例如整型数据与字符型数据以及逻辑型数据可以通用。

⑧ C语言允许直接访问物理地址，能进行位（bit）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。因此，C语言既具有高级语言的功能，又具有低级语言的许多功能，可用来编写系统软件。C语言的这种双重性，使它既是通用的程序设计语言，又是成功的系统描述语言。有人把C语言称为“高级语言中的低级语言”或“中级语言”，意为兼有高级和低级语言的特点。

按此观点可将各语言分类如下：

高级语言：如BASIC、FORTRAN、COBOL、Pascal、Ada、Modula-2。

中级语言：如C、FORTH、宏汇编。

低级语言：如汇编语言。

一般仍习惯将C语言称为高级语言，因为C程序也要通过编译、连接才能得到可执行的目标程序，这是和其他高级语言相同的。

⑨ 生成的目标代码质量高，程序执行效率高。C语言生成的目标代码一般只比汇编程序生成的目标代码效率低10%～20%。

⑩ 用C语言写的程序可移植性好（与汇编语言相比），基本上不做修改就能用于各种操作系统，如DOS、Windows、UNIX等。

上面只介绍了C语言最容易理解的一般特点，至于C语言内部的其他特点将结合以后各章的内容做具体介绍。由于这些优点，使得C语言的应用面很广。许多系统软件都用C语言编写，这主要是由于C语言的可移植性好和硬件控制能力高，表达和运算能力强。许多以前只能用汇编语言处理的问题，现在都可以改用C语言来处理。

C语言也有一定的弱点，如非强制类型、语法限制不严格，这使得编程者无法过多地依赖C语言的编译程序去查错；缺少运行时检查、数组越界检查等。

下面从应用角度出发，对C语言和其他传统的高级语言做一下简单比较。从掌握语言的难易程度来看，C语言比其他语言难一些。BASIC语言是较好的初学者入门语言，FORTRAN也比较好掌握。在科学计算领域，多用FORTRAN或PL/I；在商业和管理等数据处理领域，用COBOL语言为宜。C语言虽然也可用于科学计算和管理领域，但并不理想，因为C语言的特长不在这里。对操作系统和系统实用程序以及需要对硬件进行操作的场合，用C语言明显优于其他高级语言，有的大型应用软件也用C语言编写。从教学角度看，由于Pascal语言是世界上第一个结构化语言，曾被认为是计算机专业比较理想的教学语言，前几年在数据结构等课程的教学中一般用Pascal语

言举例。但由于 Pascal 语言难以推广到实际应用领域，因此到目前为止基本上只是教学语言。C 语言也是理想的结构化语言，且描述能力强，同样适于教学。C 语言除了能用于教学外，还有广泛的应用领域，因此更有生命力。Pascal 和其他高级语言的设计目标是通过严格的语法定义和检查来保证程序的正确性，而 C 语言则强调灵活性，使程序设计人员能有较大的自由度，以适应宽广的应用面。

总之，C 语言对程序员的要求较高。程序员使用 C 语言编写程序会感到限制少、灵活性大、功能强，可以编写出任何类型的程序。现在，C 语言已不仅仅用来编写系统软件，还可以用来编写应用软件。学习和使用 C 语言的人已越来越多。

1.3 简单的 C 程序开发

下面从最简单的程序例子来分析 C 语言的程序构成及运行情况。

【例 1-1】下面的代码是一个完整的可以运行的程序，其功能是输出一个句子。

```
/*--------------------------
          example1.c
--------------------------*/
#include <stdio.h>
main()
{
   printf ("It is a first C program.\n");        /*输出语句*/
}
```

运行结果为：

```
It is a first C program.
```

【例 1-2】下面的程序其功能是求两数的和并且输出。

```
/*--------------------------
          example2.c
--------------------------*/
#include <stdio.h>
main()
{
   int a,b,sum;                          /*定义变量*/
   a=234;                                /*赋值语句*/
   b=567;                                /*赋值语句*/
   sum=a+b;                              /*求和语句*/
   printf("sum is %d\n",sum);            /*输出语句*/
}
```

运行结果为：

```
sum is 801
```

【例 1-3】下面的程序其功能是比较两数的大小，并且输出较大的值。

```
#include <stdio.h>
main()                                   /*主函数*/
{
   int a,b,c;                            /*声明部分，定义变量*/
   scanf("%d,%d",&a,&b);                 /*输入变量 a 和 b 的值*/
   c=max(a,b)                            /*调用 max 函数，将得到的值赋给 c*/
```

```
    printf("max=%d",c);    /*输出 c 的值*/
}
int max(int x,int y)       /*定义 max 函数，函数值为整型，形式参数 x,y 为整型*/
{
    int z;                 /*max 函数中的声明部分,定义本函数中用到的变量 z 为整型*/
    if(x>y) z=x;           /*条件判断语句*/
    else z=y;
    return(z);             /*将 z 的值返回，通过 max 带回调用处*/
}
```

输入 5，17 两个数时，运行结果为：

```
max=17
```

从上面三个 C 程序可以看出，一个 C 程序由三个部分组成：

- 注释。
- 编译预处理。
- 程序主体。

C 语言的注释为“/*”与“*/”之间的内容，它可以占多行。也可以是“//”之后的内容，但是只能注释一行，换行后即为程序主体内容。

例如：

```
//*****************************
//this is a simplest program.
//*****************************
```

注释是程序员为读者做的说明，是提高程序可读性的一种手段。一般可将其分为两种：序言注释和注解性注释。前者用于程序开头，说明程序或文件的名称、用途、编写时间、编写人以及输入/输出说明等，后者用于程序中难懂的地方。

每个以符号“#”开头的行，称为编译预处理行，如#include 称为文件包含预处理命令。有关内容将在以后的章节中介绍。

#include "stdio.h"的作用是在编译之前将文件 stdio.h 的内容增加（包含）到程序中，以作为其一部分。stdio.h 是系统定义的一个“头文件”，它设置了 C 语言的 I/O 相关环境，定义输入/输出函数 printf()和 scanf()等。printf()和 scanf()的使用方法将在后面章节中介绍。

main()表示主函数，每一个 C 语言程序都必须有一个 main()函数。main()作为程序的入口，无论处在程序中的哪个位置，程序总是从 main()函数开始。函数体用花括号{}括起来，如果一个函数内有多个花括号，则最外层的一对{}为函数体的范围。描述一个函数所执行算法的过程称为函数定义。如上面例子，main()函数头和函数体构成了一个完整的函数定义。

C 程序中的一个函数总是由两部分组成：函数的说明部分和函数体。函数的说明部分处于函数的头部，包括函数类型、函数名、函数参数名（形参）及函数形参类型说明。

函数名 main 全部由小写字母构成。在 C 程序中，严格区分字母的大小写，所以在书写标识符时要注意其大小写。

在 main()函数体中，printf()（全是小写字母）是一个代表标准输出的函数，它是 C 语言预定义的函数，前面包含的头文件就是为了能在这里使用输出函数 printf()。当程序要在设备上进行输出时，就需要在程序中指定该对象。标准的输出格式是：printf("")，它表示将双引号中的数据送到显示设备上。

程序中用双引号括起来的数据“It is a first C program. \n”称为字符串常量。其中，字符“\n”表示一个换行控制符。

";" 表示一个语句的结束。每个语句和数据定义的最后必须有一个分号，分号是 C 语句的必要组成部分。

C 程序书写格式自由，一行可以写一条或多条语句，一个语句也可以分写在几行上，但是一个标识符不可以分成两行或者多行书写。同时，C 程序不像 BASIC 程序那样每行有行号，也不像 FORTRAN 或 COBOL 那样对书写格式有严格的规定。

C 语言源程序的一般形式可以直观地概括为：

```
全局变量说明
(结果类型标识符)main(参量表及说明)/*主函数*/
{
    子函数说明表
    局部变量说明表
    语句序列表
  }
(结果类型标识符)f1(形式参量表及说明)/*函数 f1*/
  参数说明
  {
    局部变量说明表
    语句序列表
  }
      ...
(结果类型标识符)fn(形式参量表及说明)/*函数 fn*/
  参数说明
  {
    局部变量说明表
    语句序列
  }
```

1.4 C 语言的运行环境

C 语言是一种编译型的程序设计语言，它采用编译的方式将源程序翻译成目的程序（机器代码）。运行一个 C 程序，从输入源程序开始，要经过编辑源程序文件(.c)、编译生成目标文件(.obj)、连接生成可执行文件（.exe）和执行四个步骤。

下面主要介绍在 Turbo C 下运行 C 语言源程序和在 Visual C++6.0 中运行 C 语言源程序。

1. Turbo C 下运行 C 语言源程序

Turbo C 是美国 Borland 公司推出的 IBM PC 系列机的 C 语言编译程序。它具有方便、直观、易用的界面和丰富的库函数。它向用户提供了集成环境，把程序的编辑、编译、连接和运行等操作全部集中在一个界面上进行，使用十分方便。

（1）Turbo C 工作环境介绍

一个 C 语言程序的实施是从进入 Turbo C 的集成开发环境开始的。进入 Turbo C 集成开发环境，一般有两种途径，即从 DOS 环境进入和从 Windows 环境进入。

① 从 DOS 环境进入。在 DOS 命令行上输入命令：

```
C>CD \TC↙(指定当前目录为 TC 子目录)
C>TC↙(进入 Turbo C 环境)
```

这时，进入 Turbo C 集成开发环境的主菜单窗口，如图 1-1 所示。

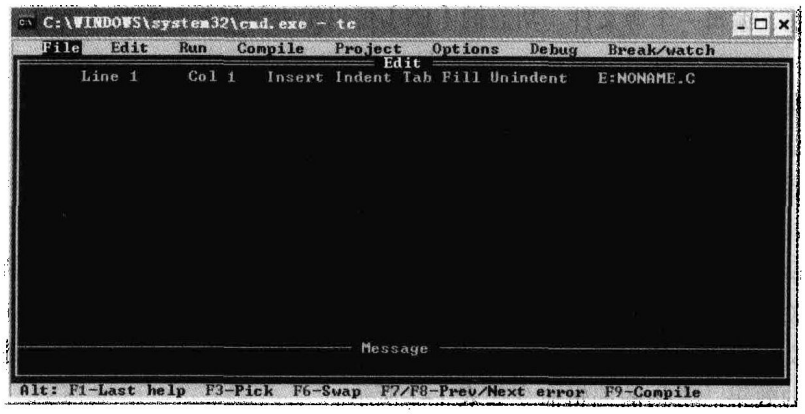

图 1-1　Turbo C 的运行界面

② 从 Windows 环境进入。在 Windows 环境中，如果本机中已安装了 Turbo C，可以在桌面上建立一个快捷方式，双击该快捷方式图标即可进入 C 语言环境。或者从“开始”菜单中选择“运行”命令，在“运行”对话框中输入“C:\TC\TC”命令，单击“确定”按钮即可。

需要说明的是，以上两种方式有一个共同的前提，即 Turbo C 的安装路径为 C:\TC，如果计算机中 Turbo C 的安装路径不同，只要在上述方式中改变相应的路径即可。

刚进入 TC 环境时，光带覆盖在 File 上，整个屏幕由四部分组成，依次为主菜单、编辑窗口、信息窗口和功能键提示行（或称快速参考行）。

a. 主菜单：显示屏的顶部是主菜单条，它提供了 8 个选择项，如表 1-1 所示。

表 1-1　TC 集成环境的主菜单

命　令	功　能
File	处理文件（装入、存盘、选择、建立、换名存盘、写盘），目录操作（列表、改变工作目录），退出 Turbo C，返回 DOS 状态
Edit	建立、编辑源文件
Run	自动编辑、连接并运行程序
Compile	编辑、生成目标文件组合成工作文件
Project	将多个源文件和目标文件组合成工作文件
Options	提供集成环境下的多种选择和设置（如设置存储模式、选择编参数、诊断及连接任选项）以及定义宏；也可记录 include、output 及 library 文件目录，保存编译任选项和从配置文件加载任选项
Debug	检查、改变变量的值、查找函数，程序运行时查看调用栈。选择程序编译时是否在执行代码中插入调试信息
Break/watch	增加、删除、编辑监视表达式，以及设置、清除、执行至断点

在主菜单中，Edit 选项仅仅是一条进入编辑器的命令。其他选项均为下拉式菜单，包含许多命令选项，使用方向键移动光带来选择某个选项时，按【Enter】键，表示执行该命令，若屏幕上弹出一个下拉菜单，则提供进一步选择。