“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定

C YUYAN CHENGXU SHEJI
SHILI JIAOCHENG

C语言程序设计实例教程（第2版）

主　编　涂玉芬　刘　芳　袁晓芳
副主编　黄　琴　白友林
主　审　向　隅

北京邮电大学出版社
www.buptpress.com

“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定

C语言程序设计实例教程

（第2版）

主　编　涂玉芬　刘　芳　袁晓芳
副主编　黄　琴　白友林
主　审　向　隅

北京邮电大学出版社
·北京·

内 容 提 要

本教程主要面向高职院校非计算机专业学生，从学生的认知规律出发，按照C语言知识的层次，由浅入深，由易到难，将学习内容分为“基础篇”和“提高篇”两个部分。“基础篇”包括第1～6章，详细介绍了C语言程序的结构和语法规则、C语言的基本数据类型与基本运算、三种基本结构的流程控制和程序设计方法、函数等内容，是设计简单C语言程序必须具备的基础知识。“提高篇”包括第7～11章，主要介绍了数组、指针、结构体等复杂数据类型的处理方法及应用、预处理和文件操作等，是编写复杂C语言程序需要掌握的相关知识。

本教程可作为高职高专院校C语言程序设计课程教材，也可作为各类培训机构的培训教材和自学者的参考书。

图书在版编目(CIP)数据

C语言程序设计实例教程 / 涂玉芬，刘芳，袁晓芳主编. --2版. --北京：北京邮电大学出版社，2015.8

ISBN 978-7-5635-3960-4

Ⅰ.①C… Ⅱ.①涂…②刘…③袁… Ⅲ.①C语言—程序设计—高等职业教育—教材 Ⅳ.①TP312

中国版本图书馆CIP数据核字(2014)第102639号

书　　名：C语言程序设计实例教程(第2版)

著作责任者：涂玉芬　刘　芳　袁晓芳　主编

责任编辑：彭　楠

出版发行：北京邮电大学出版社

社　　址：北京市海淀区西土城路10号(邮编：100876)

发 行 部：电话：010-62282185　传真：010-62283578

E-mail：publish@bupt.edu.cn

经　　销：各地新华书店

印　　刷：北京睿和名扬印刷有限公司

开　　本：787 mm×1 092 mm　1/16

印　　张：19

字　　数：470千字

印　　数：1—3 000册

版　　次：2011年1月第1版　2015年8月第2版　2015年8月第1次印刷

ISBN 978-7-5635-3960-4　　定　价：38.00元

前　言

C语言是国内外广泛推广使用的一种结构化、可编译的通用程序设计语言。C语言功能丰富，表达能力强，使用灵活方便，应用面广，目标程序效率高，可移植性好，既具有高级语言的优点，又具有低级语言的许多特点，适合于系统程序和应用程序的设计。“C语言程序设计”课程是目前各大中专院校理工科学生的必修课程。

本教程主要面向高职院校非计算机专业学生，按照C语言知识的层次，由浅入深，由易到难，将学习内容分为“基础篇”和“提高篇”两个部分。“基础篇”包括C语言程序结构、C语言的数据与运算、三种基本结构的程序设计及函数共6章，是设计简单C语言程序必须具备的基础知识；“提高篇”包括数组、指针、结构体、预处理和文件共5章，是编写复杂C语言程序需要掌握的相关知识。

教程从学生的认知规律出发，每篇篇首提供了一个综合案例，根据对案例的分析，引入学习内容，兼顾趣味性和知识性，激发学生的求知欲。在教程的每一章节，首先通过“问题描述”提出要解决的问题，通过“解决方法”给出问题处理方法，再通过“相关知识”剖析“解决方法”中涉及的知识点，启发学生思考，指导学生探究。在教程每一章的最后，再根据该章的知识要点，给出结合实际的应用举例，突出知识的实用性，加深学生对知识的理解，使学生能触类旁通，灵活运用。

本教程由武汉铁路职业技术学院涂玉芬、刘芳、袁晓芳主编，黄琴、白友林任副主编，向隅主审，其中，涂玉芬负责本教程的组织与统稿，并编写了教程的第7、8、10、11章及案例1和案例2，刘芳编写了教程的第3、5章，袁晓芳编写了教程的第1、2章及附录，黄琴编写了教程的第9章，白友林编写了教程的第4、6章。另外，参加本教程大纲讨论、案例调试等工作的人员还有廖梦虎、张理武、余辉、王德洪、陈梅等。

本教程可作为高职高专院校C语言程序设计的课程教材，也可作为各类培训机构的培训教材和自学者的参考书。

编　者

目　录

第一部分　基础篇

案例 1　简单学习机 …… 1

第 1 章　C 语言及程序设计基本知识 …… 5

1.1　程序及程序设计语言 …… 5
1.1.1　程序 …… 6
1.1.2　程序设计语言 …… 7
1.1.3　算法 …… 7
1.1.4　结构化程序设计方法 …… 8
1.2　C 程序结构 …… 9
1.2.1　C 语言的发展及特点 …… 10
1.2.2　C 语言程序格式说明 …… 11
1.3　C 程序调试 …… 11
1.4　习题与实训 …… 16

第 2 章　数据与运算 …… 19

2.1　变量定义 …… 19
2.1.1　数据类型 …… 19
2.1.2　常量 …… 20
2.1.3　变量 …… 22
2.2　数据运算 …… 25
2.2.1　类型转换 …… 25
2.2.2　算术运算 …… 26
2.2.3　赋值运算 …… 27
2.2.4　逗号运算 …… 28
2.3　习题与实训 …… 29

第 3 章　顺序结构程序设计 …… 34

3.1　数据输入 …… 34
3.1.1　赋值语句 …… 35
3.1.2　字符数据输入函数 …… 35

3.1.3 格式化输入函数 …… 35
3.2 数据输出 …… 37
3.2.1 字符数据输出函数 …… 37
3.2.2 格式化输出函数 …… 38
3.3 注释 …… 40
3.4 应用举例 …… 40
3.5 习题与实训 …… 43

第 4 章 选择结构程序设计 …… 50

4.1 关系表达式和逻辑表达式 …… 50
4.1.1 关系表达式 …… 50
4.1.2 逻辑表达式 …… 51
4.2 简单选择结构流程控制 …… 52
4.2.1 简单 if 语句 …… 53
4.2.2 复合语句 …… 54
4.2.3 空语句 …… 54
4.2.4 条件表达式 …… 55
4.3 复杂选择结构流程控制 …… 55
4.3.1 嵌套 if 语句 …… 56
4.3.2 switch 语句 …… 56
4.4 应用举例 …… 58
4.5 习题与实训 …… 63

第 5 章 循环结构程序设计 …… 72

5.1 循环结构流程控制 …… 72
5.1.1 while 语句 …… 73
5.1.2 do…while 语句 …… 74
5.1.3 for 语句 …… 75
5.2 循环结构中的跳转控制 …… 77
5.2.1 break 语句 …… 77
5.2.2 continue 语句 …… 78
5.3 多重循环结构 …… 79
5.4 应用举例 …… 81
5.5 习题与实训 …… 88

第 6 章 函数 …… 97

6.1 函数的定义和调用 …… 97
6.1.1 函数定义 …… 98
6.1.2 函数调用 …… 100

6.1.3　函数声明 …… 103
6.2　局部变量与全局变量 …… 106
6.2.1　局部变量 …… 106
6.2.2　全局变量 …… 107
6.3　动态变量与静态变量 …… 108
6.3.1　动态存储方式和静态存储方式 …… 108
6.3.2　动态变量 …… 109
6.3.3　静态变量 …… 110
6.4　内部函数与外部函数 …… 113
6.4.1　内部函数 …… 113
6.4.2　外部函数 …… 114
6.5　应用举例 …… 114
6.6　习题与实训 …… 118

第二部分　提高篇

案例 2　班级成绩管理系统 …… 113

第 7 章　数组 …… 134

7.1　一维数组 …… 134
7.1.1　一维数组的定义与引用 …… 134
7.1.2　一维数组初始化 …… 135
7.2　二维数组 …… 137
7.2.1　二维数组的定义与引用 …… 137
7.2.2　二维数组初始化 …… 139
7.3　字符数组 …… 140
7.3.1　字符数组的定义、引用与初始化 …… 141
7.3.2　字符串与字符数组 …… 142
7.3.3　字符串函数 …… 143
7.4　应用举例 …… 146
7.5　习题与实训 …… 152

第 8 章　指针 …… 159

8.1　指针的概念及基本运算 …… 159
8.1.1　指针的概念 …… 159
8.1.2　指针变量 …… 160
8.1.3　指针变量的运算 …… 161
8.2　指针与数组 …… 164
8.2.1　指针与一维数组 …… 165
8.2.2　指针与二维数组 …… 170

8.2.3 指针与字符串 …… 176
8.2.4 指针数组 …… 178
8.3 指针与函数 …… 180
8.3.1 指针作函数参数 …… 180
8.3.2 指针函数 …… 184
8.3.3 函数指针 …… 185
8.4 指向指针的指针与命令行参数 …… 186
8.4.1 指向指针的指针 …… 186
8.4.2 命令行参数 …… 188
8.5 应用举例 …… 189
8.6 习题与实训 …… 192

第9章 结构体与共用体 …… 201

9.1 结构体 …… 201
9.1.1 结构体的基本操作 …… 201
9.1.2 结构体数组 …… 206
9.1.3 结构体指针 …… 208
9.1.4 结构体与函数 …… 209
9.2 链表 …… 211
9.2.1 动态分配函数 …… 212
9.2.2 链表节点类型的定义 …… 214
9.2.3 建立链表 …… 214
9.2.4 链表的基本操作 …… 216
9.3 共用体 …… 220
9.3.1 共用体类型的定义 …… 221
9.3.2 共用体变量的定义 …… 222
9.3.3 共用体变量的引用 …… 222
9.4 枚举 …… 223
9.4.1 枚举类型的定义 …… 224
9.4.2 枚举变量的定义 …… 225
9.4.3 枚举变量的应用 …… 226
9.5 自定义类型 …… 228
9.6 应用举例 …… 228
9.7 习题与实训 …… 233

第10章 编译预处理与位运算 …… 248

10.1 编译预处理 …… 248
10.1.1 宏定义 …… 248
10.1.2 文件包含 …… 250

10.1.3　条件编译…… 252
10.2　位运算…… 254
10.3　应用举例…… 256
10.4　习题与实训…… 257

第 11 章　文件 …… 263

11.1　文件的基本概念…… 264
11.2　文件的基本操作…… 265
11.2.1　定义文件指针…… 265
11.2.2　打开与关闭文件…… 266
11.2.3　读写文件…… 268
11.2.4　定位文件…… 273
11.2.5　检测文件是否结束…… 275
11.3　应用举例…… 276
11.4　习题与实训…… 278

附录…… 286

参考文献…… 293

第一部分　基础篇

案例 1　简单学习机

问题描述

为小朋友设计一个简单的算术运算学习机，该学习机能根据小朋友对运算的要求，随机地给出一些加、减、乘、除运算的练习题供小朋友练习，并对小朋友的回答进行判断。

程序

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
void main()                    /* 函数首部 */
{
        /* 声明部分 */
    int c,s;
    char x,d;
    void menu();
    int epe(char);
    void jude(int,int);
    void end();
      /* 执行部分 */
    menu();                          /* 显示主界面 */
    scanf("%c",&x);                  /* 选择练习类型 */
    while(1)
      {
       s=epe(x);                     /* 计算机随机出题并计算正确答案 */
       scanf("%d",&c);               /* 小朋友输入自己的答案 */
       getchar();
       jude(s,c);                    /* 计算机对小朋友的答案进行对错判断 */
       printf("你想继续练习吗(y/n)?");
```

```
            d= getchar();
            if((d=='n')||(d=='N')) break;
        };
    end();
}
void menu()
{
    printf("*****************************************************\n");
    printf("*****************************************************\n");
    printf("*****                                           *****\n");
    printf("*****  小朋友,你好,我是你的学习机,我们一起努力吧!*****\n");
    printf("*****                                           *****\n");
    printf("*****  你今天想要做什么练习?                    *****\n");
    printf("*****                                           *****\n");
    printf("*****        加法练习(+)    减法练习(-)        *****\n");
    printf("*****        乘法练习(*)    除法练习(/)        *****\n");
    printf("*****                                           *****\n");
    printf("*****  请选择(输入练习后的运算符):              *****\n");
    printf("*****                                           *****\n");
    printf("*****************************************************\n");
    printf("*****************************************************\n");
}
int epe(char x)
{
    int a,b,s;
    unsigned begin_seed;
    time_t t;
    begin_seed = (unsigned)time(&t);
    srand(begin_seed);
    a=rand()%10;
    srand(begin_seed+1);
    b=rand()%10;
    switch(x)
    {
     case '+':printf("    你的练习题是:%d+%d=?",a,b);s=a+b;break;
     case '-':printf("    你的练习题是:%d-%d=?",a,b);s=a-b;break;
     case '*':printf("    你的练习题是:%d*%d=?",a,b);s=a*b;break;
```

```
        case '/':printf("      你的练习题是：%d/%d=?",a,b);s=a/b;break;
    }
    return s;
}
void jude(int s,int c)
{
    printf("*****************************************************\n");
    printf("*****************************************************\n");
    if(s==c)
    {
        printf("*****              答  对  了  !              *****\n");
        printf("*****            你  真  聪  明  !            *****\n");
    }
    else
    {
        printf("*****              答  错  了  !              *****\n");
        printf("*****          你  要  加  油  啊  !          *****\n");
    }
    printf("*****************************************************\n");
    printf("*****************************************************\n");
}
void end()
{
    printf("*****************************************************\n");
    printf("*****************************************************\n");
    printf("*****                                           *****\n");
    printf("*****          小朋友,再见了,欢迎下次练习!          *****\n");
    printf("*****                                           *****\n");
    printf("*****************************************************\n");
    printf("*****************************************************\n");
}
```

程序分析

上述 C 程序的流程如图 1-1 所示。

程序中包含 5 个函数：main()、menu()、epe()、jude()和 end()。

- main()是主函数，通过调用函数 menu()、epe()和 jude()完成功能。
- 函数 menu()完成显示主界面的功能。
- 函数 epe()完成随机出题并计算正确答案的功能。

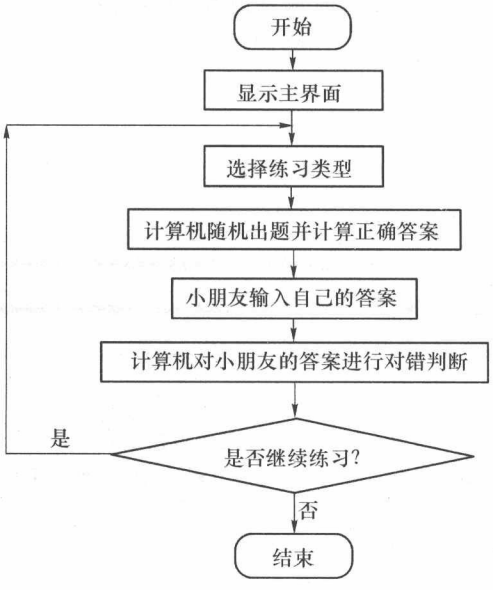

图 1-1

- 函数 jude()完成对小朋友的答案进行对错判断的功能。
- 函数 end()完成结束练习信息显示功能。

相关知识

通过上述程序分析可知,完成一个简单的 C 程序设计所需要掌握的基础知识如下:

- C 语言及程序设计基本知识;
- C 语言的数据类型与表达式;
- 顺序程序设计;
- 选择结构程序设计;
- 循环程序设计;
- 函数。

第1章　C语言及程序设计基本知识

随着计算机的普及和发展，大家对计算机这种电子设备并不陌生。计算机是一种能按照人们事先编写的程序连续、自动地工作，能对输入的数据进行加工、存储、传送，由电子和机械部件组成的电子设备。

没有程序，计算机不能做任何事情。

1.1　程序及程序设计语言

问题描述

计算机是一种具有内部存储能力和自动控制能力的电子设备，不知道如何去完成用户交给它的具体工作任务，用户需要让计算机去完成具体的工作任务时，需要写出完成该具体任务的程序，输入并存储在计算机的内部存储器中，当用户发出执行指令后，计算机就自动到计算机的内部存储器中取出程序，按程序自动执行，完成该任务。

解决方法

用户根据具体的工作任务编写出能让计算机高效地完成该任务的程序的过程，称为“程序设计”。

程序设计的一般步骤如下。

(1) 确定数据结构。

分析具体任务，确定输入数据和输出数据，确定数据的逻辑结构和存储结构。

(2) 确定算法。

根据确定的数据结构，确定解决问题的方法，即完成任务的一步一步的步骤。

(3) 编写程序。

根据确定的数据结构和算法，使用选定的计算机语言编写程序代码，简称“编程”。

(4) 调试程序。

将编写好的程序输入到计算机内存中，对程序进行测试并修正，直至程序符合任务要求。

(5) 整理文档资料。

根据数据结构和程序，整理编写相关文档资料。

【例】　从键盘输入圆半径，求圆的面积和周长，结果保留两位小数。

(1) 确定数据结构。

- 输入数据：圆半径。
- 输出数据：圆的面积、圆的周长。
- 数据的逻辑结构和存储结构：简单变量、随机存储。

(2) 确定算法。

基本方法如下:

圆的面积=π×圆半径^2

圆的周长=2×π×圆半径

详细步骤如流程图 1-1-1 所示。

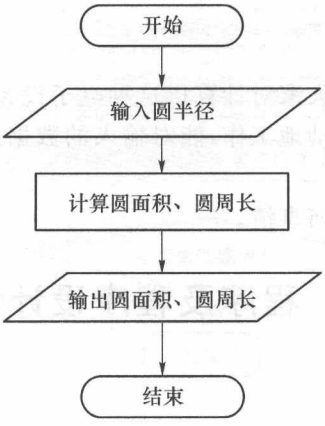

图 1-1-1

(3) 编写程序。

```
#include "stdio.h"
#include "math.h"
void main()
{
float r,area,peri;          /* r 表示圆半径、area 表示圆面积、perie 表示圆周长 */
printf("r = ");
scanf("%f",&r);
area = 3.14 * pow(r,2);    /* 函数 pow(x,y)表示 x^y */
peri = 2 * 3.14 * r;
printf("area = %.2f,peri = %.2f\n",area,peri);
}
```

(4) 调试程序。

启动 VC++6.0,进入 VC++编辑窗口,输入程序,进行编辑、编译、连接、运行,保存程序,退出 VC++6.0。

(5) 整理文档资料。

软件使用方法:启动程序运行,在光标处从键盘输入圆半径,即可在屏幕上看到结果,其中,r 表示圆半径、area 表示圆面积、perie 表示圆周长。

相关知识

1.1.1 程序

用户要让计算机去完成各种不同的工作,需要将如何完成具体工作任务的详细步骤以

计算机能执行的指令形式告诉计算机，这种以计算机能执行的指令形式出现的、能完成具体工作任务的详细步骤，称为“程序”。

1.1.2　程序设计语言

用户之间交流时使用的是用户能相互理解的语言，如汉语、英语、法语等，这些语言称为自然语言。

用户与计算机交流时需要使用计算机能理解的语言，这些语言称为程序设计语言。程序设计语言规定了编写程序时可以使用的符号集合和语法规则。程序设计语言种类繁多，根据对机器的依赖程度，可将其分为如下三大类。

1. 机器语言

机器语言是计算机唯一可以直接识别并执行的语言。机器语言所有的指令都由二进制数字 0 或 1 编码组成。

机器语言的特点是：执行速度快，不直观，不通用。

2. 汇编语言

汇编语言是机器语言的简记形式。它采用人们容易记忆的符号和标记来表示机器语言指令，使程序具有一定的可读性。

计算机不能直接识别汇编语言，汇编语言程序需要经过专用软件“汇编程序”转换为机器语言程序，才能在计算机上运行。

用汇编语言编写的程序称为源程序，其等价的机器语言表示的程序称为目标程序，“汇编程序”的功能是将汇编语言编写的源程序翻译成机器语言表示的目标程序。

汇编语言的特点是：执行速度较快，较直观，不通用。

3. 高级语言

高级语言是使用人们容易理解的自然语言和数学语言中一些简单的符号和单词组成，语句功能强大，可读性好，编程效率最高。

计算机也不能直接识别高级语言，高级语言程序同样需要经过专用软件转换为机器语言程序，才能在计算机上运行。转换方式分为两类：解释方式和编译方式。

解释方式是一边将高级语言源程序的语句解释为机器语言表示的目标代码，一边执行代码。

编译方式是将高级语言源程序翻译为机器语言表示的目标程序，形成的目标程序可以脱离其语言环境独立存在和运行。

C 语言是高级语言中的一种，其采用的转换方式是编译方式。

1.1.3　算法

1. 算法的概念

程序是以计算机能执行的指令形式出现的，能完成具体工作任务的详细步骤。因此，在编写程序前，确定完成任务的操作及其顺序是十分重要的。这种为解决某一特殊问题而采取的确定而有限的操作步骤，称为算法。

有了一个好的算法，就可以用任何一种程序设计语言将算法转换成计算机能执行的程序，因此程序是算法的计算机语言描述。

2. 算法的特征

一个算法应具备以下 5 个基本特征。

• 确定性

算法中的每一个操作步骤必须有确切的含义,不能产生二义性,对任意确定的条件,算法只有唯一的执行路径,使相同的输入产生相同的输出。

• 可行性

算法中的操作都是可以通过有限次执行已实现的基本运算来完成的。

• 有穷性

一个算法应包含有限的步骤,每一个步骤都可在有限的时间内完成。

• 有零个或多个输入

一个算法可以没有输入。

• 有一个或多个输出

一个算法应至少有一个输入。

3. 算法的描述

算法有多种表示方法,常用的有自然语言、流程图和伪码。

流程图是描述算法的很好的工具,包括传统流程图和 N-S 流程图。对程序设计的初学者来说,传统流程图具有形象直观、简单方便、可读性好等特点。

构成传统流程图的基本符号如图 1-1-2 所示。

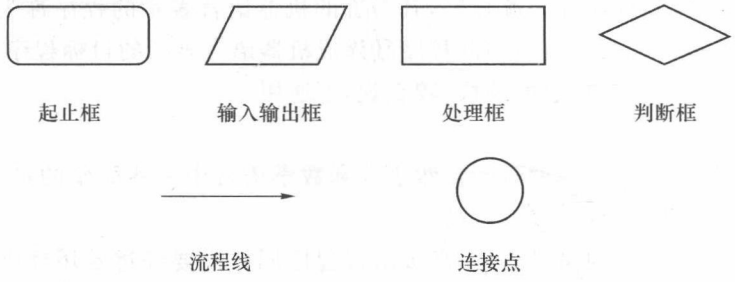

图 1-1-2

1.1.4 结构化程序设计方法

结构化程序设计(structured programming)的概念最早是由迪克斯特拉(E. W. Dijikstra)于 1965 年提出的,是软件发展的一个重要里程碑。他的主要观点是采用自顶向下、逐步求精,使用 3 种基本控制结构构造程序,以模块功能和处理过程为中心进行详细设计。

结构化程序设计语言的特点是程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护及调试。

程序的 3 种基本控制结构包括顺序结构、选择结构和循环结构。

1. 顺序结构

顺序结构中的语句是按书写顺序执行的,即语句的执行顺序与书写顺序一致。顺序结构可用如图 1-1-3 所示的流程图表示。