



世纪高职高专规划教材
高等职业教育规划教材编委会专家审定

C YUYAN CHENGXU SHEJI
SHILI JIAOCHENG

C语言程序设计实例教程

涂玉芬 刘芳 主编
黄琴 白友林 副主编
向隅 主审



北京邮电大学出版社
www.buptpress.com

21 世纪高职高专规划教材

高等职业教育规划教材编委会专家审定

C 语言程序设计实例教程

涂玉芬 刘 芳 主 编

黄 琴 白友林 副主编

向 隅 主 审

北京邮电大学出版社

· 北 京 ·

内 容 简 介

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。

本书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程序设计培训教材和自学者的自学参考书。

图书在版编目(CIP)数据

C 语言程序设计实例教程/涂玉芬,刘芳主编. --北京:北京邮电大学出版社,2011.1

ISBN 978-7-5635-2518-8

I. ①C… II. ①涂…②刘… 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(2010)第 246357 号

书 名: C 语言程序设计实例教程

主 编: 涂玉芬 刘 芳

责任编辑: 王晓丹

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: 北京忠信诚胶印厂

开 本: 787 mm×1 092 mm 1/16

印 张: 17.25

字 数: 429 千字

印 数: 1—3 000 册

版 次: 2011 年 1 月第 1 版 2011 年 1 月第 1 次印刷

ISBN 978-7-5635-2518-8

定 价: 32.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

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。

本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,指导学生探究,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,使学生能够触类旁通,灵活运用。

本教程由武汉铁路职业技术学院涂玉芬、刘芳主编,黄琴、白友林任副主编,向隅主审,张理武、余辉、廖梦虎、刘世荣、叶文、贾华生、何玉梅参编。其中,涂玉芬编写了第 8 章,刘芳编写了第 3、5 章,黄琴编写了第 9 章,白友林编写了第 4、6 章,张理武编写了第 7 章,余辉编写了第 11 章,廖梦虎编写了第 1 章,刘世荣编写了第 2 章,叶文编写了第 10 章,贾华生编写了案例 1 和案例 2,何玉梅编写了附录。另外,参加本教程大纲讨论及案例调试的人员还有王德洪、陈梅等。

本书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程序设计培训教材和自学者的自学参考书。

编 者
2010. 8

目 录

第一部分 基础篇

案例 1:一个简单的学习机	1
第 1 章 C 程序结构及 C 程序调试	5
1.1 C 程序结构	5
1.2 C 程序调试	6
1.3 习题与实训	10
第 2 章 数据与运算	13
2.1 变量定义	13
2.1.1 数据类型	13
2.1.2 常量	14
2.1.3 变量	15
2.2 数据运算	18
2.2.1 算术运算	18
2.2.2 赋值运算	20
2.2.3 逗号运算	21
2.3 习题与实训	22
第 3 章 顺序结构程序设计	27
3.1 数据输入	27
3.1.1 赋值语句	28
3.1.2 字符数据输入函数	28
3.1.3 格式化输入函数	28
3.2 数据输出	30
3.2.1 字符数据输出函数	30
3.2.2 格式化输出函数	31
3.3 注释	32
3.4 应用举例	33
3.5 习题与实训	35
第 4 章 选择结构程序设计	42
4.1 描述判断条件	42

4.1.1 关系表达式	42
4.1.2 逻辑表达式	43
4.2 简单选择结构流程控制	44
4.2.1 简单 if 语句	45
4.2.2 复合语句	46
4.2.3 空语句	46
4.2.4 条件表达式	47
4.3 复杂选择结构流程控制	47
4.3.1 嵌套 if 语句	48
4.3.2 switch 语句	48
4.4 应用举例	50
4.5 习题与实训	53
第 5 章 循环结构程序设计	62
5.1 循环结构流程控制语句	62
5.1.1 while 语句	63
5.1.2 do...while 语句	64
5.1.3 for 语句	65
5.2 多重循环结构	67
5.3 循环结构中的跳转语句	69
5.3.1 break 语句	70
5.3.2 continue 语句	71
5.4 应用举例	71
5.5 习题与实训	78
第 6 章 函数	87
6.1 函数的定义和调用	87
6.1.1 函数定义	88
6.1.2 函数调用	90
6.1.3 函数声明	94
6.2 变量的作用域及生存期	96
6.2.1 局部变量与全局变量	96
6.2.2 动态变量与静态变量	98
6.3 习题与实训	103

第二部分 提高篇

案例 2: 班级成绩管理系统	113
-----------------------	------------

第 7 章 数组	119
-----------------	------------

7.1 一维数组	119
7.1.1 一维数组的定义与引用	120

7.1.2 一维数组初始化	121
7.1.3 应用举例	121
7.2 二维数组	125
7.2.1 二维数组的定义与引用	125
7.2.2 二维数组初始化	126
7.2.3 应用举例	127
7.3 字符数组	130
7.3.1 字符数组的定义、引用与初始化	131
7.3.2 字符串与字符数组	132
7.3.3 字符串函数	133
7.3.4 应用举例	136
7.4 习题与实训	137
第8章 指针	144
8.1 指针的概念及基本运算	144
8.1.1 指针的概念	144
8.1.2 指针变量	145
8.1.3 指针变量的运算	146
8.2 指针与数组	149
8.2.1 指针与一维数组	150
8.2.2 指针与二维数组	155
8.2.3 指针与字符串	160
8.2.4 指针数组	163
8.3 指针与函数	166
8.3.1 指针作函数参数	166
8.3.2 指针函数	170
8.3.3 函数指针	171
8.4 指向指针的指针与命令行参数	172
8.4.1 指向指针的指针	172
8.4.2 命令行参数	174
8.5 习题与实训	176
第9章 结构体与共用体	185
9.1 结构体	185
9.1.1 结构体的基本操作	185
9.1.2 结构体数组	190
9.1.3 结构体指针	193
9.1.4 结构体与函数	195
9.2 链表	197

9.2.1 动态分配函数	199
9.2.2 链表节点类型的定义	200
9.2.3 建立链表	201
9.2.4 链表的基本操作	202
9.3 共用体	207
9.3.1 共用体类型的定义	207
9.3.2 共用体变量的定义	208
9.3.3 共用体变量的引用	209
9.4 枚举	210
9.4.1 枚举类型的定义	210
9.4.2 枚举变量的定义	211
9.4.3 枚举变量的应用	212
9.4.4 自定义类型	214
9.5 习题与实训	215
第 10 章 编译预处理与位运算	230
10.1 编译预处理	230
10.1.1 文件包含	230
10.1.2 宏定义	230
10.2 位运算	233
10.3 习题与实训	235
第 11 章 文件	241
11.1 文件的基本概念	242
11.2 文件的基本操作	243
11.2.1 定义文件指针	243
11.2.2 打开文件与关闭文件	244
11.2.3 读写文件	245
11.2.4 定位文件	251
11.2.5 检测文件是否结束	253
11.3 习题与实训	253
附录	261
参考文献	268

第一部分 基础篇

案例 1: 一个简单的学习机



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");
    printf(" ***** \n");
}

```



程序分析

上述 C 语言程序的流程如图 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。
- 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。

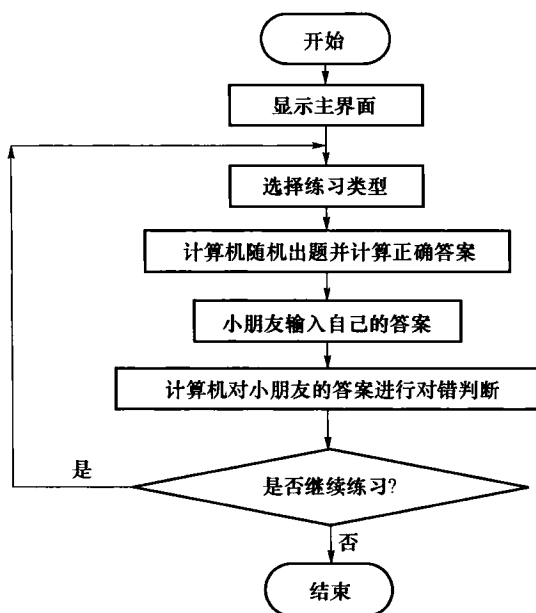


图 1-1



相关知识

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

- C 语言程序的基本结构;
- C 语言程序调试方法;
- 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 程序结构及 C 程序调试

1.1 C 程序结构



问题描述

使用 C 语言编写程序有一定的格式要求。编写 C 语言程序应按格式规范进行。



解决方法

```
void main()                /* 函数首部 */
{                          /* 函数体 */
    int c,s;               /* 声明部分 */
    char x,d;
    ...
    menu();               /* 执行部分 */
    scanf("%c",&x);
    ...
}
```



相关知识

1. C 语言程序结构

- (1) C 程序由函数构成。
- (2) 函数包括函数首部和函数体。
- (3) 每个函数体必须用一对花括号“{”和“}”括起来,包括声明部分和执行部分。
- (4) 声明部分用来对函数中使用的变量和函数作说明。
- (5) 执行部分由基本语句组成,确定完成函数功能的操作。

2. C 语言程序格式说明

- (1) 每个基本语句的结尾必须要用“;”作为终止符。
- (2) 每个程序必须有一个而且只能有一个称为主函数的 main() 函数。
- (3) main() 函数没有固定的位置。
- (4) 程序一般用小写字母书写。
- (5) 注释部分包含在“/*”和“*/”之间,可以出现在程序的任何地方,在编译时忽略。

1.2 C 程序调试



问题描述

编写的 C 语言程序需要在计算机上调试,以找出程序中的错误,修正错误,保证程序的正确。



解决方法

(1) 启动 VC++6.0,进入 VC++ 编辑窗口

选择“开始”→“程序”→“Microsoft Visual Studio 6.0”菜单,单击“Microsoft Visual C++ 6.0”,选择“文件(File)”菜单,单击“新建(New)”,在弹出的“新建(New)”对话框中选择“文件(File)”选项卡,再选择“C++ Source File”,在“文件(File)”下面的文本框中输入源文件名(注意一定要加上扩展名.c),在“目录(Location)”下面的选择框中选择源文件的存放目录,如图 1-1-1 所示,单击“确定”按钮。

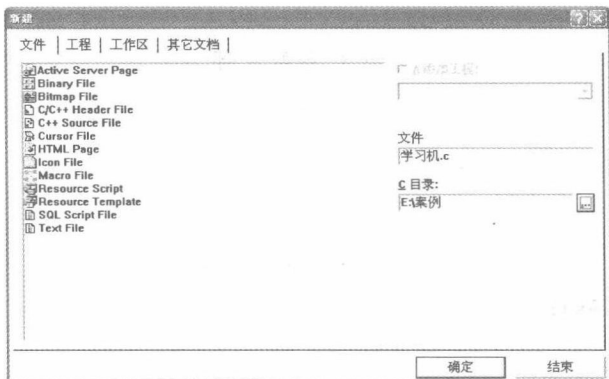


图 1-1-1

(2) 编辑 C 语言源程序

在 VC++ 编辑窗口的编辑区输入源程序,选择“文件(File)”菜单,单击“保存(Save)”,如图 1-1-2 所示。



图 1-1-2

(3) 编译源程序

选择“编译(Build)”菜单,单击“编译(Compile)学习机.c”,如图 1-1-3 所示。



图 1-1-3

(4) 连接目标程序

选择“编译(Build)”菜单,单击“构件(Build)学习机.c”,如图 1-1-4 所示。

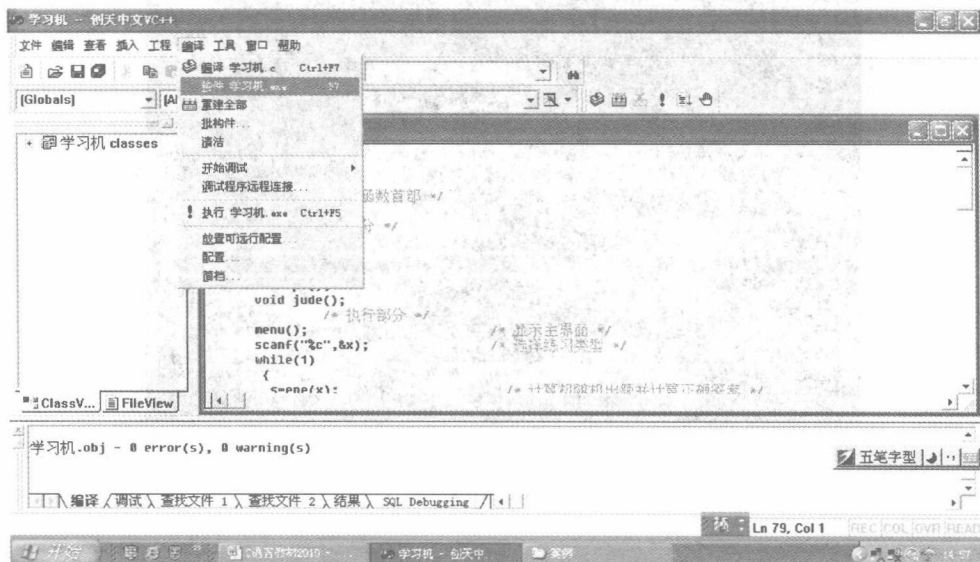


图 1-1-4

(5) 运行可执行程序

选择“编译(Build)”菜单,单击“执行(Execute)学习机.c”,如图 1-1-5 所示。



图 1-1-5

程序运行结果如图 1-1-6 所示,保存程序,退出 VC++6.0。



图 1-1-6



相关知识

C 语言源程序的上机调试流程如图 1-1-7 所示,一般包括编辑、编译、连接和执行 4 个步骤。

(1) 编辑。将 C 语言源程序通过键盘输入到计算机中,修改源程序,以文件形式存盘,源程序的扩展名为.C。

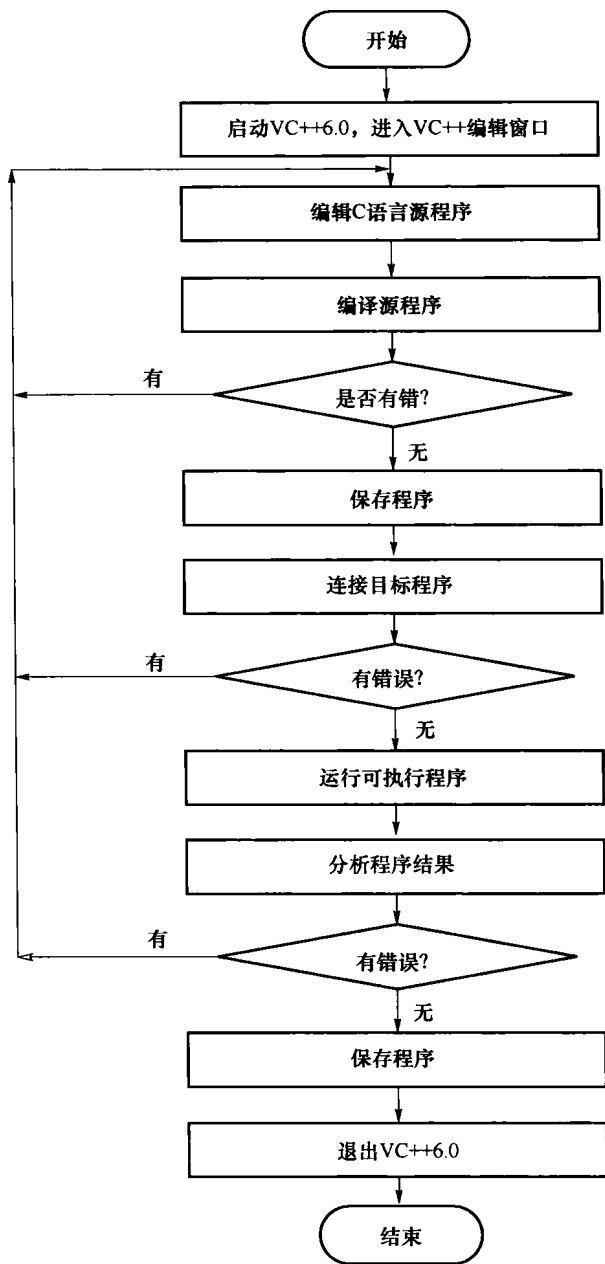


图 1-1-7

(2) 编译。通过编辑将源程序输入到计算机后,经过 C 语言编译器将其生成目标程序,得到的目标文件以 .OBJ 为文件扩展名。

(3) 连接。将经过编译后生成的目标文件和库函数或其他目标程序连接成可执行的目标程序,其扩展名是 .EXE。

(4) 执行。连接后源程序的目标程序就是可执行文件了。在 DOS 系统下,只要键入可执行文件名,按回车键即可。即最后执行的是连接后生成的 .EXE 文件。