



高职高专“十一五”计算机类专业规划教材

C语言 案例教程

赵睿 主编

C YUYAN ANLI JIAOCHE



机械工业出版社
CHINA MACHINE PRESS



高职高专“十一五”计算机类专业规划教材

C 语言案例教程

主 编 赵 睿

副主编 李模刚 张太芳 王 忠



机械工业出版社

本书以 VC++ 6.0 为学习环境,内容涵盖了最新全国计算机等级考试二级考试大纲“C 语言程序设计”的全部内容。全书共 12 章,包括 C 语言的程序结构与基础知识、数据类型及其运算、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、编译预处理、指针、结构体与共用体、位运算、文件操作。每章均由学习目标、本章小结、实训案例、习题组成,部分章节还有调试技术等知识的介绍。

本书适合作为高职高专 C 语言程序设计课程的教材,也可作为计算机考试的辅导教材。本书配有电子教案供教师使用,可发邮件至 wangyx@mail.machininfo.gov.cn 邮箱索取。

图书在版编目 (CIP) 数据

C 语言案例教程/赵睿主编. —北京:机械工业出版社,
2007.1

高职高专“十一五”计算机类专业规划教材

ISBN 978-7-111-20750-4

I. C… II. 赵… 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 (2007) 第 005193 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:李学锋 版式设计:张世琴 责任校对:姚培新

封面设计:马精明 责任印制:洪汉军

北京京丰印刷厂印刷

2007 年 2 月第 1 版·第 1 次印刷

184mm×260mm·15 印张·367 千字

标准书号:ISBN 978-7-111-20750-4

定价:23.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
销售服务热线电话:(010) 68326294

购书热线电话:(010) 88379639 88379641 88379643

编辑热线电话:(010) 68354423

封面防伪标均为盗版

前 言

C语言接近自然语言，易懂好学，适合作为培养学生程序设计思想的第一门语言。几乎所有院校的计算机及相关专业都把C语言程序设计作为专业基础课来设置；全国计算机二级、三级考试都要求掌握C语言，这些足以说明C语言是一门非常重要的语言。

全书共12章，包括C语言的程序结构与基础知识、数据类型及其运算、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、编译预处理、指针、结构体与共用体、位运算、文件操作。每章均由学习目标、授课内容、小结、实训案例、习题组成，部分章节还有调试技术等知识的介绍。

本书具有如下特点：

1. 体现了理论“必需、够用”的基本原则，去掉了偏难、不太常用知识的讲授，适合高职高专学生学习。

2. 采用案例式教学思想来组织教材，全书配有50多个趣味性、实践性很强的实例贯穿全书，提高了学生的学习兴趣。

3. 本着边讲边练的目的，每章都安排了实训案例，有利于提高教学质量，加强学生的实际动手能力和计算机应用能力。

4. 为了涵盖全国计算机等级考试二级考试大纲“C语言程序设计”的全部内容，每章都精选了大量的、有针对性的习题。

5. 为了满足不同读者的需要，书中介绍了Microsoft Visual C++ 6.0及Turbo C 2.0两种集成开发环境。

本书编者均从事多年计算机语言的教学工作，具有丰富的理论与实践经验。现根据学生和教师的反馈，精心汇集资料编成此书。本书适合作为高职高专C语言程序设计课程的教材，也可作为计算机考试的辅导教材。

编写分工：本书由赵睿任主编并统稿，李模刚、张太芳、王忠任副主编。其中第1~4章由赵睿编写，第5~8章及附录A~C由张太芳编写，第9~11章及附录D、附录E由李模刚编写，第12章由王忠编写。在本书编写的过程中，汝宇林、王庆岭、史文权、王诚、黄建华、周晓康、王萌等老师提供了部分资料和习题，在此一并表示感谢。

限于编者水平有限，书中难免存在错误和不足，恳请广大读者批评指正。

编 者

目 录

前言

第1章 C语言的程序结构与基础知识

基础知识

1.1 C语言的发展史和特点

1.1.1 C语言的起源和发展

1.1.2 C语言的特点

1.2 C语言程序总体结构

1.2.1 示例程序

1.2.2 C语言程序特点

1.2.3 C语言程序的一般形式

1.2.4 C语言程序的书写格式

1.3 C语言程序的编程环境

1.3.1 Turbo C 编程环境

1.3.2 Microsoft Visual C++ 6.0 集成

开发环境

本章小结

【实训案例】

习题1

第2章 数据类型及其运算

2.1 数据类型

2.2 常量和变量

2.2.1 常量

2.2.2 变量

2.2.3 整型数据

2.2.4 实型数据

2.2.5 字符型数据

2.3 运算符和表达式

2.3.1 算术运算符和算术表达式

2.3.2 赋值运算符和赋值表达式

2.3.3 关系运算符和关系表达式

2.3.4 逻辑运算符和逻辑表达式

2.3.5 逗号运算符和逗号表达式

2.3.6 条件运算符和条件表达式

2.4 类型转换

本章小结

【实训案例】

习题2

第3章 顺序结构程序设计

3.1 算法

3.1.1 算法的含义

3.1.2 算法的描述

3.2 C语句介绍

3.2.1 控制语句

3.2.2 操作运算语句

3.3 字符数据输入/输出处理

3.3.1 getchar 函数

3.3.2 putchar 函数

3.4 格式输入/输出

3.4.1 printf 函数

3.4.2 scanf 函数

3.5 顺序结构程序应用举例

本章小结

【实训案例】

习题3

第4章 选择结构程序设计

4.1 if 语句

4.1.1 单分支结构

4.1.2 if-else 双分支结构

4.1.3 else-if 多分支结构

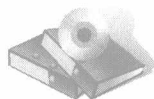
4.2 switch-case 开关语句

4.3 选择结构程序应用举例

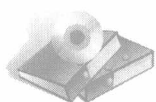
本章小结

【程序调试】

【实训案例】



习题 4	62	7.4.1 形参和实参的特点	108
第 5 章 循环结构程序设计	66	7.4.2 数组作为函数参数	109
5.1 goto 语句	66	7.5 函数的嵌套调用	111
5.2 while 语句	67	7.6 函数的递归调用	113
5.3 do-while 语句	68	7.7 变量的作用域	114
5.4 for 语句	69	7.7.1 局部变量	114
5.5 循环嵌套	71	7.7.2 全局变量	115
5.6 break 语句和 continue 语句	72	7.8 动态存储变量和静态 存储变量	118
5.6.1 break 语句	72	7.8.1 自动变量	118
5.6.2 continue 语句	73	7.8.2 外部变量	118
5.7 综合实例分析	74	7.8.3 静态变量	119
本章小结	76	7.8.4 寄存器变量	120
【实训案例】	76	7.9 外部函数和内部函数	121
习题 5	79	7.9.1 内部函数	121
第 6 章 数组	83	7.9.2 外部函数	121
6.1 一维数组	83	本章小结	122
6.1.1 一维数组的定义	83	【实训案例】	122
6.1.2 一维数组元素的引用	84	习题 7	127
6.1.3 一维数组的初始化	84	第 8 章 编译预处理	132
6.1.4 一维数组的应用	84	8.1 什么是编译预处理	132
6.2 二维数组	86	8.2 宏定义	132
6.2.1 二维数组的定义	86	8.2.1 不带参数的宏定义	132
6.2.2 二维数组的引用	87	8.2.2 带参数的宏定义	133
6.2.3 二维数组的初始化	88	8.2.3 宏定义的终止	134
6.2.4 二维数组应用举例	90	8.3 文件包含	135
6.3 字符数组与字符串	91	8.4 条件编译	137
6.3.1 字符数组	91	本章小结	138
6.3.2 字符串及其使用	93	【实训案例】	139
6.3.3 字符数组的输入/输出	94	习题 8	140
6.3.4 常用的字符串处理函数	94	第 9 章 指针	144
6.3.5 字符数组应用举例	96	9.1 指针与指针变量	144
本章小结	98	9.1.1 指针	144
【实训案例】	99	9.1.2 指针变量	144
习题 6	101	9.1.3 定义指针变量	145
第 7 章 函数	105	9.1.4 指针变量的赋值	146
7.1 C 语言程序的特点	105	9.1.5 指针变量的运算	147
7.2 函数定义	105	9.1.6 指针变量作函数参数	149
7.3 函数调用	107	9.2 数组与指针	151
7.4 函数参数以及参数传递	108		



9.2.1 指向数组的指针变量	151	【实训案例】	182
9.2.2 指向数组元素的指针变量	151	习题 10	186
9.2.3 数组作函数参数	152	第 11 章 位运算	189
9.3 字符串与指针	155	11.1 位运算符	189
9.4 指针与函数	157	11.1.1 “与”运算	189
9.4.1 指针型函数	157	11.1.2 “或”运算	190
9.4.2 函数的指针及指向函数 的指针变量	158	11.1.3 “异或”运算	190
本章小结	159	11.1.4 “取反”运算	191
【实训案例】	159	11.1.5 “左移”运算	191
习题 9	163	11.1.6 “右移”运算	192
第 10 章 结构体与共用体	170	11.2 复合赋值运算	192
10.1 结构体类型及其变量的 定义和引用	170	11.3 位段	193
10.1.1 结构体类型的定义	170	11.3.1 位段的定义	193
10.1.2 结构体类型变量的定义、 引用和初始化	171	11.3.2 位段的引用	194
10.2 结构体数组的定义、 初始化和引用	174	本章小结	195
10.2.1 结构体数组的定义 和初始化	174	【实训案例】	195
10.2.2 结构体数组的成员变量引用 ..	174	习题 11	198
10.3 指向结构体类型数据 的指针	176	第 12 章 文件操作	200
10.3.1 指向结构体变量的指针	176	12.1 C 语言的文件系统	200
10.3.2 指向结构体数组的指针	176	12.1.1 文件内容的存储形式	200
10.3.3 结构体类型数据作 函数形参	177	12.1.2 缓冲文件系统	201
10.4 共用体类型变量的定义 和引用	178	12.2 文件的打开与关闭	201
10.4.1 共用体类型及共用体 变量的定义	178	12.2.1 文件的打开	201
10.4.2 共用体变量的引用	179	12.2.2 文件的关闭	202
10.5 枚举类型	180	12.3 文件的读写	203
10.5.1 枚举类型的定义	180	12.3.1 字符读写函数	203
10.5.2 枚举类型变量的定义 和引用	181	12.3.2 数据块读写函数	205
本章小结	182	12.3.3 格式读写函数	207
		本章小结	208
		【实训案例】	209
		习题 12	214
		附录 A 部分习题参考答案	217
		附录 B ASCII 字符编码一览表	219
		附录 C 常用库函数选摘	220
		附录 D 运算符和结合性	225
		附录 E Turbo C 编译、连接时错误 和警告信息	226
		参考文献	231

第 1 章 C 语言的程序结构与基础知识

学习目标

- 1) 了解 C 语言的发展。
- 2) 理解 C 语言的主要特点。
- 3) 掌握 C 语言程序的开发过程。
- 4) 掌握程序结构。
- 5) 掌握上机环境。

1.1 C 语言的发展史和特点

1.1.1 C 语言的起源和发展

C 语言是与 UNIX 操作系统互相依存的，二者相互促进得以发展。UNIX 的第一个版本是用汇编语言实现的。1970 年 Ken Thompson 在 BCPL 语言基础上开发了 B 语言，并用 B 语言写了 UNIX 操作系统和绝大多数实用程序。D.M.Ritchie 于 1971 年在 B 语言的基础上开发了 C 语言，并于 1972 年开始使用。1973 年 K.Thompson 和 D.M.Ritchie 把 UNIX 系统用 C 语言重写了一遍。此后，C 语言又经过多次改进，直到 1975 年用 C 语言编写的 UNIX 操作系统第 6 版公诸于世之后，C 语言才引起广泛的重视，其目前已成为最流行的程序设计语言之一。

随着微型计算机的日益普及，大量的 C 语言工具得以问世。美国国家标准协会 (ANSI) 于 1983 年专门成立了一个委员会，为 C 语言制定了 ANSI 标准。当时比较流行的是 Turbo C，它不仅满足 ANSI 的 C 语言标准，还提供一个集成开发环境，同时还按传统方式提供了一个命令行编译程序版本以满足不同用户的需要。随着 Windows 编程的兴起，Borland C 和 Microsoft C 受到用户的欢迎。目前比较流行的是兼容 C 语言的 Microsoft Visual C++ 6.0 及 Borland C++ 集成环境。

1.1.2 C 语言的特点

一种语言之所以存在和发展，总是有不同于其他语言的特点。C 语言的特点是多方面的，大体可归结为以下几点。

1. C 语言是中级语言

C 语言常被称为“中级语言”。这并不意味着 C 语言功能差、难以使用，或者不如 BASIC、Pascal 这些语言那样完善，也不是说 C 语言类似于汇编语言，或涉及到使用汇编语言时出现的问题。把 C 语言定义为中级语言，只是意味着它把高级语言的基本结构与低级语言的实用性两者结合起来。

作为一种中级语言，C 语言允许直接访问物理地址，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。

2. C 语言是结构语言

C 语言的主要结构成分是函数。在 C 语言中，函数是基本的结构模块，所有的程序活动内容都包含在其中。函数可以在程序中被定义完成独立的任务，独立地编译成代码，这样可以实现程序的模块化。函数建立后，可以在各种情况下正确运行，而不必担心它对程序的其他部分产生不良影响。

3. C 语言简洁灵活，易于学习和应用

C 语言在表达方法上明了易懂、简单易行。C 语言仅包含 32 个关键字，9 种控制语句，程序书写形式自由，压缩了一切不必要成分。C 语言把一般语言的许多成分都通过显式函数调用来实现，使得编译程序小而精。

4. C 语言具有很强的构造数据类型的能力

程序员可以在简单类型的基础上，根据需求构造各种自定义类型（如数组、指针、结构和联合等），尤其是指针类型数据，使用起来比 Pascal 语言更为灵活、多样。

5. C 语言生成的目标代码质量高，程序执行效率高

C 语言在代码效率方面可以和汇编语言相媲美，一般只比汇编语言生成目标代码低 10% ~ 20%。

C 语言的优点很多，但也有一些不足和缺点。例如，运算优先级太多，不便于记忆，有些还与常规约定有所不同；类型检查太弱，增加了不安全因素。

1.2 C 语言程序总体结构

1.2.1 示例程序

下面通过学习几个 C 语言程序的简单例子，分析一下 C 语言程序的特性。

【例 1-1】 在屏幕上输出“Hello!”信息。

```
#include <stdio.h>
void main ()
{
    printf("Hello!");    /* 输出“Hello!” */
}
```

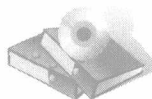
输出结果：

Hello!

这是一个完整的 C 语言程序。“/*”与“*/”及其之间的内容是对程序的注释，用以对语句进行说明，输入程序时可以不输入。

其中 main 表示“主函数”。“{”与“}”之间的语句是主函数的内容，是程序的主体，也称函数体。所有的 C 语言程序都必须包含有一个主函数。程序从主函数的第一行语句开始执行，到最后一条语句结束。

printf(“Hello!”)的功能是在屏幕上显示“Hello!”，是通过 printf 函数完成的。printf 函



数是 C 语言提供的按指定格式进行标准输出的函数，其功能是输出由双引号括起来的字符序列。在其中可以包含控制字符，“\n”是换行控制符，表示该符号之前的内容输出完毕后换行显示后续内容。

C 语言的函数分为两类：系统本身提供的库函数和用户自定义函数，库函数又称标准函数。库函数定义在相应的头文件（头文件的后缀是 .h）中，如果要调用这些库函数，要在源程序最前面使用 #include 语句将相应的头文件包含进来，然后在程序中就可调用这些库函数。printf 函数的头文件是 stdio.h，在程序中应使用 #include <stdio.h> 语句，然后就可以在程序中使用 printf 函数实现输出功能。

【例 1-2】 输入两个数，在屏幕上显示输入的内容和最大的数。

```
#include <stdio.h>
int max(int x,int y); /* Max 函数声明 */
void main()
{
    int a,b,m;
    scanf("%d,%d",&a,&b);
    m = max(a,b); /* 用户自定义函数的调用 */
    printf("a = %d,b = %d,m = %d",a,b,m);
}
int max(int x,int y) /* 用户自定义函数,求两个数中的最大值 */
{
    int m;
    if (x > y)
        m = x;
    else
        m = y;
    return m;
}
```

程序运行时输入“8, 4”

输出结果：

a = 8, b = 4, m = 8

程序中包括主函数和 max 函数。在主函数中，第一行是说明部分，后面多行语句是执行部分。用户自定义函数同库函数一样，可以在主函数中调用。

scanf 函数用来读取用户从键盘输入的值。scanf 函数和 printf 函数的使用方法将在 3.4 节介绍，这里不再赘述。

1.2.2 C 语言程序特点

从以上示例可以看出，C 语言程序有以下特点：

1) 每条语句后面都有“;”作为终止符，它是 C 语句结束符。



2) 每个程序必须有且只能有一个主函数。程序由一个主函数和多个用户自定义函数组成, 它们统称为程序体。

3) 在 C 语言中, 大、小写字母是有区别的。

4) 注释部分在 “/*” 和 “*/” 之间, 在编译时被 C 语言的编译器忽略。

1.2.3 C 语言程序的一般形式

C 语言程序的一般形式如下所示, 其中 f1 ~ fn 代表用户定义的函数。

```
编译预处理  
全程变量说明  
void main ()  
{  
    局部变量说明  
    程序段  
}  
返回类型 f1 (形式参数说明)  
{  
    局部变量说明  
    程序段  
}  
返回类型 f2 (形式参数说明)  
{  
    局部变量说明  
    程序段  
}  
:  
返回类型 fn (形式参数说明)  
{  
    局部变量说明  
    程序段  
}
```

说明:

1) C 语言变量在使用之前必须先定义其数据类型, 未经定义的变量不能使用。定义变量类型的语句必须放在可执行语句前面。

2) 程序段由多行语句组成。

3) 形式参数说明用于传值, 见函数调用部分。

4) 局部变量、全程变量的用法含义, 见 7.7 节。

5) 花括号内的内容统称为函数体。

6) 用户定义的函数名由用户命名, 命名应符合标识符的命名规则 (见第 2 章内容)。

7) 用户自定义函数可置于主函数之前, 也可置于主函数之后。若在主函数之后, 要在



主函数之前对被调用函数进行函数声明。

1.2.4 C 语言程序的书写格式

C 语言程序书写比较灵活，一条语句中只要不把单词分开，可随意分隔。但编程者都应养成良好的书写习惯，以便提高程序的可读性。对 C 语言程序的书写格式习惯上有如下要求：

- 1) C 语言程序中每行可写一条语句，也可写多条语句，一般一行写一条语句。
- 2) C 语言程序中每条语句的末尾必须加一个分号，分号用来表示一条语句的结束。
- 3) C 语言程序中的花括号必须成对出现。
- 4) C 语言程序的书写要注意适当的缩进，以便使程序更加清晰易读。

5) 在书写程序时，要习惯使用注释，所注释的内容只能阅读，而不参与编译和运行。使用注释可增加程序的可读性。

1.3 C 语言程序的编程环境

C 语言程序的开发过程有编辑、编译、连接和运行四个阶段。首先，将源程序输入机器，形成源程序文件（.c）。编辑源程序文件，原则上可以使用任何编辑器。然后，由 C 语言编译程序对源程序进行编译，将源程序翻译成机器代码语言的目标程序文件（.obj）。若编译过程中发现错误，将得不到目标文件，这时要根据错误信息，返回到编辑状态修改错误，再编译。其后，由 C 语言连接程序对目标程序进行连接，将目标程序连接成可执行程序文件（.exe）。最后，运行可执行程序，若输出结果不正确，说明源程序中的算法错误，要返回到编辑状态修改算法。再编译、连接、运行和检查结果，反复进行这四个过程直到运行结果正确为止。在 C 语言的最初应用阶段，这些工作需要分步进行，现在一般都使用集成环境。所谓集成环境就是将 C 语言的编辑、编译、连接和运行都集成到一个综合环境中。下面主要介绍两种比较常用的 C 语言集成开发环境 Microsoft Visual C++ 6.0 和 Turbo C。

1.3.1 Turbo C 编程环境

1. 进入 Turbo C 编程环境

找到 Turbo C 的安装目录，运行 TC 程序，进入 Turbo C 编程环境。

2. 程序的编辑、保存、调试与运行

(1) 编辑程序 按下“Alt + E”组合键，在编辑窗口内出现“_”光标，此时就可以输入程序了。录入程序时要仔细，注意区分大小写、中文标点符号和英文标点符号。

(2) 保存程序 按下“F2”键，若是第一次命名，会出现如图 1-1 所示对话框，询问是否要修改文件名。在对话框中输入文件名（如 E:\zhang\abc.c）即可。



图 1-1 输入文件名示意图



(3) 编译程序 程序录入完后,按下“Alt + F9”组合键,TC 调试环境将编译输入的程序。如果程序没有语法错误,编译成功,可得到目标程序;若有错误则修改源程序,错误提示说明如图 1-2 所示。修改后再次按下“Alt + F9”组合键进行编译。

屏幕中 Compiling 对话框显示的是编译时遇到的语法错误 (Errors) 和逻辑上可能存在的错误警告 (Warnings)、编译程序行数、文件名称等信息。Warnings 一般不影响连接和运行过程的进行,但运行结果可能有错。灰色亮条显示错误及警告的位置,Message 框显示的是具体的错误及其在当前文件中的行数,Edit 框上标注出出错的大概位置,可以按上下箭头按钮浏览各种错误和警告。按回车键返回编辑状态,检查并修改错误;若按回车键不能返回编辑状态,可能是连接错误,此时,可将 Message 框中的灰色亮条移动至最上面一条再按回车键,方可返回编辑状态。



图 1-2 程序编译示意图

(4) 运行程序 按下“Ctrl + F9”组合键就可以连接、运行程序了,按“Alt + F5”键就可查看程序在输出屏幕上的输出结果。

1.3.2 Microsoft Visual C++ 6.0 集成开发环境

1. 进入 Visual C++ 6.0 集成开发环境

选中 Microsoft Visual Studio 6.0 程序组的“Microsoft Visual C++ 6.0”选项,启动 Visual C++ 6.0 开发环境,在提示界面上单击“结束”按钮,进入 Visual C++ 6.0 开发环境。

2. 建立控制台应用程序

Visual C++ 也提供了编制 DOS 程序的方法,这就是控制台程序。利用 Visual C++ 提供的向导工具,编写 C 语言程序源文件。具体步骤如下:

1) 选中 File (文件) 菜单的“New (新建)”命令,弹出如图 1-3 所示的“新建”对话框,并自动选中“工程”选项卡。

2) 在“工程”选项卡中选择“Win32 Console Application”选项,并在右边“工程”输入



框中输入项目名称（用户自己定义）。

3) 若想改变项目的存放位置，在“位置”输入框中输入你自己的文件夹名称，这个文件夹实际上就是你编写的源程序存放位置。

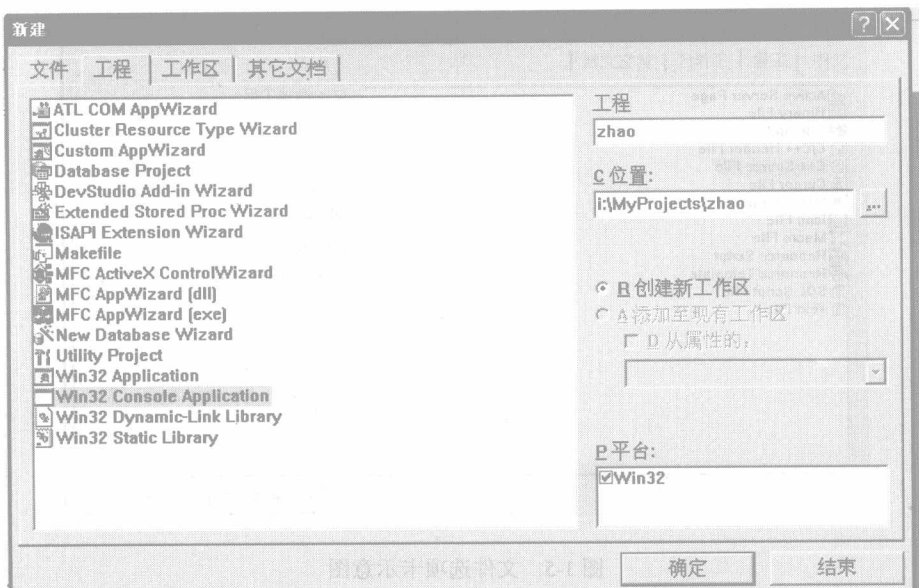


图 1-3 “工程”选项卡

4) 选择完毕，单击“确定”按钮即可进入如图 1-4 所示的 Win32 Console Application 制作向导第一步。

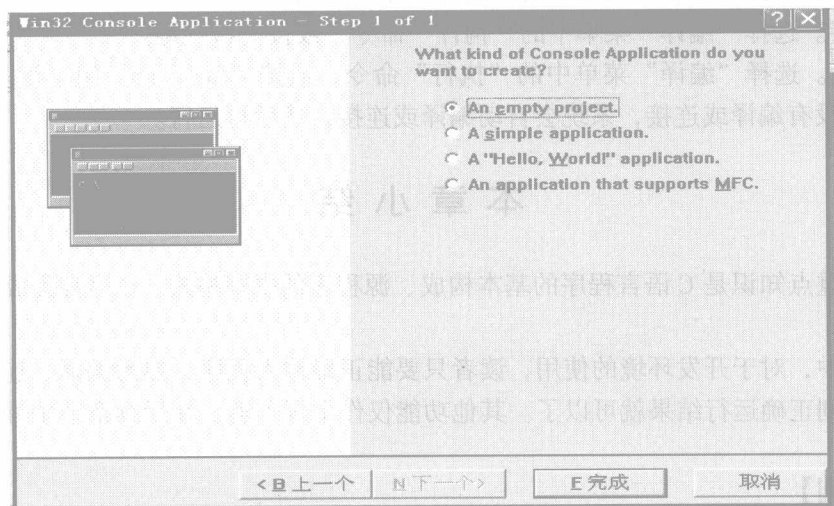
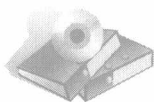


图 1-4 Win32 Console Application 制作向导第一步

5) 要编辑 C 语言程序文件应选择“An empty project”选项。单击“完成”按钮，出现一个信息框，再单击“确定”按钮，完成空项目的建立。

6) 再从文件菜单中选择新建命令，并自动选中 File（文件）选项卡。选中“C++”



Source File”选项，在右方的文件输入框中输入 C 语言程序文件名（如 abc.c）。这里输入的文件名一定要加后缀 .c，系统默认的后缀是 .cpp。单击“确定”按钮，返回集成环境，开始输入并编辑源程序，如图 1-5 所示。

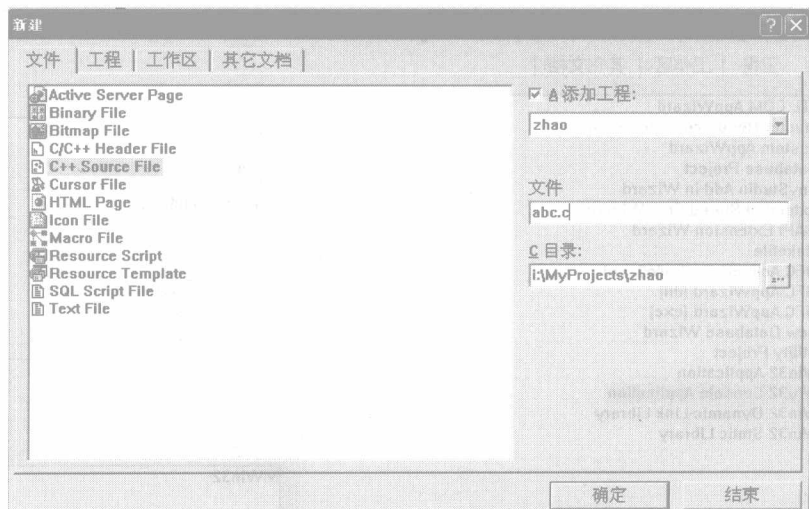


图 1-5 文件选项卡示意图

说明：在 Visual C++ 环境中，若使用标准输入输出函数，源程序中必须包含 `stdio.h` 库文件。所有函数的声明和定义，在函数名前必须写返回类型，没有返回值的函数，返回类型为 `void`。

3. 编译、连接和运行

- 1) 编译。选择“编译”菜单中的“编译”命令，或按“Ctrl + F7”组合键。
- 2) 连接。选择“编译”菜单中的“构件”命令，或按“F7”键。
- 3) 运行。选择“编译”菜单中的“执行”命令，或按“Ctrl + F5”组合键。在执行时，若源程序还没有编译或连接，系统会自动编译或连接，然后再运行。

本章小结

本章的重点知识是 C 语言程序的基本构成、源程序的书写格式、C 语言集成开发环境的使用。

在本章中，对于开发环境的使用，读者只要能正确输入程序，熟悉编译、连接、运行等操作并能得到正确运行结果就可以了。其他功能仅作个别了解，在后续章节中慢慢熟悉。

【实训案例】

实训目的与要求

- 1) 掌握一种 C 语言程序的开发环境。
- 2) 掌握 C 语言应用程序的开发过程。



- 3) 掌握 C 语言应用程序的最基本结构。
- 4) 练习输入并运行一个简单的 C 语言程序。
- 5) 运行一个自己编写的程序。

实训示例

【示例 1】 实现在屏幕上显示如下两行文字：

Hello, students!

I am students!

```
#include <stdio.h>
void main()
{
    printf("Hello,students! \n");
    printf("I am students!");
}
```

【示例 2】 实现两个整型数相加，并显示结果。

```
#include <stdio.h>
void main()
{
    int a = 12, b = 45, sum;
    sum = a + b;
    printf("sum = %d", sum);
}
```

输出结果：

sum = 57

习 题 1

1-1 选择题

1. 一个 C 语言程序的执行是从 ()。
 - A) 本程序的主函数开始，到主函数结束
 - B) 本程序文件的第一个函数开始，到本程序文件的最后一个函数结束
 - C) 本程序的主函数开始，到本程序文件的最后一个函数结束
 - D) 本程序文件的第一个函数开始，到本程序主函数结束
2. 以下叙述正确的是 ()。
 - A) 在 C 语言程序中，主函数必须位于程序的最前面
 - B) C 语言程序的每行中只能写一条语句
 - C) C 语言本身没有输入输出语句
 - D) 在对一个 C 语言程序进行编译的过程中，可发现注释产生的拼写错误



3. 以下叙述不正确的是 ()。

- A) 一个 C 语言程序可由一个或多个函数组成
- B) 一个 C 语言程序必须包含一个主函数
- C) C 语言程序的基本组成单位是函数
- D) 在 C 语言程序中, 注释说明只能位于一条语句的后面

4. C 语言规定: 在一个源程序中, 主函数的位置 ()。

- A) 必须在最开始
- B) 必须在系统调用的库函数的后面
- C) 可以任意
- D) 必须在最后

5. 一个 C 语言程序是由 ()。

- A) 一个主程序和若干子程序组成
- B) 函数组成
- C) 若干过程生成
- D) 若干子程序组成

1-2 填空题

1. C 语言程序的基本单位是_____。

2. 一个 C 语言程序只能包括一个_____。

3. 在一个 C 语言程序中, 注释部分两侧的分界符分别为_____和_____。

4. 在 C 语言程序中, 基本输入操作是由库函数_____完成的, 输出操作是由库函数_____完成。

1-3 问答题

- 1. 什么是函数体?
- 2. 什么是编译、连接?
- 3. 程序文档至少包含哪些内容?

1-4 编程题

- 1. 输入并运行示例程序, 熟悉开发 C 语言程序的方法和步骤。
- 2. 参照示例 1, 编写一个 C 语言程序, 输出以下信息:

```
* * * * *
```

```
Hello, teacher!
```

```
* * * * *
```

3. 编写程序输出以下图案:

```
*
```

```
* * *
```

```
* * * * *
```

```
* * *
```

```
*
```