

程序设计案例教程

(C语言版)

主编 赵秀岩 于晓强

CHENGXU
SHEJI ANLI
JIAOCHENG

 高等教育出版社
HIGHER EDUCATION PRESS

程序设计案例教程（C 语言版）

Chengxu Sheji Anli Jiaocheng(C Yuyan Ban)

主 编 赵秀岩 于晓强
副主编 康 丽 房 媛



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

TP312C
2296

内容简介

本书为高等学校计算机程序设计课程教材,全书共分十一章。本书一改传统教材以“知识点”为主线的编写方法,采用以“案例”为主线,案例引导知识点的编写形式。本书重点介绍了程序设计的过程、方法以及计算思维的基本思想。本书以C语言作为程序设计的实现语言,详细介绍了C语言的数据类型与运算、程序设计基本结构、数组、函数、指针、其他构造数据类型、文件等内容。本书每一章都精心设计了经典案例,案例设计围绕本章涉及知识点来完成,让学生在编程实践中来认识和理解相关知识,使得学生对知识点的认识和理解都更加直观、深刻。同时每个案例中的“分析与思考”、“程序流程”环节注重学生计算思维能力的培养;“拓展训练”和“技能提高”环节注重学生举一反三和实践能力的培养。

本教材适合作为高等学校本科生计算机程序设计课程教材,还可以用于初学者自学,也可以作为培训机构的培训教材。

图书在版编目(CIP)数据

程序设计案例教程:C语言版/赵秀岩,于晓强主编.
—北京:高等教育出版社,2014.2

ISBN 978-7-04-039359-0

I.①程… II.①赵…②于… 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013039号

策划编辑 吴勇 周征	责任编辑 段会青 周征	封面设计 姜磊	版式设计 杜微言
插图绘制 尹莉	责任校对 刘娟娟	责任印制 毛斯璐	

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100120
印 刷 北京北苑印刷有限责任公司
开 本 787mm×1092mm 1/16
印 张 24.75
字 数 600千字
购书热线 010-58581118

咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>
版 次 2014年2月第1版
印 次 2014年2月第1次印刷
定 价 49.80元



本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物料号 39359-00

前 言

随着信息技术的发展,计算机应用技能以及计算思维能力已成为当代大学生所必备的基本素质之一。多数高校选择开设程序设计基础这样的课程,以培养学生利用计算机解决专业领域研究与应用问题的能力。

传统的程序设计教材注重编程语言的语法及其规则介绍,学生依照教材完成学习过程,虽然可以精通高级编程语言的语法规则,但是对解决问题的思路、问题求解过程的能力培养却十分薄弱,甚至不具备应用计算机独立解决实际问题的思维能力。同时,传统的程序设计教材在内容设计上往往将理论与实验分开编写,造成理论知识点介绍及所使用的案例与实验教材中相应的实验内容之间没有紧密的关联设计。这会使学生既对理论部分知识点的理解和认识感到困难,又对实验部分感到无从下手,只能照猫画虎,一知半解地完成实验。这样的情况,往往使得我们对学生的实践动手能力的培养事倍功半。

本教材正是笔者在教学实践中总结了传统教材在上述两方面的缺陷和不足编写而成的。首先,在教材的内容结构上,改变了传统教材以“语法”为主的内容组织,采取以“案例”为核心的组织方式。在教学方法上以“提出问题—分析问题—解决问题—总结问题—应用问题”为主线,旨在引导和帮助学生建立解决问题的常规思路,掌握程序设计的思维过程。

其次,在教材的形式组织上,实现了“理论实践一体化”的无缝设计,每个知识单元都设计成“案例组”的形式。每一组案例涉及三个程序,第一个是理解程序,主要由教师讲解;第二个是模仿程序,问题难度、算法都与第一个程序十分类似,由学生模仿练习;第三个是提高程序,在前面两个程序的基础上,难度和算法略有变化和提升,由学生独立完成。这样就会在潜移默化中培养学生在面对复杂问题的时候,采用“从简到繁”的解决方法和思维方式。

本书突破了传统教材理论与实践分别编写组织的特点,将理论教学和实践教学完美结合在了一起,如果程序设计课程能够在机房授课,即保证学生人手一机的情况下,那么本教材将是教学的合适选择。即使是在多媒体教室授课,也可以选择本教材,同样可以取得良好的教学效果。

本书由大连工业大学计算机基础教研室教师编写完成。其中第1、2章由康丽编写,第3、4章由邵利编写,第5章由刘英编写,第6章由房媛编写,第7章由赵秀岩编写,第8章由于晓强编写,第9章由王美航编写,第10、11章由贺晓阳编写,全书由赵秀岩老师统稿。

由于编写时间紧张,加之编者水平有限,书中难免有疏漏和错误,敬请各位读者和同行批评指正。

编 者

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010)58581897 58582371 58581879

反盗版举报传真 (010)82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

目 录

第 1 章 概述	1	第 3 章 顺序结构与输入输出	54
1.1 基本概念	1	3.1 字符输入输出函数	54
1.2 初识 C 程序	2	3.1.1 getchar 函数概述	54
1.3 C 程序的基本结构	3	3.1.2 putchar 函数概述	55
1.4 运行 C 程序	5	3.1.3 字符输入输出函数的应用 ...	55
1.4.1 Turbo C 2.0	5	3.2 格式输入输出函数	58
1.4.2 VC++ 6.0	7	3.2.1 scanf 函数概述	58
1.4.3 Win-TC	13	3.2.2 printf 函数概述	61
本章总结	16	3.2.3 scanf 函数和 printf 函数的 特殊格式控制	62
第 2 章 程序设计基础	17	3.2.4 格式输入输出函数的综合 应用	64
2.1 算法	17	本章总结	68
2.2 数据类型	19	第 4 章 选择结构	70
2.2.1 常量和变量	20	4.1 if 语句	70
2.2.2 整型数据	22	4.1.1 if 的单分支形式	70
2.2.3 实型	25	4.1.2 if 的双分支形式	73
2.2.4 字符型	28	4.1.3 if 的多分支形式	76
2.3 运算符和表达式	32	4.2 switch 语句	82
2.3.1 算术运算符和算术 表达式	34	4.3 条件语句的嵌套使用	87
2.3.2 自增、自减运算符	35	4.3.1 if 语句嵌套的一般形式	87
2.3.3 关系运算符和关系表达式 ...	36	4.3.2 if 语句嵌套的应用	87
2.3.4 逻辑运算符和逻辑表达式 ...	38	本章总结	91
2.3.5 赋值运算符和赋值表达式 ...	40	第 5 章 循环结构	92
2.3.6 逗号运算符和逗号表达式 ...	42	5.1 while 循环语句	92
2.3.7 条件运算符	44	5.2 do-while 循环语句	95
2.3.8 位运算符	45	5.3 for 循环语句	98
2.3.9 sizeof 运算符	48	5.4 for 循环语句的特殊格式	101
2.3.10 各类型数据间的混合 运算	49	5.5 用 goto 语句和 if 语句构成的 循环语句	103
2.4 宏常量与宏替换	51		
本章总结	53		

5.6 循环的嵌套	105	7.6.3 变量的存储类型与 生存期	178
5.7 break 语句和 continue 语句	108	7.6.4 动态存储类型	179
5.7.1 break 语句	108	7.6.5 静态存储类型	180
5.7.2 continue 语句	109	7.6.6 寄存器存储类型	181
5.8 经典程序举例	111	7.6.7 外部存储类型	181
本章总结	117	7.7 函数的综合应用	184
第 6 章 数组	118	本章总结	188
6.1 数组的本质	118	第 8 章 指针	189
6.2 一维数组	120	8.1 指针的概念及指针变量	190
6.2.1 一维数组的定义	120	8.1.1 指针的概念	190
6.2.2 一维数组的引用	121	8.1.2 指针变量的定义	191
6.2.3 一维数组的初始化	123	8.1.3 指针变量的赋值	191
6.2.4 一维数组的输入与输出	124	8.1.4 指针运算符	192
6.2.5 一维数组的应用	126	8.2 指针的运算	196
6.3 二维数组	132	8.2.1 指针的赋值运算	196
6.3.1 二维数组的定义	132	8.2.2 指针的算术运算	196
6.3.2 二维数组的引用和初 始化	134	8.2.3 指针的关系运算	197
6.3.3 二维数组的应用	136	8.3 指针与数组	197
6.4 字符数组和字符串	139	8.3.1 指向一维数组元素的指针 变量	197
6.4.1 字符数组的定义、引用和 初始化	139	8.3.2 通过指针引用数组元素	198
6.4.2 字符串处理函数	141	8.3.3 指针与字符串	201
6.4.3 字符数组的应用	147	8.3.4 指向多维数组元素的指针 变量	205
本章总结	151	8.4 指针与函数	206
第 7 章 函数	152	8.4.1 指针变量作为函数参数	206
7.1 函数的定义	153	8.4.2 数组名作为函数参数	208
7.2 函数的声明	159	8.4.3 返回指针值的函数	211
7.2.1 库函数的声明	159	* 8.4.4 指向函数的指针	213
7.2.2 自定义函数的声明	160	8.5 指向指针的指针变量	215
7.3 函数的参数	161	8.6 指针与二维数组	215
7.4 函数的返回值	167	8.6.1 二维数组的结构	215
7.5 函数的调用关系	169	8.6.2 数组指针	216
7.6 全局变量与局部变量	173	8.6.3 指针数组	219
7.6.1 局部变量	173	* 8.6.4 指针与二维字符数组	220
7.6.2 全局变量	174	8.7 main 函数的参数	224

本章总结	227	11.1.2 标准流和重定向	283
第 9 章 结构体	228	11.1.3 文件是什么	284
9.1 结构体类型的定义	229	11.2 标准 I/O	285
9.2 结构体变量	230	11.2.1 打开文件	285
9.3 结构体数组	234	11.2.2 模式	286
9.4 结构体与函数	239	11.2.3 关闭文件	287
9.5 结构体与指针	244	11.2.4 文件结束检测函数	288
9.6 结构体成员变量为结构体类型	249	11.3 文件 I/O	289
9.7 链表	252	11.3.1 格式化 I/O	289
9.7.1 链表的定义	253	11.3.2 字符的 I/O	291
9.7.2 链表的基本操作	256	11.3.3 字符串的 I/O	294
本章总结	272	11.3.4 数据块的 I/O	296
第 10 章 共用体与枚举	273	11.4 文件的随机读写	298
10.1 共用体类型	274	11.5 文件应用案例	299
10.1.1 共用体类型的定义	274	本章总结	302
10.1.2 共用体变量的定义	274	附录 I 经典习题及答案	303
10.2 枚举类型	278	附录 II 常用运算符优先级和结合性 ...	379
10.2.1 枚举类型的定义	278	附录 III 常用库函数	380
10.2.2 枚举变量的定义	278	附录 IV 常用字符 ASCII 码值对照表	384
本章总结	282	参考文献	385
第 11 章 文件	283		
11.1 流	283		
11.1.1 文件指针	283		

第 1 章 概述

当今世界计算机已无处不在,从军事、科研到生产、生活、各行各业都离不开计算机;从学习、工作到休闲、娱乐,计算机以各种形式成为了人们不可缺少的“伴侣”。当你使用计算机进行数据计算或者游戏娱乐的时候,有没有想过这些功能是如何实现的呢?答案显然是通过运行程序实现的。若要处理数据就要运行数据处理程序,若要打游戏就要运行游戏程序。那么,这些程序是哪里来的呢?你有没有想过自己动手编写一个程序让计算机去运行?那么,现在就让我们开始神奇的编程之旅吧!

本章主要介绍 C 程序的基本概念和运行方法。

1.1 基本概念

1. 程序

编程之前,首先要知道什么是程序。程序(Program)是为了完成特定任务而用计算机语言编写的命令序列的集合。

2. 计算机语言

编程需要使用计算机语言。计算机语言是人与计算机进行交流的工具,主要分为三大类:机器语言、汇编语言和高级语言。

(1) 机器语言

完全由 0 和 1 构成的机器指令,可直接执行。但含义不清,难学难记,难理解,同一个指令,不同型号的机器使用的 0、1 串可能不同。

(2) 汇编语言

为了方便理解和记忆,减轻编程人员的工作量,出现了汇编语言。汇编语言是由一些含义相对清晰的英文字母、符号来替代相应功能的二进制串指令,如“ADD”代表相加等。汇编语言不能够直接运行,需要由翻译程序将其转换成机器语言才能够运行。汇编语言依旧依赖计算机硬件,不同型号的硬件汇编指令可能不同。这种针对特定硬件的汇编指令,执行效率比较高,但移植性不好,更换硬件可能要更改程序甚至重新编程。

(3) 高级语言

汇编语言的可读性和可移植性不够好,人们期待使用更接近人类自然语言习惯、在不同型号的机器上通用的编程语言,高级语言应运而生。高级语言是和人类自然语言(英语)较为接近的编程语言,具有良好的可读性,不依赖具体的计算机型号和类型,通用性较强,是目前大多数编程者的选择。高级语言有很多种,如 C、C++、Java、C# 等。高级语言也不能够直接运行,需要借助“翻译程序”转换为机器语言。高级语言的出现,使得编程不再是极少数专业人士的专属。而大量编程爱好者的加入,又极大地推动了计算机技术的飞速发展。

3. C 语言

C 语言是一种高级语言,诞生于世界著名的贝尔实验室。C 语言自 1972 年问世以来,经过

多年的发展,现已成为一种功能强大、应用广泛的高级语言。它既有高级语言的基本特征,又具有汇编语言的部分特征,因而成为经久不衰的经典编程工具。同其他语言相比,C 语言具有以下特点:语言简洁紧凑,运算符及数据结构丰富;有结构化的控制语句但语法限制不很严格;可以进行位操作且生成目标代码质量高、可移植性好等。

与所有高级语言一样,C 语言需要经过“翻译程序”转换为机器语言才能够运行。C 程序的执行过程如图 1.1 所示。

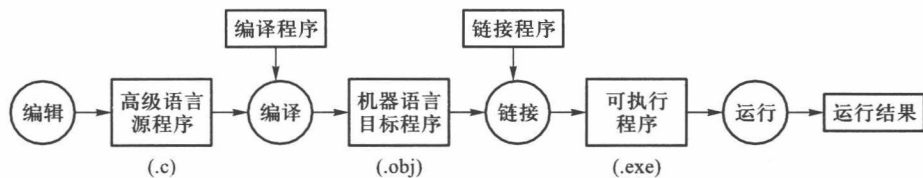


图 1.1 C 程序的执行过程

- (1) 编辑:使用编辑器录入程序代码,并保存成扩展名.c 的源程序文件。
- (2) 编译:用编译程序将源程序翻译成机器语言程序,即生成扩展名为.obj 的目标程序文件。
- (3) 链接和运行:通过链接程序将引用的所有函数链接到目标程序中,生成扩展名为.exe 的可执行程序文件,运行即有相应结果。

1.2 初识 C 程序

了解了基本概念后,本节通过两个案例来初步认识 C 程序。

【案例 1.1】输出文字。

一、案例设计目的

✓ 初步认识 C 程序。

二、案例主要代码

```
#include<stdio.h>
void main()
{
    printf("我的编程之旅开始啦!\n");
}
```

三、案例运行结果(图 1.2)

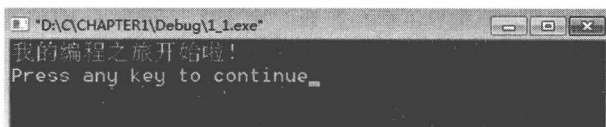


图 1.2 案例 1.1 的运行结果

因为 C 语言的语法和语句更接近自然语言的表述习惯,其程序可读性更好。请阅读下面的程序,猜一下它的功能。

【案例 1.2】请猜程序功能。

一、案例设计目的

✓ 猜测 C 程序功能,体会 C 程序的可读性。

二、案例主要代码

```
#include<stdio.h>
void main( )
{
    int a,b,c;
    a=300;b=18;
    c=a+b;
    printf("%d\n",c);
}
```

三、拓展训练

观察案例 1.1、1.2 两个程序,总结二者的异同。模仿案例 1.2,编写实现两个数相减(如 300 减 18)的程序。

四、技能提高

模仿案例 1.2,编写实现三个数相加的程序。

1.3 C 程序的基本结构

从案例 1.1 和案例 1.2 两个程序不难总结出,程序都有相同部分 `main( ){ }`。

【案例 1.3】编写求两个数平均值的程序。

一、案例设计目的

✓ 掌握 C 程序的基本构成。

✓ 理解 C 程序的编写规则。

二、案例主要代码

```
(注:为方便分析说明,本书后续部分程序添加了行标号,运行时不需要)
行 1:  /*求两个数的平均值*/
行 2:  #include<stdio.h>
行 3:  void main( )           //主函数首部
行 4:  {                     //“{”表示函数体开始
行 5:      int x,y,p;
```

```
行 6:      x=60;y=40;
行 7:      p=(x+y)/2;
行 8:      printf("%d\n",p);
行 9:  }
```

//“}”表示函数体结束

三、案例运行结果(图 1.3)

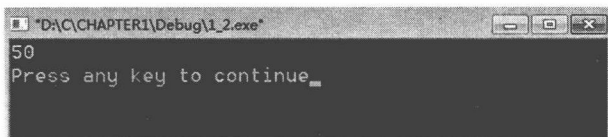


图 1.3 案例 1.3 的运行结果

四、案例小结

1. 主函数 main

C 程序由函数构成,一个 C 程序可以有多个函数,但必须有且只能有一个 main(主函数);程序从 main 函数开始执行。

2. 函数

函数由函数首部和函数体组成。其中,函数体为{ }扩起来的部分;函数首部的完整形式为, 函数类型 函数名称(函数参数类型 函数参数 1,函数参数类型 函数参数 2,……)

案例中 main 函数的首部是行 3“void main()”。其中,main 是函数名称;函数名称前面的 void 是函数类型;因为没有函数参数,所以“()”中为空。紧跟在函数首部后的左花括号“{”表示函数体开始,相应的右花括号“}”表示函数体结束,即该函数结束。有关函数的详细介绍见后续函数章节。

3. 语句

函数体内为语句,包括说明性语句、可执行语句,所有语句必须以分号“;”结束。

4. 注释

注释是不参与程序执行的内容,可以出现在程序的任何位置。注释部分对程序执行没有任何影响,只是供编程者查看。常用的注释方式有以下两种:

(1) “/* */”形式

用“/*”和“*/”括起来,可用于行中部分注释、单行或多行注释,如案例 1.3 中“/* 求两个数的平均值 */”为注释。

(2) “//”形式

用“//”实现单行注释(个别编译环境不支持,如 Win-TC),表示从“//”开始该行后面均为注释部分,如案例 1.3 中的“//主函数首部”等。

5. 书写格式

C 语言书写格式自由,一条语句可以占一行或多行,一行也可以有一条或多条语句;区分大小写,多习惯用小写字母;程序书写格式虽未规定,但多采用缩进格式;所有标点符号必须为英文格式,否则不能识别。

以上部分内容,读者也许暂时不能完全理解,但在以后的学习过程中,将会不断有所体会。

1.4 运行 C 程序

前几节介绍了 3 个 C 程序,并分析出每个程序的运行结果。本节将介绍如何在计算机上运行 C 程序。C 程序的开发主要包括编辑、编译、链接和运行几个步骤。常用的高级语言开发工具集成了编辑需要的编辑器、编译需要的编译程序、链接需要的链接程序。下面简要介绍常见的 C 程序开发工具 Turbo C 2.0、VC++ 6.0 和 Win - TC。

1.4.1 Turbo C 2.0

打开 tc2 文件夹找到并运行 tc.exe 文件,看到软件的版本信息窗口后按回车(Enter)键即进入 Turbo C 2.0 开发环境。主界面包括菜单栏、编辑区、信息区、提示栏,如图 1.4 所示。



图 1.4 Turbo C 2.0 主界面

菜单栏包括 File、Edit、Run、Compile、Project 等菜单项。Edit 的功能是激活编辑区进入编辑状态,除此项外其他各项都有下拉子菜单。Turbo C 2.0 环境下鼠标不可用,完全用键盘。选择菜单项可使用 Alt 键加上菜单项首字母,结合上下左右键切换菜单项。

以案例 1.1 的执行为例,在 Turbo C 2.0 中运行程序的主要步骤如下:

一、编辑

按 Alt+F 组合键或在 File(文件)菜单项选中的状态按下回车键,在子菜单中通过 ↓ 键移动选择 New 并按回车键,光标切换到编辑区域即可开始编辑程序,如图 1.5 所示。如果需要打开已有 C 程序,选择 Load 并按回车键,键入要打开文件的路径名称后按回车键即可。

在编辑区域输入案例 1.1 程序。由于 Turbo C 2.0 不支持中文,所以需要修改汉字为英文。编辑完毕应保存程序,可使用 File 菜单下的 Save(保存)命令或快捷键 F2。同其他软件一样,首次保存会弹出保存路径窗口,输入要保存的位置和文件名后按回车键即完成保存,如图 1.6 所示。

二、编译

完成程序的输入后,选择 Compile 菜单中的 Compile to Obj 菜单项进行编译。编译完成后会出现编译结果窗口,如果错误数不为 0,则表示程序中有错误需要修改,如图 1.7 所示。

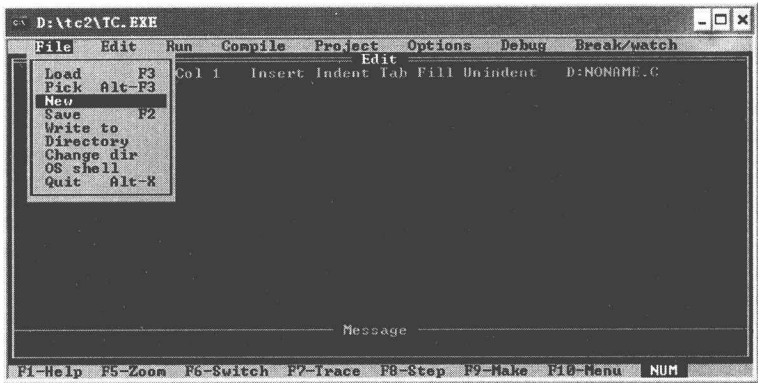


图 1.5 新建文件

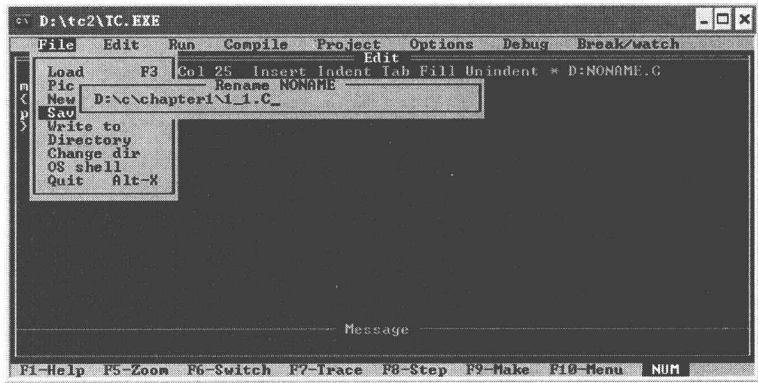


图 1.6 设置保存路径

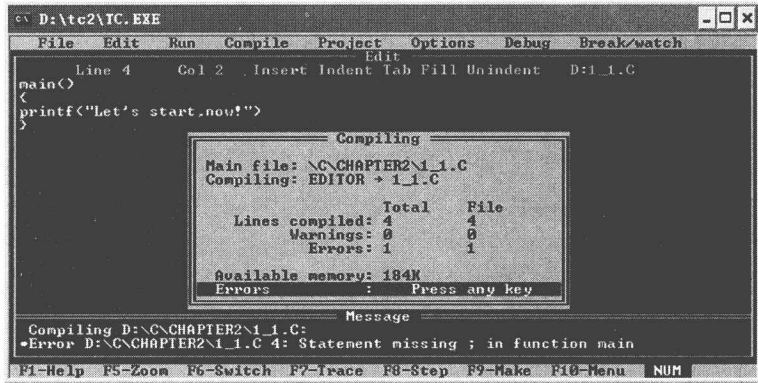


图 1.7 编译结果

按回车键可看到信息窗口高亮显示错误提示信息,同时在编辑区域高亮显示错误可能出现的位置,如图 1.8 所示。

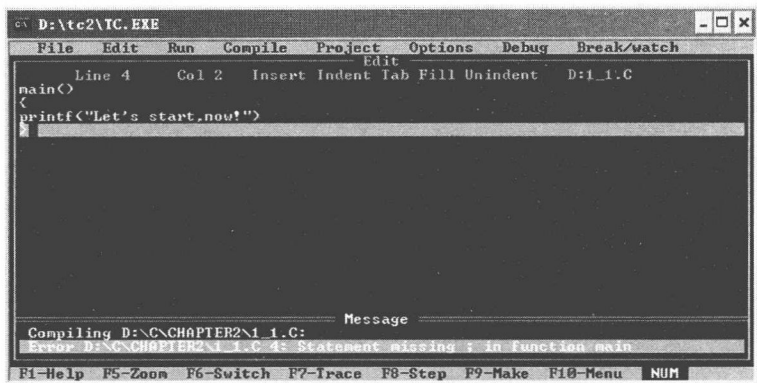


图 1.8 编译错误提示

阅读错误提示“Statement missing ;in function main”，观察编辑区错误提示行，可以看出缺少了语句结束标志“;”。按回车键即可对程序进行修改，将光标移到 printf 语句的末尾添加“;”，再次编译；若有错误请继续修改，直到显示错误数为 0。

三、链接

编译无误后，便可以开始链接。链接的方法是：选择菜单 Compile 中的 Link Exe file 菜单项。链接后也会有结果提示，通常是没有错误的。

四、运行

链接的下一步运行，通过 Run 菜单下的 Run 菜单项或组合键“Ctrl+F9”完成。也可以省略编译和链接直接使用 Run 菜单项或 Ctrl+F9 组合键，系统会自动依次进行编译、链接、运行。

程序正常运行后并没有任何提示，需要通过 Run 菜单下的 User screen 菜单项或组合键 Alt+F5 查看结果，按回车键可回到编辑界面。案例 1.1 的运行结果如图 1.9 所示。

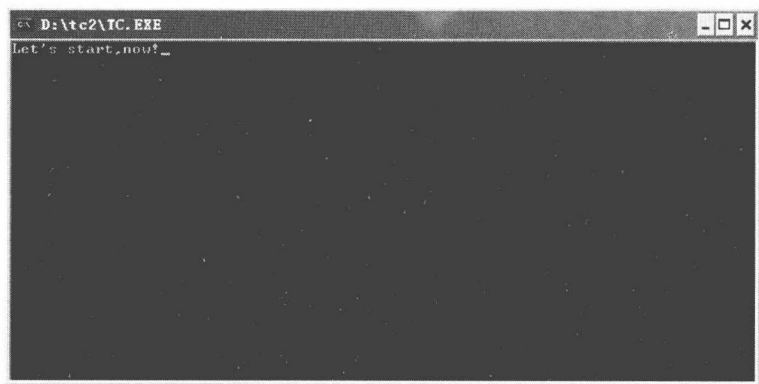


图 1.9 案例 1.1 运行结果

1.4.2 VC++ 6.0

Visual C++(简称 VC)是 Microsoft 公司开发的一个功能强大的可视化集成编程环境，其主界面如图 1.10 所示。本书后续章节所有案例程序均以 VC 6.0 为编程工具。

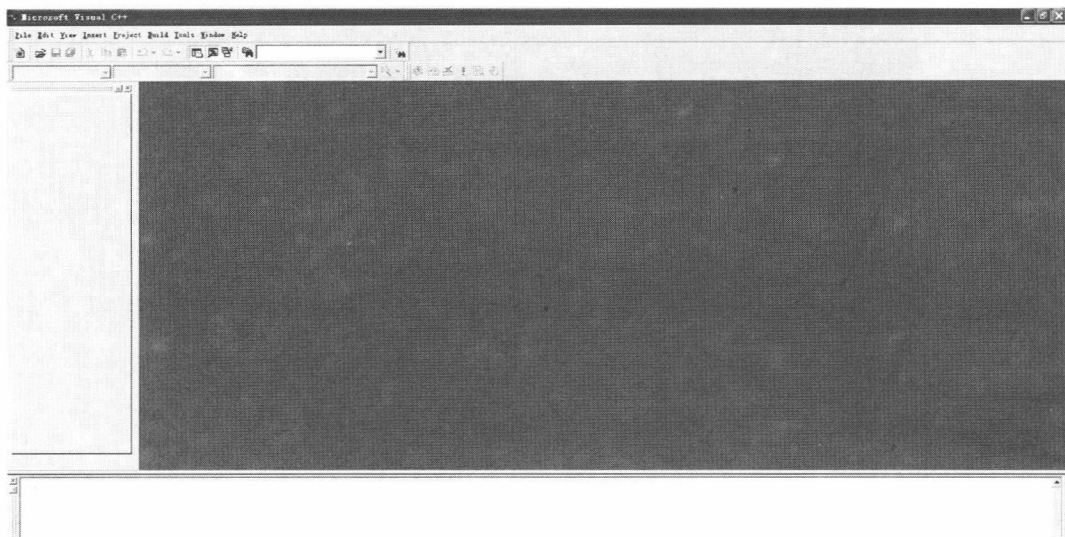


图 1.10 VC 6.0 主界面

以案例 1.2 的执行为例,在 VC 6.0 中运行程序的主要步骤如下:

一、编辑

1. 新建工程

选择 File 菜单下的 New 菜单项,弹出 New(新建)对话框;在 Projects 选项卡中选择 Win32 Console Application 项,并设定 Project name(工程名称)及 Location(存储位置),单击 OK 按钮,如图 1.11 所示。

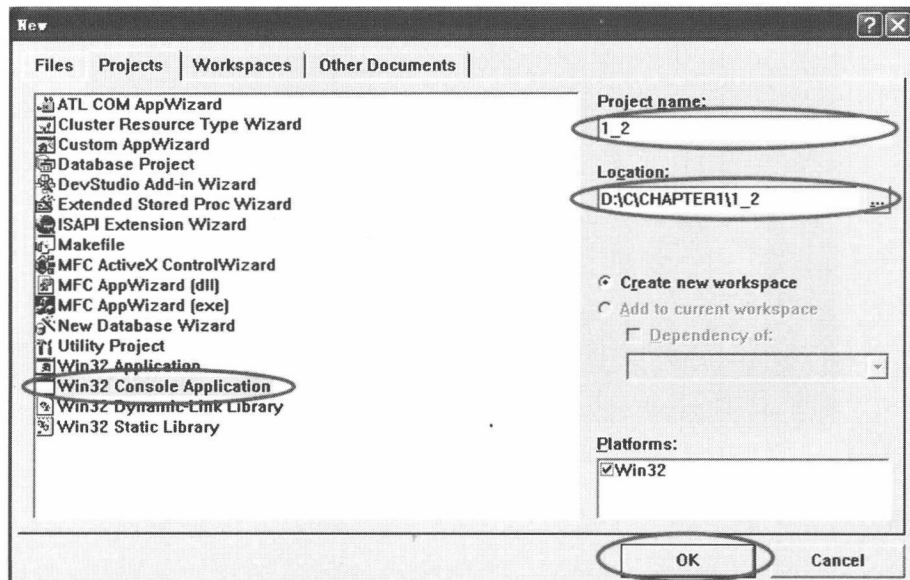


图 1.11 新建工程

在弹出的对话框中选择 An empty project 单选按钮,单击 Finish 按钮,如图 1.12 所示。

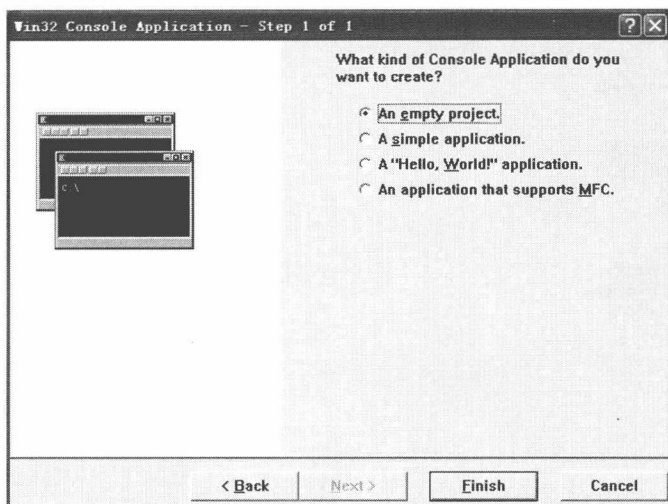


图 1.12 选择工程类型

在弹出的 New Project Information(新工程信息)窗口中单击 OK 按钮,完成工程创建。编程环境变为图 1.13 所示,标题栏显示工程名,并在编辑区域左侧出现了工程信息窗口。

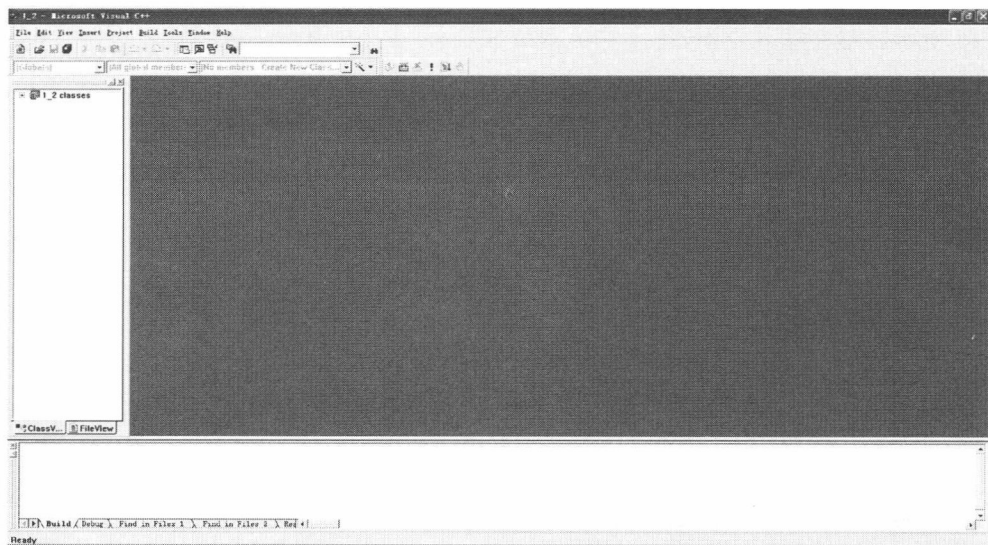


图 1.13 工程创建完成

2. 新建文件

在完成工程创建的主窗口中(如图 1.13 所示),选择 File 菜单下 New 菜单项,弹出 New 对话框。在选择 Files 选项卡后选择 C++ Source File 项,如图 1.14 所示,并在右侧设定文件名,