



普通高等教育“十三五”规划教材

C语言 程序设计 实训教程

C YUYAN CHENGXU SHEJI SHIXUN JIAOCHENG

主 编 陈 鑫
副主编 陕粉丽
参 编 路 璐 李慧玲 马 强



北京邮电大学出版社
www.buptpress.com

普通高等教育“十三五”规划教材

C 语言程序设计实训教程

主 编 陈 鑫

副主编 陕粉丽

参 编 路 璐 李慧玲 马 强



北京邮电大学出版社
www. buptpress. com

内 容 简 介

本书以 Visual Studio 6.0 作为 C 语言程序的集成开发环境,以循序渐进、深入浅出的写作思想,系统地介绍了 C 语言的基本知识和程序设计方法。

全书共分为 11 章,内容包括:C 语言基础知识,C 语言程序设计概述与数据类型,顺序、选择和循环三种基本结构的程序设计方法,数组、函数和指针等 C 语言程序设计重点和难点内容,结构体和共用体两种复杂数据结构程序设计,文件的概念和文件的基本操作方面的知识,最后以学生成绩管理系统为例,介绍 C 语言程序设计的基本编程方法与技巧。

本书可作为高等院校计算机专业和非计算机专业学生学习 C 语言程序设计的教材(并且有利于读者进一步学习 C++),也可作为参加计算机等级考试的读者的学习与参考书。

图书在版编目(CIP)数据

C 语言程序设计实训教程 / 陈鑫主编. -- 北京:北京邮电大学出版社, 2018.2

ISBN 978-7-5635-5341-9

I. ①C… II. ①陈… III. ①C 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 306220 号

书 名: C 语言程序设计实训教程

著作责任者: 陈 鑫 主编

责任编辑: 满志文

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: 保定市中华美凯印刷有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 12.25

字 数: 302 千字

版 次: 2018 年 2 月第 1 版 2018 年 2 月第 1 次印刷

ISBN 978-7-5635-5341-9

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 语言程序设计是目前高等院校中最为基本,也是最为核心的课程。通过该课程的教学,一是让学生掌握一种高级程序设计语言,二是使学生了解程序设计的思想和方法,培养程序设计的实践能力。要学会利用 C 语言去解决实际问题,单凭编写一些简单的小程序训练是无法解决的。特别是对于初次接触 C 语言的学生或者个人而言,按照 C 语言的知识体系去学习和实践,也许能够对 C 语言的知识有比较全面的了解,但是学完后如果要马上去开发一些小的系统项目,还是会感到力不从心。本书力图按照知识构建的思路来引导和训练学生学习使用 C 语言,切实提高 C 语言程序设计的学习效率和应用水平。只有学好了 C 语言,才有可能进一步学习 C++、数据结构等高级软硬件课程。本书以讲解基本知识,提高基本技能为宗旨,将程序设计基本技能与技巧组织在教材中。

本书第 1 章 C 语言概述,介绍了 C 语言的基本概念以及运行 C 语言程序的上机步骤。第 2 章数据类型与运算符,介绍了 C 语言的数据类型、运算符和表达式。第 3 章顺序结构程序设计,第 4 章选择结构程序设计,第 5 章循环结构程序设计,这三章分别详细介绍了程序设计的三种基本结构、语法及应用。第 6 章函数,介绍了函数的定义、调用、变量的作用域及存储类别。第 7 章数组,介绍了一维数组、二维数组和字符数组的定义、初始化及应用。第 8 章指针,介绍了指针的定义和运算。第 9 章用户自定义数据类型,介绍结构体和共用体的定义及应用。第 10 章文件,介绍文件的基本操作。第 11 章综合项目,以学生成绩管理系统为例,介绍 C 语言程序设计的基本编程方法与技巧。

各章分工如下:第 1、2 章由陕粉丽编写,第 3、4、5 章由陈鑫编写,第 6、7 章由路璐编写,第 8、9 章由李慧玲编写,第 10、11 章由马强编写。全书由陈鑫担任主编,并负责统稿。由于编者水平有限,书中有不当之处恳请读者批评指正。

编 者

目 录

第 1 章 C 语言概述	1
1.1 程序设计语言的发展	1
1.2 C 语言的发展及其特点	2
1.2.1 C 语言出现的历史背景	2
1.2.2 C 语言的特点	2
1.3 开发环境	3
1.3.1 开发工具介绍	3
1.3.2 Visual Studio 安装	3
1.3.3 Visual Studio 主界面	5
1.4 案例分析:几个简单的 C 语言程序	6
1.4.1 C 语言程序举例	6
1.4.2 C 语言程序的结构	8
1.5 C 语言程序的运行步骤	9
1.6 C 程序设计的任务	9
习题	10
第 2 章 数据类型与运算符	11
2.1 数据存储的原理	11
2.1.1 进制	11
2.1.2 进制转换	13
2.1.3 原码、反码、补码	15
2.2 常量和变量	16
2.2.1 常量	16
2.2.2 变量	18
2.3 关键字和标识符	19
2.3.1 关键字	19
2.3.2 标识符	20
2.4 运算符和表达式	21
2.4.1 算术运算符与表达式	21
2.4.2 运算符的优先级与结合性	22
2.4.3 赋值运算符与表达式	22

2.4.4 逗号运算符与表达式	23
2.5 不同类型数据间的转换	24
习题	25
第3章 顺序结构程序设计	26
3.1 数据的输入和输出	26
3.1.1 格式化输出函数 printf	26
3.1.2 字符数据输出函数 putchar	32
3.1.3 格式化输入函数 scanf	33
3.1.4 字符数据输入函数 getchar	35
3.2 C语言的基本语句	36
3.3 顺序结构程序应用举例	37
习题	38
第4章 选择结构程序设计	39
4.1 关系运算符与表达式	39
4.1.1 关系运算符及其优先次序	39
4.1.2 关系表达式	40
4.2 逻辑运算符与表达式	40
4.2.1 逻辑运算符及其优先次序	40
4.2.2 逻辑表达式	41
4.3 条件运算符与表达式	43
4.4 if 条件语句	44
4.5 选择结构的嵌套	49
4.6 switch 多分支选择语句	51
4.7 选择结构程序应用举例	53
习题	55
第5章 循环结构程序设计	56
5.1 while 循环语句	56
5.2 do...while 循环语句	58
5.3 for 循环语句	61
5.4 几种循环的比较	64
5.5 循环结构的嵌套	65
5.6 跳转语句	65
5.6.1 continue 语句	65
5.6.2 break 语句	67
5.7 循环结构程序应用举例	67
习题	73

第 6 章 函数	74
6.1 库函数	74
6.2 函数的定义和返回值	75
6.2.1 函数定义	75
6.2.2 函数的返回值	76
6.3 函数的调用	77
6.3.1 函数的调用方式	77
6.3.2 函数调用时的语法规则	78
6.4 函数的声明	80
6.4.1 函数声明的形式	81
6.4.2 函数声明的位置	81
6.5 函数调用时的数据传递	82
习题	83
第 7 章 数组	85
7.1 一维数组的定义和引用	85
7.1.1 一维数组的定义	85
7.1.2 一维数组元素的引用	86
7.1.3 一维数组的初始化	87
7.2 二维数组的定义和引用	89
7.2.1 二维数组的定义	89
7.2.2 二维数组元素的引用	90
7.2.3 二维数组的初始化	90
7.3 字符数组	92
7.3.1 字符数组的定义	92
7.3.2 字符数组的初始化	92
7.3.3 有关字符串处理的函数	93
习题	96
第 8 章 指针	97
8.1 指针的基本概念	97
8.2 变量与指针	97
8.2.1 指针变量的定义	98
8.2.2 指针变量的引用	99
8.2.3 指针变量作为函数参数	101
8.3 一维数组与指针	103
8.3.1 指向数组元素的指针变量	103
8.3.2 指向数组的指针的相关运算	104

8.3.3 通过指针引用数组元素	105
8.3.4 数组作函数的参数	108
8.4 二维数组与指针	113
8.4.1 二维数组的地址	113
8.4.2 指向二维数组元素的指针	114
8.4.3 行指针变量	115
8.5 字符串与指针	116
8.5.1 字符串的表示与引用	116
8.5.2 字符串指针作函数参数	118
8.6 返回指针值的函数	118
8.7 指针数组	119
8.7.1 指针数组概述	119
8.7.2 指向指针的指针	121
8.8 函数的指针和指向函数的指针变量	122
第 9 章 用户自定义数据类型	126
9.1 结构体类型	126
9.1.1 结构体类型的定义	126
9.1.2 结构体变量	128
9.1.3 结构体数组	131
9.1.4 结构体指针	134
9.2 共用体数据类型	139
9.2.1 共用体类型的定义	139
9.2.2 共用体变量的引用	140
9.2.3 共用体的应用	141
9.3 枚举数据类型	143
9.4 自定义类型	144
习题	146
第 10 章 文件	148
10.1 文件概述	148
10.1.1 文件的分类	148
10.1.2 文件指针	149
10.2 文件的打开与关闭	149
10.2.1 文件的打开(fopen 函数)	150
10.2.2 文件关闭函数(fclose 函数)	151
10.3 文件的读写	151
10.3.1 字符读写函数 fgetc 和 fputc	152
10.3.2 字符串读写函数 fgets 和 fputs	155

10.3.3 数据块读写函数 fread 和 fwrite	157
10.4 文件的随机读写	159
10.4.1 文件定位	160
10.4.2 文件的随机读写	160
10.5 文件检测函数	161
10.5.1 文件结束检测函数 feof 函数	161
10.5.2 读写文件出错检测函数	162
10.5.3 文件出错标志和文件结束标志置 0 函数	162
10.6 本章小结	162
习题	162
第 11 章 综合项目——学生成绩管理系统	164
11.1 设计要求	164
11.2 总体设计	164
11.3 详细设计	165
11.4 系统运行界面	174
附录 A 常用字符与 ASCII 代码对照表	177
附录 B C 语言中的关键字	178
附录 C 运算符的优先级和结合性	179
附录 D 常用 ANSI C 标准库函数	180

第 1 章 C 语言概述

1.1 程序设计语言的发展

自 20 世纪 60 年代以来,世界上公布的程序设计语言已有上千种之多,但是只有很小一部分得到了广泛的应用。从发展历程来看,程序设计语言可以分为四代。

(1) 第一代机器语言

机器语言是由二进制 0、1 代码指令构成,不同的 CPU 具有不同的指令系统。机器语言程序难编写、难修改、难维护,需要用户直接对存储空间进行分配,编程效率极低。这种语言已经被渐渐淘汰了。

(2) 第二代汇编语言

汇编语言指令是机器指令的符号化,与机器指令存在着直接的对应关系,所以汇编语言同样存在着难学难用、容易出错、维护困难等缺点。但是汇编语言也有自己的优点:可直接访问系统接口,汇编程序翻译成的机器语言程序效率高。从软件工程角度来看,只有在高级语言不能满足设计要求,或不具备支持某种特定功能的技术性能(如特殊的输入/输出)时,汇编语言才被使用。

(3) 第三代高级语言

高级语言是面向用户的、基本上独立于计算机种类和结构的语言。其最大的优点是:形式上接近于算术语言和自然语言,概念上接近于人们通常使用的概念。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令。因此,高级语言易学易用,通用性强,应用广泛。

(4) 第四代非过程化语言

4GL 是非过程化语言,编码时只需说明“做什么”,不需描述算法细节。

数据库查询和应用程序生成器是 4GL 的两个典型应用。用户可以用数据库查询语言(SQL)对数据库中的信息进行复杂的操作。用户只需将要查找的内容在什么地方、根据什么条件进行查找等信息告诉 SQL,SQL 将自动完成查找过程。应用程序生成器则是根据用户的需求“自动生成”满足需求的高级语言程序。真正的第四代程序设计语言应该说还没有出现。所谓的第四代语言大多是指基于某种语言环境上具有 4GL 特征的软件工具产品,如 System Z、PowerBuilder、FOCUS 等。第四代程序设计语言是面向应用,为最终用户设计的一类程序设计语言。它具有缩短应用开发过程、降低维护代价、最大限度地减少调试过程中出现的问题以及对用户友好等优点。

1.2 C 语言的发展及其特点

C 语言是国际上广泛流行的计算机高级语言。它是一种用途广泛、功能强大、使用灵活的过程性编程语言,既可用于编写应用软件,又能用于编写系统软件。因此 C 语言问世以后得到迅速推广。自 20 世纪 90 年代初,C 语言在我国开始推广以来,学习和使用 C 语言的人越来越多,成了学习和使用人数最多的一种计算机语言,绝大多数理工科大学都开设了 C 语言程序设计课程。掌握 C 语言成为计算机开发人员的一项基本功。

C 语言发展如此迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。许多著名的系统软件,如 Windows(C,C++)、Linux(C)、UNIX(C)都是由 C 语言编写的。

1.2.1 C 语言出现的历史背景

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL(Basic Combined Programming Language)发展演变而来。在 1970 年,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言,最后导致了 C 语言的问世。

随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准。

1.2.2 C 语言的特点

C 语言有以下一些主要特点。

(1) 简洁紧凑、灵活方便

C 语言一共只有 32 个关键字,9 种控制语句,程序书写自由,主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元。

(2) 运算符丰富

C 语言的运算符包含的范围很广泛,共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 语言的运算类型极其丰富。表达式类型多样化,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。

(3) 数据类型丰富

C 语言的数据类型有: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。它能用来实现各种复杂的数据类型的运算,并引入了指针概念,使程序效率更高。另外 C 语言具有强大的图形功能,支持多种显示器和驱动器。且计算功能、逻辑判断功能强大。

(4) C是结构式语言

结构式语言的显著特点是代码及数据的分隔化,即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、维护以及调试。C语言是以函数形式提供给用户的,这些函数可方便的调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

(5) C语言的语法限制不太严格,程序设计自由大

一般的高级语言语法检查比较严,能够检查出几乎所有的语法错误。而C语言允许程序编写者有较大的自由度。

(6) C语言允许直接访问物理地址,可以直接对硬件进行操作

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,能够像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元,可以用来写系统软件。

(7) C语言程序生成代码质量高,程序执行效率高

一般只比汇编程序生成的目标代码效率低10%~20%。

(8) C语言适用范围大,可移植性好

C语言有一个突出的优点就是适合于多种操作系统,如DOS、UNIX,也适用于多种机型。对操作系统和系统使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,许多大型应用软件都是用C语言编写的。

C语言绘图能力强,具有可移植性,并具备很强的数据处理能力,因此适于编写系统软件,三维、二维图形和动画,它是数值计算的高级语言。

1.3 开发环境

1.3.1 开发工具介绍

Visual C++是Microsoft公司的Visual Studio开发工具箱中的一个C++程序开发包。Visual Studio提供了一整套开发Internet和Windows应用程序的工具,包括Visual C++、Visual Basic、Visual FoxPro、Visual InterDev、Visual J++以及其他辅助工具,如代码管理工具Visual SourceSafe和联机帮助系统MSDN。Visual C++包中除包括C++编译器外,还包括所有的库、例子和为创建Windows应用程序所需要的文档。

从最早期的1.0版本发展6.0版本,到最新的.NET版本,Visual C++已经有了很大的变化,在界面、功能、库支持方面都有许多的增强。6.0版本在编译器、MFC类库、编辑器以及联机帮助系统等方面都比以前的版本做了较大改进。

Visual C++一般分为三个版本:学习版、专业版和企业版,不同的版本适合于不同类型的开发。本书中安装的实验环境是基于企业版的。

1.3.2 Visual Studio 安装

下载一个Visual Studio 6.0的安装包。下载完成之后,解压,找到安装文件,单击进入安装。

第一步,弹出安全警告之后,单击“运行”按钮,继续安装。

第二步,进入安装向导,单击“下一步”按钮,继续安装,如图 1-1 所示。

第三步,阅读协议,选择“接受协议”,单击“下一步”按钮,继续安装,如图 1-2 所示。

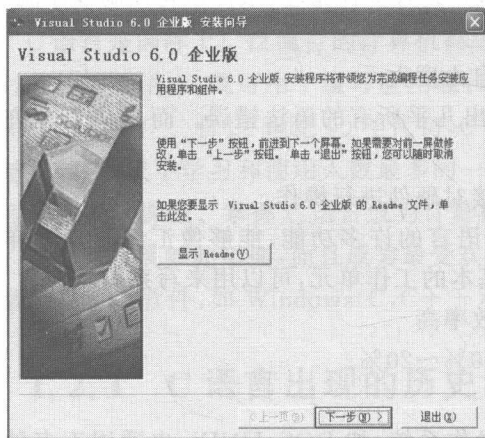


图 1-1



图 1-2

第四步,输入产品号和用户 ID,单击“下一步”按钮,继续安装,如图 1-3 所示。

第五步,选择企业版安装选项,通常选择“自定义”,单击“下一步”,继续安装,如图 1-4 所示。

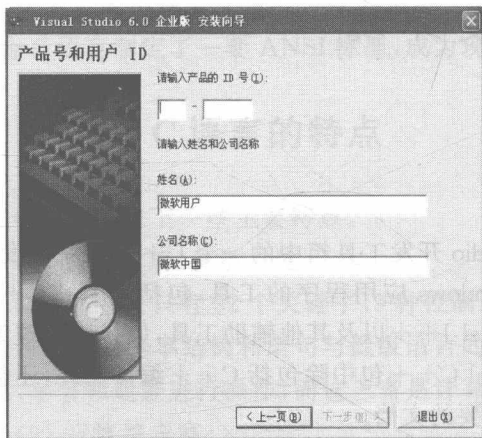


图 1-3

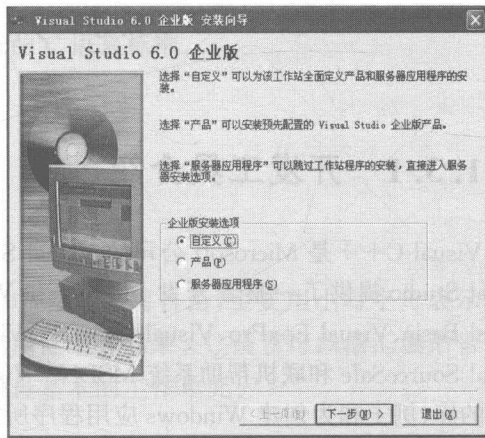


图 1-4

第六步,进入准备安装状态,可根据需要进行相应选项的选择,单击“继续”按钮,进入安装,如图 1-5 所示。

第七步,安装环境变量,为操作方便,选择注册环境变量,单击“确定”按钮,如图 1-6 所示。

第八步,进入安装过程,安装结束后。确认创建桌面快捷方式。Visual Studio 6.0 成功安装完成,双击桌面上 Visual Studio 6.0 图标,就能顺利进入 Visual Studio 6.0 集成环境,如图 1-7 所示。

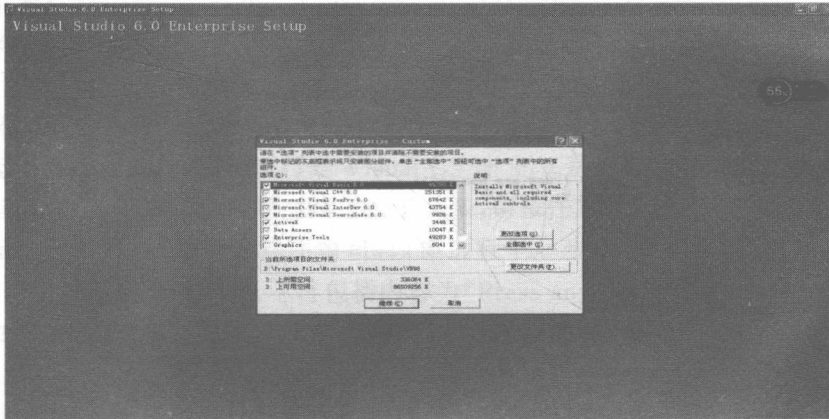


图 1-5

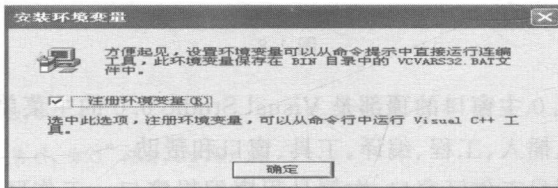


图 1-6

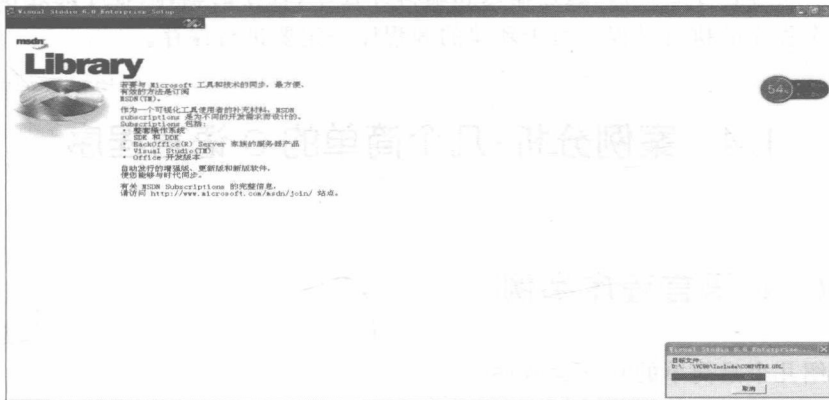


图 1-7

1.3.3 Visual Studio 主界面

为了能使用 Visual Studio 6.0 集成环境, 必须事先在所用的计算机上安装 Visual Studio 6.0 系统。在安装后最好在桌面上设立 Visual Studio 6.0 的快捷方式图标, 以方便使用。

双击桌面上 Visual Studio 6.0 图标, 就能进入 Visual Studio 6.0 集成环境, 屏幕上出现 Visual Studio 6.0 的主窗口, 如图 1-8 所示。为了方便读者, 本书介绍的是 Visual Studio 6.0 中文版。

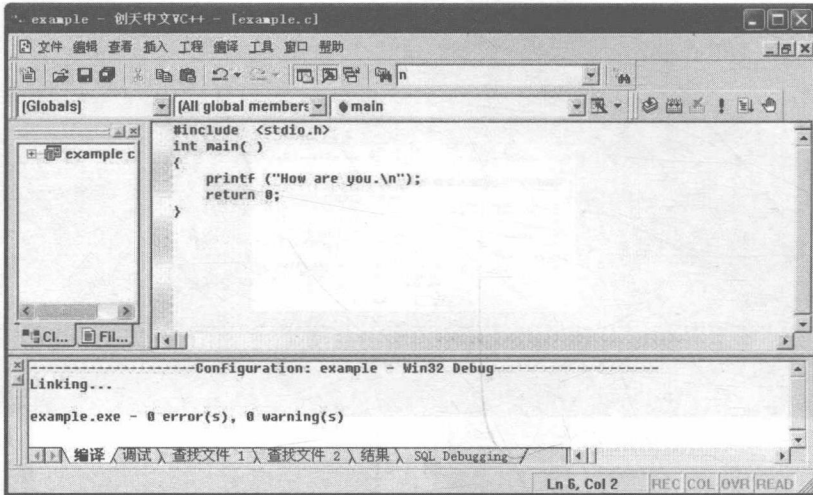


图 1-8

在 Visual Studio 6.0 主窗口的顶部是 Visual Studio 6.0 的主菜单栏。其中包含 9 个菜单项：文件、编辑、查看、插入、工程、编译、工具、窗口和帮助。

主窗口的左侧是项目工作区窗口，右侧是程序编辑窗口。工作区窗口用来显示所设定的工作区的信息，程序编辑窗口用来输入和编辑源程序。

在新建一个源程序后，可以通过输入和编辑源程序、程序的编译、程序的连接和程序的执行完成一个程序的执行过程。对于新建的源程序一定要进行保存。

1.4 案例分析：几个简单的 C 语言程序

1.4.1 C 语言程序举例

下面介绍几个最简单的 C 语言程序。

例 1.1 要求在屏幕上输出以下一行信息。

How are you.

解题思路：在主函数中用 printf 函数原样输出以上文字。

编写程序：

```
#include <stdio.h>           //这是编译预处理指令
int main()                   //定义主函数
{                             //函数开始的标志
    printf("How are you.\n"); //输出所指定的一行信息
    return 0;                 //函数执行完毕时返回函数值 0
}                             //函数结束的标志
```

运行结果:

How are you.

Press any key to continue

以上运行结果是在 Visual C++ 6.0 环境下运行程序时屏幕上得到的显示。其中第 1 行是程序运行后输出的结果,第 2 行是 Visual C++ 6.0 系统在输出完运行结果后自动输出的一行信息,告诉用户:“如果想继续进行下一步,请按任意键”。当用户按任意键后,屏幕上不再显示运行结果,而是返回程序窗口,以便进行下一步工作(如修改程序)。为节省篇幅,本书在以后显示运行结果时,不再包括内容为“Press any key to continue”的行。

例 1.2 本程序的功能是对从键盘输入的长方体的长、宽、高三个整型量求其体积的值。

解题思路:用一个函数来计算长方体的体积。在主函数中调用此函数并输出结果。

编写程序:

```
#include <stdio.h>
int main()
{
    float x,y,z,v;          /* 定义整型变量 */
    scanf("%d,%d,%d",&x,&y,&z);
                                /* 调用标准函数,从键盘输入 x,y,z 的值 */
    v = volume(x,y,z);      /* 调用 volume 函数,计算体积 */
    printf("v = %d\n",v);
}
int volume(int a, int b, int c) /* 定义 volume 函数 */
{
    int p;                  /* 定义函数内部使用的变量 p */
    p = a * b * c;         /* 计算体积 p 的值 */
    return(p);            /* 将 p 值返回调用处 */
}
```

运行结果:

5,8,6

v = 240

程序分析:

在本例中,main 函数在调用 volume 函数时,将实际参数 x、y、z 的值分别传送给 volume 函数中的形式参数 a、b、c。经过执行 volume 函数得到一个结果(即 volume 函数中变量 p 的值)并把这个值赋给变量 v。

例 1.3 求两个整数中的较大者。

解题思路:用一个函数来实现求两个整数中的较大者。在主函数中调用此函数并输出结果。

编写程序:

```
#include <stdio.h>
```

```

//主函数
int main()                //定义主函数
{                          //主函数体开始
    int max(int x,int y);  //对被调用函数 max 的声明
    int a,b,c;            //定义变量 a,b,c
    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\n",c); //输出 c 的值
    return 0;             //返回函数值为 0
}                          //主函数体结束
//求两个整数中的较大者的 max 函数
int max(int x,int y)
{                          //定义 max 函数,函数值为整型,形式参数 x 和 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;         //若 x>y 成立,将 x 的值赋给变量 z
    else z = y;           //否则(即 x>y 不成立),将 y 的值赋给变量 z
    return(z);            //将 z 的值作为 max 函数值,返回到调用 max 函数的位置
}

```

运行结果:

89,56

max = 89

程序分析:

上面例中程序的功能是由用户输入两个整数,程序执行后输出其中较大的数。本程序由两个函数组成,主函数和 max 函数。函数之间是并列关系。可从主函数中调用其他函数。max 函数的功能是比较两个数,然后把较大的数返回给主函数。max 函数是一个用户自定义函数。因此在主函数中要给出说明(程序第三行)。可见,在程序的说明部分中,不仅可以有变量说明,还可以有函数说明。在程序的每行后用 /* 和 */ 括起来的,或用 // 标记的内容为注释部分,程序不执行注释部分。

上例中程序的执行过程是,首先在屏幕上显示提示串,请用户输入两个数,回车后由 scanf 函数语句接收这两个数送入变量 a,b 中,然后调用 max 函数,并把 a,b 的值传送给 max 函数的参数 x,y。在 max 函数中比较 a,b 的大小,使 max 函数中的变量 z 得到一个值(即 x 和 y 中大者的值),通过 return(z) 把 z 的值作为 max 函数值返回给主函数的变量 c,最后在屏幕上输出 c 的值。

1.4.2 C 语言程序的结构

通过以上几个程序例子,可以看到一个 C 语言程序的结构有以下特点:

(1) 一个程序由一个或多个源程序文件组成。一个规模较小的程序,往往只包括一个