



工业和信息化
人才培养规划教材

Industry And Information
Technology Training
Planning Materials

高职高专计算机系列

C语言程序设计 案例教程

C Language Program Design
Case Tutorial

朱作付 龙浩 © 主编

王勇 王方杰 左丹霞 © 副主编

- + 以学生成绩**管理系统**的项目设计贯穿整个 **C 语言知识** 的学习过程
- + 以知识**够用为原则**，学习重点放到 **C 语言程序项目设计** 上
- + 本着**先练习后总结**的原则，将结构化**程序设计和算法**的学习后移
- + 课后习题兼顾了学生对**计算机等级考试**的内容需求



工业和信息化
人才培养规划教材

Industry And Information
Technology Training
Planning Materials

高职高专计算机系列

C语言程序设计 案例教程

C Language Program Design
Case Tutorial

朱作付 龙浩 © 主编

王勇 王方杰 左丹霞 © 副主编

人民邮电出版社

北京

C语言程序设计案例教程 / 朱作付, 龙浩主编. —
北京: 人民邮电出版社, 2014. 11
工业和信息化人才培养规划教材. 高职高专计算机系
列
ISBN 978-7-115-36325-1

I. ①C… II. ①朱… ②龙… 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178531号

内 容 提 要

本书是一本针对C语言学习的基础性教材,通过一个完整的项目设计将C语言的基本知识衔接起来,实现即学即用。主要内容包括:C语言简介、C语言语法基础、顺序结构程序设计、分支结构程序设计、循环结构程序设计、数组、函数、结构体、算法与项目设计、指针、文件和位运算等。

本书针对高职高专学生的特点,做到理论知识适用、够用,专业技能实用、管用。本书遵循先练习后总结的原则,将结构化程序设计和算法的学习后移,让学生在具有一定程序设计能力的基础上再进行算法的学习,有利于克服学生学习C语言的畏难情绪,提高学生的学习效果。

本书课后习题形式多样,在注重编程练习的基础上兼顾了学生对计算机等级考试的内容需求,有利于学习者的自我检查和学习提高。本书选编了关于计算机语言和程序设计方面的一些短文,有利于提高学生计算机知识素养。

本书适合作为高职高专院校计算机及相关专业的教材,也可作为程序开发人员的参考用书。

-
- ◆ 主 编 朱作付 龙 浩
 - 副 主 编 王 勇 王方杰 左丹霞
 - 责任编辑 王 平
 - 责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京中新伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
 - 印张: 16.25 2014年11月第1版
 - 字数: 413千字 2014年11月北京第1次印刷
-

定价: 37.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316
反盗版热线: (010)81055315

前 言 PREFACE

近年来,教育部先后颁发了《关于提高高等职业教育教学质量的若干意见》等多个文件,大力推进高等职业教育基于项目化、案例化的课程教学改革工作。但是,如何将知识的学习和项目化、案例化教学方法有机地结合在一起,不同的课程有各自不同的特点。我们既不能为了推进项目化、案例化而简单地将原有的知识体系进行分割和包装,又不能固守传统不顾高职院校学生的学习特点进行过多的理论阐述而忽略动手能力的培养。因此,本书将结合编者实际教学经验,本着由易到难的原则,逐步引导学生通过项目设计来学习 C 语言的知识,并通过项目设计逐步提高编程能力和技巧。

C 语言作为计算机类专业学生学习程序设计的入门课程,对于广大学生学习后续课程具有较大的影响。因此,我们从学生学习的角度,在知识学习的基础上逐步开展项目设计,让大家克服学习过程的畏难情绪,并通过项目设计逐步提高学生学习 C 语言的兴趣和成就感。本书通过一个完整的项目设计将 C 语言的基本知识衔接起来,实现即学即用。

本书的主要特点如下:

1. 以同学们比较熟悉的学生成绩管理系统的项目设计贯穿整个 C 语言知识的学习过程。本着由易到难的原则,逐步引导学习者通过自己编写程序来加深对 C 语言知识的运用能力,并通过项目设计逐步提高学生学习的兴趣。
2. 对 C 语言的知识介绍以够用为原则,删除了一些对于高职院校学生比较难以理解的内容,而将学习重点放到 C 语言程序项目设计上,注重提高学生的动手能力,让学生敢于写程序,克服对编程的畏难心理。
3. 本着先练习后总结的原则,将结构化程序设计和算法的学习后移,让学生在具有一定程序设计能力的基础上再进行算法的学习,有利于帮助学生克服学习 C 语言的畏难情绪,提高学生的学习效果。
4. 课后习题形式多样,在注重编程练习的基础上兼顾了学生对计算机等级考试的内容需求,有利于学生课后的自我检查和学习提高。
5. 注重对学生计算机知识素养的培养,选编了关于计算机语言和程序设计方面的短文,有利于学生拓展计算机程序设计方面的知识,提高对学习计算机语言和编程的兴趣。

本书由朱作付、龙浩老师担任主编,王勇、王方杰、左丹霞老师任副主编,具体分工如下:第 1 章、第 2 章由王勇老师编写,第 3 章、第 4 章、第 5 章、第 10 章由龙浩老师编写,第 6 章、第 7 章、第 8 章、第 9 章、第 11 章由朱作付老师编写,第 12 章由王方杰老师编写,附录 1、2、3、4 由左丹霞老师编写,徐超、王侠、葛红美、臧博、张雪松等老师参与了本书的习题编校工作,全书由朱作付老师总揽成稿。中国矿业大学计算机学院王潜平教授和徐州华社信息技术有限公司的叶志江经理担任本书的主审。在本书的编写过程中得到了徐州华社信息技术有限公司部分工程师的帮助,在此一并感谢。

由于编者水平所限,书中难免存在错误与不足,敬请读者批评指正。

编 者

2014 年 4 月

目 录 CONTENTS

第 1 章 C 语言简介 1

1.1 C 语言的发展	2	1.5 C 语言的字符集	8
1.2 Visual C++6.0 集成开发环境	2	1.6 C 语言的标识符与关键字	9
1.2.1 Visual C++6.0 简介	2	1.7 C 语言的特点	10
1.2.2 Visual C++6.0 操作步骤	3	小结与提示	10
1.3 简单的 C 程序结构分析	6	知识拓展	11
1.4 书写程序时应遵循的规则	8	习题与项目练习	12

第 2 章 C 语言语法基础 13

2.1 C 语言的数据类型	14	2.5.1 字符常量	20
2.2 常量与变量	14	2.5.2 转义字符	20
2.2.1 常量	14	2.5.3 字符变量	21
2.2.2 变量	15	2.6 各类数值型数据之间的混合运算	22
2.2.3 变量初始化	16	2.7 算术运算符和算术表达式	24
2.3 整型数据	16	2.7.1 C 运算符简介	24
2.3.1 整型常量	16	2.7.2 算术运算符和算术表达式	25
2.3.2 整型变量	18	2.8 赋值运算符和赋值表达式	26
2.4 实型数据	18	2.9 逗号运算符和逗号表达式	28
2.4.1 实型常量	18	小结与提示	28
2.4.2 实型变量	19	知识拓展	29
2.5 字符型数据	20	习题与项目练习	29

第 3 章 顺序结构程序设计 33

3.1 C 语句概述	34	3.5.1 printf 函数 (格式输出函数)	37
3.2 赋值语句	34	3.5.2 scanf 函数 (格式输入函数)	40
3.3 C 语言中数据输入输出的实现	36	3.6 顺序结构程序设计举例	41
3.4 字符数据的输入输出	36	小结与提示	42
3.4.1 putchar 函数 (字符输出函数)	36	知识拓展	43
3.4.2 getchar 函数 (键盘输入函数)	37	习题与项目练习	44
3.5 格式输入与输出	37		

第 4 章 分支结构程序设计 50

4.1 分支结构程序概述	51	4.2.1 关系运算符及其优先次序	51
4.2 关系运算符和表达式	51	4.2.2 关系表达式	52

4.3 逻辑运算符和表达式	52	4.4.3 条件运算符和条件表达式	58
4.3.1 逻辑运算符及其优先次序	52	4.5 switch 语句	60
4.3.2 逻辑运算的值	53	4.6 选择结构程序设计举例	62
4.3.3 逻辑表达式	53	小结与提示	65
4.4 if 语句	54	知识拓展	65
4.4.1 if 语句的 3 种形式	54	习题与项目练习	66
4.4.2 if 语句的嵌套	58		

第 5 章 循环结构程序设计 75

5.1 循环程序设计概述	76	5.8 break 和 continue 语句	82
5.2 goto 语句	76	5.8.1 break 语句	82
5.3 while 语句	76	5.8.2 continue 语句	82
5.4 do while 语句	77	5.9 循环结构程序设计举例	83
5.5 for 语句	78	小结与提示	86
5.6 循环的嵌套	80	知识拓展	86
5.7 几种循环的比较	81	习题与项目练习	87

第 6 章 数组 97

6.1 一维数组的定义和引用	98	6.3.1 字符数组的定义	106
6.1.1 一维数组的定义	98	6.3.2 字符数组的初始化	106
6.1.2 一维数组元素的引用	99	6.3.3 字符数组的引用	106
6.1.3 一维数组的初始化	100	6.3.4 字符串和字符串结束标志	107
6.1.4 一维数组编程练习	101	6.3.5 字符数组的输入输出	107
6.2 二维数组的定义和引用	103	6.3.6 字符串处理函数	109
6.2.1 二维数组的定义	103	6.3.7 数组程序举例	111
6.2.2 二维数组元素的引用	103	小结与提示	113
6.2.3 二维数组的初始化	104	知识拓展	113
6.3 字符数组	106	习题与项目练习	114

第 7 章 函数 118

7.1 函数概述	119	7.5 数组作为函数参数	124
7.2 函数定义的一般形式	120	7.6 局部变量和全局变量	128
7.3 函数的参数和函数的值	122	7.6.1 局部变量	128
7.3.1 形式参数和实际参数	122	7.6.2 全局变量	129
7.3.2 函数的返回值	123	7.7 函数程序设计举例	131
7.4 函数的调用	123	小结与提示	131
7.4.1 函数调用的一般形式	123	知识拓展	132
7.4.2 函数调用的方式	124	习题与项目练习	132

第8章 结构体 136

8.1 结构体的定义	137	8.3.1 结构体变量的赋值	139
8.2 结构体类型变量的说明与表示方法	137	8.3.2 结构体变量的初始化	140
8.2.1 结构体类型变量的说明	137	8.4 结构体数组的定义	140
8.2.2 结构体变量成员的表示方法	139	小结与提示	142
8.3 结构体变量的赋值与初始化	139	知识拓展	142
		习题与项目练习	143

第9章 算法与项目设计 147

9.1 算法概述	148	9.6.1 简易学生成绩管理系统功能描述	154
9.2 简单算法举例	148	9.6.2 简易学生成绩管理系统各模块的程序流程图	155
9.3 算法的特性	150	9.6.3 简易学生成绩管理系统程序设计	157
9.4 怎样表示一个算法	151	小结与提示	161
9.4.1 用自然语言表示算法	151	知识拓展	161
9.4.2 用流程图表示算法	151	习题与项目练习	163
9.4.3 3种基本结构	152		
9.4.4 用N-S流程图表示算法	153		
9.5 结构化程序设计方法概述	154		
9.6 简易学生成绩管理系统的设计	154		

第10章 指针 164

10.1 指针的基本概念	165	10.6 结构指针变量的说明和使用	183
10.2 变量的指针和指向变量的指针变量	165	10.6.1 指向结构变量的指针	183
10.2.1 指针变量的定义	166	10.6.2 指向结构数组的指针	185
10.2.2 指针变量的引用	166	10.6.3 结构指针变量作函数参数	186
10.2.3 指针变量作为函数参数	169	10.7 有关指针的数据类型和指针运算的小结	187
10.3 数组指针和指向数组的指针变量	171	10.7.1 有关指针的数据类型的小结	187
10.3.1 指向数组元素的指针	171	10.7.2 指针运算的小结	187
10.3.2 通过指针引用数组元素	172	10.8 动态存储分配	188
10.3.3 数组名作函数参数	174	10.9 链表的概念	189
10.4 字符串的指针和指向字符串的指针变量	179	10.10 枚举类型	192
10.4.1 字符串的表示形式	179	10.10.1 枚举类型的定义和枚举变量的说明	192
10.4.2 使用字符串指针变量与字符串数组的区别	181	10.10.2 枚举类型变量的赋值和使用	194
10.5 函数指针变量	182	小结与提示	195
		知识拓展	196
		习题与项目练习	196

第 11 章 文件 206

11.1 C 文件概述	207	fprintf	215
11.2 文件指针	207	11.5 文件的定位与随机读写	216
11.3 文件的打开与关闭	208	11.5.1 文件定位	216
11.3.1 文件的打开 (fopen 函数)	208	11.5.2 文件的随机读写	217
11.3.2 文件的关闭 (fclose 函数)	210	11.6 文件检测函数	218
11.4 文件的读写	210	11.7 C 库文件	218
11.4.1 字符读写函数 fgetc 和 fputc	210	小结与提示	219
11.4.2 字符串读写函数 fgets 和 fputs	212	知识拓展	219
11.4.3 数据块读写函数 fread 和 fwrite	214	习题与项目练习	222
11.4.4 格式化读写函数 fscanf 和			

第 12 章 位运算 229

12.1 位运算概述	230	12.1.6 右移运算	232
12.1.1 按位与运算	230	12.2 位域 (位段)	233
12.1.2 按位或运算	230	小结与提示	235
12.1.3 按位异或运算	231	知识拓展	235
12.1.4 求反运算	231	习题与项目练习	236
12.1.5 左移运算	231		

附录 238

附录 1 C 语言关键字	238	附录 3 常用库函数	241
附录 2 运算符和结合性	239	附录 4 ASCII 码对照表	249

参考文献 251

PART 1

第1章
C语言简介

教学目标

通过本章学习,使学生了解C语言的基本知识和编译环境。



教学要求

知识要点	能力要求	关联知识
C语言的发展和基本特点	了解C语言的发展现状及其特点	高级程序设计语言
C语言程序的基本结构	(1) 了解C语言的基本结构 (2) 正确书写C程序	基本的顺序结构及其书写格式
C语言的基本字符集合关键字	掌握C语言的关键字	main、for、if、break等关键字的学习
C语言的集成开发环境	熟悉C语言编译环境	C语言程序的创建、编译、连接、执行



重点难点

- C语言程序的基本结构
- C语言的基本字符集合关键字
- C语言的集成开发环境

C语言程序设计是计算机及相关专业学生最重要的一门专业基础课程之一,通过对C语言的学习,可以培养学生良好的编程思维能力和编程方法,为今后更深入地学习计算机语言打好基础。

1.1 C语言的发展

自从 1946 年世界上第一台真正意义上的计算机诞生以来, 计算机技术的发展已经深刻地改变了人类社会的生活。计算机技术的发展既有工业技术发展带来的计算机硬件的变化, 更包含计算机软件的发展所带来的应用层面的变化, 而计算机软件的发展则得益于计算机语言的发展。在计算机语言的发展过程中, 20 世纪 70 年代初问世的 C 语言是一种重要的、被广泛应用的计算机编程语言。

C 语言的原型是 Algol 60 语言 (也称为 A 语言)。1963 年, 剑桥大学将 Algol 60 语言发展成为 CPL (Combined Programming Language) 语言。1967 年, 剑桥大学的 Martin Richards 对 CPL 语言进行了简化, 于是产生了 BCPL 语言。1970 年, 美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改, 并为它起了一个有趣的名字 “B 语言”。1973 年, 美国贝尔实验室的 D.M.Ritchie 在 B 语言的基础上最终设计出了一种新的语言, 他取了 BCPL 的第 2 个字母作为这种语言的名字, 这就是 C 语言。1977 年, Dennis M.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年, Brian W.Kernighan 和 Dennis M.Ritchie 出版了名著《The C Programming Language》, 从而使 C 语言成为目前世界上最广泛流行的高级程序设计语言。

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。1983 年, 美国国家标准协会 (ANSI) 委任一个委员会 X3J11 对 C 语言进行标准化。经过长期艰苦的过程, 该委员会的工作于 1989 年 12 月 14 日正式被批准为 ANSI X3.159-1989 并于 1990 年春天颁布, 成为现行的 C 语言标准。

目前最流行的 C 语言有以下几种:

- ① Microsoft C 或称 MS C;
- ② Borland Turbo C 或称 Turbo C;
- ③ AT&T C。

这些 C 语言版本不仅实现了 ANSI C 标准, 而且在此基础上各自做了一些扩充, 使之更加方便、完美。早期的 C 语言主要用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识, 到了 20 世纪 80 年代, C 语言开始进入其他操作系统, 并很快在各类大、中、小和微型计算机上得到了广泛的使用, 成为最优秀的程序设计语言之一。

1.2 Visual C++6.0 集成开发环境

1.2.1 Visual C++6.0 简介

目前, C 语言的集成开发环境主要有两种: 一种是基于 DOS 操作系统支持下的 Turbo C 2.0 系统; 一种是在 Windows 环境下的 Visual C++ 系统。由于国家计算机等级考试的环境一般为 Visual C++ 系统, 且在实际程序调试过程中使用 Visual C++ 系统更为方便, 因此本书不再介绍 Turbo C 2.0 的操作方法, 有兴趣的读者可以在网上进行查找和学习。Visual C++ 系统是一个功能强大的可视化软件开发工具。自 1993 年 Microsoft 公司推出 Visual C++1.0 后, 随着其新版本的不断问世, Visual C++ 已成为专业程序员进行软件开发的首选工具。目前比较流行的是 Visual C++6.0 系统。

Visual C++6.0 是目前比较流行的 C 语言程序开发工具,它不仅是一个 C++编译器,而且是一个基于 Windows 操作系统的可视化集成开发环境(Integrated Development Environment, IDE)。Visual C++6.0 由许多组件组成,包括编辑器、调试器以及程序向导 AppWizard、类向导 ClassWizard 等开发工具。这些组件通过一个名为 Developer Studio 的组件集成为和谐的开发环境。关于 Visual C++6.0 的详细信息,请读者查阅相关教材。本书不作过多描述。

1.2.2 Visual C++6.0 操作步骤

1. 启动 Visual C++ 6.0 环境

和一般的应用程序启动方法相似,下面我们简单地介绍一下 Visual C++6.0 的启动过程,大家在以后的学习过程中,通过实际操作很容易掌握。

单击“开始”—“程序”—“Microsoft Visual Studio 6.0”—“Microsoft Visual C++ 6.0”命令,启动 Visual C++ 6.0,初始启动画面如图 1.1 所示,启动后主窗口如图 1.2 所示。

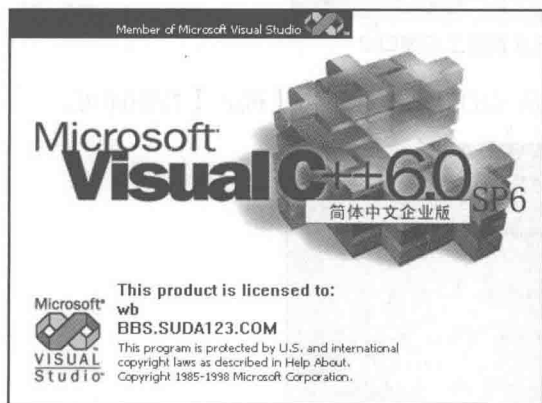


图 1.1 Visual C++ 6.0 启动画面

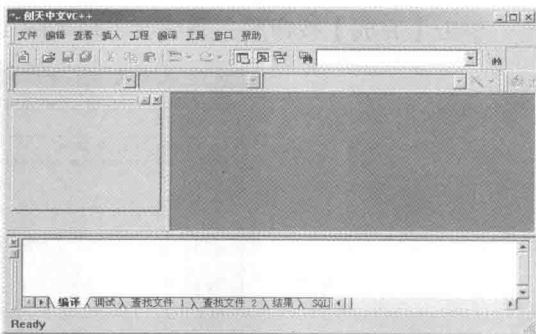


图 1.2 Visual C++ 6.0 初始窗口

2. 编辑源程序文件

(1) 建立新工程项目。在 Visual C++ 6.0 环境下,编辑 C 语言源程序文件首先需要建立一个工程项目,具体操作如下。

① 单击【文件】—【新建】命令,弹出【新建】对话框。

② 单击【工程】选项卡,单击【Win32 Console Application】选项,在【工程】文本框中输入工程项目名,如【gc1】,在【位置】框输入或选择新项目所在位置,单击【确定】按钮,如图 1.3 所示。

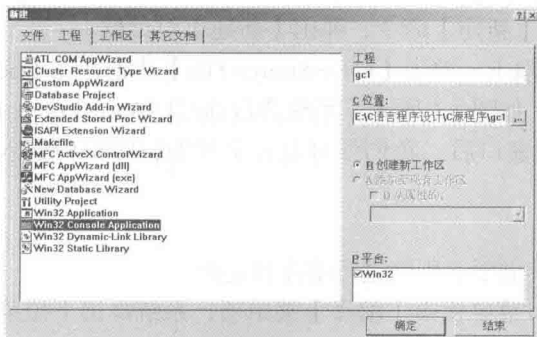


图 1.3 Visual C++ 6.0 新建工程窗口 1

在单击【确定】按钮以后，系统会弹出如图 1.4 所示的对话框，我们只要单击【完成】按钮即可。

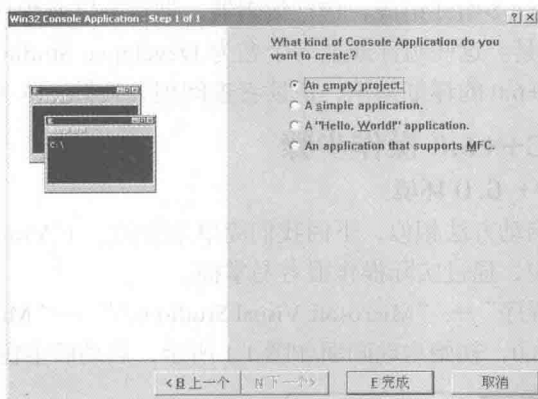


图 1.4 Visual C++ 6.0 新建工程窗口 2

单击【完成】按钮后，系统会出现如图 1.5 所示的对话框，单击【确定】按钮即可。

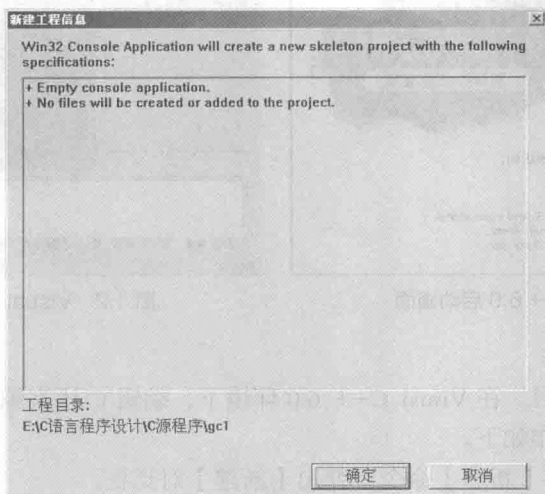


图 1.5 Visual C++ 6.0 新建工程窗口 3

(2) 建立新工程项目中的文件。完成上述操作后，系统会返回初始界面，这时候我们按如下步骤开始建立新工程下的 C 程序文件。

- ① 单击【文件】—【新建】命令，弹出【新建】对话框。
- ② 选择【文件】选项卡。单击【C++ Source File】选项，在【文件】文本框中输入文件名，单击【确定】按钮，如图 1.6 所示。系统会自动返回 Visual C++ 6.0 初始窗口。
- ③ 显示文件编辑区窗口后，我们就可以在文件编辑区窗口输入源程序文件，如图 1.7 所示。

3. 编译和连接

在输入完源程序后，需要对程序进行编译和连接。

方法一：选择主窗口菜单栏中【编译】菜单项，系统弹出下拉菜单，选择【构件】菜单命令。

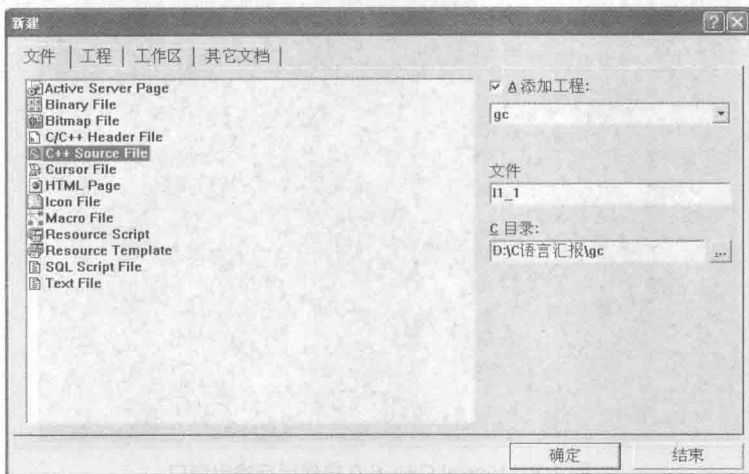


图 1.6 Visual C++ 6.0 新建文件窗口

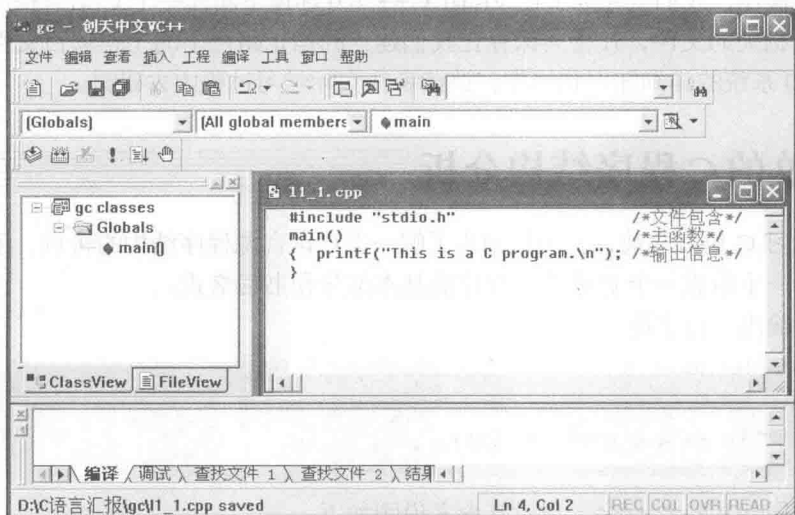


图 1.7 Visual C++ 6.0 源文件输入窗口

方法二：单击主窗口编译工具栏上的【Build】按钮进行编译和连接。

(1) 系统对程序文件进行编译和连接，生成以项目名称命名的可执行目标代码文件扩展名为.exe。

(2) 编译连接过程中，系统如发现程序有语法错误，则在输出区窗口中显示错误信息，给出错误的性质、出现位置和错误的原因等。如果双击某条错误信息，编辑区窗口右侧会出现一个箭头，指示再现错误的程序行。用户据此对源程序进行相应的修改，并重新编译和连接，直到通过为止。

4. 执行

方法一：单击【编译】菜单中【执行】命令。

方法二：单击主窗口编译工具栏上的【Build Execute】按钮来执行编译连接后的程序。

运行成功，屏幕上输出执行结果，并提示信息：“Press any key to continue”，如图 1.8 所示。此时按 Enter 键系统将返回 Visual C++ 6.0 主窗口。

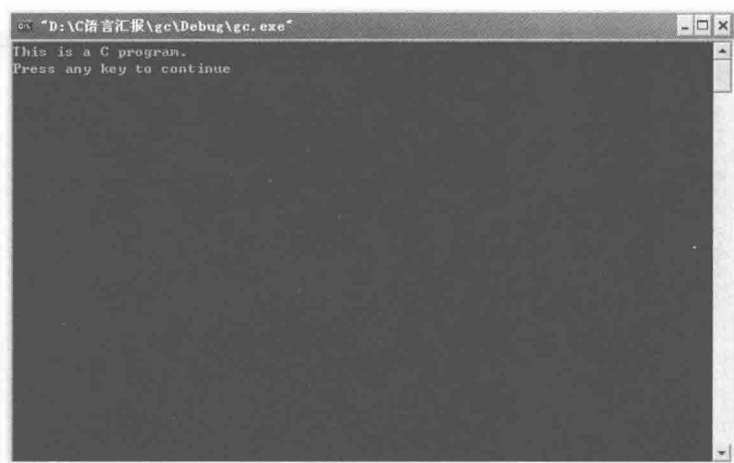


图 1.8 Visual C++ 6.0 程序运行输出窗口

通过上述操作，我们就完成了在 Visual C++ 6.0 环境下新建一个 C 语言程序的过程，系统会自动生成相关的文件，并将其保存在我们建立的指定路径下的工程项目文件夹中。关于 Visual C++ 6.0 系统的详细知识和操作，请读者在实际学习过程中去研讨。

1.3 简单的 C 程序结构分析

在正式学习 C 语言之前，我们先初步了解一下 C 语言源程序结构的特点。下面我们通过几个程序了解一下组成一个 C 语言源程序的基本部分和书写格式。

【例 1.1】输出一行字符。

```
#include<stdio.h>
void main()
{
    printf("This is a C program! \n");
}
```

这是一个最基本的 C 程序，现对其含义说明如下。

(1) #include<stdio.h> 是一个系统函数，其作用是用来输入和输出信息，在以后的章节中会加以介绍，读者在本章不需要过多关注。

(2) main 是主函数的函数名，表示这是一个主函数，主函数的函数名必须为 main。

(3) 每一个 C 源程序都必须有且只能有一个主函数（main 函数）。

(4) 一对“{ }”是一段程序或整个程序的起始和终止标志，必须成对出现。

(5) “printf("This is a C program! \n");”是一个函数调用语句，printf 函数的功能是把要输出的内容送到显示器去显示。

本程序的运行结果是在计算机上显示输出一句话：“This is a C program!”。概括地说，一个 C 函数必须有一个函数名字，有用“{ }”括起来的函数体，通俗地说叫函数包含的内容，也就是指函数用来处理什么信息。特别声明一点，任何 C 程序都必须要有有一个名字为 void main 的主函数，当然，还可以有其他函数，这些知识我们以后会学习到。

【例 1.2】求和。

```
#include<stdio.h>
void main()
```



```

{
    int x,y,sum;                /*定义 x、y、sum 三个变量的类型为整型 */
    x=1;                        /* 给 x 赋值 1 */
    y=2;                        /* 给 y 赋值 2 */
    sum=x+y;                    /* 把 x 和 y 的和赋值给 sum */
    printf("x 和 y 的和等于 %d\n",sum); /* 输出变量 sum 的值 */
}

```

这个程序比第一个程序要复杂一些，上面介绍过的就不再说明了，注意与刚才程序不同的地方。

(1) `int x,y,sum;` 用来定义变量的类型，`int` 为整型数据的定义符号，也是一个系统的保留字；`x`、`y`、`sum` 是 3 个变量，表示 3 个整型数。

(2) `x=1; y=2; sum=x+y;` 是 3 条赋值语句，其作用是把“=”后面的数值赋给前面的变量。

(3) `printf("x 和 y 的和等于 %d\n",sum);` 中包含了更多的信息，不仅要输出“`x 和 y 的和等于`”这些字符，还要输出 `sum` 的值。

(4) 每一行后面的 `/* */` 之间的内容为注释内容，用来说明每一行的含义，程序在运行时不会执行，其作用是让他人了解程序语句所包含的信息。也可以使用 `//` 作为注释语句的开头。

本程序的作用是求两个数的和，运行结果是：`x` 和 `y` 的和等于 3。

在本例题中的主函数体中又分为两部分，一部分为说明部分，另一部分为执行部分。说明是指变量的类型说明。由于例题 1.1 中未使用任何变量，因此无说明部分。C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部分，本例中，说明部分定义了 3 个变量 `x`、`y`、`sum`，说明部分后的 4 行为执行部分或称为执行语句部分，用以完成程序的功能。

【例 1.3】求两个数中的较大值。

```

#include<stdio.h>
int max(int a,int b);          /*函数说明*/
void main()                   /*主函数*/
{
    int x,y,z;                 /*变量说明*/
    int max(int a,int b);      /*函数说明*/
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);       /*输入 x、y 值*/
    z=max(x,y);                /*调用 max 函数*/
    printf("maxmum=%d",z);     /*输出*/
}
int max(int a,int b)           /*定义 max 函数*/
{
    if(a>b) return a;
    else return b;             /*把结果返回主调函数*/
}

```

我们再来看看这个程序与前面两个程序不同的地方。

(1) 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。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。关于函数的详细内容将在后面介绍。

(2) `printf("input two numbers:\n");` 中的信息我们使用英文，因为计算机程序的主要语言环境是英文，前面两个例题中的中文信息是为了便于初学者理解程序。当然，目前的 C 语言编译环境可以使用中文符号，但我们仍然建议读者在学习编程时尽可能地使用英文表示相关的

信息。具体的字符信息要依据开发程序的要求。

(3) `scanf` 函数的功能是将数值输入到相应的变量存储单元,即给变量输入数值。本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。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 `x`、`y` 中,然后调用 `max` 函数,并把 `x`、`y` 的值传送给 `max` 函数的参数 `a`、`b`。在 `max` 函数中比较 `a`、`b` 的大小,把大者返回给主函数的变量 `z`,最后在屏幕上输出 `z` 的值。

通过上述 3 个例题,我们可以初步了解一些关于 C 程序的结构知识。

(1) 一个 C 语言源程序可以由一个或多个函数组成。

(2) 一个源程序必须有一个且只能有一个 `void main` 函数,即主函数。

(3) 在源程序中可以有预处理命令 (`include` 命令仅为其中的一种),预处理命令通常应放在程序的最前面。

(4) 在进行变量类型说明和每一个语句后面都必须以分号结尾。但预处理命令,函数头和花括号“`}`”之后不能加分号。

(5) 各种类型标识符与其他标识符之间必须至少加一个空格隔开,否则会出现错误信息。如 `void main` 不能写成 `voidvoid main`, `int x` 不能写成 `intx` 等。

1.4 书写程序时应遵循的规则

为了便于阅读和修改程序,在进行程序书写时应遵循以下规则。

(1) 要尽可能让每一个说明和一个语句占一行。

(2) 用“`{}`”括起来的部分,通常表示了程序的某一层结构。“`{}`”一般与该结构语句的第一个字母对齐,并单独占一行。

(3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,使程序呈递进式结构,以便看起来更加清晰,增加程序的可读性,也便于修改程序中的错误信息。

在以后的学习中会逐步体会到遵循这些规则的重要性,希望初学者一定要从开始编程时就力求遵循这些规则,以养成良好的编程风格。

1.5 C 语言的字符集

任何一种语言的组成都可以分为两部分,第一是这种语言所使用的基本符号,第二是这种语言的基本规则。字符是组成语言的最基本的元素,C 语言字符集由字母、数字、空格、标点和特殊字符组成。

(1) 字母:小写字母 `a~z` 共 26 个;大写字母 `A~Z` 共 26 个。

(2) 数字: `0~9` 共 10 个

(3) 空白符:空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时,只起间隔作用,编译程序对它们忽略不计。因此,在程序中使用空白符与否,对程序的编译不发生影响,但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

(4) 标点符号、运算符和特殊字符。

这些内容在后续的学习过程中会逐步接触到,不要急于记忆。

需要特别说明的是，正如在前面例题中看到的，在 C 语言的字符串常量和注释中还可以使用汉字或其他可表示的图形符号，但它们不是 C 语言基本符号。

1.6 C 语言的标识符与关键字

1. 标识符

在 C 语言中，标识符是指在程序中使用的变量名、函数名、标号等。除库函数的函数名由系统定义外，其余都由用户自定义。C 语言规定，标识符只能是字母 (A~Z, a~z)、数字 (0~9)、下划线 (_) 组成的字符串，并且其第一个字符必须是字母或下划线。

以下标识符是合法的：

a, x, x3, BOOK_1, sum5

以下标识符是非法的：

3s 以数字开头

s*T 出现非法字符*

-3x 以减号开头

bowy-1 出现非法字符- (减号)

在使用标识符时还必须注意以下几点。

(1) 标准 C 语言不限制标识符的长度，但它受各种版本的 C 语言编译系统限制，同时也受到具体机器的限制。例如，在某版本 C 中规定标识符前 8 位有效，当两个标识符前 8 位相同时，则被认为是同一个标识符。

(2) 在标识符中，大小写是有区别的。如 BOOK 和 book 是两个不同的标识符。

(3) 标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便于阅读理解，做到“见名知义”，一般使用对应的英文单词、单词缩写或单词组合。

2. 关键字

关键字是由 C 语言规定的具有特定意义的字符串，通常也称保留字，是由 C 语言自身系统使用的，如表 1.1 所示用户定义的标识符不能与关键字相同，否则会出现错误。

表 1.1

C 语言的关键字

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	static	sizeof	struct	switch	typedef	union
unsigned	void	volatile	while			

C 语言的关键字分为以下几类。

(1) 类型说明符：用于定义、说明变量、函数或其他数据结构的类型。如前面例题中用到的 int 等。

(2) 语句定义符：用于表示一个语句的功能。如例 1.3 中用到的 if else 就是条件语句的语句定义符。