

C语言 程序设计教程

黄迎久 庞润芳 主 编
赵军富 徐 扬 胡晓燕 贾 茹 副主编



+



+



+



- ◆ 以基础理论—实用技术—实训为主线
- ◆ 按照教与学的实际需要取材谋篇
- ◆ 精心设置了“小型案例实训”，旨在培养学生的实践能力
- ◆ 配备丰富的免费教学资源——电子教案、习题答案和与课程配套的学习系统



全国高等院校应用型创新规划教材·计算机系列

C 语言程序设计教程

黄迎久 庞润芳 主 编

赵军富 徐 扬 胡晓燕 贾 茹 副主编

清华大学出版社

北 京

内 容 简 介

本书以 Visual C++ 6.0 为开发平台,全面介绍了 C 语言的基础知识、程序结构及程序编写技巧。全书共分为 12 章,包括引言、C 语言基础知识、顺序结构程序设计、选择结构程序设计、循环结构程序设计、函数、预处理、数组、指针、结构体与共用体、文件、位运算。

本书内容丰富、结构合理、思路清晰、语言简练流畅、示例翔实,在准确讲解概念的基础上力求通俗易懂,大量引入实例,分析程序设计思路,旨在培养学生的实践动手能力。每章均配有丰富的例题和习题,并附有参考答案。

本书既可作为普通高校非计算机专业计算机公共课的教材,也可作为学习 C 语言程序设计的自学用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计教程/黄迎久,庞润芳主编.——北京:清华大学出版社,2016

(全国高等院校应用型创新规划教材·计算机系列)

ISBN 978-7-302-41852-8

I. ①C… II. ①黄… ②庞… 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(2015)第 252090 号

责任编辑:秦 甲 杨作梅

封面设计:杨玉兰

责任校对:周剑云

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.25 字 数:419 千字

版 次:2016 年 1 月第 1 版 印 次:2016 年 1 月第 1 次印刷

印 数:1~2500

定 价:38.00 元

产品编号:064417-01

前 言

“C 语言程序设计”是高校计算机专业和理工科专业重要的公共基础课程之一。C 语言的功能强大，使用灵活方便，移植性强，兼有高级语言和低级语言的特点，利用 C 语言可以编写系统软件和应用软件。

作者根据多年的教学经验，结合高校非计算机专业计算机基础教学的最新大纲，在分析国内外多种同类教材的基础上，编写了本书。本书力求通过理论联系实际，采用计算思维的方法，引导和启发学生掌握思考和解决问题的方法，达到举一反三的目的。

本书共分 12 章，主要内容如下：

第 1 章主要介绍 C 语言的发展过程及特点、C 语言的结构特点、C 语言程序的执行过程。

第 2 章主要介绍 C 语言的数据类型、标识符、常量和变量、运算符和表达式等内容。

第 3 章主要介绍算法的基本概念、C 语言的基本语句、数据的输入和输出等内容。

第 4 章主要介绍 if 结构、if...else 结构、if 语句的嵌套、switch 语句。

第 5 章主要介绍 while 语句、do...while 语句、for 语句、循环的嵌套、break 和 continue 语句在循环结构中的应用等内容。

第 6 章主要介绍函数的基本概念、变量的作用域和存储类型、内部函数和外部函数、函数的递归调用等内容。

第 7 章主要介绍宏定义、文件包含、条件编译等内容。

第 8 章主要介绍数组概述、一维数组、二维数组、字符数组与字符串等内容。

第 9 章主要介绍指针概述、指针变量与简单变量的关系、指针与数组、指针作为函数的参数、函数的返回值为指针、指向指针的指针等内容。

第 10 章主要介绍结构体类型、结构体数组、结构体变量与函数、共用体数据类型、枚举数据类型、链表的概念等内容。

第 11 章主要介绍 C 文件的基本概念和分类、文件的基本操作、文件的定位、出错的检测等内容。

第 12 章主要介绍位的取反运算、左移运算、右移运算、与运算、或运算、异运算、复合位运算等内容。此章为选讲章节。

本书内容丰富、结构合理、思路清晰、语言简练流畅，书中所有示例都已在 Visual C++ 6.0 环境下调试并运行通过。为了帮助读者更好地掌握 C 语言，每章末尾都安排了类型丰富的习题，并且随书提供了自主开发的“C 语言辅导学习系统”软件。辅导学习系统中有“学习指导”、“单元练习”、“综合练习”以及“经典算法”等模块，紧密结合教学内容，已应用于教学实践多年，深受师生好评。读者依据本书循序渐进地学习，可以巩固基本知识、培养实践能力、增强对基本概念的理解和解决实际问题的能力，能够比较容易地掌握 C 语言的主要用法。

本书获内蒙古科技大学教材建设项目资助，由内蒙古科技大学计算机教学基地的教师

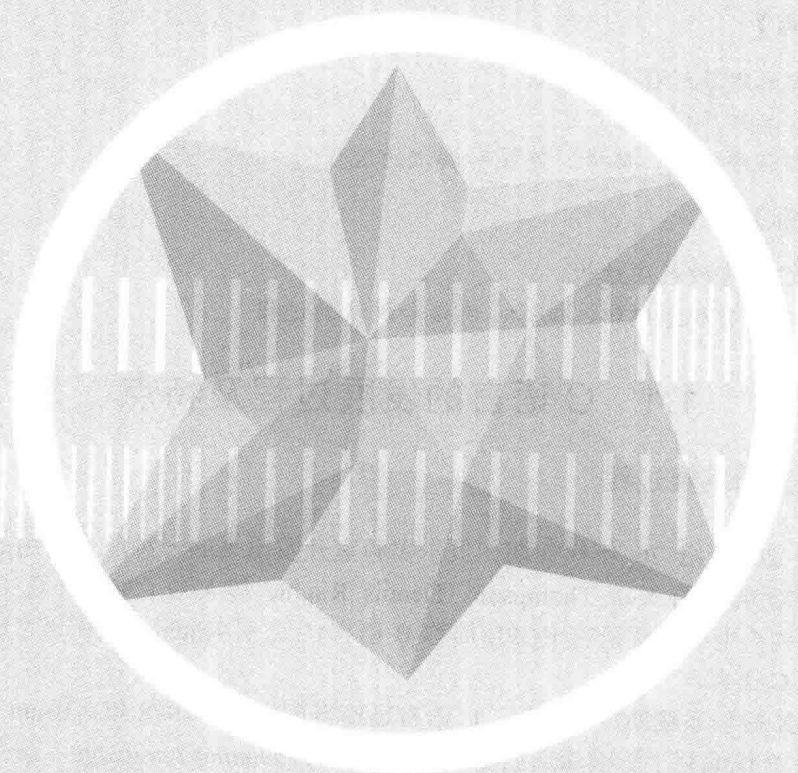
第 1 章 引言	1
1.1 C 语言的发展过程及特点	2
1.1.1 C 语言的发展过程	2
1.1.2 C 语言的特点	2
1.2 简单的 C 语言程序	3
1.3 C 语言程序的结构	5
1.4 C 语言程序的执行	6
1.4.1 编译源程序	6
1.4.2 链接目标程序	6
1.4.3 C 语言开发工具简介	7
本章小结	12
习题	13
第 2 章 C 语言基础知识	15
2.1 C 语言的数据类型	16
2.1.1 整型数据	16
2.1.2 浮点型数据	18
2.1.3 字符型数据	19
2.2 标识符、常量和变量	20
2.2.1 标识符	20
2.2.2 常量	21
2.2.3 变量	23
2.3 运算符与表达式	24
2.3.1 概述	24
2.3.2 算术运算符及算术表达式	25
2.3.3 赋值运算符及赋值表达式	26
2.3.4 自增、自减运算符	27
2.3.5 关系运算符及关系表达式	28
2.3.6 逻辑运算符及逻辑表达式	29
2.3.7 位运算符及位运算	30
2.3.8 条件运算符及条件表达式	31
2.3.9 逗号运算符及逗号表达式	32

2.3.10 数据类型的转换	32
2.4 小型实训案例	33
本章小结	34
习题	34
第 3 章 顺序结构程序设计	37
3.1 算法	38
3.1.1 算法的概念	38
3.1.2 算法的表示	39
3.2 结构化程序设计的三种基本结构	40
3.3 C 语言的基本语句	40
3.4 数据的输入/输出	42
3.4.1 格式输出函数 printf()	42
3.4.2 格式输入函数 scanf()	43
3.4.3 格式字符	44
3.4.4 字符输入/输出函数	47
3.5 程序应用实例	48
3.6 小型实训案例	50
本章小结	51
习题	51
第 4 章 选择结构程序设计	55
4.1 if 结构	56
4.1.1 if 语句	56
4.1.2 if...else 语句	57
4.1.3 if 语句的嵌套	58
4.2 switch 语句	60
4.2.1 switch 语句的一般格式	60
4.2.2 break 语句在 switch 语句中 的应用	62
4.3 程序应用实例	63
4.4 小型实训案例	65
本章小结	66

习题	66	第 7 章 预处理	127
第 5 章 循环结构程序设计	71	7.1 宏定义	128
5.1 概述	72	7.2 文件包含	130
5.2 while 语句	72	7.3 条件编译	131
5.3 do...while 语句	76	本章小结	133
5.4 for 语句	77	习题	133
5.5 循环的嵌套	79	第 8 章 数组	137
5.6 break 语句与 continue 语句	80	8.1 概述	138
5.6.1 break 语句	80	8.2 一维数组	139
5.6.2 continue 语句	81	8.2.1 一维数组的定义	139
5.6.3 break 语句与 continue 的 比较	83	8.2.2 一维数组的初始化	140
5.7 几种循环的比较	83	8.2.3 一维数组元素的引用	140
5.8 程序应用实例	83	8.2.4 一维数组的应用	142
5.9 小型实训案例	89	8.3 二维数组	147
本章小结	90	8.3.1 二维数组的定义	147
习题	90	8.3.2 二维数组的初始化	148
第 6 章 函数	95	8.3.3 二维数组的引用	149
6.1 函数概述	96	8.3.4 二维数组的应用	149
6.1.1 函数的定义	97	8.4 字符数组与字符串	152
6.1.2 函数的声明和调用	100	8.4.1 字符数组	152
6.1.3 函数的参数传递	105	8.4.2 字符串	153
6.2 变量的作用域和存储类型	107	8.4.3 字符串处理函数	155
6.2.1 变量的作用域	108	8.4.4 字符数组的应用	159
6.2.2 变量的存储类型	110	8.5 数组作函数参数	160
6.3 内部函数与外部函数	114	8.5.1 数组元素作函数实参	160
6.4 函数的递归调用	116	8.5.2 数组名作函数实参	161
6.5 程序应用实例	117	8.6 小型实训案例	162
6.6 小型实训案例	119	本章小结	163
本章小结	120	习题	164
习题	121	第 9 章 指针	171
		9.1 指针概述	172

9.1.1 指针变量的定义.....	172	10.4 共用体数据类型.....	212
9.1.2 指针变量的初始化.....	173	10.4.1 共用体类型的声明.....	212
9.1.3 指针变量的引用.....	174	10.4.2 共用体变量的定义、 初始化及引用.....	213
9.1.4 指针的运算.....	174	10.5 枚举数据类型.....	215
9.2 指针变量与简单变量的关系.....	175	10.5.1 枚举类型的定义.....	215
9.3 指针与数组.....	177	10.5.2 枚举变量的定义.....	215
9.3.1 指向一维数组的指针.....	177	10.5.3 有关枚举型数据的操作.....	216
9.3.2 指向多维数组的指针.....	178	10.6 链表的概念.....	216
9.3.3 字符指针.....	182	10.6.1 动态分配内存.....	217
9.3.4 指针数组.....	185	10.6.2 单链表的建立.....	218
9.4 指针作为函数的参数.....	186	10.6.3 从单链表中删除节点.....	220
9.5 函数的返回值为指针.....	188	10.6.4 向链表中插入节点.....	220
9.6 指向函数的指针.....	189	10.7 小型实训案例.....	222
9.6.1 指向函数的指针变量定义.....	189	本章小结.....	223
9.6.2 函数指针变量的使用.....	190	习题.....	223
9.7 指向指针的指针.....	191	第 11 章 文件.....	229
9.8 程序应用实例.....	192	11.1 C 语言文件的基本概念与分类.....	230
9.9 小型实训案例.....	194	11.1.1 文件的基本概念.....	230
本章小结.....	195	11.1.2 文件的分类.....	230
习题.....	196	11.2 文件的基本操作.....	231
第 10 章 结构体与共用体.....	201	11.2.1 文件类型指针.....	231
10.1 结构体类型.....	202	11.2.2 打开文件函数 <code>fopen()</code>	232
10.1.1 结构体的概念.....	202	11.2.3 关闭文件函数 <code>fclose()</code>	233
10.1.2 结构体类型的定义.....	202	11.3 文件的读写.....	233
10.1.3 结构体变量的定义.....	204	11.3.1 字符读/写函数.....	234
10.1.4 结构体变量的初始化.....	205	11.3.2 字符串读/写函数.....	237
10.1.5 结构体变量成员的引用.....	206	11.3.3 数据块读/写函数.....	237
10.2 结构体数组.....	208	11.3.4 格式化读/写函数.....	238
10.2.1 结构体数组的定义.....	208	11.4 文件的定位.....	239
10.2.2 结构体数组的初始化.....	208	11.4.1 <code>rewind()</code> 函数.....	239
10.2.3 结构体数组的引用.....	209	11.4.2 <code>fseek()</code> 函数.....	239
10.3 结构体变量与函数.....	210		

11.4.3 ftell()函数.....	240	12.5 按位或运算.....	251
11.5 出错的检测.....	241	12.6 按位异或运算.....	252
11.6 小型实训案例.....	242	12.7 复合位运算符.....	253
本章小结.....	243	本章小结.....	254
习题.....	243	习题.....	254
第 12 章 位运算.....	247	附录 A C 语言的关键字.....	257
12.1 按位取反运算.....	248	附录 B ASCII 码表.....	259
12.2 按位左移运算.....	249	附录 C 常用库函数.....	260
12.3 按位右移运算.....	249	附录 D C 运算符的优先级与结合性.....	267
12.4 按位与运算.....	250		



第 1 章

引 言



【本章要点】

- C 语言的发展过程及其特点
- C 语言的程序结构
- C 语言程序编辑、编译、连接和调试的过程

【学习目标】

- 掌握 C 语言的程序结构
- 掌握 C 语言程序编辑、编译、连接和调试的过程

1.1 C 语言的发展过程及特点

1.1.1 C 语言的发展过程

C 语言是 20 世纪 70 年代初期在贝尔实验室开发出来的一种广为使用的编程语言。1970 年, 贝尔实验室的 Ken Thompson、Dennis Ritchie 等人为了重新开发 UNIX 操作系统, 设计了一种小型的编程语言 BCPL(简称 B 语言), 之后 Ritchie 对 B 语言进行了升级, 并将其命名为 C 语言。

随着 UNIX 操作系统的广泛使用, C 语言迅速得到推广。1978 年, Brian Kernighan 和 Dennis Ritchie 合作编写了影响深远的名著 *The C Programming Language*, 该书成为 C 语言程序员必读的“圣经”, 书中介绍的 C 语言也成为后来 C 语言各版本的基础, 被称为标准的 C 语言。随着 C 语言的迅速普及, 其也在不断地变化, 即不时地增加新特性并去除一些旧特性, 于是 C 语言需要一个全面、准确的最新描述开始成为“共识”。

1983 年美国国家标准协会(ANSI)制定了 C 语言标准——83 ANSI C, 1988 年又制定了新的 C 语言标准——ANSI X3.159—1989(简称 C89)。1990 年国际标准化组织(ISO)通过了 C89 标准, 并将其作为 ISO/IEC 9899:1990 国际标准, 简称 C90; 1999 年国际标准化组织又对 C 语言标准进行了修订, 在原 C 语言特征的基础上, 增加了一些面向对象的特征, 命名为 ISO/IEC 9899:1999, 简称 C99。目前大多数软件厂商提供的 C 语言编译系统均是以 C89 为标准开发的, 本书也以 C89 为基础进行介绍。

1.1.2 C 语言的特点

C 语言是一种结构化的程序设计语言, 兼有高级语言和低级语言的功能。C 语言的特点表现在以下几个方面。

- (1) C 语言允许直接访问物理地址, 能进行位操作, 可以对硬件直接操作。
- (2) 移植性强。用 C 语言编写的程序几乎不用修改就能适用于各种型号的计算机及各种操作系统。

- (3) 运算符和数据结构丰富, 功能强大。
- (4) 生成的目标代码质量高, 程序执行效率高。

C 语言的上述特点使得其不仅适用于编写系统软件(如操作系统、编译系统等), 也适

用于编写应用软件(如图形处理、信息处理等),因此成为最流行的程序设计语言之一。

尽管 C 语言拥有众多优点,但同样也有一些缺点。由于 C 语言放松了语法检查以及出众的灵活性,因此提高了 C 语言编程出错的概率。

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,国内绝大多数高校都开设了“C 语言程序设计”课程。随着 Internet 技术的飞速发展,一些新型的编程语言也似雨后春笋般地孕育而生,如 C++、Java、PHP、C#等,但这些编程语言大多数都是以 C 语言为基础或借鉴了 C 语言,因此,学好 C 语言对学习其他计算机编程语言大有帮助。

1.2 简单的 C 语言程序

下面介绍几个实例,从中总结一下 C 语言的程序结构。

【实例 1-1】 编写程序,在屏幕上显示“Hello World!”。

程序如下:

```
#include<stdio.h>
void main()
{
    /*在屏幕上输出 Hello World! */
    printf("Hello World!\n");
}
```

程序运行结果:

```
Hello World!
```

程序说明:

(1) #include: 称为“文件包含命令”,其作用是把系统目录下的头文件 stdio.h 包含到本程序中,成为程序的一部分。

C 语言提供的头文件中包含各种标准库函数的函数原型,在程序中调用某个库函数时,就必须将该函数原型所在的头文件包含进来。本程序包含的头文件是 stdio.h(stdio 是 standard input & output(标准输入/输出)的缩写),该文件里的函数主要用于处理数据流的标准输入/输出。

(2) main: 主函数的名字,每一个 C 语言程序只允许有一个 main 函数;main 函数之前的 void 表示此 main 函数是“空类型”,即执行此函数后不产生一个函数值(有的函数执行后会产生一个函数值,如 cos(x))。

(3) /* ...*/: 表示注释语句,即程序中的说明文字,是不被 C 语言系统执行的语句。位于“/*”和“*/”之间的所有内容都属于注释语句,可以写在一行之内,也可以写在多行之内。

(4) printf(): C 语言系统库函数,其函数原型在头文件 stdio.h 中,该函数的功能是将其圆括号中的内容输出到显示器上,其圆括号中的双引号所引的内容会按原样输出,其中的“\n”是换行符。

(5) main()函数中的内容必须放在函数体中,即 main()函数下方的一对花括号中。

 提示:

- 包含头文件的命令尽量不要忽略,虽然有的时候不影响程序的运行,但希望学习者在开始学习的时候就能养成良好的书写习惯。
- `main()` 函数之前的 `void` 可以忽略不写。

【实例 1-2】 计算表达式 $1+3+5+7+9$ 的值。

程序如下:

```
#include<stdio.h>
void main( )
{
    int sum;
    sum=1+3+5+7+9;
    printf("sum=%d\n",sum);
}
```

程序运行结果:

```
sum=25
```

程序说明:

- (1) 在 `main` 函数里定义了一个整型变量 `sum`。
- (2) 语句 “`sum=1+3+5+7+9;`” 是将表达式 “ $1+3+5+7+9$ ” 的计算结果赋给变量 `sum`。
- (3) 语句 “`printf(“sum=%d”,sum);`” 是将变量 `sum` 的结果输出到计算机屏幕上,其中双引号中的格式字符 “`%d`” 对应的是双引号之后的变量 `sum` 的值。

 提示: 本程序中 `printf()` 的双引号里出现了格式字符 `%d`, 其作用是输出一个整数类型的值, 输出的对象是位于双引号之后的整型变量 `sum`。

【实例 1-3】 求两个整数中的较大者。

程序如下:

```
#include<stdio.h>
int max(int x,int y);
void main( )
{
    int a,b,c;
    a=12;
    b=34;
    c=max(a,b);
    printf("max=%d",c);
}
int max(int x, int y)
{
    int z;
    if(x>y) z=x;
    else z=y;
    return (z);
}
```

程序运行结果:

```
max=34
```

程序说明:

(1) 本程序中包括两个函数, 主函数 `main` 和被调用的函数 `max`。`max` 函数的作用是判断 `x` 和 `y` 中的较大值。`max` 函数中的 `return` 语句将 `z` 的值返回给主调函数 `main`, 返回值通过函数名 `max` 带回到 `main` 函数中调用 `max` 函数的位置。

(2) 程序的第 2 行是对函数 `max` 的声明。

(3) 程序的第 8 行调用 `max` 函数, 在调用时将实际参数 `a` 和 `b` 的值分别传给 `max` 函数中的形式参数 `x` 和 `y`, 经过执行 `max` 函数, 其返回值返回给 `main` 函数中的变量 `c`。

(4) 程序的第 9 行输出变量 `c` 的值。

(5) 程序的第 11 行至第 17 行是 `max` 函数的具体定义。

1.3 C 语言程序的结构

前面给出的 C 语言程序实例虽然结构和功能简单, 但其中都包含了 C 语言程序的基本组成部分, 从中可以得到以下结论:

(1) C 程序是由函数构成的。一个 C 语言程序必须包含一个 `main` 函数, 或者一个 `main` 函数和若干个其他函数。因此, 函数是 C 语言的基本单位。被调用的函数可以是系统提供的库函数(如 `printf()` 函数), 也可以是用户自己编写设计的函数(如实例 1-3 中的 `max()` 函数)。

(2) C 程序的函数由两部分组成, 即函数首部和函数体。

① 函数首部, 即函数的第一行, 包括函数类型、函数名称、函数参数(形式参数)名称和参数类型。函数名称后面必须跟一对圆括号, 括号内写明函数的参数类型和参数名称, 函数也可以没有参数, 如 `main()`。实例 1-3 中 `max` 函数的首部为

<code>int</code>	<code>max (</code>	<code>int</code>	<code>x,</code>	<code>int</code>	<code>y)</code>
↓	↓	↓	↓	↓	↓
函数类型	函数名称	参数类型	参数名称	参数类型	参数名称

② 函数体, 即函数首部下方花括号内的部分。若函数体有多个花括号, 则以最外层的一对花括号包含的内容为函数体的范围。

函数体一般包括两部分:

- 声明部分。这部分要定义所要使用的变量和对所要调用的函数进行声明。如实例 1-3 中 `main` 函数对变量的定义语句 “`int a,b,c;`”。
- 执行部分。这部分由若干条语句组成。

在某些情况下也可以没有声明部分, 如实例 1-1; 甚至可以既无声明部分也无执行部分, 如:

```
void tmp( )
{ }
```

这是一个空函数，什么也不执行。

(3) C 程序总是从 main 函数开始执行，直至 main 函数中最后一条执行语句为止，与 main 函数的位置无关。

(4) C 程序书写格式自由。一行内可以写若干条语句，一条语句也可以分写在多行上。

(5) 每条语句和数据声明的最后都必须带一个分号，即使是程序中的最后一条语句也要带上分号。

我们将通过后续章节的学习，了解 C 语言程序的语法规则、基本表达式、控制结构语句的作用，并通过了解模块化程序设计的思想和方法，逐步掌握 C 语言程序的设计方法。

1.4 C 语言程序的执行

用 C 程序语言编写的程序称为源程序(Source Program)，由于计算机只能识别和执行由 0 和 1 组成的二进制指令，为了使计算机能够执行编程语言的源程序，首先要将源程序翻译成二进制的“目标程序”(Object Program)，这个过程称为“编译”(Compile)；然后还要将目标程序和系统提供的函数与其他目标程序链接起来，得到计算机可以执行的程序(称为“可执行程序”)，这个过程称为“链接”(Link)。

1.4.1 编译源程序

C 语言源程序的扩展名为.c，必须将其编译成目标程序，再将目标程序链接成可执行程序，才能在计算机上运行。C 语言源程序的编译过程如图 1-1 所示，由词法分析、语法分析和代码生成三部分组成。



图 1-1 C 语言源程序的编译过程

1.4.2 链接目标程序

C 源程序经过编译后所生成的目标程序尽管是机器语言的形式，但却不是计算机可以执行的方式，此时的目标程序还只是一些松散的机器语言，要想得到可执行的程序，就得将它们链接起来。

编程语言的链接工作由链接器(Linker)来完成。链接器的任务就是将目标程序链接成可执行的程序(或称“载入模块”)，这种可执行的程序是一种可储存在磁盘存储器上的文件。

提示:

- 并不是每一个目标程序都可以链接成可执行程序。
- 在应用系统中，只允许一个源程序中包含一个 `main()` 函数。

C 语言源程序一旦生成了可执行程序，就可以反复地被加载执行，而不再需要重新编译、链接；如果修改了源程序，也不会影响已生成的可执行程序，需要对修改后的源程序重新编译和链接，生成一个新的可执行程序。

1.4.3 C 语言开发工具简介

为了提高编程的工作效率，目前大多采用集成开发工具来开发 C 语言程序。集成开发工具是一个经过整合的软件系统，它将编辑器、编译器、链接器和其他软件单元集合在一起，在这个工具里，程序员可以很方便地对程序进行编辑、编译、链接以及跟踪程序的执行过程，以便查找程序中的问题。

适合 C 语言的集成开发工具有很多，如 Visual C++、Dev C++、GCC、Visual Studio 等。这些集成开发工具各具特色，分别适用于 Windows 环境和 Linux 环境。常用的 C 语言集成开发工具的基本特点和运行环境如表 1-1 所示。

表 1-1 常用的 C 语言开发工具

开发工具	运行环境	基本特点
Visual C++	Windows	(1) 符合标准 C 语言(即 C89 标准) (2) 各系统具有一些扩充内容 (3) 能开发 C 语言程序(集程序编辑、编译、链接、调试、运行于一体)
Dev C++	Windows	
GCC	Linux	
Visual Studio	Windows	

从表 1-1 中可以看出，目前流行的 C 语言开发工具都适合于开发 C++ 程序。因为 C++ 语言是在借鉴、继承 C 语言的基础上开发出来的编程语言，C 语言的基本表达式、基本结构和基本语法等同样适合 C++ 语言，因此，这些开发工具同样适用于 C 语言。

由于我国计算机等级考试采用的 C 语言开发工具是 Visual C++ 6.0，而且国内各大学的计算机实验室一般都安装了此开发工具，因此本书也采用 Visual C++ 6.0 作为 C 语言开发工具，书中的所有实例、习题都是在 Visual C++ 6.0 的环境下调试和运行的。下面来介绍在 Visual C++ 6.0 环境中如何编辑、编译、链接和运行 C 语言源程序。

1. 安装和运行 Visual C++ 6.0

进入 Visual C++ 的安装文件夹，找到 `setup.exe` 文件并双击，即可启动安装进程，在输入序列号和用户信息后，就进入了安装进程。

安装完 Visual C++ 6.0 后，就可以进入该集成环境编辑 C 源程序了。为了方便使用该系统，最好在桌面上创建 Visual C++ 6.0 快捷图标，用鼠标双击该快捷图标就能启动 Visual C++ 6.0。

Visual C++ 6.0 集成环境主窗口如图 1-2 所示。主窗口的顶部是 Visual C++ 的主菜单

栏, 包含 9 个主菜单项: File(文件)、Edit(编辑)、View(浏览)、Insert(插入)、Project(项目)、Build(建造)、Tools(工具)、Window(窗口)和 Help(帮助)。主窗口的左侧是项目工作区, 右侧是程序编辑窗格, 下方是调试信息窗格。项目工作区用于显示所设定的工作区的信息, 程序编辑窗格用于输入和编辑源程序, 调试信息窗格用于显示调试源程序时出现的提示信息。

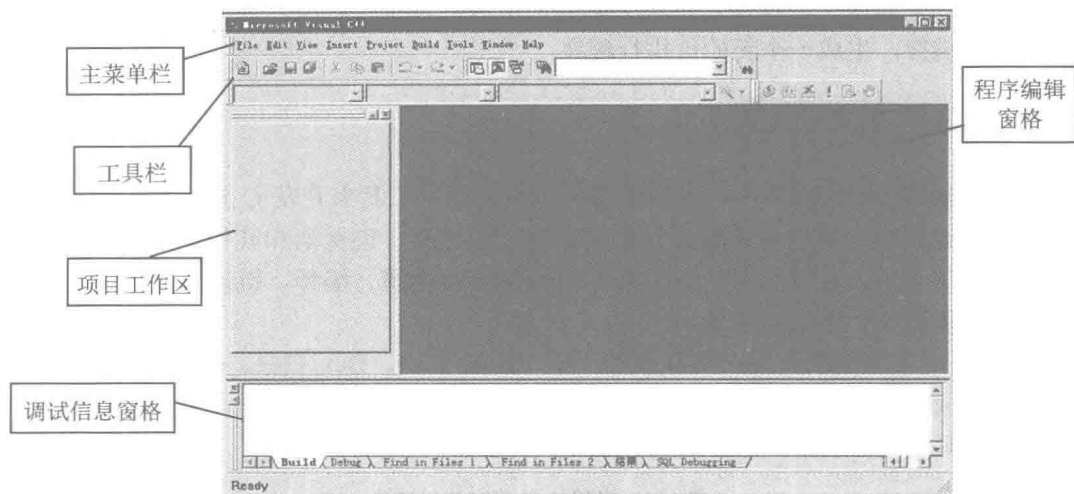


图 1-2 Visual C++ 6.0 集成环境主窗口

2. 编辑源程序

1) 新建源程序

在 Visual C++主窗口的主菜单栏中选择 File(文件)菜单项, 然后选择 New(新建)命令, 如图 1-3 所示。

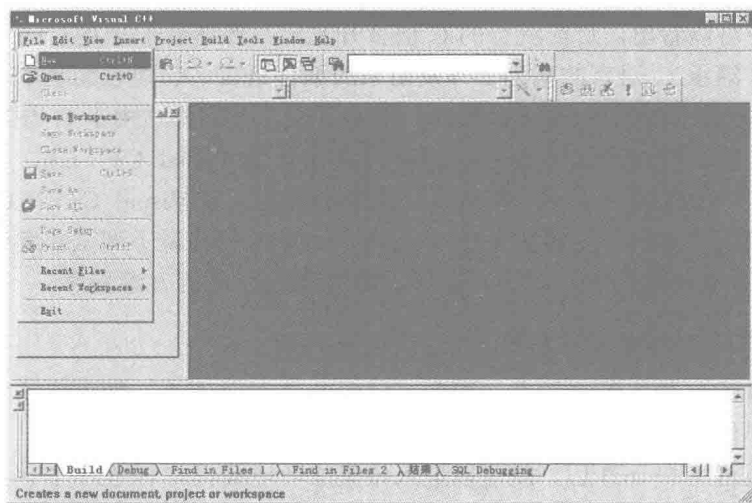


图 1-3 新建源程序

在屏幕上会出现一个 New 对话框，切换到 Files 选项卡，在左侧的列表框中选择 C++ Source File 项，表示要建立新的 C++ 源程序文件；然后在右侧的 File 文本框中输入准备编辑的源程序文件名称，如 c1.c，在 Location 文本框中输入源程序文件的存储路径，如 C:\example；最后单击 OK 按钮，编辑的源程序就以 c1.c 为文件名存放在 C 盘的 example 文件夹下了，如图 1-4 所示。当然，也可以指定其他的文件名和存储路径。

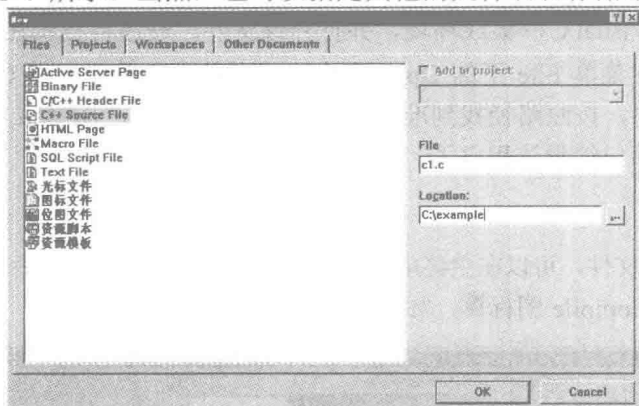


图 1-4 New 对话框

 **提示：** 由于我们编辑的是 C 程序，所以在 File 文本框中输入的源程序文件的扩展名必须是.c；若遗漏扩展名，则 Visual C++ 系统默认该源程序文件为 C++ 源程序文件(C++ 源程序文件的扩展名为.cpp)。C++ 源程序与 C 源程序有些区别，有可能会影响接下来的“编译”和“链接”过程。

回到主窗口后，由于前面已指定了源程序文件名和路径，因此主窗口的标题栏中显示出“C:\example\c1.c”，可以看到光标在程序编辑窗格中闪动，表示程序编辑窗格已经激活，可以输入和编辑源程序了，如图 1-5 所示。

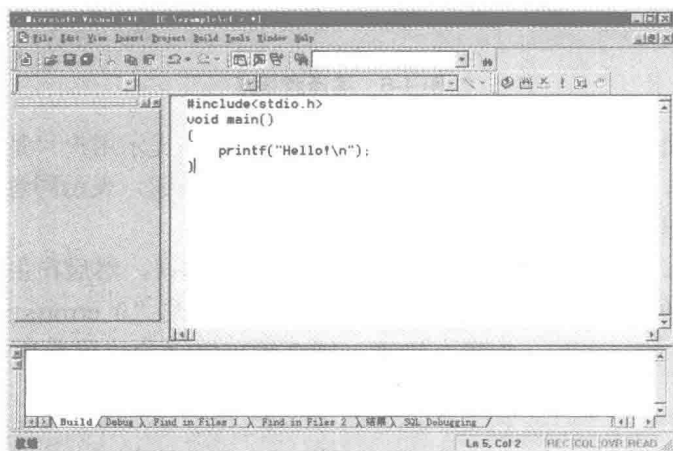


图 1-5 输入和编辑源程序