



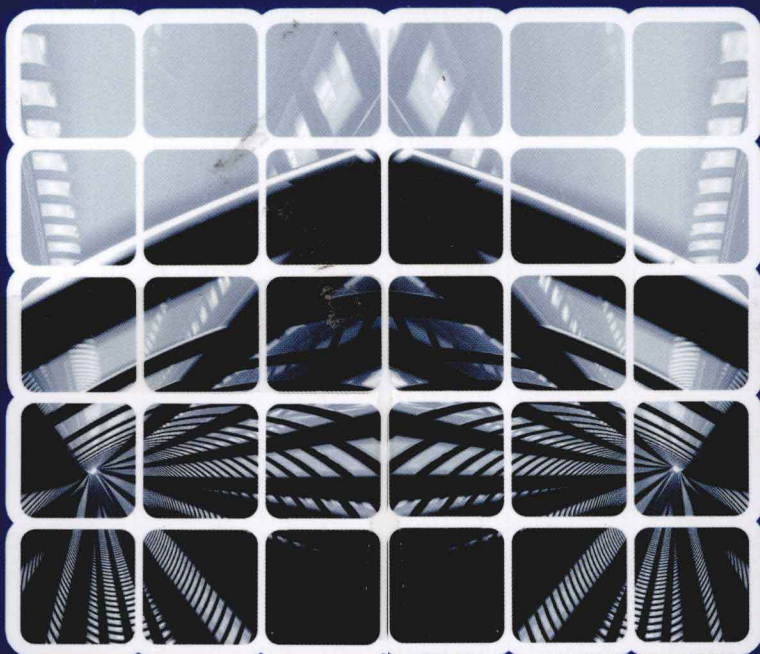
面向应用型高校“十二五”规划教材

The C Programming Language

C语言程序设计

李俊主编

- 配有自动评测考试系统
- 理论与实践紧密融合
- 案例新颖、趣味性强



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

面向应用型高校“十二五”规划教材

C 语言程序设计

李 俊 主编

张小莉 郭宇周 尹胜彬 高立敏 厉志红 编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书由浅入深、循序渐进地介绍 C 语言程序设计的思路和方法。全书共 13 章,分为教学篇和学习篇两部分。教学篇系统地介绍基于 Visual C++ 的 C 语言开发环境、数据类型与表达式、基本输入/输出语句、流程控制、模块化程序设计方法和文件系统的基本操作。学习篇主要内容包括 Visual C++ 6.0 环境下的 C 语言上机实践、练习题、全国计算机等级考试试题及答案。本书内容翔实、案例新颖、结构清晰、重点明确,以丰富有趣的案例驱动知识点教学。本书配有自动评测考试系统,提供相关教学资源(含电子课件)。

本书适合于作为高等院校计算机程序设计课程的教材,也可作为计算机程序设计培训和各种计算机等级考试的参考教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C 语言程序设计/李俊主编. —北京:电子工业出版社, 2012.1

面向应用型高校“十二五”规划教材

ISBN 978-7-121-15480-5

I. ①C… II. ①李… 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(2011)第 259290 号

策划编辑:谭海平

责任编辑:章海涛

印 刷:涿州市京南印刷厂

装 订:涿州市桃园装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1 092 1/16 印张:21 字数:525 千字

印 次:2012 年 1 月第 1 次印刷

定 价:37.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

C 语言是目前国际上广泛流行的一种结构化的程序设计语言,兼具高级语言和低级语言的功能,提供类型丰富、使用灵活的基本运算和数据类型,具有较高的可移植性。C 语言不仅适用于开发系统软件,而且是开发应用软件和进行大规模科学计算的常用程序设计语言。

全书分为教学篇和学习篇两部分。教学篇系统地介绍基于 Visual C++ 的 C 语言开发环境、数据类型与表达式、基本输入/输出语句、流程控制、模块化程序设计方法和文件系统的基本操作。学习篇主要内容包括 Visual C++ 6.0 环境下的 C 语言上机实践、练习题、全国计算机等级考试试题及答案。本书内容翔实、案例新颖、结构清晰、重点明确、可操作性强,以丰富有趣的案例驱动知识点教学。

本书具有如下主要特点:

1. 知识点精练,适合短学时教学

现在,各高校都在对课程学时进行压缩。“C 语言程序设计”课程知识点繁多,如何让读者在短时间内理解 C 语言程序设计的精髓呢?为了解决这个问题,作者对各章节中的知识点进行了提炼,删减了一些不常用甚至几乎从来不用知识点。因此,本书能够满足短学时教学的需要。

2. 案例新颖、趣味性强

书中的每个案例都由作者精心设计,趣味性较强,通过这些案例,不仅可以提高读者学习的兴趣,也可以使读者对所学知识点达到举一反三的效果,从而使得读者更深刻理解所学的知识点。

3. 通过综合应用融合知识点

很多 C 语言教材都是独立地介绍 C 语言的知识点,这样就会造成读者无法将 C 语言的各知识点融为一个整体。为了解决这个问题,本书的各知识点都附有精彩的案例和综合应用,这些综合应用将本章的知识点与前面各章的知识点综合起来,使读者能够直观地将这些知识点融为一体。

4. 提高读者分析问题和独立思考问题的能力

读者在学习的过程中,经常会遇到这样的问题:教材的例子能看懂,教师讲的内容也能听明白,但是遇到问题时自己无从下手。为了解决这个问题,作者在编写每个案例时,都先对案例进行分析,提高读者分析问题的能力;然后编写代码,并在代码中给出大量注释;最后在案例之后提出一些思考问题,提高读者独立思考问题的能力。

5. 内容安排循序渐进、由易到难

本书内容安排循序渐进、由易到难,教学篇共 10 章。第 1 章介绍 C 语言的基本知识和开发环境的使用。第 2 章介绍 C 语言基本数据类型与表达式。第 3 章介绍 C 语言的输入/输出语句。第 4、5 章介绍 C 语言的控制结构。第 6 章介绍函数和模块化程序设计的思想。第 7、8 章介绍数组和指针的应用。第 9 章介绍结构体和共用体类型。第 10 章介绍 C 程序文件操作。

本书作者具有多年的 C 语言程序设计和相关专业课程的教学经验。本书由李俊主编并进行总体设计,参加本书编写的有厉志红、高立敏、郭宇周、尹胜彬、张小莉等。

由于作者的水平有限,书中错误和不妥之处在所难免,敬请读者批评指正。

本书为教师提供自动评测考试系统和相关教学资源(含电子课件、源代码、习题答案等),请登录华信教育资源网(<http://www.hxedu.com.cn>),注册后进行下载。

作 者

目 录

上篇 教学篇

第1章 C语言概述3	本章小结.....25
1.1 C语言的发展及特点.....3	习题2.....25
1.1.1 C语言的发展.....3	第3章 输入、输出语句26
1.1.2 C语言的特点.....4	3.1 C语言基本语句.....26
1.2 C语言程序的基本结构.....4	3.2 字符数据的输入和输出.....28
1.3 C语言字符集、标识符和关键字.....6	3.3 格式化输入与输出.....29
1.3.1 C语言字符集.....6	3.3.1 格式化输出函数 printf.....29
1.3.2 C语言标识符与关键字.....6	3.3.2 格式化输入函数 scanf.....33
1.4 C语言程序的开发环境.....7	3.4 综合应用.....35
1.4.1 C语言开发过程.....7	本章小结.....36
1.4.2 Microsoft Visual C++ 6.0 集成 开发环境.....7	习题3.....37
本章小结.....10	第4章 选择控制结构38
习题1.....10	4.1 算法.....38
第2章 基本数据类型与表达式11	4.1.1 算法的概念.....38
2.1 C语言的数据类型.....11	4.1.2 算法的组成要素.....39
2.1.1 数据类型概述.....11	4.1.3 算法的描述.....40
2.1.2 整数类型.....12	4.2 if语句.....41
2.1.3 实数类型.....12	4.2.1 单分支的if语句.....41
2.1.4 字符类型.....12	4.2.2 双分支的if语句.....42
2.2 常量与变量.....13	4.2.3 多分支的if语句.....43
2.2.1 常量.....13	4.2.4 if语句的嵌套.....45
2.2.2 变量.....15	4.3 条件运算符.....46
2.3 运算符和表达式.....16	4.4 switch语句.....47
2.3.1 算术运算符和算术表达式.....16	4.5 综合应用.....49
2.3.2 赋值运算符和赋值表达式.....17	本章小结.....51
2.3.3 关系运算符和关系表达式.....19	习题4.....52
2.3.4 逻辑运算符与逻辑表达式.....20	第5章 循环控制结构54
2.3.5 条件运算符、逗号运算符和 求字节运算符.....22	5.1 循环结构算法.....54
2.4 运算符的优先级.....23	5.2 while语句.....55
2.5 数据类型转换.....23	5.3 do...while语句.....57
2.5.1 自动类型转换.....23	5.4 for语句.....58
2.5.2 强制类型转换.....24	5.5 break语句和continue语句.....60
	5.5.1 break语句.....60

5.5.2 continue 语句	61	7.6 综合应用	112
5.6 循环结构的嵌套	62	本章小结	114
5.7 综合应用	63	习题 7	115
本章小结	66	第 8 章 指针	116
习题 5	66	8.1 指针与指针变量	116
第 6 章 函数与宏替换	68	8.1.1 指针的概念	116
6.1 模块化设计	68	8.1.2 指针变量的定义与初始化	117
6.2 函数的定义与调用	69	8.1.3 指针运算	118
6.2.1 函数的定义	69	8.2 指针与数组	121
6.2.2 函数的调用	70	8.2.1 一维数组的指针表示法	121
6.2.3 参数的传递	73	8.2.2 二维数组的指针表示法	124
6.3 函数的递归调用	74	8.3 指针与字符串	127
6.4 变量的存储类型、作用域	77	8.3.1 字符串的指针表示方法	127
6.4.1 变量的作用域	77	8.3.2 字符串数组与指针数组	129
6.4.2 变量的存储类型	80	8.4 指针与函数	131
6.5 宏替换	83	8.4.1 指针作为函数的形参	131
6.5.1 宏替换概述	83	8.4.2 指针型函数	136
6.5.2 文件包含	86	8.5 指向指针的指针变量*	137
6.6 综合应用	87	8.6 综合应用	138
本章小结	89	本章小结	142
习题 6	89	习题 8	142
第 7 章 数组	91	第 9 章 结构体与共用体	144
7.1 概述	91	9.1 结构体类型的定义	144
7.2 一维数组	92	9.2 结构体变量	145
7.2.1 一维数组的定义	92	9.2.1 结构体变量的定义	145
7.2.2 一维数组的初始化	93	9.2.2 结构体变量的使用	147
7.2.3 一维数组的引用	93	9.2.3 结构体变量的初始化	148
7.2.4 一维数组应用	94	9.3 结构体数组	149
7.3 二维数组	98	9.3.1 结构体数组的定义	149
7.3.1 二维数组的定义	98	9.3.2 结构体数组的初始化	150
7.3.2 二维数组的初始化	99	9.3.3 结构体数组的使用	150
7.3.3 二维数组的引用	99	9.4 结构体类型指针	151
7.3.4 二维数组应用	101	9.4.1 指向结构体变量的指针	151
7.4 字符数组与字符串	103	9.4.2 指向结构体数组的指针	153
7.4.1 字符数组的定义与初始化	103	9.5 结构体与函数	154
7.4.2 字符串的概念及存储	103	9.5.1 结构体变量作为函数参数	154
7.4.3 字符数组的输入输出	104	9.5.2 结构体指针变量作为函数参数	155
7.4.4 字符串处理函数	105	9.5.3 函数的返回值为结构体类型	156
7.4.5 字符数组应用	108	*9.6 链表	157
7.5 数组作为函数参数	109	9.6.1 链表概述	157

9.6.2 内存动态管理函数	158	10.2.2 文件的打开	176
9.6.3 链表的基本操作	160	10.2.3 文件的关闭	177
9.7 共用体类型	165	10.3 文件读写函数	178
9.7.1 共用体类型与共用体变量	165	10.3.1 读写文件字符函数	178
9.7.2 共用体变量的使用	166	10.3.2 读写文件字符串函数	180
9.8 综合应用	169	10.3.3 格式化读写	182
本章小结	172	10.3.4 块读写	183
习题 9	173	10.4 文件定位和随机读写	185
第 10 章 文件	174	10.5 文件检测函数	187
10.1 文件概述	174	10.6 综合应用	188
10.2 文件的打开和关闭	175	本章小结	191
10.2.1 文件指针	175	习题 10	192

下篇 学习篇

第 11 章 上机实践	195	第 13 章 全国计算机等级考试二级试题	
11.1 熟悉 C 语言程序开发环境	195	汇编及参考答案	265
11.2 C 语言基本数据类型与表达式	198	13.1 2008 年 4 月全国计算机等级考试	
11.3 C 语言输入/输出语句	199	二级 C 语言笔试试题	265
11.4 选择控制结构	200	参考答案	273
11.5 循环控制结构	203	13.2 2008 年 9 月全国计算机等级考试	
11.6 函数与宏替换	205	二级 C 语言笔试试题	274
11.7 数组	209	参考答案	282
11.8 指针	214	13.3 2009 年 3 月全国计算机等级考试	
11.9 结构体与共用体	221	二级 C 语言笔试试题	283
11.10 文件	225	参考答案	291
第 12 章 练习题	228	13.4 2009 年 9 月全国计算机等级考试	
12.1 C 语言程序设计概述	228	二级 C 语言笔试试题	292
12.2 数据类型、运算符与表达式	228	参考答案	300
12.3 顺序结构	230	13.5 2010 年 3 月全国计算机等级考试	
12.4 选择结构	232	二级 C 语言笔试试题	301
12.5 循环结构	236	参考答案	311
12.6 函数与宏替换	240	13.6 2010 年 9 月全国计算机等级考试	
12.7 数组	243	二级 C 语言笔试试题	312
12.8 指针	247	参考答案	321
12.9 结构体与共用体	253	附录 A ASCII 码表	322
12.10 文件	256	附录 B C 语言运算符优先级和结合方向	323
12.11 综合习题	259	附录 C C 语言常见的出错信息	324
		参考文献	328

上 篇
教 学 篇

第1章 C语言概述

在众多的程序设计语言中，C语言作为一种高级程序设计语言，具有方便性、灵活性和通用性等特点，同时，它还向程序员提供了直接操作硬件的功能，兼具低级语言的特点，适合各种类型软件的开发。因此，C语言是深受程序设计人员欢迎的编程语言。

本章主要介绍C语言程序设计的发展及其特点，描述了C语言的基本结构、字符集及C语言程序的开发环境等内容。

本章重点：

- ☒ C语言程序的基本结构
- ☒ Visual C++ 6.0 集成开发环境的使用

1.1 C语言的发展及特点

1.1.1 C语言的发展

C语言是国际上广泛流行的、很有发展前途的计算机高级语言，是一种编译型的语言，其发展是一个充实和完善的过程。

C语言是在B语言的基础上发展起来的，它的根源可以追溯到ALGOL 60。1960年出现的ALGOL是一种面向问题的高级语言，它离硬件比较远，不宜用来编写系统程序。1963年，英国的剑桥大学推出了CPL（Combined Programming Language）。CPL在ALGOL 60的基础上更接近硬件一些，但规模比较大，难以实现。1967年，英国剑桥大学的Martin Richards对CPL做了简化，推出了BCPL（Basic Combined Programming Language）。1970年，美国贝尔实验室的Ken Thompson以BCPL为基础，又做了进一步简化，设计出了简单且接近硬件的B语言（取BCPL的第一个字母），并用B语言写了第一个UNIX操作系统，在PDP-7上实现。1971年，在PDP-11/20上实现了B语言，并写了UNIX操作系统。但B语言过于简单，功能有限。1972年至1973年间，贝尔实验室的D.M.Ritchie在B语言的基础上设计出了C语言（取BCPL的第二个字母）。C语言既保持了BCPL和B语言的优点（精练、接近硬件），又克服了它们的缺点（过于简单、数据无类型等）。最初的C语言只是为描述和实现UNIX操作系统提供一种工作语言而设计的。1973年，K.Thompson和D. M. Ritchie两人合作把UNIX的90%以上用C改写，即UNIX第5版。原来的UNIX操作系统是1969年由美国的贝尔实验室的K.Thompson和D. M. Ritchie开发成功的，是用汇编语言写的。

随着UNIX的日益广泛使用，C语言也迅速得到了推广。1978年以后，C语言先后移植到大、中、小、微型机上。而且此时的C语言出现了不同的版本，并将Brain W. Kernighan和Dennis M. Ritchie合著的名著《C程序设计语言》作为C语言的标准。1983年，美国标准化协会（ANSI）又制定了新的标准，称为ANSI C。现在的C语言已风靡全世界，成为世界上最广泛的几种计算机语言之一。

1.1.2 C 语言的特点

C 语言之所以能被推广并被广泛使用, 概括地说, 主要有如下特点。

1. 简洁紧凑、灵活方便

C 语言只有 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。

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. 数据结构丰富

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, 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 语言是结构化语言

结构化语言的显著特点是代码及数据的分隔化, 即程序的各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、维护及调试。C 语言主要由函数组成, 这些函数可方便地调用, 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 语言语法限制不太严格, 程序设计自由度高

虽然 C 语言也是强类型语言, 但其语法比较灵活, 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特点, 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。

7. C 语言程序生成代码质量高, 程序执行效率高

C 语言程序的执行效率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

可移植性是指, 可以把为某种计算机编写的软件运行在另一种计算机或操作系统上, 如在 DOS 下写的程序, 可以方便地在 Windows 2000 下运行, 这个程序就是一个可移植的程序。C 语言程序具有较高移植性。C 语言不包含依赖硬件的输入/输出机制, 其输入/输出功能是由独立于 C 语言的库函数来实现的, 这样就使 C 语言程序本身不依赖于硬件系统, 也便于在不同的机器系统间移植。

1.2 C 语言程序的基本结构

任何一种程序设计语言都具有特定的语法规则和规定的表达方法。一个程序只有严格按照语言规定的语法和表达方式编写, 才能保证编写的程序在计算机中能正确地执行, 同时这样也便于阅读和理解。为了了解 C 语言的基本程序结构, 先介绍一个简单的 C 程序。

【例 1.1】已知两个整数 5 和 7，求这两个数的乘积，并将结果显示出来。

```
#include <stdio.h>           //标准输入输出头文件
main()                       //主函数
{
    void OutStar();          //声明函数
    int a,b,c;               //定义 3 个整型变量
    a=5;b=7;                 //变量赋值
    c=a*b;                   //算术运算并赋值
    OutStar();               //调用 OutStar 函数
    printf("c=%d\n",c);      //输出结果
    OutStar();               //调用 OutStar 函数
}
void OutStar()               //定义 OutStar 函数，void 指定该函数不返回值
{
    printf("\n*****\n");
}
```

通过上面的例子，可以看出以下几点。

1. C 语言程序由函数组成

C 语言程序为函数模块结构，所有 C 语言程序都是由一个或多个函数构成的，其中必须且只能有一个主函数 `main()`。程序从主函数开始执行，当执行到调用函数的语句时，程序将转入被调用的函数中执行，执行结束后，再返回主函数中继续执行，直至程序执行结束。C 程序的函数由编译系统提供的标准函数（如 `printf`、`scanf` 等）和由用户自己定义的函数（如 `OutStar()`）组成。

函数的基本形式如下：

```
返回值类型 函数名（形式参数）
{
    数据说明部分；
    语句部分；
}
```

其中，函数头包括函数返回值类型、函数名和圆括号中的形式参数，如果函数没有形式参数，圆括号中的形式参数为空（如 `void OutStar()` 函数）。函数体包括函数体内使用的数据说明和执行函数功能的语句，花括号“{”和“}”表示函数体的开始和结束。

2. C 语言注释方法

在 Visual C++ 6.0 中，语句的注释内容可以写在“//”后面，实现单行注释，也可以通过“/*”和“*/”，实现多行注释。

注释语句不参与程序的编译和运行，只起到说明的作用，提高程序的可读性。

3. C 语言的基本语法规则

- ① 在 C 语言中，语句的结束标记为分号“;”，每条语句都必须以分号“;”结束。
- ② C 语言中严格区分大小写字母，如 `OutStar()` 和 `outstar()` 将被 C 语言视为不同的函数名称。

4. 用预处理命令 `#include` 可以包含有关的文件信息

C 语言提供了多个头文件，这些头文件分类包含了各类标准函数的原型说明，需要用到某些标准库函数时，只需将对应的头文件用 `#include` 语句包含在程序的首部就可以直接使用了。头文件的扩展名一般为“.h”，头文件名称可以用“<>”括起来，也可以使用双引号“”引起来。

1.3 C 语言字符集、标识符和关键字

1.3.1 C 语言字符集

任何一种程序设计语言所使用的字符都是固定、有限的。要使用一种程序设计语言编写程序，必须符合该语言的规定的、能被计算机系统识别的字符。

C 语言中规定的字符集包括英文字母、阿拉伯数字及其他一些符号，具体归纳如下。

- ① 英文字母：大小写各 26 个，共 52 个。
- ② 阿拉伯数字：0~9，共 10 个。
- ③ 下画线：_。
- ④ 其他特殊符号：主要指运算符和其他符号。
 - 括号（6 个）：(,), [,], {, }。
 - 算术运算符（7 个）：+, -, *, /, %, ++, --。
 - 关系运算符（6 个）：>, <, ==, !=, >=, <=。
 - 逻辑运算符（3 个）：&&, ||, !。
 - 位操作运算符（6 个）：&, |, ~, ^, <<, >>。
 - 赋值运算符（11 个）：=, +=, -=, *=, /=, %=, &=, |=, ^=, >>=, <<=。
 - 条件运算符：?:。
 - 逗号运算符：,。
 - 指针运算符（2 个）：*, &。
 - 求字节数运算符：sizeof。
 - 特殊运算符：->, ..。

1.3.2 C 语言标识符与关键字

1. 标识符

所谓标识符，是指常量、变量及用户自定义函数等的名称。在 C 语言中，标识符的定义必须满足以下规则：

- 所有标识符必须由一个字母（a~z, A~Z）或下画线（_）开头。
- 标识符的其他部分可以由字母、下画线或数字（0~9）组成。
- 大小写字母表示不同意义，即代表不同的标识符。
- 标识符只有前 32 个字符有效。
- 标识符不能使用 C 语言关键字。

下面列举几个合法和不合法的标识符：

- 合法的标识符：t1、_t1、t_1、day、IF。
- 不合法的标识符：M.1、1k、m?1、5*a、if。

2. 关键字

所谓关键字，就是已被 C 语言本身使用，不能作其他用途使用的字。例如，关键字不能用做变量名、函数名等。C 语言包含以下关键字：

auto	asm	break	case	cdecl	char
const	continue	default	do	double	else
enum	extern	far	float	for	goto
huge	if	int	interrupt	long	near
pascal	register	return	short	signed	sizeof
static	struct	switch	typedef	union	unsigned
void	volatile	while	_cs	_ds	_es
_ss					

1.4 C 语言程序的开发环境

1.4.1 C 语言开发过程

开发一个 C 语言程序，要经过编辑（程序录入）、编译和连接后才能生成可执行程序，运行可执行程序后输出结果。

1. 编辑

程序员用任意编辑软件（编辑器）将编写好的 C 语言程序输入计算机，并以扩展名为“.c”的文本文件的形式保存在计算机的磁盘上，生成 C 语言源文件。

2. 编译

程序编译是指将编辑好的源文件翻译成二进制目标代码的过程。编译过程是使用 C 语言提供的编译程序（编译器）完成的。不同操作系统下的各种编译器的使用命令不完全相同，使用时应注意计算机环境。编译时，编译器首先要对源程序中的每个语句检查语法错误，发现错误时，就在屏幕上显示错误的位置和错误类型的信息。此时，要再次调用编辑器进行查错修改。然后进行编译，直至排除所有语法和语义错误。正确的源程序文件经过编译后在磁盘上生成目标文件。

3. 连接

程序编译后产生的目标文件是可重定位的程序模块，不能直接运行。连接就是把目标文件和其他分别进行编译生成的目标程序模块（如果有）及系统提供的标准库函数连接在一起，生成可以运行的可执行文件的过程。连接过程使用 C 语言提供的连接程序（连接器）完成，生成的可执行文件存在磁盘中。

4. 运行

生成可执行文件后，即可在操作系统控制下运行。若执行程序后达到预期目的，则 C 程序的开发工作到此完成。否则，要进一步检查修改源程序，重复编辑—编译—连接—运行的过程，直到取得预期结果为止。大部分 C 语言都提供一个独立的开发集成环境，可将上述四步连贯起来。下面主要介绍 Visual C++ 6.0 开发环境的使用，本书实例均在 Visual C++ 6.0 开发环境下开发、调试。

1.4.2 Microsoft Visual C++ 6.0 集成开发环境

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 C++ 软件包中的 Developer Studio 是一个集成开发环境，集成了各种开发工具和 Visual C++ 编译器。程序员可以在不离开该环境的情况下编辑、编译、调试和运行一个应用程序。

1. 创建一个新的 C 语言的文件

(1) 启动 Microsoft Visual C++ 6.0

可以通过“开始”菜单、桌面快捷方式或者快速启动工具栏等方式启动 Microsoft Visual C++ 6.0, 如图 1.1 所示。

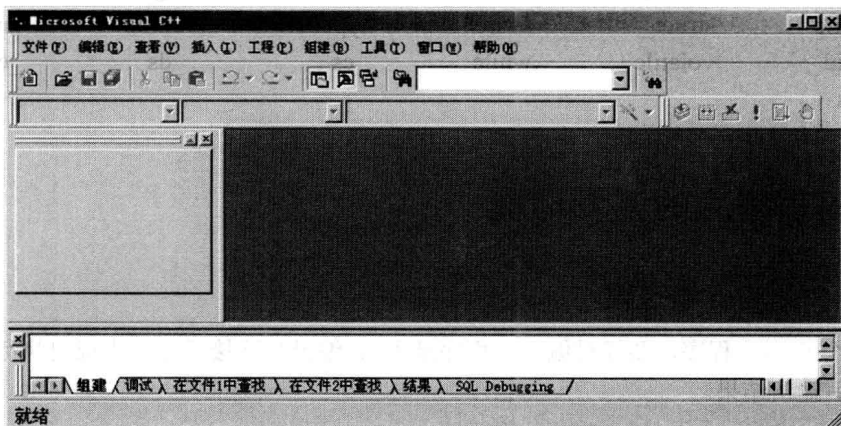


图 1.1 Visual C++ 6.0 启动界面

(2) 新建文件

选择“文件”菜单的“新建”命令，打开如图 1.2 所示的“新建文件”对话框。

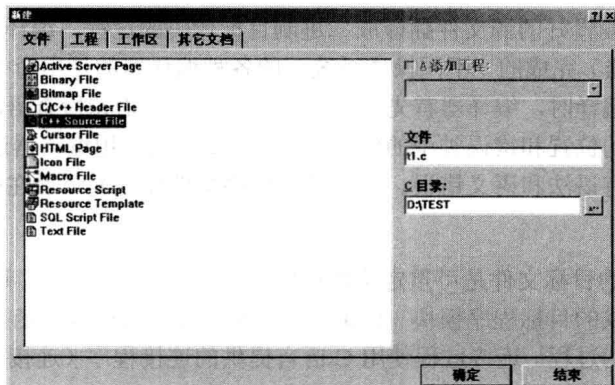


图 1.2 “新建文件”对话框

在图 1.2 所示的“新建文件”对话框中选择“C++ Source File”，在“文件名”框中输入文件名，在“目录”中选择保存该文件的文件夹目录，然后单击“确定”按钮。

注意：文件的扩展名可以输入“.c”或者“.cpp”，默认为“.cpp”。

2. 编写 C 语言源程序

新建一个文件后，可以在 Visual C++ 程序编辑窗口中编写 C 语言源程序，如图 1.3 所示。

3. 编译、连接和运行

编写好源程序后，可以单击“组建”工具栏的  图标，对源程序进行编译；如果编译没有错误，可以单击“组建”工具栏的  图标，对源程序进行连接；如果连接没有错误，可以单击“组建”工具栏的  图标，运行程序。



图 1.3 “程序编辑”窗口

如果编译过程出错，在组建窗口中将显示出错信息，如图 1.4 所示。用户可以双击出错信息定位出错的语句。

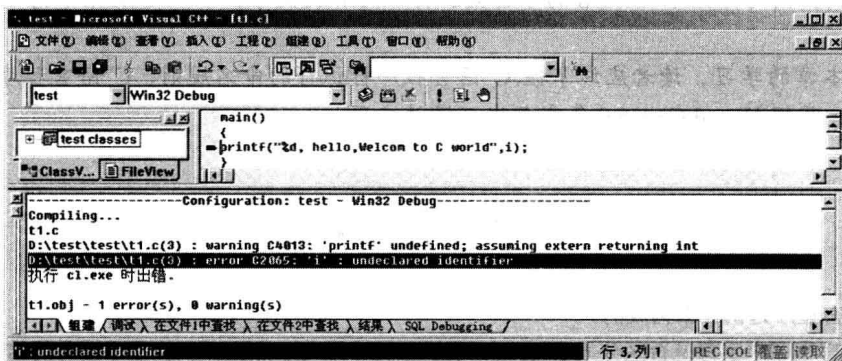


图 1.4 编译出错信息

4. 关闭工作空间

在 Visual C++ 6.0 中，调试运行完一个应用程序后，用户需要通过“文件”菜单的“关闭工作空间”命令关闭当前工作空间，然后才能新建或打开新的 C 语言程序文件。

5. 打开 C 语言源程序文件

(1) 打开文件

可以通过 Visual C++ 6.0 的“文件”菜单的“打开”命令或标准工具栏的图标，打开一个 C 语言源程序文件。

注意：在打开一个 C 语言源程序文件时，应该关闭当前工作空间，否则，连接时会出错。

(2) 编译、连接和运行

编译一个打开的 C 语言源程序文件时，单击“组建”工具栏的图标进行编译时，将弹出如图 1.5 所示的“自动新建工作空间”对话框。

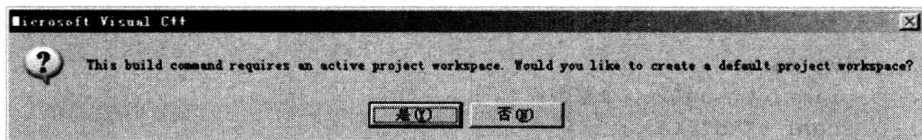


图 1.5 自动创建工作空间

在图 1.5 自动创建工作空间对话框中,单击“是(Y)”按钮后,将弹出如图 1.6 所示的“自动创建工程”对话框。在自动创建工程对话框中,单击“是(Y)”按钮后,即完成对 C 语言源程序的编译,然后通过单击“组建”工具栏的图标和图标,即可以完成连接和运行操作。

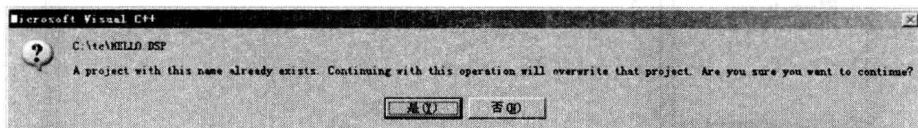


图 1.6 自动新建工程

本章小结

本章介绍了 C 语言的发展及特点,通过一个简单的 C 语言程序事例介绍了 C 语言程序的组成和基本结构,同时介绍了 C 语言的字符集、标识符和关键字,以及 C 语言程序的开发步骤和开发环境。

通过对本章的学习,读者应该掌握 C 语言程序标识符的命名规则、C 语言程序的基本结构和一些约定,掌握 Visual C++ 6.0 集成开发环境的应用。

习 题 1

1. C 语言程序是由什么构成的?
2. C 语言程序中字符是否区分大小写?如何为 C 程序语句添加注释?C 程序语句的结束标志是什么字符?
3. 下列标识符中哪些是 C 语言中的合法标识符?
A123 123A _A123 _123 #A_123
If c.d for FOR a*b
4. 在一个 C 语言程序中,有且只有一个_____函数。
5. 在 Visual C++ 6.0 集成环境下,输入下列程序并查看运行结果。

(1)

```
main()
{
    printf("\n-----\n");
    printf("\n    欢迎进入 C 语言世界!\n");
    printf("\n    C 语言世界很精彩\n");
    printf("\n-----\n");
}
```

(2)

```
main()
{
    int a;
    printf("\n 输入一个整数: ");
    scanf("%d",&a);
    printf("\n 您输入的整数是 %d",a);
}
```