

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21 世纪普通高校计算机公共课程规划教材

# 新概念编程C语言篇

王薇 编著

清华大学出版社



21世纪普通高校计算机公共课程规划教材

# 新概念编程C语言篇

王薇 编著

清华大学出版社  
北京

## 内 容 简 介

《新概念编程 C 语言篇》是对 C 语言的一种解读,本书在解析概念,编写例题、习题方面有着比较新颖的特点。从顺序结构编写计算器封皮开始,到选择结构完善计算器功能,再到用循环结构设计计算器无数次使用的功能,使读者由浅入深地对 C 语言编程有一个深刻的认识,从而达到培养编程兴趣、开发编程能力的效果。

本书对结构体的讲解更是独到,有利于读者的理解和使用。同时,在位运算中,根据编写程序的需求,专门编写了有关文件加密、解密的方法。最后,还编写了项目文件,配合学生课程设计,以及合作编程之需要,同时为配合计算机等级考试提出了许多解题思路和解题方法。

本书语言流畅,善于解析,举例独特,通俗易懂,是初学者最易理解和使用的教科书。本书可作为高等院校本科及自学者的教材。

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

新概念编程 C 语言篇/王薇编著. —北京:清华大学出版社,2010.7

(21 世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-22486-0

I. ①新… II. ①王… 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(2010)第 067645 号

责任编辑:闫红梅 王冰飞

责任校对:梁 毅

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, [jsjic@tup.tsinghua.edu.cn](mailto:jsjic@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:9 字 数:225 千字

版 次:2010 年 7 月第 1 版 印 次:2010 年 7 月第 1 次印刷

印 数:1~3000

定 价:19.00 元

---

产品编号:032008-01

# 出版说明

---

随着我国改革开放的进一步深化,高等教育也得到了快速发展。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度。通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”。通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选编。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人: 梁颖 liangying@tup.tsinghua.edu.cn

# 前 言

---

《新概念编程 C 语言篇》是一本关于 C 语言编程的实用教材。本书立足于为初步接触编程的人员提供指导和帮助,是一本编程的入门教材,适合于高等院校本科生、高等职业技术学院、相关培训机构及自学者使用。

本书作者多年从事非计算机专业计算机语言教学工作,因此深刻了解到非计算机专业学生在学习 C 语言编程的过程中所产生的问题与困惑。书中尽量用简化的语言解释比较复杂的概念及理论。例如,利用两杯饮料互换的实例解释了数字排序的道理,利用中介公司的原理解释困扰学生的指针概念等。

所谓“新概念编程”,旨在提高读者的理解力,培养读者的编程兴趣,积累编程经验,从而激发读者开发程序的创造热情和提高读者的编程能力。本书为初学者提供了很多动手机会以及实现自我的可能。读者在学习过程中,会不知不觉地就掌握了许多诸如计算、统计、文件管理及文件加密、解密技术等知识。

本书作者通过大量阅读国外知名大学的优秀教材及国内著名学者的教材,结合中国学生学习编程的实际情况,力求适合中国学生的学习习惯和中国学生的思维方式。在编写教材时,摆脱以往大量使用英文为变量、文件名取名的方式从而导致中国学生困惑的情况,相反,多处采用汉语拼音形式,让读者一看就懂,一做就明白。对复杂语句都用汉语加以注释,使读者上手快、兴趣浓,让读者可低门槛入门、无障碍学习编程。对问题有分析、有提示,以便于读者理解。

本书 C 语言环境采用的是 Microsoft Visual C++ 6.0,方便读者直接使用汉语,以适合当前诸如期末考试或计算机等级考试系统之需求。

《新概念编程 C 语言篇》一书是作者的一次尝试,因为时间紧,编写过程中肯定会有许多缺点和错误,恳请读者批评指正!

编 者

2010 年 3 月

# 目 录

---

|                                |    |
|--------------------------------|----|
| <b>第 1 章 简述</b> .....          | 1  |
| 1.1 C 语言历史 .....               | 1  |
| 1.2 C 语言的特点 .....              | 1  |
| 1.3 C 语言程序介绍 .....             | 1  |
| 1.4 C 语言环境 .....               | 5  |
| 习题 1 .....                     | 9  |
| <b>第 2 章 数据类型、常量与变量</b> .....  | 10 |
| 2.1 数据类型.....                  | 10 |
| 2.2 常量与变量.....                 | 10 |
| 2.2.1 常量与变量的定义 .....           | 10 |
| 2.2.2 书写规则 .....               | 11 |
| 2.3 常量与变量在各种数据类型中的作用.....      | 11 |
| 2.3.1 整型 .....                 | 11 |
| 2.3.2 浮点型 .....                | 15 |
| 2.3.3 字符型 .....                | 16 |
| 习题 2 .....                     | 18 |
| <b>第 3 章 进制换算、运算及操作符</b> ..... | 19 |
| 3.1 进制换算.....                  | 19 |
| 3.2 运算与操作符.....                | 22 |
| 3.2.1 算术运算符及表达式 .....          | 22 |
| 3.2.2 关系运算符及表达式 .....          | 24 |
| 3.2.3 逻辑运算符及表达式 .....          | 25 |
| 3.2.4 赋值操作符及表达式 .....          | 26 |
| 3.2.5 递增递减操作符及表达式 .....        | 27 |
| 3.2.6 逗号操作符及表达式 .....          | 28 |
| 3.2.7 测试长度操作符 sizeof .....     | 28 |
| 习题 3 .....                     | 31 |

|                      |    |
|----------------------|----|
| <b>第 4 章 程序设计结构</b>  | 32 |
| 4.1 顺序结构             | 32 |
| 4.2 选择结构             | 33 |
| 4.2.1 if()语句的使用      | 34 |
| 4.2.2 三目运算           | 37 |
| 4.2.3 switch()语句的使用  | 38 |
| 4.3 循环结构             | 40 |
| 习题 4                 | 49 |
| <b>第 5 章 数组</b>      | 51 |
| 5.1 整型数组             | 51 |
| 5.1.1 一维数组的定义及使用     | 51 |
| 5.1.2 二维数组的定义及使用     | 53 |
| 5.2 浮点型数组            | 55 |
| 5.3 字符型数组            | 57 |
| 5.4 字符串操作函数          | 59 |
| 习题 5                 | 63 |
| <b>第 6 章 函数</b>      | 65 |
| 6.1 函数的定义            | 65 |
| 6.2 函数的调用            | 66 |
| 6.3 函数的声明            | 67 |
| 6.4 自定义函数种类          | 70 |
| 6.5 函数的嵌套            | 72 |
| 6.6 函数的递归            | 73 |
| 习题 6                 | 74 |
| <b>第 7 章 结构体与共用体</b> | 75 |
| 7.1 结构体              | 75 |
| 7.1.1 定义结构体          | 75 |
| 7.1.2 结构体初始化         | 76 |
| 7.1.3 结构体的应用         | 76 |
| 7.2 共用体              | 79 |
| 习题 7                 | 81 |
| <b>第 8 章 指针</b>      | 82 |
| 8.1 指针、指针变量的定义       | 82 |
| 8.2 指针的特点            | 83 |

|                                      |            |
|--------------------------------------|------------|
| 8.3 指针的运用.....                       | 86         |
| 8.3.1 指针运算 .....                     | 86         |
| 8.3.2 指针与数组 .....                    | 87         |
| 8.3.3 指针与函数 .....                    | 88         |
| 8.3.4 指针与结构体 .....                   | 89         |
| 习题 8 .....                           | 90         |
| <b>第 9 章 预处理及动态分配存储 .....</b>        | <b>91</b>  |
| 9.1 宏定义.....                         | 91         |
| 9.2 宏定义易混淆的问题.....                   | 93         |
| 9.3 文件包含.....                        | 94         |
| 9.4 条件编译.....                        | 95         |
| 9.5 动态分配存储.....                      | 95         |
| 习题 9 .....                           | 98         |
| <b>第 10 章 文件的管理.....</b>             | <b>99</b>  |
| 10.1 文件的建立.....                      | 99         |
| 10.2 相关函数的使用 .....                   | 101        |
| 10.2.1 getw 和 putw 函数的使用 .....       | 101        |
| 10.2.2 fscanf 和 fprintf 函数的使用 .....  | 103        |
| 10.2.3 fgetc 和 fputc 函数 .....        | 104        |
| 10.2.4 fseek、ftell 和 rewind 函数 ..... | 105        |
| 习题 10 .....                          | 107        |
| <b>第 11 章 位运算 .....</b>              | <b>108</b> |
| 11.1 按位与“&”操作符及操作 .....              | 108        |
| 11.2 按位或“ ”操作符及操作.....               | 109        |
| 11.3 异或“^”操作符及操作.....                | 110        |
| 11.4 取反“~”操作符及操作 .....               | 112        |
| 11.5 位移操作 .....                      | 112        |
| 习题 11 .....                          | 114        |
| <b>第 12 章 项目文件 .....</b>             | <b>115</b> |
| 12.1 函数变量类型 .....                    | 115        |
| 12.2 变量的存储类别 .....                   | 118        |
| 12.2.1 静态存储方式与动态存储方式 .....           | 118        |
| 12.2.2 寄存器变量 .....                   | 119        |
| 12.2.3 用 extern 声明外部变量 .....         | 120        |
| 12.3 内部函数和外部函数 .....                 | 121        |

|                     |     |
|---------------------|-----|
| 习题 12 .....         | 123 |
| 附录 A C 语言库函数 .....  | 124 |
| 附录 B C 语言关键字 .....  | 129 |
| 附录 C 操作符优先级列表 ..... | 130 |
| 附录 D ASCII 码表 ..... | 132 |
| 参考文献 .....          | 133 |

## 1.1 C 语言历史

20 世纪 60 年代初期,一种计算机语言 ALGOL 风靡欧洲,这种语言首次使用结构化编程概念受到了许多科学家的推崇。接下来,类似的语言不断被推出。1967 年 Martin Richard 开发了用于编写软件的语言 BCPL (Basic Combined Programming Language), 1970 年 Ken Thompson 利用 BCPL 语言开发出新的语言并命名为 B 语言。利用这种 B 语言,早期版本的 UNIX 操作系统在美国贝尔实验室被开发出来。

1972—1973 年期间,美国贝尔实验室的 Dennis. Ritchie 在 ALGOL、BCPL 和 B 语言基础上设计出了 C 语言软件。直接的效果是 UNIX 操作系统的最新版本不断更新,从而引起了人们对 C 语言的广泛关注,以至于 C 语言作为编程语言历经三十多年经久不衰,同时得到了美国国家标准化协会(ANSI)的发展与扩充。我们今天学习的就是 ANSI C 语言。

## 1.2 C 语言的特点

C 语言具有以下特点:

- (1) 语言表达清晰、灵活,运行速度快。
- (2) 运算功能强,并提供了 32 种运算符,可实现其他高级语言难以实现的运算。
- (3) 数据结构丰富,提供基本数据类型、结构类型、结构体和共用体等。
- (4) 生成代码质量较高,一般只比汇编语言程序的代码效率低 10%~20%。
- (5) 可移植性好,基本上能用于各种操作系统。
- (6) C 语言是由多种函数组成的,学习 C 语言主要研究各种函数之间及变量之间的关系。

## 1.3 C 语言程序介绍

例 1.1 编写一个程序计算矩形面积。

```
include <stdio.h>          /* 调用标准库函数文件 */
main(){                    /* 主函数 */
int a,b,s;                 /* 定义整型变量 */
a = 10;                    /* 给变量赋值 */
```

```

b=6;
s=a*b;           /* 确定变量之间的关系(即定义矩形面积公式) */
printf("%d",s);   /* 输出矩形面积 */
}

```

运行结果:

60

分析以上程序,解释如下:

- (1) “# include <stdio. h>”是命令行,要写在程序的最前面。
- (2) “# include <stdio. h>”是调用库文件,结束时不需要加“;”。
- (3) “<stdio. h>”是 C 语言自身提供的一个标准输入、输出库。(std 表示 standard, i 表示 input, o 表示 output, h 表示 head)

在程序中,main 是主函数的名称,主函数的名称是固定不变的。函数在 C 语言中的书写特点是:由函数名称+“( )”组成。在程序中不管使用了多少个函数,主函数是唯一的。主函数是整个程序的灵魂,所有要解决和实现的功能都要在主函数中进行调用。因为命令、语句很多,所以用函数体“{ }”把语句装起来,这样在书写时我们会发现,主函数里面的内容总是被“{ }”圈在一起,否则运行时就会出错。

“int a, b, s;”(同类型变量在定义类型时,这样书写比较简单)语句含义是定义整型变量。编程时常会遇到许多不同类型的数据,在 C 语言中,会用不同类型去定义变量来满足处理不同类型数据的要求。这些问题会在后面的章节中详细介绍。

“a=10; b=6;”语句是给变量赋值。a=10;就是把整数 10 赋值给变量 a, b=6;就是把整数 6 赋值给变量 b。“=”在这里不是“等于”的意思,而只是一个赋值符号。

在 C 语言中要表达“等于”的意思用“==”符号。一个变量被定义或者被赋值之后也就是一个语句的结束,用“;”符号表示结束。如果忘记书写结束符号,程序在编译时系统会被提醒缺少结束符号“;”。

printf()是标准库文件中的输出函数。在这个函数的“( )”里,由两部分内容组成:

```
printf("      ",      );
```

两部分之间用“,”隔开。前部分确定要输出变量的类型及要说明的部分,用“ ”把内容括起来。后部分书写要输出变量的名称。例如:

```
printf(" s =   %d \n " , s );
```

“s=”是说明部分,“%d”是变量“s”输出时的类型。“s”是变量名称。

C 语言对变量管理非常严格,使用变量之前一定要定义变量的类型,输出之前要确定用什么类型进行输出。可是,同一个变量在定义类型和输出类型时,书写格式是不同的。例如,事先定义表达面积的变量为整型,即“int s;”,但是在输出函数中要确定 s 这个变量用整型输出时,它的书写格式是 %d 而不是 int。

这一点,初学者极容易困惑,这里特别提醒大家,分清定义变量类型和输出变量类型的书写格式,参见表 1.1。

表 1.1 printf 函数输出类型格式表

| 类型与格式  | 说 明                                       |
|--------|-------------------------------------------|
| %d     | 整型输出 (以数据实际长度输出)                          |
| %ld    | 长整型输出                                     |
| %f     | 单精度输出                                     |
| %lf    | 双精度输出                                     |
| %c     | 字符型输出                                     |
| %s     | 字符串输出                                     |
| %o     | 八进制输出                                     |
| %0x    | 十六进制输出 (字母小写)                             |
| %0X    | 十六进制输出 (字母大写)                             |
| %u     | 无符号输出                                     |
| %e     | 数据以指数形式输出, 字母小写                           |
| %E     | 数据以指数形式输出, 字母大写                           |
| %md    | m 是数字, 整型数据在 m 个位内输出, 数据少于 m 位数左补空格       |
| %m.nf  | 浮点型数据在 m 个位内输出, 小数保留 n 位, 数据少于 m 位数左补空格   |
| %ms    | 字符串在 m 个位内输出, 数据少于 m 位数左补空格               |
| %m.ne  | 数据以指数形式在 m 个位内输出, 小数保留 n 位, 数据少于 m 位数左补空格 |
| %-md   | 整型数据在 m 个位内输出, 数据少于 m 位数右补空格              |
| %-m.nf | 浮点型数据在 m 个位内输出, 小数保留 n 位, 数据少于 m 位数右补空格   |
| %-ms   | 字符串在 m 个位内输出, 数据少于 m 位数右补空格               |
| %-m.ne | 数据以指数形式在 m 个位内输出, 小数保留 n 位数据少于 m 位数右补空格   |

### 例 1.2 例 1.1 程序的改进。

```

include <stdio.h>                /* 调用标准库函数文件 */
main(){                          /* 主函数 */
int a,b,s;                       /* 定义整型变量 */
printf("请输入矩形的长和宽: \n"); /* 提醒用户在运行时输入变量的值 */
scanf(" %d %d ", &a, &b);        /* 使用输入函数, 相当于给变量赋值 */
s = a * b;                       /* 确定变量关系 (即定义矩形面积公式) */
printf("s = %d\n", s);          /* 用整型输出矩形面积 */
}

```

运行结果:

请输入矩形的长和宽:

```

10    6
s = 60

```

在改进的程序里使用了输入函数 scanf()。这个函数的括号里面也分两部分。两部分中间用“,”号隔开。前一部分用引号把要输入变量的类型确定, 后一部分书写要输入的变量名称。

```
scanf("    ", &    )
```

值得注意的是, 输入的每个变量名称前要加上“&”符号。这是因为程序在运行的过程

中,要到内存地址中去取输入的值,这是输入函数必须使用的符号,大家在使用中不要忽略。

使用输入函数后会发现:没使用输入函数之前,只能算出一个矩形面积,使用之后,可以算出任意一个矩形面积。scanf()函数输入类型格式见表 1.2。

表 1.2 scanf()函数输入类型格式表

| 类型与格式 | 说 明                    |
|-------|------------------------|
| %d    | 整型输入                   |
| %ld   | 长整型输入                  |
| %hd   | 短整型输入                  |
| %f    | 单精度输入                  |
| %lf   | 双精度输入                  |
| %c    | 字符型输入                  |
| %s    | 字符串输入                  |
| %o    | 无符号八进制整数输入             |
| %0x   | 无符号十六进制数输入(字母小写)       |
| %0X   | 无符号十六进制数输入(字母大写)       |
| %u    | 无符号数据输入                |
| %e    | 数据以实数形式输入,字母小写,作用与%f相同 |
| %E    | 数据以实数形式输入,字母大写,作用与%f相同 |
| %g    | 数据以实数形式输入,字母小写,作用与%f相同 |
| %G    | 数据以实数形式输入,字母大写,作用与%f相同 |
| %md   | 整型数据在 m 个位内输入          |

**提示:** 在 scanf()函数中定义输入类型时,尽量保持类型与类型格式之间不加空格或逗号,如果加了空格或逗号,必须提醒用户在输入时加上相应的空格或逗号,否则,程序运行时会提示错误信息。

例如:

```
scanf("%d%d",&a&b);
```

运行时输入: 6 8

```
scanf("%d %d",&a&b);
```

运行时输入: 6 8

```
scanf("%d,%d",&a&b);
```

运行时输入: 6,8

同时,scanf()的格式部分不可以像 printf()函数那样有说明部分。

例如:

```
scanf("a= %d b= %d",&a&b);
```

这样程序将无法运行,会提示错误信息。

C 语言中需要注释的部分使用“/ \* \* /”。程序在运行时,注释部分会被跳过去,但是不会影响程序的运行。程序的每一条语句,都要经过编译器(compile)编译成二进制的机

器指令。由 C 语言构成的机器指令序列被称为 C 源程序。按 C 语言的语法编写 C 程序的过程,被称为 C 语言代码编写。

C 语言源程序经过 C 语言编译之后会生成一个目标文件,即后缀为 .obj 的二进制文件。最后还要由“连接命令”link 把 .obj 文件与 C 语言提供的各种库函数连接起来生成一个可执行文件,即后缀为 .exe 的文件。如果在 DOS 系统下,此文件在运行时,只要输入文件名字即可运行,而不用输入后缀 .exe。

C 语言的编写者创建了许多库函数。编程者所要做的是了解它们、熟悉它们、正确地运用它们。例如,一般情况下,直接调用标准库函数 #include <stdio.h> 就可以解决一些基本运算。如果想进一步运用 C 语言的代码来解决更多的数学运算,那么可以在程序开头调用数学方面的库函数,就可以进行三角函数、微积分等运算。书写如下:

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

如果想要绘图,还可以调用 C 语言提供的有关绘图方面的库函数。  
代码如下:

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

然后便可以画点、线以及各种图形。因为事先 C 语言软件都已经把函数设计好了,编程者要做的只是调用各种图形函数和按照自己的喜好来使用。有关这方面的内容在后面的章节将会详细介绍。

## 1.4 C 语言环境

本书编程所采用的环境是 Microsoft Visual C++ 6.0。有关使用方法介绍如下。

### 1. 安装 Microsoft Visual C++ 6.0

首先,(以 Windows XP 系统为例)应该在驱动器上建一个文件夹。例如,在 D 驱动器上建一个文件夹,名字叫做“C++环境”,然后把 C++ 文件复制到该文件夹内,双击 setup.exe 安装指令就可以把 C++ 环境安装好。

### 2. 打开 C++ 6.0 环境,编写一个 C 语言程序,然后编译、查看运行结果

图 1.1 就是打开 C++ 环境的初始界面。

### 3. 编写一个 C 语言程序

单击 File 菜单选择 New 命令,如图 1.2 所示。

打开 New 对话框,如图 1.3 所示。在 New 对话框中选择 Files 选项卡,之后选择 C++ Source File,在右侧 File 下面文本框中输入程序名称,别忘记名称之后加写扩展名.c,如果忘记加写扩展名,C++ 环境在编译过程中会自动加上一个扩展名.cpp。这样做不会影响程序的效果,但是表明程序不是正宗的 C 语言程序,而是 C++ 程序,该程序在 C 语言环境中将不能运行。在 Location 下面文本框中输入路径,存放编写的程序。最好事先建立一个文件夹,把自己编写的程序放入这个文件夹内,方便管理和查找。注意,编写的程序不要和安装程序放置在一个文件夹内,这样会很乱,不易区分是安装程序还是自己创建的程序。

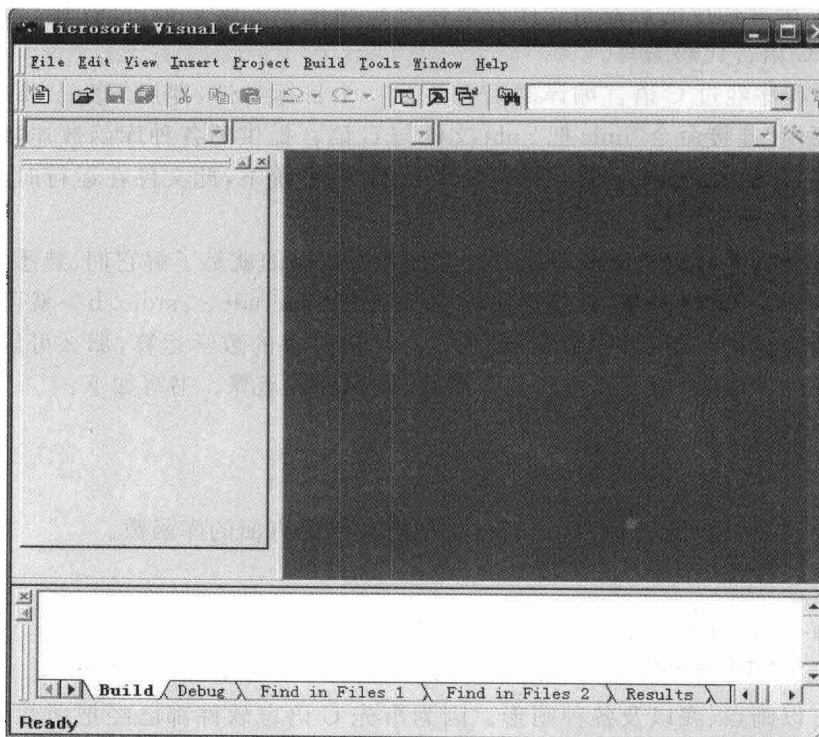


图 1.1 C++环境初始界面

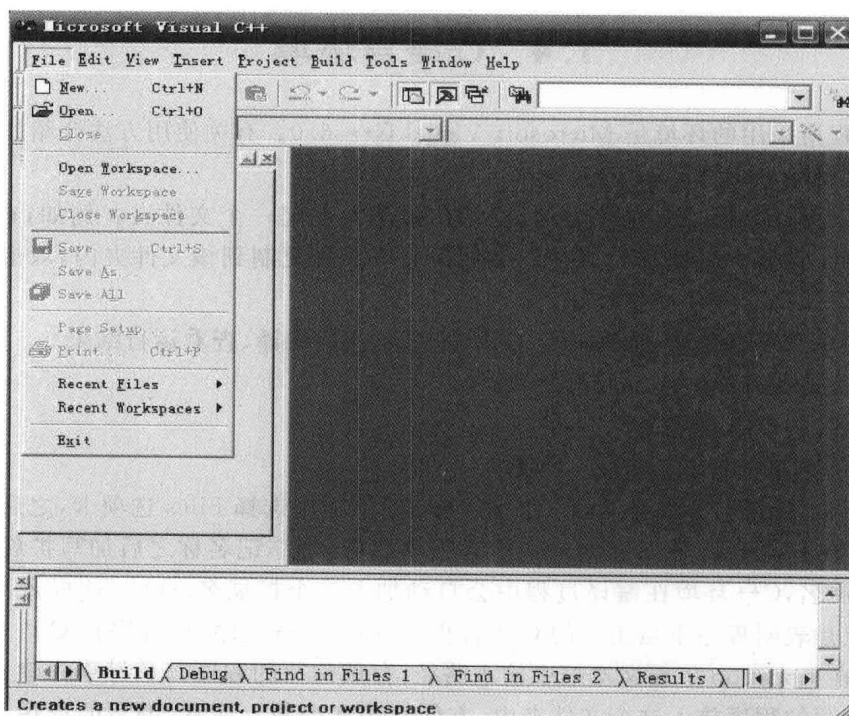


图 1.2 File 菜单

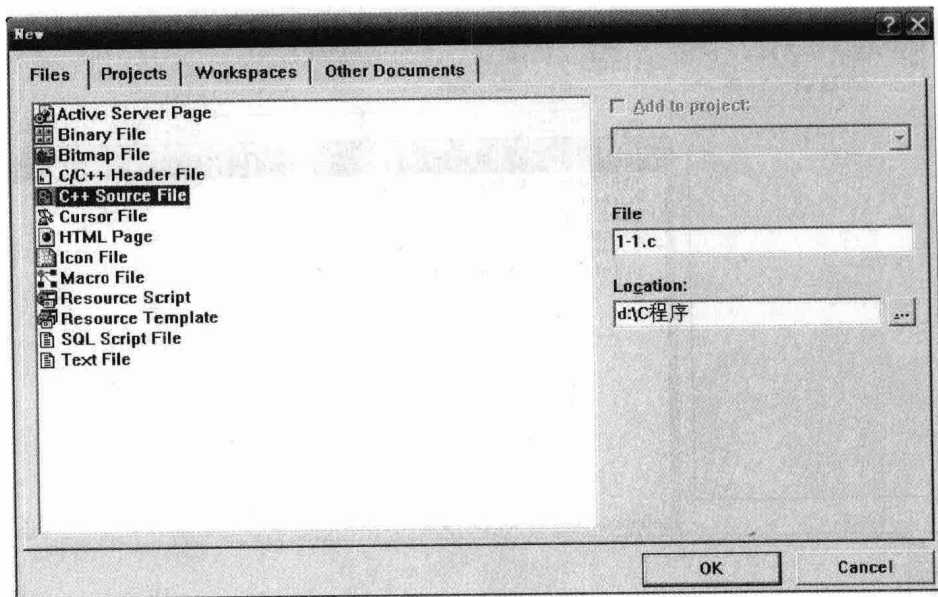


图 1.3 New 对话框

例如,在 D 驱动器上先建一个文件夹,名字叫“C 程序”,在 Location 下面文本框右侧有一个按钮 。单击这个按钮查找自己建立的“C 程序”文件夹。

如图 1.4 所示,在 Drives 下面的下拉列表中找到 D 驱动器,再从上面的列表框中找到“C 程序”文件夹,然后单击 OK 按钮。

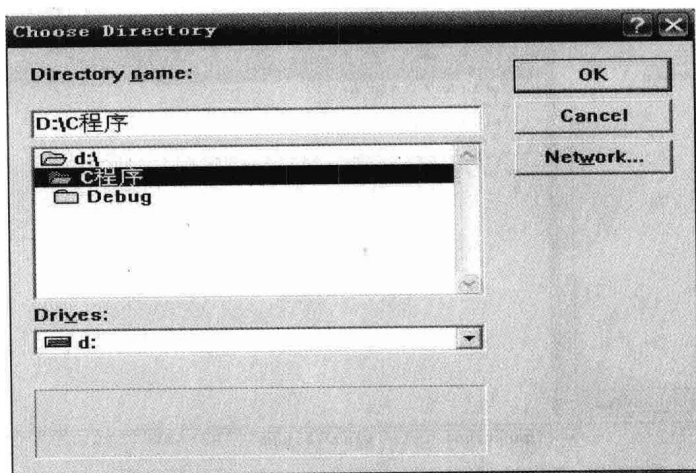


图 1.4 创建“C 程序”文件夹

再单击 New 对话框中的 OK 按钮进入编辑状态,如图 1.3 所示。

在编辑窗口内编写一个简单程序,如图 1.5 所示。

编写结束后,选择菜单命令 Build|Build 进行编译,对于所有的提问都选择 yes 即可。如果顺利,系统不会出现错误提示,否则,系统会在最下面的窗口提示错误信息,如图 1.6 所示。