



普通高等教育“十二五”规划教材
中国科学院教材建设专家委员会规划教材

C语言程序设计

何丽 余平 主编



科学出版社

普通高等教育“十二五”规划教材
中国科学院教材建设专家委员会规划教材

C 语言程序设计

何 丽 余 平 主编

胡 勇 刘云杰 陈 勇 副主编

科 学 出 版 社

北 京

内 容 简 介

本书根据教育部计算机基础课程教学指导分委员会制定的“高级语言程序设计”课程的教学基本要求和全国高等学校非计算机专业学生计算机等级考试教学大纲要求,以 C 语言为主要内容,介绍了结构化程序设计的方法。

本书主要内容有 C 语言概述、C 语言程序基础、结构化程序设计、数组、函数、指针、结构体和共用体、文件。各章均包含了具体的知识内容与操作实例。另外,附录中有 C 语言要使用的 ASCII 码表、运算符表和常用函数介绍。

本书内容丰富、由浅入深,语言通俗易懂,适合于高等学校作为教材使用,也可供从事计算机应用和开发的各类人员使用。

本书每章后面有丰富的习题,供读者练习使用。

本书配有电子教案及丰富的网络教学资源,请访问精品课程网站——计算机基础教学网站 (<http://jsjjc.cqnu.edu.cn>)。

图书在版编目 (CIP) 数据

C 语言程序设计/何丽,余平主编. —北京:科学出版社,2011

(普通高等教育“十二五”规划教材·中国科学院教材建设专家委员会规划教材)

ISBN 978-7-03-032850-2

I. ① C… II. ① 何… ② 余… III. ① C 语言—程序设计—高等职业教育—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2011) 第 240086 号

责任编辑:吕燕新 李太铨 艾冬冬/责任校对:耿耘

责任印制:吕春珉/封面设计:耕者设计工作室

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

铭浩彩色印装有限公司印刷

科学出版社发行 各地新华书店经销

*

2012年1月第 一 版 开本:787×1092 1/16

2012年1月第一次印刷 印张:14

字数:320 000

定价:25.00 元

(如有印装质量问题,我社负责调换(俊杰))

销售部电话 010-62142126 编辑部电话 010-62138978-8102

版权所有,侵权必究

举报电话:010-64030229; 010-64034315; 13501151303

前 言

随着计算机技术的飞速发展,出现了很多高级程序设计语言,其中 C 语言是推广应用最迅速的。而今, C 语言不仅为计算机专业工作者使用,而且为广大计算机应用人员所喜爱和使用,学习 C 语言已经成为社会和个人对掌握程序设计思想的需要。

C 语言是一种理想的结构化程序设计语言,支持自顶向下、逐步求精的程序设计技术,通过 C 语言函数结构可以方便地实现程序的模块化。C 语言功能丰富、表达能力强、程序执行效率高、可移植性好。C 语言既具有高级计算机程序设计语言的特点,同时又具有汇编语言的部分特点,因此它具有较强的系统处理能力。在 C 语言基础上发展起来的面向对象程序设计语言,如 C++、Java、C#等与 C 语言有许多共同的特征。掌握好 C 语言,对学习和应用这些面向对象程序设计语言有极大的帮助, C 语言也为非计算机专业的计算机基础教学提供了一个非常优秀的程序设计平台。

本书根据教育部计算机基础课程教学指导分委员会制定的“高级语言程序设计”课程的教学基本要求,本着系统、准确、合理和实用的基本原则,由长期从事计算机基础课程教学工作的教师,结合丰富的教学经验编写而成。本书具有知识结构合理、条理清楚,描述准确、文字流畅,内容由浅入深、循序渐进,实例丰富、突出应用的特点,且采用案例驱动的教学方式,注重培养学生的实际操作能力,为读者学习如何利用计算机处理信息和掌握程序设计技术打下坚实的基础。

本书内容共分为 8 章。第 1 章介绍 C 语言的发展、基本特点和实验环境,第 2 章介绍 C 语言程序基础,第 3 章介绍 C 程序基本结构,第 4 章介绍数组,第 5 章介绍函数,第 6 章介绍指针,第 7 章介绍结构体和共用体,第 8 章介绍文件的相关知识。全书所有程序均在 Visual C++ 6.0 系统中调试通过。

本书第 1、2 章由重庆师范大学刘云杰编写,第 3 章由重庆师范大学何丽编写,第 4 章及附录 A~C 由重庆师范大学余平编写,第 5 章由重庆师范大学蒋明宇编写,第 6 章由重庆师范大学陈勇编写,第 7、8 章由重庆交通大学胡勇编写。全书由重庆师范大学的何丽、余平主编并负责组织、策划、统稿等工作。

本书在编写过程中,得到了重庆师范大学同仁和科学出版社的大力支持,在此表示最衷心的感谢!

由于编者水平有限,书中难免存在不足或疏漏之处,欢迎广大读者批评指正。

编 者

2011 年 11 月于重庆

目 录

前言

第 1 章 C 语言概述	1
1.1 C 语言出现的历史背景	1
1.2 C 语言的基本特点	2
1.3 C 语言的发展	3
1.3.1 K&R C	3
1.3.2 ANSI C 和 ISO C	4
1.3.3 C99	4
1.3.4 C1X	5
1.4 C 语言的程序结构	5
1.5 C 语言程序上机环境	7
第 2 章 C 语言程序基础	11
2.1 变量和常量	11
2.1.1 变量名	11
2.1.2 基本数据类型	12
2.1.3 常量	13
2.1.4 变量的说明	16
2.2 运算符和表达式	17
2.2.1 算术运算符	17
2.2.2 关系运算符与逻辑运算符	18
2.2.3 自增运算符与自减运算符	18
2.2.4 按位运算符	19
2.2.5 赋值运算符	21
2.2.6 逗号运算符	21
2.2.7 条件运算符	21
2.2.8 运算符优先级与求值顺序	22
2.3 类型转换	23
2.4 预处理	24
2.4.1 宏定义和宏展开	24
2.4.2 文件包含	25
习题	25
第 3 章 结构化程序设计	27
3.1 顺序结构程序设计	27
3.1.1 赋值语句	28
3.1.2 输入/输出函数	28

3.2 选择结构程序设计	31
3.2.1 if 语句	31
3.2.2 条件运算符	36
3.2.3 switch 语句	37
3.2.4 选择结构程序设计举例	38
3.3 循环结构程序设计	40
3.3.1 四种循环结构	40
3.3.2 循环的嵌套	48
3.3.3 break 与 continue 语句	50
3.3.4 循环结构程序设计举例	51
习题	54
第4章 数组	60
4.1 一维数组	60
4.1.1 一维数组的定义	60
4.1.2 数组中元素的引用	61
4.1.3 一维数组的初始化	61
4.1.4 一维数组应用举例	61
4.2 二维数组	65
4.2.1 二维数组的定义	65
4.2.2 二维数组的引用	65
4.2.3 二维数组的初始化	66
4.2.4 二维数组应用举例	66
4.3 字符数组	69
4.3.1 字符数组的定义和赋值	69
4.3.2 字符串的输入和输出	70
4.3.3 字符串常用处理函数	71
4.3.4 字符串应用举例	72
习题	77
第5章 函数	82
5.1 函数简述	82
5.2 函数定义和调用	83
5.2.1 函数的定义和声明	83
5.2.2 函数的调用和参数传递	85
5.2.3 数组作为函数参数	88
5.3 函数的嵌套调用和递归调用	94
5.3.1 函数的嵌套调用	94
5.3.2 函数的递归调用	96
5.4 变量的作用域和生存期	99
5.4.1 变量的作用域	99

5.4.2 变量的生存期	102
5.5 内部函数和外部函数	106
5.5.1 内部函数	106
5.5.2 外部函数	106
习题	106
第6章 指针	115
6.1 地址与指针	115
6.2 指针变量	116
6.2.1 指针变量的定义	116
6.2.2 指针变量的引用	116
6.3 指针变量作为函数参数	122
6.4 指针与数组	126
6.4.1 数组名是一个常量指针	126
6.4.2 数组元素的指针表示	127
6.4.3 指向数组元素的指针变量	129
6.5 数组名作函数参数	134
6.6 字符串与指针变量	140
6.6.1 字符串的表示	140
6.6.2 字符串指针作函数的参数	142
6.6.3 字符串指针变量与字符数组的区别	144
6.7 函数指针变量	144
6.8 指针型函数	145
6.9 指针数组和指向指针的指针	147
6.9.1 指针数组	147
6.9.2 指向指针的指针	151
6.9.3 main 函数的参数	154
6.10 有关指针的数据类型和指针运算的小结	155
6.10.1 常见指针类型	155
6.10.2 指针的运算	156
6.10.3 void 指针类型	156
习题	157
第7章 结构体和共用体	162
7.1 结构体的概念与定义	162
7.1.1 结构体简况	162
7.1.2 定义结构体类型变量的方法	163
7.1.3 结构体变量的初始化与引用	164
7.1.4 结构体变量的输入和输出	165
7.2 结构体数组	165
7.2.1 结构体数组的定义	166

7.2.2 结构体数组的初始化	166
7.2.3 结构体数组的引用	166
7.3 指针与结构体	168
7.3.1 指向结构体变量的指针	168
7.3.2 指向结构体数组的指针	170
7.3.3 用结构体变量和指向结构体的指针作函数参数	172
7.4 动态存储分配—链表	174
7.4.1 链表的建立	174
7.4.2 链表的遍历	177
7.4.3 链表的插入与删除	177
7.5 共用体	180
7.5.1 共用体的概念	180
7.5.2 共用体变量的引用	181
7.5.3 共用体变量的应用	182
7.6 枚举类型	184
7.7 用 typedef 定义类型	186
习题	187
第 8 章 文件	191
8.1 文件概述及文件类型指针	191
8.1.1 文件概述	191
8.1.2 文件 (FILE) 类型指针	192
8.2 文件的打开与关闭	193
8.2.1 文件的打开 (fopen 函数)	193
8.2.2 文件的关闭 (fclose 函数)	194
8.3 文件的读写	194
8.3.1 输入和输出一个字符	194
8.3.2 输入和输出一个字符串	197
8.3.3 格式化的输入和输出	198
8.3.4 按“记录”的方式输入和输出	199
8.4 文件的定位及出错检测	201
8.4.1 文件的定位	201
8.4.2 文件操作的出错检测	201
习题	202
附录 A ASCII 码表	205
附录 B 运算符表	206
附录 C 常用函数表	207
主要参考文献	214

第 1 章

C 语言概述

C 语言是一种通用的程序设计语言，它不受限于任何一种操作系统和机器。由于它很适合用来编写编译器和操作系统等系统软件，C 语言被称为“系统编程语言”。C 语言同样适合于编写不同领域中的大多数程序。现在世界上许多软件都是在 C 语言及其衍生的各种语言的基础上开发出来的。C 语言的特点包括简洁的表达式、流行的控制流和数据结构、丰富的运算符集等。C 语言既具有一般高级语言的特性——语言本身不依赖于特定硬件，程序可读性和可移植性强，又具有低级语言的优势——能实现类似于汇编语言的对内存地址和二进制位的操作功能；同时，C 语言限制少，通用性强，这些都使得 C 语言使用起来更方便、效率更高。

1.1 C 语言出现的历史背景

C 语言最初是由贝尔实验室的 Dennis Ritchie（全名 Dennis MacAlistair Ritchie）为 UNIX 操作系统设计的，并在 DEC PDP-11 计算机上实现。Dennis Ritchie 因此被誉为“C 语言之父”，他还和 Ken Thompson（全名 Kenneth Lane Thompson）一起开发了 UNIX 操作系统。作为 UNIX 操作系统和 C 语言的设计者、开发者，Dennis Ritchie 与同为 UNIX 发明人的 Ken Thompson 一起，于 1983 年获得了由美国计算机学会（ACM）颁发的“图灵奖”。同时，美国计算机学会在 1983 年还设立一个名为“软件系统奖”的新奖项，首届“软件系统奖”的获得者也是 Dennis Ritchie 和 Ken Thompson。1999 年，因为开发了 C 语言和 UNIX 操作系统，两人共同获得了美国国家技术奖章。

C 语言和 UNIX 在发展过程可以说是相辅相成，谈到 C 语言的历史就不得不谈到 UNIX。UNIX 操作系统源自 MAC 项目的 Multics 系统。MAC 项目开始于 1964 年，最早是由麻省理工（MIT）主导、通用电气和贝尔实验室共同参与的一个项目，旨在开发出一个名为 Multics 的系统。Multics 系统的目标是要向大的用户团体提供对计算机的同时访问，支持强大的计算数据存贮，以及允许用户在需要的时候很容易地共享他们的数据。遗憾的是，MAC 并不是一个成功的项目，由于开发费用投入太大，同时也由于它的一些超前的想法和设计使得系统过于庞杂，糅合了太多特性，开发周期过长，系统迟迟不能提交。贝尔实验室于 1969 年退出了该项目，通用电气的计算机部门也于 1970 年被 Honeywell 收购。Honeywell 买下通用电气的计算机部门后接管了 Multics 项目，对它进行了持续的开发和发布。直到 1985 年，Honeywell 才停止了对 Multics 的支持。当时

的大学、企业和政府部门已经有相当多数量的服务器运行着 Multics 系统, Honeywell 停止了对 Multics 的支持后, 用户不得不将服务器迁移到其他的系统上。已知的最后一台运行 Multics 的服务器是在加拿大国防部, 于 2000 年 10 月 30 日关闭。

尽管 Multics 没有成功, 但它在计算机领域中却占有重要的地位, 它的设计思想影响了其后一系列的操作系统, 这其中就包括 UNIX。在 MAC 项目之初, Dennis Ritchie 和 Ken Thompson 作为贝尔实验室的工程师都被派往麻省理工学院参与了该项目。1969 年, 贝尔实验室退出 MAC 项目后, Dennis Ritchie 和 Ken Thompson 返回了贝尔实验室, 回到贝尔实验室后 Dennis Ritchie 和 Ken Thompson 决定将 Multics 进行改写, 让它在较小的规模上实现, 以体现“小既是美”的设计哲学, 这一设计思想也影响到后来对 C 语言的设计, 包括 UNIX 的名字, 最早名为 Unics, 意即 Uniplexed Information and Computing Service, 以对应 Multics 的 MULTiplexed Information and Computing System。

最初的 UNIX 是用汇编语言编写的, 一些应用是由叫做 B 语言的解释型语言和汇编语言混合编写的。B 语言在进行系统编程时不够强大, 所以 Dennis Ritchie 和 Ken Thompson 对其进行了改造, 并与 1971 年共同开发了 C 语言。C 语言的很多重要概念来源于 Martin Richards 开发的 BCPL 语言。1973 年 Dennis Ritchie 和 Ken Thompson 用 C 语言重写了 UNIX。在当时, 为了实现最高效率, 系统程序都是由汇编语言编写, 所以 Dennis Ritchie 和 Ken Thompson 此举是极具大胆创新和革命意义的。用 C 语言编写的 UNIX 代码简洁紧凑、易移植、易读、易修改, 为此后 UNIX 的发展奠定了坚实基础。1977 年, Dennis 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》, 极大地推动了 C 语言的发展, 使 C 语言移植到其他机器时所需要做的工作大大简化。目前, C 语言编译器普遍存在于各种不同的操作系统中。C 语言的设计也影响了许多后来的编程语言, 例如 C++、C#、Java 等。

1.2 C 语言的基本特点

C 语言是面向过程的结构化程序设计语言, 它的设计的目标是能够提供这样一种程序设计语言:

- 具有良好的可移植性。
- 能像汇编语言一样可以直接通过存储器地址访问存储单元。
- 编译后能产生有效率的机器指令, 做到代码优化。
- 只需要尽量小的运行时态库 (run-time library) 就可以支持所编写的程序运行。

遵循这样的设计思路, C 语言既提供了许多类似于汇编语言的低级处理功能, 又保持了良好的跨平台特性, 同时利用 C 语言开发的程序还具有非常高的执行效率。

作为结构化程序设计语言, C 语言提供了对变量作用域 (variable scope) 和递归的支持。在 C 语言中, 所有的可执行代码都包含在名为“函数”的程序模块里。从整体结构来讲, 由 C 语言编写的程序就是由一个一个的函数组成的。尤其需要注意的是, 在 C 语言中, 函数的参数始终是通过值来传递的, 称为“值传递” (pass-by-value)。C 语言

没有直接提供现代编程语言中的“引用传递”(pass-by-reference)，但是可利用在函数参数中传递指针的值来实现“引用传递”。这里特别要强调的是，在函数参数中传递指针仍然是“值传递”，传递的是指针的值，指针的值也就是存储单元的地址。

除此以外，C 语言还包括以下一些基本特点：

(1) C 语言简洁、紧凑、使用方便、灵活。程序书写形式自由，每条语句用分号作为结束标志。程序中字母区分大小写。

(2) C 语言提供了很多类型，其基本类型包括字符、具有多种长度的整数和浮点数等。另外还有通过指针、数组、结构和联合派生的各种数据类型，并允许用户自定义类型。

(3) 运算符丰富，表达式多样化。包括赋值和函数调用都作为表达式，每个表达式都可以作为程序中的一条语句。

(4) 指针提供了与具体机器无关的地址运算。

(5) C 语言采用静态类型系统，但它并不提供对类型的强制检查。这意味着，每个变量虽然有一个类型，但不同类型的变量可以进行相互赋值以及作比较、算术等运算。编译器允许通过隐式或显式类型转换来实现以上情况。例如，一个字符型变量可以接收一个整数值对它赋值，也可以跟一个整数进行大小比较和加减运算。

(6) C 语言为实现结构良好的程序提供了完备的控制流结构，如条件判断(if...else)、多分支(switch)、循环(while、for、do...while)等。

(7) 函数可以返回基本类型、结构、联合或指针类型的值。任何函数都可以递归调用。

(8) 在编译的预处理阶段对程序文本进行替换、包含其他源文件以及进行条件编译。

(9) C 语言本身没有提供输入/输出功能，要实现输入/输出必须通过调用函数来实现。类似地，C 语言也没有直接提供诸如内存块分配、内存回收、文件访问等功能，所有这些高层机制都是通过函数来实现的。

1.3 C 语言的发展

1.3.1 K&R C

1978 年，Brian Kernighan 和 Dennis Ritchie 出版了《*The C Programming Language*》一书。这本书里面介绍的 C 语言标准被程序员们非正式地叫 K&R C 语言，K&R 就是两位作者名字的缩写。在 ANSI（美国国家标准协会）制定 C 语言标准之前，C 语言的定义就是 K&R C。

在 1989 年 ANSI C 标准颁布后的很多年，K&R 依然是许多 C 语言编译器支持的最低标准要求，只要遵循 K&R C 编写的代码，就能够通过编译器的编译。

K&R C 出版后，包括 AT&T 在内的一些厂商在发布 C 语言编译器时，还加入了一些新的特性，包括以下几方面：

- void 函数。

- 函数返回值可以是结构类型或联合类型。
- 可以对结构类型的变量直接赋值。
- 支持枚举类型。

随着C语言的流行,越来越多的厂商针对不同的操作系统都发布了C语言编译器,但是它们或多或少都做了一些自己的扩展,没有一个统一的标准,包括附带发布的C语言标准函数库也没有统一,这就使得C语言的标准化工作越来越迫切。

1.3.2 ANSI C 和 ISO C

1983年,ANSI(美国国家标准协会)成立了一个委员会,其目的是制定“一个无歧义的且与具体机器无关的C语言定义”,同时又要保持C语言原有的“精神”。最后的结果就是1988年完成的ANSI C语言标准。该标准于1989年被ANSI批准通过(ANSI X3.159—1989)。遵循这个标准的C语言通常被称做“ANSI C”或“标准C”,有时候也叫做“C89”。

1990年,ANSI C标准被ISO(国际标准化组织)采纳,形成ISO/IEC 9899:1990标准,这就是被称为C90的C语言标准。C89和C90只在个别规定上有很细微的差别,我们通常将C89和C90认为是同一个标准。

C89标准是基于K&R C制定的,语言本身只做了较少的改动。对大部分程序员来说,最重要的变化是函数声明和函数定义的新语法。在C89中,函数声明可以包含描述函数实际参数的信息,这使得编译器很容易检测到因参数不匹配而导致的错误。

C89标准还对语言做了一些细微的改进,允许对结构变量直接赋值;支持枚举类型;支持单精度的浮点数运算;对预处理操作做了详尽的说明。

该标准还为C语言定义了标准函数库,描述了诸如读写文件、格式化输入/输出、内存分配和字符串操作等函数。还定义了一系列的标准头文件,它们为访问函数声明和数据类型声明提供了统一的标准,确保了兼容性和可移植性。

C89标准被当前的绝大多数编译器所支持。目前大部分的C语言程序也都是基于C89标准来编写的。

1.3.3 C99

在ANSI标准确立后,在很长一段时间内C语言的规范没有大的变动。1995年,针对C90的标准修正案发表,修正了C90中的细节,并增加了对国际字符集的支持。1999年,ISO发布了新的C语言标准ISO/IEC 9899:1999,也就是通常所说的C99。

C99的特性如下:

- 增加了对编译器的限制。比如源程序每行要求至少支持到4095字节,变量名函数名的要求支持到63字节(extern要求支持到31字节)。
- 增强了预处理功能。
- 支持//开头的单行注释。
- 增加了新关键字 restrict, inline, _Complex, _Imaginary, _Bool。
- 支持 long long, long double _Complex, float _Complex 等类型。

- 支持不定长的数组，即数组长度可以在运行时决定。比如利用变量作为数组长度。声明时使用 `int a[var]` 的形式。不过考虑到效率和实现，不定长数组不能用在全局或 `struct` 与 `union` 里。
- 修改了 `/`、`%` 处理负数时的定义，这样可以给出明确的结果。例如在 C89 中有些编译器 `-22/7=-3`，`-22%7=-1`，有些编译器 `-22/7=-4`，`-22%7=6`，而 C99 中明确为 `-22/7=-3`，`-22%7=-1`，只有一种结果。
- 取消了函数返回类型默认为 `int` 的规定。

现在编译器大多数只部分支持了 C99。

1.3.4 C1X

随着 C++0X 标准的制定工作开展，为了增强对 C++ 的兼容性，对 C 语言标准新的修订工作也于 2007 年开始，这就是被大家称为 C1X 的标准。

C1X 中增加的内容包括：支持匿名结构；增强对 Unicode 的支持；提供对泛型宏的支持，允许宏根据参数的类型来生成不同的内容；提供对多线程的支持；提供边界检查功能等。

1.4 C 语言的程序结构

学习程序设计语言的目的就是要使用它来编写程序。从编写 C 语言的源程序开始，到它最后生成可以执行的程序，要经历编译和链接两个阶段，如图 1-1 所示。先选择编辑器书写 C 语言源代码，并以文本的方式保存为扩展名为 `.c` 的源程序文件。然后通过编译器将文本方式的源文件编译成二进制的机器指令，并生成扩展名为 `.obj` 的目标文件。最后再通过链接器，将本程序的目标文件和要用到的库文件以及要用到的其他目标文件一起链接起来，生成扩展名为 `.exe` 的可执行文件。



图 1-1 生成程序的过程

下面从书写一个最简单的 C 语言程序入手来分析 C 语言程序的基本构成。

```

#include <stdio.h>

void main()
{
    printf("Hello, the world!\n");    /*打印输出*/
}
  
```

该程序通过编译、链接生成可执行程序后，执行该程序会在屏幕打印输出字符串：

```
Hello, the world!
```

下面对本程序涉及内容做些说明。一个C语言程序，无论其大小，都是由函数构成的。一个主程序中只能有一个main函数。main是一个特殊的函数名，它代表了程序的入口地址，程序在执行时要从main函数的起点开始执行。

在main函数之前，有一条预处理命令#include <stdio.h>用于告诉编译器在本程序中要包含输入/输出库的头文件信息。需要注意的是，预处理命令并不会在程序运行的过程中执行，而是在编译阶段由编译器来处理。程序在执行时，始终从main函数的起点开始执行，而不管它处于程序中的什么位置。

函数中的语句用一对花括号括起来，本程序中main仅包含下面一条语句：

```
printf("Hello, the world!\n");
```

每条语句用分号作为结束标志。上面这条语句调用了一个名为printf的函数。C语言本身没有输入/输出语句，输入和输出都是由库函数scanf和printf等函数来完成。调用函数时，只需要使用函数名加上用圆括号括起来的参数表就可以了。上面这条语句将字符串"Hello, the world!\n"作为参数调用printf函数。printf是一个用于打印输出的函数，在标准输入/输出函数库中。因此，在使用它之前要用#include <stdio.h>包含标准输入/输出库的头文件，以便获得该函数的声明信息。

用双引号括起来的字符序列被称为字符串或字符串常量，如"Hello, the world!\n"就是一个字符串。另外，在C语言中字符\n表示换行。在printf函数的参数中，只能用\n表示换行，不能用程序语句中本身的换行来代替\n，例如：

```
printf("Hello,  
the world!\n");
```

在程序执行后，并不能在屏幕中输出：

```
Hello,  
the world!
```

相反，它只会使该程序在编译时产生一条错误信息。如果要想产生如上所述的输出效果，可以使用：

```
printf("Hello, \nthe world!\n");
```

在printf函数的参数所接收到的字符串中，只要一遇到\n就会产生一个换行。

另外注意到，在printf("Hello, the world\n");的右边还出现了一个/*打印输出*/。C语言中可以使用/*.....*/对程序中的任何部分作注释。注释中的内容不会被编译成可执行代码，只是在源程序中起到注释的作用。在编译时，编译器遇到/*.....*/就会把/*.....*/里的内容忽略掉。

了解了C语言源程序的基本结构后，可知在编写C语言程序的时候一般包括以下几个部分：

(1) 分析问题。仔细分析所要解决的具体问题，明确解决问题的具体方法和步骤，抽象出问题的数学模型。

(2) 确定数据结构和算法。确定所用的数据结构，选择一种算法描述方式将解决问题的步骤明确化。

(3) 编辑程序。选择编辑环境编写程序，一般选择在 C 语言编程环境中进行编辑，编辑好的文件称为源程序文件（文件的扩展名为.c）。

(4) 编译程序。源程序文件不能直接运行，必须用 C 语言的编译程序进行编译，生成计算机能识别的二进制代码组成的目标文件（文件的扩展名为.obj）。在编译过程中进行语法检查，如果源程序中存在错误，会给出相应错误提示，例如“f.obj - 1 error(s), 2 warning(s)”，只有当源文件中不存在错误时才生成目标文件。

(5) 调试运行。程序通过编译后还不可直接运行，需要将程序与编程环境中的库函数进行链接（link），形成可执行文件（文件的扩展名为.exe）。

运行可执行文件，查看运行结果。如果源程序中存在语义上或设计上的错误，运行结果就会不正确，需要重新检查修改源程序、再编译、再运行，直至运行结果正确，这个过程就称为调试。

调试的方法主要有单行调试和设置断点调试。单行调试也称为单步跟踪，一般用于程序行数较少、程序结构较简单的程序；断点调试一般用于较复杂的程序，在设置断点后，程序运行到断点所在位置会自动停止，这时可以通过查看变量的值来判断程序有无错误。

(6) 保存文件。编辑源程序的过程中随时可以保存文件，调试运行程序前或退出 C 语言环境前必须保存文件，以便以后对源程序进一步维护和修改。

1.5 C 语言程序上机环境

启动 VC 6.0 后进入的界面如图 1-2 所示。

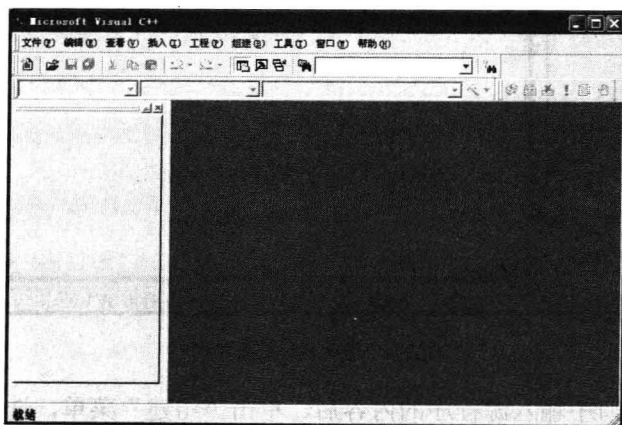


图 1-2 VC 6.0 的界面

然后将写好的程序输入到计算机中，采用的方法是：单击“文件”菜单，选择“新建”命令，则弹出图 1-3 所示的界面。

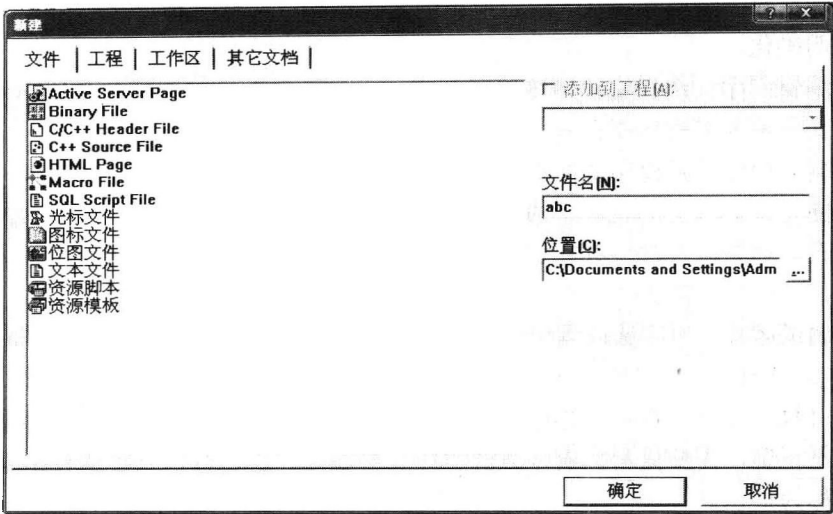


图 1-3 VC 6.0 的“新建”界面

选择“文件”选项卡，再选择 C++ Source File ,在文件名位置处输入源程序的名字，位置处选择保存的位置，单击“确定”按钮后进入图 1-4 所示的界面。

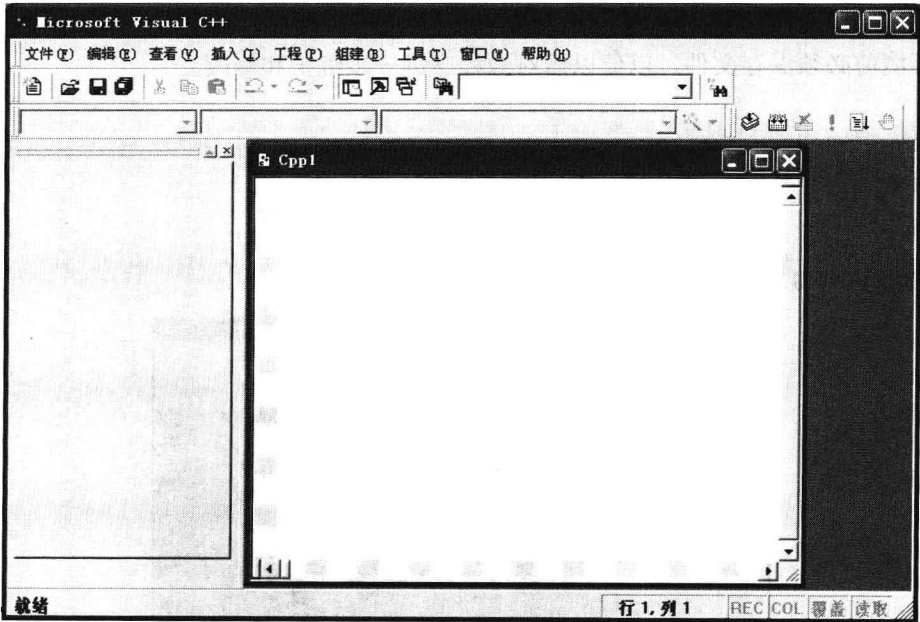


图 1-4 源程序编辑界面

在中间编辑窗口中输入源程序的内容后，单击“组建”菜单，选择“编译”命令，或单击工具栏上箭头所指的按钮后出现图 1-5 所示的界面。



图 1-5 选择“编译”命令后

再单击“组建”菜单，选择“执行”命令，或单击工具栏上箭头所指的按钮，如图 1-6 所示，之后出现图 1-7 所示的界面。

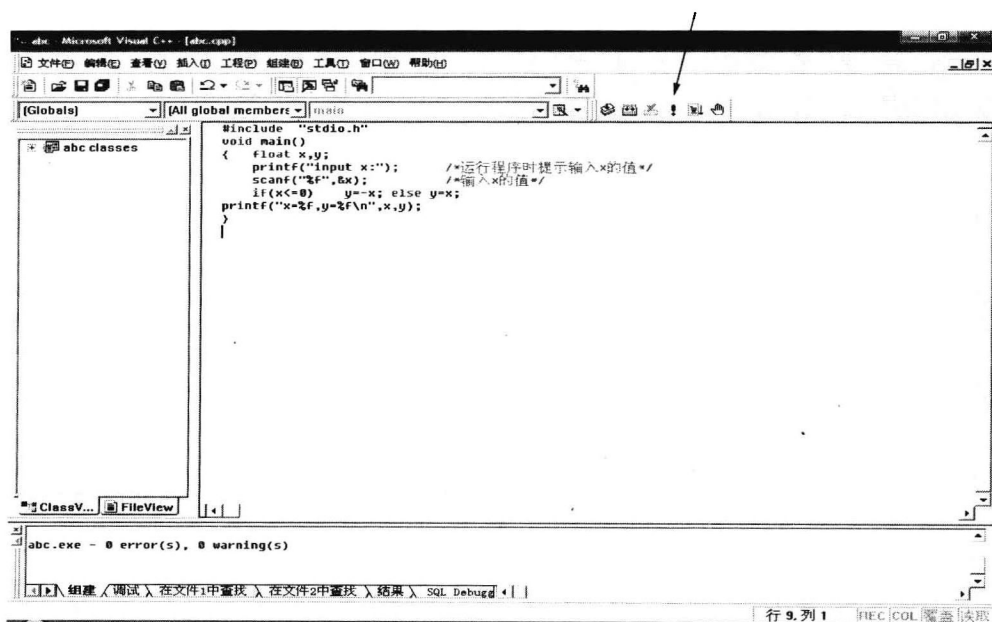


图 1-6 选择“执行”命令