



普通高等教育“十一五”国家级规划教材 计算机系列教材

C语言 程序设计教程

李含光 郑关胜 编著

清华大学出版社





普通高等教育“十一五”国家级规划教材 计算机系列教材

李含光 郑关胜 编著

C语言 程序设计教程

清华大学出版社

北京

内 容 简 介

C 语言是国内外广泛使用的计算机语言,是理工科学生应该掌握的一种程序设计工具。本书针对高等学校理工科 C 语言程序设计教学和等级考试的要求,结合作者多年的教学实践,经过精心策划、准确定位,以清晰的概念和大量的典型实例,深入浅出地介绍了 C 语言的基本知识、运算符和表达式、常见的算法、流程控制、函数、数组、指针、预编译命令、结构体与共用体、文件操作。同时还讲述了程序设计的思想和规范,并在附录中提供了上机实验指导。

本书结构合理、逻辑性强、通俗易懂,不仅可作为计算机专业和其他理工科专业的教材,也可作为等级考试和自学的参考书。

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 / 李含光, 郑关胜编著. —北京: 清华大学出版社, 2011.1

ISBN 978-7-302-24471-4

I. ①C… II. ①李… ②郑… 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 265238 号

责任编辑: 魏江江 李 晔

责任校对: 白 蕾

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

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京富博印刷有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 21.25 字 数: 519 千字

版 次: 2011 年 1 月第 1 版 印 次: 2011 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 35.00 元

产品编号: 038162-01

前言

C 语言是计算机软件开发领域中最流行的程序设计语言之一,它不仅语法简洁,结构清晰,而且还有程序效率高、可移植性好等特点,在系统级软件开发领域深受广大程序员的喜爱。C 语言从诞生到现在已经得到了很大的发展。从最初的为编写 UNIX 操作系统而设计并在实验室内部使用的高级程序设计语言,到现在普遍采用的标准 C, C 语言逐渐走向通用化和标准化。简洁的语法、丰富的表达能力、灵活的运算符和数据结构、可执行代码效率高以及不同系统下的可移植性强等特点,使得 C 语言备受重视,在程序设计语言中有着不可替代的地位。C 语言已经成功帮助人们开发出了很多大型的系统软件和应用软件,著名的 UNIX 操作系统 90%以上的代码是用 C 语言编写的。

C 语言在我国也得到了广泛的应用,许多学校将 C 语言列入学生的必修课程。此外, C 语言也是我国计算机等级考试中可选的高级程序设计语言之一。为了更加有效地开展计算机程序设计语言的教育,不断提高学生对 C 程序设计认识、实践与应用的能力,我校教师在长期的 C 语言课程教学过程中,努力探索、大胆实践,在注重理论知识教学的同时,不断强化实验教学环节,形成了一整套行之有效的教学方法,并确立了具有自身特色的教学思想。作为长期教学与实践经验的总结,编写并出版了本书。

本书系统、全面地介绍了 C 语言程序设计的方法,共分 9 章。主要介绍了 C 语言的基本数据处理、基本输入输出、程序流程控制、函数、数组、指针、预编译命令、结构体与共用体及文件管理等内容。本书内容全面、结构合理,通过实例对 C 语言的语法要点进行了详尽的阐述。本书既可供 C 语言初学者学习使用,也可供有一定经验的软件开发人员学习参考。

本书覆盖了 C 语言的主要语言点。为了使读者能够真正掌握 C 语言,本书在介绍 C 语言的各个语言点时力求做到通俗易懂,并结合了很多实例,使读者能够真正加以运用。在教材编写过程中,我们以创新教学模式、践行“授人以渔”的教学方法,以强调厚基础重实践为原则,以师生易于教、善于教、易于学、乐于学为教与学的目标,对传统教材的体系结构进行了调整,以分散难点,突出重点。

除本教材编写人员外,我校从事 C 语言程序设计教学的课程组教师对教材编写提出了许多富有建设性的意见与建议,并对文本校对等工作做了大量贡献,特别是王定成、聊灵芝和范春年老师,为本书的编写提供了具有指导意义的帮助。对此,我们深表谢意。

另外,本教材的出版得到了南京信息工程大学教材教务处教材基金的大力支持,清华大学出版社为本教材的出版和发行工作给予了大力的支持与帮助,在此一并表示感谢!

由于作者水平所限,书中难免存在不当之处,甚至错误,恳请广大读者批评指正。

作者

2010 年 10 月

目 录

第 1 章 概述	1
1.1 C 语言的发展历史	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的主要标准	2
1.2 C 语言程序的结构	2
1.3 C 语言程序的特点	4
1.4 C 语言程序的开发方法	5
1.4.1 程序	5
1.4.2 程序设计和程序设计语言	5
1.4.3 程序开发方法	5
1.5 C 语言程序上机调试	6
1.5.1 编写源程序	6
1.5.2 源程序文件编译	7
1.5.3 目标文件连接	8
1.5.4 执行	8
本章小结	9
习题 1	10
第 2 章 数据类型、运算符和表达式	11
2.1 C 语言的数据类型	11
2.2 常量与变量	11
2.2.1 标识符命名	11
2.2.2 常量	13
2.2.3 变量	14
2.3 整型数据	14
2.3.1 整型常量	14
2.3.2 整型变量	14
2.4 浮点型数据	16
2.4.1 浮点常量	16
2.4.2 浮点变量	16
2.5 字符型数据	17
2.5.1 字符常量	17

2.5.2	字符串常量	18
2.5.3	转义字符	18
2.5.4	字符变量	19
2.6	运算符和表达式	19
2.6.1	算术运算符	19
2.6.2	自增和自减运算符	20
2.6.3	关系运算符和逻辑运算符	21
2.6.4	位操作符	22
2.6.5	条件运算符	23
2.6.6	逗号操作符	24
2.6.7	赋值运算符	24
2.6.8	复合赋值运算符	25
2.6.9	关于优先级的小结	25
2.7	表达式求值	27
2.7.1	整型表达式	27
2.7.2	赋值表达式	28
2.7.3	关系表达式和逻辑表达式	29
	本章小结	30
	习题 2	31
第 3 章	算法和控制语句	34
3.1	算法初步	34
3.1.1	算法的概念	34
3.1.2	算法的表示	35
3.1.3	算法举例	36
3.1.4	三种基本结构	38
3.2	C 语言的标准输入和输出	39
3.2.1	格式化输入输出	40
3.2.2	其他输入输出	43
3.2.3	C 语言语句	44
3.2.4	顺序结构程序设计	45
3.3	条件语句	46
3.3.1	if 语句	46
3.3.2	if...else 语句	49
3.3.3	if...else if 语句	50
3.3.4	条件语句的嵌套	55
3.3.5	条件语句的应用	58
3.4	多分支语句	61
3.4.1	switch 多分支语句	61



3.4.2 多分支语句的嵌套	63
3.4.3 多分支语句应用	64
3.5 循环语句	66
3.5.1 while 循环语句	67
3.5.2 do...while 循环语句	69
3.5.3 for 循环语句	72
3.5.4 循环语句的嵌套	75
3.6 转移语句	80
3.6.1 goto 语句	80
3.6.2 continue 和 break 语句	80
3.7 综合应用	82
本章小结	90
习题 3	91
第 4 章 函数	103
4.1 函数概述	103
4.2 函数的声明和定义	104
4.2.1 函数的类型说明	104
4.2.2 函数的定义	104
4.3 函数的参数和返回值	105
4.3.1 函数的形式参数和实际参数	105
4.3.2 函数的返回值	107
4.4 函数的调用	108
4.4.1 函数调用的一般形式	108
4.4.2 函数调用方式	110
4.4.3 函数的嵌套调用	112
4.4.4 函数递归调用	114
4.5 局部变量和全局变量	119
4.5.1 局部变量	119
4.5.2 全局变量	121
4.6 变量的存储类型	123
4.6.1 动态存储与静态存储	123
4.6.2 auto 变量	123
4.6.3 static 变量	124
4.6.4 register 变量	125
4.6.5 用 extern 声明外部变量	126
4.7 外部函数和内部函数	127
4.7.1 内部函数	127
4.7.2 外部函数	127

4.8 综合应用	129
本章小结	132
习题 4	133
第 5 章 数组	142
5.1 一维数组	142
5.1.1 一维数组的定义	143
5.1.2 一维数组元素的引用	145
5.1.3 数组的初始化	147
5.1.4 向函数传递一维数组	149
5.1.5 一维数组应用	153
5.2 二维数组	158
5.2.1 二维数组的声明	158
5.2.2 二维数组的初始化	159
5.2.3 二维数组应用	161
5.3 字符数组	164
5.3.1 字符数组的定义	165
5.3.2 字符数组元素的引用	165
5.3.3 字符数组元素的初始化	165
5.3.4 字符串的表示	166
5.3.5 字符数组的输入与输出	168
5.3.6 字符串处理函数	171
5.3.7 字符数组应用	174
5.4 多维数组	175
5.4.1 多维数组的定义	175
5.4.2 多维数组的初始化	176
※5.5 变长数组	177
5.5.1 不指定维长的数据初始化	177
5.5.2 可变长数组及定义	177
5.6 数组应用举例	178
本章小结	181
习题 5	181
第 6 章 指针	189
6.1 指针与指针变量	189
6.2 指针变量的定义与引用	191
6.2.1 指针变量的定义	191
6.2.2 指针变量的引用	192
6.3 指针运算符与指针表达式	194

6.3.1	与指针运算相关的运算符与表达式	194
6.3.2	指针变量作函数的参数	198
6.3.3	返回指针的函数	200
6.4	指针与数组	201
6.4.1	指针与一维数组	201
6.4.2	指针与二维数组	205
6.4.3	数组指针作函数的参数	208
6.4.4	指针与字符数组	213
6.5	指针的地址分配	216
6.6	指针数组	218
6.7	指向指针的指针	222
6.8	指向函数的指针变量	224
6.9	main 函数的参数	225
	本章小结	228
	习题 6	228
第 7 章	预编译命令	235
7.1	概述	235
7.1.1	宏定义	235
7.1.2	无参宏定义	235
7.2	带参宏定义	238
7.3	include 命令	241
7.4	条件编译	242
	本章小结	244
	习题 7	244
第 8 章	结构体与共用体	248
8.1	结构体的声明和结构体变量定义	249
8.1.1	结构体的声明	249
8.1.2	结构体变量的声明	250
8.1.3	结构体成员的引用	252
8.1.4	结构体变量的赋值	255
8.2	结构体数组和结构体指针	255
8.2.1	结构体数组的定义和数组元素的引用	255
8.2.2	结构体数组元素的赋值及引用	256
8.2.3	指向结构体变量的指针	258
8.2.4	结构体数组和结构体指针的应用	260
8.3	结构体与函数	261
8.3.1	结构体变量作为函数参数	261
8.3.2	函数的返回值	263
8.4	链表及其应用	266

8.4.1	链表结点的定义	266
8.4.2	链表的建立	269
8.4.3	链表的插入	271
8.4.4	链表的删除	272
8.5	共用体	273
8.5.1	共用体的定义	273
8.5.2	共用体变量的引用	275
8.6	枚举类型和自定义类型	276
8.6.1	枚举类型的定义和枚举变量的说明	276
8.6.2	枚举类型变量的赋值和使用	276
8.7	类型定义符 typedef	278
8.8	位段	279
8.8.1	位域的定义和位域变量的说明	279
8.8.2	位域的使用	280
8.9	结构体的综合应用	282
	本章小结	284
	习题 8	285
第 9 章	文件系统	290
9.1	文件和流的概念	290
9.1.1	文件的定义	290
9.1.2	流的定义	290
9.1.3	文件的模式	291
9.2	文件的使用	292
9.2.1	文件的打开与关闭	292
9.2.2	文件的读写操作	294
9.2.3	文件指针的定位	302
9.2.4	出错的检测	304
9.3	文件系统应用举例	305
	本章小结	307
	习题 9	308
附录 A	实验指导	311
附录 B	常用字符与 ASCII 码对应表	324
附录 C	运算符优先级和结合性	325
附录 D	C 语言关键字	327
附录 E	C 语言常用函数表	328

第1章

概述

教学目标

- (1) 了解 C 语言的发展历史和标准。
- (2) 掌握 C 语言程序的结构。
- (3) 了解 C 语言的特点。
- (4) 初步认识程序设计方法和程序设计一般步骤。
- (5) 掌握 C 语言程序编译、连接和运行过程。

1.1 C 语言的发展历史

1.1.1 C 语言的发展

C 语言是使用非常广泛的一种计算机编程语言,是公认的最重要的几种编程语言之一,它既可以用来编写系统软件,也可以用来编写应用软件,被称作“低级语言中的高级语言,高级语言中的低级语言”。

C 语言的根源可以追溯到 ALGOL 60, ALGOL 60 结构严谨,非常注重语法和程序结构,但与计算机硬件相距甚远,不适合编写系统软件。1963 年英国剑桥大学在 ALGOL 60 的基础上推出了更接近计算机硬件的 CPL 语言,但由于 CPL 语言太复杂,不容易实现,1967 年剑桥大学的 Martin Richards 对 CPL 语言进行简化,推出了 BCPL 语言。1970 年 Bell 实验室的 Ken Thompson 以 BCPL 为基础,设计了更简单更接近硬件的 B 语言,由于 B 语言是一种解释性语言,功能结构不够强,为了更好地适应系统软件的设计要求,1972 年 Bell 实验室的 Dennis.M.Ritchie 设计了 C 语言,它既保持了 BCPL 语言和 B 语言的优点,又克服了它过于简单、没有数据类型等缺点。1973 年 Ken Thompson 和 Dennis.M.Ritchie 用 C 语言改写了 UNIX 代码,并在 PDP-II 计算机上实现,并奠定了 UNIX 系统的基础。1978 年 Brian Kernighan 和 Dennis.M.Ritchie 合著的 *The C Programming Language* 出版,该书对 C 语言进行了描述,并不断改进和完善,特别是在 20 世纪 80 年代以后,随着个人计算机的普及,C 语言已经成为很多程序员最喜爱的编程语言之一,它被用来编写操作系统、编译程序、数据库管理系统,以及 CAD、过程控制以及图形图像处理等程序。

1.1.2 C 语言的主要标准

C 语言的灵活性、丰富性、可移植性很快得到了普遍的承认，接着适合于各种不同操作系统和不同机种的 C 语言编译系统相继出现。

1. C89 标准（1989）

1982 年美国国家标准局(ANSI)语言标准化委员会成立了一个委员会开始进行 C 语言的标准化工作，它指定了一个标准并在 1989 年被正式采用，即美国国家标准 X3.159—1989，也被称作“ANSI C”。1990 年，国际标准化组织 ISO/JEC JTC1/SC22/WG14 采纳了 C89，做了少量编辑性修改后，以国际标准 ISO/IEC 9899:1990 发布，通常称其为“C90”，它同 C89 基本相同。

2. C95 标准（1995）

WG14 对 C89 做了两处技术修订（缺陷修订）和一个扩充。通过增加新的库函数（iso646.h、wctype.h、wchar.h），改变了 printf()/scanf() 函数的一些新格式代码，上述修改得到的结果，被人们称为“C89 增补 1”或“C95”。

3. C99 标准（1999）

1995 年开始，WG14 开始着手对 C 标准做全面修订，于 1999 年完成并获得通过，形成正式的 C 语言标准，命名为 ISO/IEC 9899:1999，简称“C99”。C99 标准中增加了复数运算，扩展了整数的类型、变长数组、布尔类型，对浮点类型有更好的支持，增加了 C++ 风格的注释。

1.2 C 语言程序的结构

C 语言是一门程序设计语言，要学好 C 语言，就必须用它来编写程序，因此掌握 C 语言程序的结构，是学习 C 语言的关键所在，下面就以一个简单的例子说明 C 语言程序的结构。

例 1.1 在计算机屏幕上输出“Hello, World!”。

```
/* This is first C program */
# include <stdio.h>
int main()
{
    printf(("Hello , World!"));    /* 输出一串字符 */
    return 0;                      /* 向操作系统返回一个数字 0 */
}
```

经过编译连接后，生成可执行的机器代码，运行结果为：

Hello , World!

从上述程序可以看出 C 语言程序由以下几部分组成：

(1) `/* This is first C program */`：在 C 语言中规定，由`/*...*/`组成的是注释，它可以单独占一行，也可以放在 C 语言语句的后面，只是对程序进行必要的说明，并不产生可执行代码，而且也不检查其中字符的拼写错误。

(2) `#include <stdio.h>`：预处理命令（文件包含）。在 C 语言中，以“#”开始的语句一般称作预处理语句，其目的是把文件 `stdio.h` 的内容嵌入到语句位置处。

(3) `int main()`：`main()`是函数，括号里可以包含参数，如果没有参数可以不写或用 `void` 表示，`int` 是 `main()`的返回值。在 C 语言中规定，任何 C 语言程序必须包含一个 `main()`函数，而且只能有一个 `main()`函数。

(4)

```
{  
    ...  
}
```

表示函数的开始和结束，在 C 语言中花括号“{ }”必须成对出现，且匹配。

(5) `printf("Hello , World!")`：在计算机屏幕上输出“Hello , World!”，“;”是语句结束标志。在 C 语言中，每个语句结束，都要在语句末尾加上“;”。另外，由于 C 语言中没有输入输出语句，其输入输出都是用函数来完成的。

(6) `return 0`：是函数 `main()`的返回值，返回 0，表示 `main()`正常结束。

例 1.2 输入两个整数，求它们的乘积。

```
/* 求两个整数的乘积 */  
#include <stdio.h>  
int product (int ,int );           /* 声明后面将要使用函数 product()*/  
int main (void)  
{  
    int x,y,s;                     /* 声明后面使用的变量 x,y,s 是整型的*/  
    scanf("%d %d",&x,&y)           /* 从键盘上输入两个整数 x, y*/  
    s = product(x,y);              /* 调用 product()进行计算，并用 s 接收*/  
    printf("The mul is:%d", s);    /* 输出 s 的值*/  
    return 0;  
}  
int product (int a,int b)          /* 函数 product()的定义*/  
{  
    int mul;                       /* 定义一个整数 mul，用于存放积*/  
    mul = a * b;                   /* 将 a 和 b 求乘积，并把结果送 mul*/  
    return mul;                   /* 返回 mul 的值到调用者*/  
}
```

程序运行结果：

输入 4 □ 5 回车 (□表示空格)：

The mul is:20

从上面的程序可以看出，C 语言的程序结构如下：

(1) 必要的注释语句，可以使程序阅读更清楚，它既可以单独占一行，也可以在一行的后面。

注意：C 语言的注释不能嵌套，即 `/* /*这是程序说明 */ */` 是错误的。

(2) C 语言是由函数组成的，可以由一个或多个函数组成，它是组成 C 语言的基本单位，所以把 C 语言称作函数语言。

(3) 每个 C 语言程序有且只有一个 `main()` 函数，它的位置可以任意，但 C 语言语句的执行总是从 `main()` 函数开始，到 `main()` 函数结束。

(4) 预处理不是 C 语言的语句，它后面不能加 “;” 表示结束。

(5) C 语言没有专门的输入输出语句，是用函数来实现的，如例 1.2 中，`scanf()` 和 `printf()` 就是标准格式化输入输出函数。

1.3 C 语言程序的特点

C 语言能够生存和发展，并具有较强的生命力，主要是 C 语言具有以下特点：

(1) C 语言是比较“低级”的语言：C 语言允许直接访问物理内存，能够进行位 (bit) 操作，这使 C 语言在运行系统程序时，显得非常有效，而原来通常用汇编语言来编写，现在用 C 语言代替汇编语言，使程序员可以减轻负担、提高效率，且程序具有更好的移植性。

(2) 语言简洁、紧凑、灵活：C 语言共有 32 个关键字、9 种控制语句，程序书写自由，主要用小写字母来表示，压缩了许多不必要的成分。

(3) 运算符丰富：C 语言有 34 种运算符，并把括号、赋值、强制类型转换都作为运算符来处理，从而使 C 语言的运算符非常丰富，表达式类型多样化。

(4) 语法限制不太严格，程序自由度大：对数组越界不做检查，由编写者自己保证程序的正确，对变量的使用比较灵活，程序员应该仔细检查编写的程序，不要过分依赖 C 语言编译程序去检查错误。

(5) C 语言是结构化设计语言：C 语言在结构上类似于 ALGOL 60、Pascal 等结构化语言，C 语言的主要成分就是函数，可以对一个程序中的各任务分别定义和编码，使程序模块化，而在函数的外部只需了解函数的功能，而将实现的细节隐藏起来，对于设计得好的函数能够正确地工作而对程序的其他部分不产生副作用。另外 C 语言还提供了多种结构化的控制语句，如循环 `for`、`do...while`、`while` 语句，用于分支控制的 `if`、`if...else`、`switch` 语句等，可以满足结构化程序设计的要求。

(6) C 语言是程序员的语言：很多程序语言是专为某一领域设计的，如 FORTRAN 是为工程师设计的，COBOL 是为商业人员设计的，Pascal 是为教学设计的，BASIC 是为非程序员设计的。但 C 语言是为专业程序员设计的，最初是为编写 UNIX 操作系统而设计的，这是因为 C 语言具有很少的限制、很小的强求、程序设计自由度大，方便的控制结构，独立的函数，紧凑关键字集合和较高的程序执行效率，用 C 语言编写的程序可以获得高效的机器代码，只比汇编语言编写的程序效率低 10%~20%，但其结构又具有 Pascal 的特点。

C语言能得到广泛的使用主要是因为程序员喜欢它,但现在它不再是程序员的专利了,非专业人员经过学习和实践也能熟练掌握,现在很多不同专业的非计算机专业人员都在使用C语言。

1.4 C语言程序的开发方法

1.4.1 程序

所谓程序就是一系列遵循一定规则组织起来完成指定任务的代码或指令序列,一般计算机的程序主要描述两个部分:其一是描述问题所涉及的每个对象及它们之间的关系;其二是描述处理这些对象的规则。

1.4.2 程序设计和程序设计语言

程序设计就是根据所完成任务,设计解决问题的步骤和数据对象之间的关系,然后编写相应的程序代码,并测试该代码的正确性,直到能够得到正确的运行结果为止。通常程序设计应遵循一定的方法和原则,而不是个人随意编写。良好的程序设计风格是程序具备可靠性、可读性、可维护性的保证。在编写程序代码时,程序员必须按照一定的规范来描述问题的解决方案和步骤,这种规范就是程序设计语言。计算机程序设计语言也具有一定的基本规则,具有固定的语法格式、特定的语义和使用环境,并且比通常的语言要求更严格,不能出现二义性。

1.4.3 程序开发方法

用计算机语言解决问题,首先要掌握一定的程序开发方法和步骤,一般程序开发方法如下:

(1) 明确问题的需求:就是把问题陈述清楚,明白解决问题需要什么,消除不重要的方面,集中解决根本问题。

(2) 分析问题:明确问题要处理的数据(即输入)和希望得到的结果(即输出),解决方案的约束条件和任何附加需求。例如,给定一个学生的高等数学、英语、计算机三门课程的成绩,求平均成绩。

要处理的数据(输入):三门课程的成绩。

希望得到的结果(输出):平均成绩。

根据问题的需求,可用公式计算:平均成绩=(高等数学+英语+计算机)÷3。

(3) 设计:开发出一套解决问题的方法和步骤(算法),并验证该方法和步骤可以按预期解决问题,这一步往往是解决问题中最困难的一部分(本书第3章讲述一些常见的算法)。一般来说,不要一开始就解决每一个细小的问题,相反使用自顶向下的设计方法。

即首先列出需要解决的主要步骤或子问题，然后通过解决每个子问题来解决原始问题。用计算机来解决问题，一般至少包含以下子问题：获取数据、执行计算和输出结果。

(4) 实现：将算法写成程序，将每一个算法步骤转化为编程语言的一条或多条语句。

(5) 测试：测试整个程序来验证它是否按预期工作，但是不能只依赖于一次测试的情况，要利用不同数据运行程序若干次来确定程序在每一种情况均能正确工作。

(6) 维护：通过修改程序来改正以前未检测到的错误。

1.5 C 语言程序上机调试

C 语言虽然是一种容易被人们理解的程序设计语言，但它不能直接被计算机执行，必须先把 C 语言的源程序翻译成机器语言才能被直接执行，因此用 C 语言编写程序，必须经过一定的步骤才能完成，下面就以 Microsoft Visual C++ 6.0（简称 VC++6.0）为环境，说明 C 语言的运行过程。

1.5.1 编写源程序

一个 C 语言源程序是以文件为单位进行编译，并且以文本格式存储在计算机的文件系统中的，C 语言的源文件名可以自己定义，文件扩展名（后缀名）为“.C”（VC++6.0 环境下扩展名为“.CPP”），下面利用 VC++6.0 提供的编辑程序，说明将编写的 C 语言程序存入计算机的步骤。

1. 启动 Microsoft Visual C++ 6.0

在 Windows 的“开始”菜单中选“程序”，在 Microsoft Visual C++ 6.0 程序组中选择 Microsoft Visual C++ 6.0，就启动了 Microsoft Visual C++ 6.0，其主界面如图 1-1 所示。

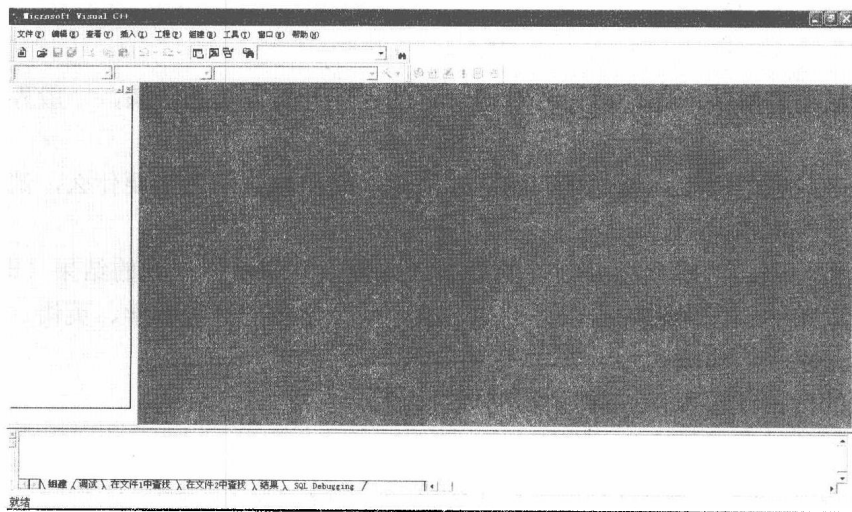


图 1-1 VC++6.0 主界面

2. 进入编辑环境并编写源程序

选择“文件”菜单中的“新建”命令，打开“新建”对话框，选择“文件”选项卡，再选择 C++ Source File，如图 1-2 所示。单击“确定”按钮后进入编辑源程序环境并输入源程序，如图 1-3 所示。

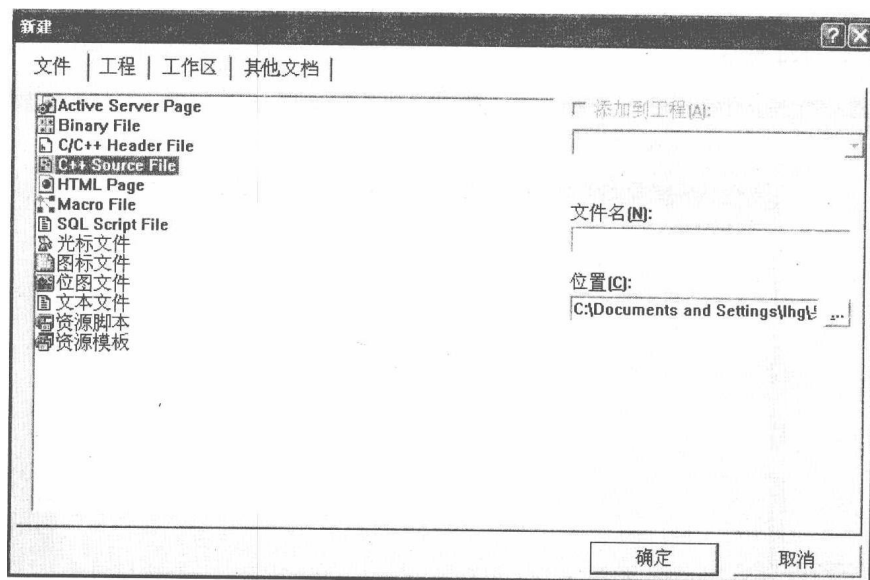


图 1-2 新建 VC++6.0 源程序

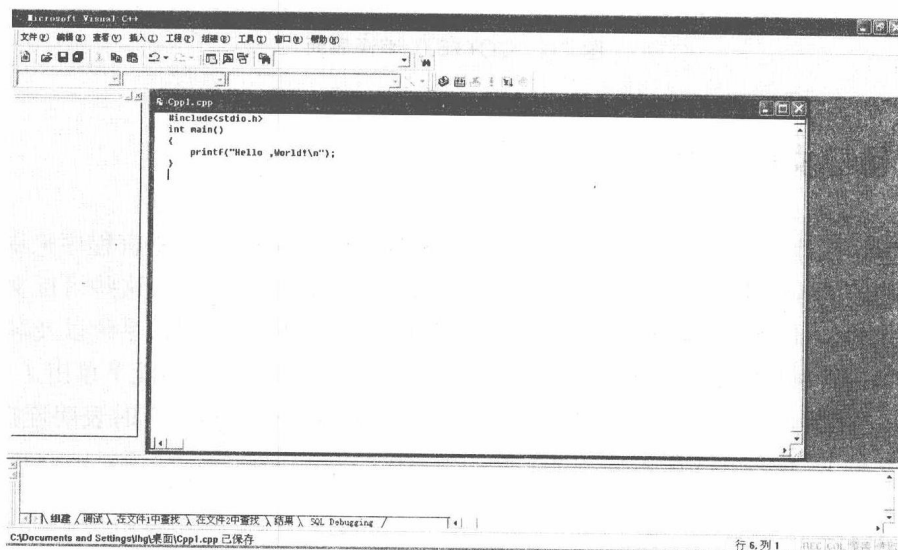


图 1-3 C 语言源程序编辑窗口

1.5.2 源程序文件编译

建立并输入源程序之后就可以用 VC++6.0 提供的编译程序（编译器）对其进行编译，