

高等职业院校国家技能型紧缺人才培养工程规划教材
· 计算机应用与软件技术专业



电子·教育

C语言 程序设计基础

苏传芳 主编 王路群 主审



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

高等职业院校国家技能型紧缺人才培养工程规划教材·计算机应用与软件技术专业

C 语言程序设计基础

苏传芳 主编

王路群 主审

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

开拓一种程序设计课程新的教学模式,让学生在短时间内掌握 C 语言程序设计的方法,是本书的目的。

本书采用以实例为先导、注重实际应用的教学模式。在编写中针对高职院校学生的知识能力和接受水平,采取了精选内容、分散难点、由浅入深的写作方法,通过大量的例题和实用而有趣的实训和课程设计,帮助学生提高兴趣和积极性,以掌握复杂的概念。

在章节的分布上,力求做到合理安排易于接受,为此,本书打破常规将指针的概念分解到各个章节,贯穿全书,既能自然表现指针在程序设计中的应用,又分解了难点,有利于学生对指针概念的掌握。在例题的选取上,采用先易后难、结合实际,注重实训和课程设计与知识点的结合。以提高学习者的动手能力。

本书共 10 章:第 1、2、3 章介绍了 C 语言的基本知识及程序设计结构;第 4、5 章着重讲解了数组和函数的概念;第 6 章对 C 语言的预编译处理及应用进行了分析;第 7、8、9 章结合具体应用介绍了结构体、共同体、位运算及文件的相关内容;第 10 章是综合所学知识,进行一个实用的项目设计。本书的章节内含有实训、课程设计。本书内容实用、通俗易读、体系合理。可作为高等职业院校学生的 C 语言课程的教材,也适用于培训和读者自学。

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 语言程序设计基础 / 苏传芳主编. —北京:电子工业出版社, 2004.9

高等职业院校国家技能型紧缺人才培养培训工程规划教材·计算机应用与软件技术专业

ISBN 7-121-00311-2

I. C… II. 苏… III. C 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 091655 号

责任编辑:赵江晨

印 刷:北京天宇星印刷厂

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

经 销:各地新华书店

开 本:787×1092 印张:18.25 字数:479 千字

印 次:2004 年 9 月第 1 次印刷

印 数:6 000 册 定价:23.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

出版说明

高等职业教育是我国高等教育体系的重要组成部分,也是我国职业教育体系的重要组成部分。社会需求是职业教育发展的最大动力。根据劳动市场技能人才的紧缺状况和相关行业人员需求预测,教育部会同劳动和社会保障部、国防科工委、信息产业部、交通部、卫生部启动了“职业院校制造业和现代服务业技能型紧缺人才培养培训工程”,明确了高等职业教育的根本任务是要从劳动力市场的实际需要出发,坚持以就业为导向,以全面素质为基础,以能力为本位,把提高学生的职业能力放在突出的位置,加强实践教学,努力造就数以千万计的制造业和现代服务业一线迫切需要的高素质技能型人才,并且优先确定了“数控技术应用”、“计算机应用与软件技术”、“汽车运用与维修”、“护理”等四个专业领域,在全国选择确定200多所高职院校作为承担技能型紧缺人才培养培训工程示范性院校,其中计算机应用与软件技术专业79所,软件示范性高职院校35所,数控技术应用专业90所,汽车运用与维修专业63所。为了加快技能型人才培养培训工程,教育部决定,在3~5年内,高职院校学制要由3年逐步改为2年。

为了适应高等职业教育发展与改革的新形势,电子工业出版社在国家教育部、信息产业部有关司局的支持、指导和帮助下,充分调研,探索符合高等职业教育教学模式、教学方法、学制改革的新教材的出版,于2004年4月3~13日在南京分别召开了计算机应用与软件技术、数控技术应用、汽车运用与维修等3个专业的教材研讨会。参加会议的150多名骨干教师来自全国近100多所高职院校,很多教师是双师型的教师,具有丰富的一线教学经验和实践经验。会议根据教育部制定的3个专业的高职两年制培养建议方案,确定了主干课程和基础课程选题共61个;其中,计算机应用与软件技术专业30个,数控技术应用专业12个,汽车运用与维修专业18个。

这批教材的编写指导思想是以两年制高等职业教育技能应用型人才为培养目标,明确了职业岗位对专业核心能力和一般专业能力的要求,重点在于培养学生的技术运用能力和岗位工作能力,并围绕核心能力的培养形成系列课程链路。教材编写注重技能性、实用性,加强实验、实训、实习等实践环节。教材的编写内容和学时数较以往教材有根本的变化,不但对教材内容系统地进行了精选、优化和压缩,而且适当考虑了相应的职业资格证书的课程内容,有利于学生在获得学历证书的同时,顺利获得相应的职业资格证书,增强学生的就业竞争能力。为了突出教学效果,这批教材将配备电子教案,重点教材将配备多媒体电子课件。

这批教材按照两年制高职教学计划编写。第一学期教学所用的基础教材将于2004年9月前出版。第二学期及之后的教材大部分将于2004年12月前出版。这批教材是伴随着高等职业教育的改革与发展而问世的,可满足当前两年制高等职业教育教学的需求,所存在的一些不尽如人意之处,将在今后的教学实践中不断修订、完善和充实。我们将在教育部和信息产业部的指导和帮助下,一如既往地依靠业内专家,与科研、教学、产业第一线人员紧密结合,加强合作,与时俱进,不断开拓,为高等职业教育提供优质的教学资源和服务。

电子工业出版社
高等职业教育教材事业部
2004年8月

前 言

C 语言已经成为各类高等职业院校计算机专业程序设计的入门语言。由于 C 语言具有概念复杂、规则繁多、使用灵活、容易出错等特点,使不少初学者在学习中感到难以在短时间内掌握它。现在,为发展软件产业,教育部批准建立了 35 所示范性高职软件学院,学制为两年。针对没有任何计算机基础知识的初学者,如何让他们在短时间内掌握 C 语言是编写本书的目的。

本书总结了编者多年的高职教学经验,以实例为先导,注重在应用中学习,以有趣的实际应用例子来提高学习者的兴趣,帮助学生理解和接受枯燥的概念和规则。我们在编写中充分考虑到学生的知识能力和接受水平,按照精选内容、分散难点、由浅入深的写作思路,通过大量的例题和实训帮助学生循序渐进地学习,同时配有相应的课程设计。

全书在章节的分布上,力求做到合理安排、易于接受。教过 C 语言课程的老师都知道,指针是学生们最难接受和掌握的内容。为此,本书打破常规将指针的概念分解到各个章节,贯穿全书,既能自然表现指针在程序设计中的应用,又分解了难点,有利于学生对指针概念的理解。

在例题的选取上,先易后难,结合实际,注意实例和实训与知识点和算法的结合,从而提高学生的实际动手能力。

本教材对学时数不作统一要求,根据各院校和学生的实际状况,教师可自行安排。学习过程中要注意实例、知识点、实训的同步进行。根据学时情况,有的实训和课程设计,在教师加以指点后,学生可放到课余时间做。第 10 章内容可根据学时及专业,酌情选择。

本书由安徽电子信息职业技术学院苏传芳主编,武汉软件职业学院王路群主审。本书第 2 章由安徽电子信息职业技术学院李媛媛编写;第 4 章由安徽电子信息职业技术学院杨军编写;第 5 章及附录由安徽电子信息职业技术学院朱正月编写;第 7 章由山东信息职业技术学院季昌武编写;第 8 章和第 9 章由沈阳职业技术学院关智编写;第 1 章、第 6 章和第 10 章由安徽电子信息职业技术学院朱士明编写;第 3 章由苏传芳编写。苏传芳负责全书总体设计和最后修改定稿。

开拓一种新的计算机语言课程的教学模式是本书编写的宗旨。但由于水平有限,编写时间较紧,本书肯定存在许多不足之处,恳请各位专家及读者批评指正。

编 者

2004 年 7 月

目 录

第 1 章 C 语言概述	(1)
1.1 C 语言产生的背景	(1)
1.2 C 语言的特点	(2)
1.3 C 语言的程序结构	(2)
1.4 C 程序上机步骤	(5)
实训 1 认识 C 语言程序	(9)
习题 1	(10)
第 2 章 数据类型、运算符与表达式	(11)
2.1 C 语言的数据类型	(11)
2.2 常量与变量	(12)
2.2.1 直接常量和符号常量	(12)
2.2.2 变量	(14)
2.3 整型数据	(14)
2.3.1 整型常量的表示方法	(14)
2.3.2 整型变量	(15)
实训 2 使用整型数据	(18)
2.4 实型数据	(19)
2.4.1 实型常量的表示方法	(19)
2.4.2 实型变量	(20)
实训 3 使用实型数据	(22)
2.5 字符型数据	(23)
2.5.1 字符常量	(23)
2.5.2 字符变量	(24)
2.5.3 字符串常量	(25)
实训 4 使用字符型数据	(25)
2.6 算术运算符和算术表达式	(26)
2.7 赋值运算符和赋值表达式	(28)
2.8 逗号运算符和逗号表达式	(31)
实训 5 使用运算符和表达式	(32)
2.9 变量的地址和指向变量的指针	(34)
2.9.1 变量的地址	(34)
2.9.2 变量的指针和指向变量的指针变量	(34)
2.9.3 指针运算符和取地址运算符	(35)
实训 6 指针的初步应用	(36)
本章小结	(37)
习题 2	(37)

第3章 基本程序结构	(41)
3.1 程序的3种基本结构	(41)
3.1.1 结构化程序设计	(41)
3.1.2 C语言的语句	(43)
3.2 赋值语句	(44)
3.3 数据的输入/输出	(45)
3.3.1 字符数据的输入/输出函数——getchar()函数/putchar()函数	(45)
3.3.2 格式输出函数——printf函数	(47)
3.3.3 格式输入函数——scanf函数	(50)
实训7 使用输入/输出函数	(53)
3.4 顺序结构程序设计	(56)
3.5 选择结构程序设计	(57)
3.5.1 关系运算符和关系表达式	(57)
3.5.2 逻辑运算符与逻辑表达式	(58)
3.5.3 if语句和switch语句	(60)
实训8 if语句和switch语句的使用	(66)
3.6 循环结构程序设计	(71)
3.6.1 goto语句及goto语句构成的循环	(72)
3.6.2 while语句、do···while语句和for语句	(73)
实训9 while语句、do···while语句和for语句的使用	(79)
课程设计1 模拟ATM取款机界面	(86)
本章小结	(89)
习题3	(89)
第4章 数组	(95)
4.1 一维数组	(95)
4.1.1 一维数组的定义、引用和初始化	(96)
4.1.2 数组与指针	(99)
实训10 一维数组的应用	(101)
4.2 二维数组	(105)
4.2.1 二维数组的定义	(105)
4.2.2 二维数组元素的引用	(106)
4.2.3 二维数组的初始化	(107)
4.2.4 二维数组与指针	(108)
实训11 二维数组的应用	(110)
4.3 字符数组	(113)
4.3.1 字符数组的定义、引用和初始化	(113)
4.3.2 字符串的使用	(114)
4.3.3 字符串处理函数	(116)
实训12 英文打字练习程序	(119)
4.4 指针数组和指向指针的指针	(122)

4.4.1 指针数组的概念	(123)
4.4.2 指向指针的指针	(123)
4.4.3 指针数组作为 main 函数的形参	(124)
实训 13 指针的应用	(125)
课程设计 2 用高斯消去法解线性方程组	(126)
本章小结	(129)
习题 4	(129)
第 5 章 函数	(132)
5.1 函数定义与调用	(132)
5.1.1 函数定义	(132)
5.1.2 函数调用	(135)
5.1.3 函数返回值	(139)
实训 14 建立和使用函数	(140)
5.2 形参与实参	(142)
5.2.1 函数的值调用和引用调用	(143)
实训 15 值参数和地址参数的传递	(144)
5.2.2 函数的参数	(146)
实训 16 函数参数传递的形式	(148)
5.3 返回指针值的函数和指向函数的指针	(150)
5.3.1 返回指针值的函数	(150)
5.3.2 指向函数的指针	(153)
5.4 函数嵌套调用与递归调用	(155)
5.4.1 函数嵌套调用	(155)
5.4.2 函数递归调用	(157)
实训 17 嵌套与递归调用的实现	(158)
5.5 局部变量和全局变量	(161)
5.5.1 局部变量	(161)
5.5.2 全局变量	(162)
5.5.3 变量存储类别	(163)
实训 18 局部变量和全局变量的使用	(167)
5.6 内部函数与外部函数	(168)
实训 19 多文件程序编译运行	(170)
课程设计 3 马跳日字问题的实现	(172)
本章小结	(176)
习题 5	(177)
第 6 章 编译预处理	(180)
6.1 宏定义	(181)
6.1.1 字符串的宏定义	(181)
6.1.2 带参数的宏定义	(183)
6.2 文件包含处理	(184)

6.3 条件编译	(186)
实训 20 定义宏和使用宏	(188)
本章小结	(190)
习题 6	(191)
第 7 章 结构体和共用体	(192)
7.1 结构体	(192)
7.1.1 结构体定义、引用和初始化	(193)
7.1.2 结构体数组和结构体指针	(198)
实训 21 结构体的应用	(202)
7.2 共用体	(209)
实训 22 定义使用共用体	(211)
7.3 枚举类型	(214)
实训 23 定义使用枚举类型	(215)
7.4 类型定义	(216)
课程设计 4 单链表插入、删除、查找等操作的实现	(217)
本章小结	(224)
习题 7	(224)
第 8 章 位运算	(227)
8.1 位运算概念	(227)
8.2 位运算符	(227)
8.3 位段	(231)
实训 24 左右循环移位的实现及位操作	(233)
课程设计 5 使用位运算符解析 IP 地址	(236)
本章小结	(237)
习题 8	(237)
第 9 章 文件	(239)
9.1 文件类型指针	(241)
9.2 文件的打开与关闭	(242)
9.3 文件的读/写操作	(244)
实训 25 文件加密程序的实现及文件的读/写操作	(247)
9.4 文件定位与出错检测	(252)
9.4.1 文件定位函数——fseek()函数	(252)
9.4.2 出错检测函数——ferror()函数	(253)
实训 26 加解密数据库程序及文件定位操作	(253)
9.5 其他文件函数	(255)
课程设计 6 给程序加上行号	(256)
本章小结	(258)
习题 9	(258)
第 10 章 项目实践 文件压缩与解压缩	(260)
10.1 预备知识——数据压缩的原理	(260)

10.2 结构设计..... (263)

10.3 功能函数的实现..... (264)

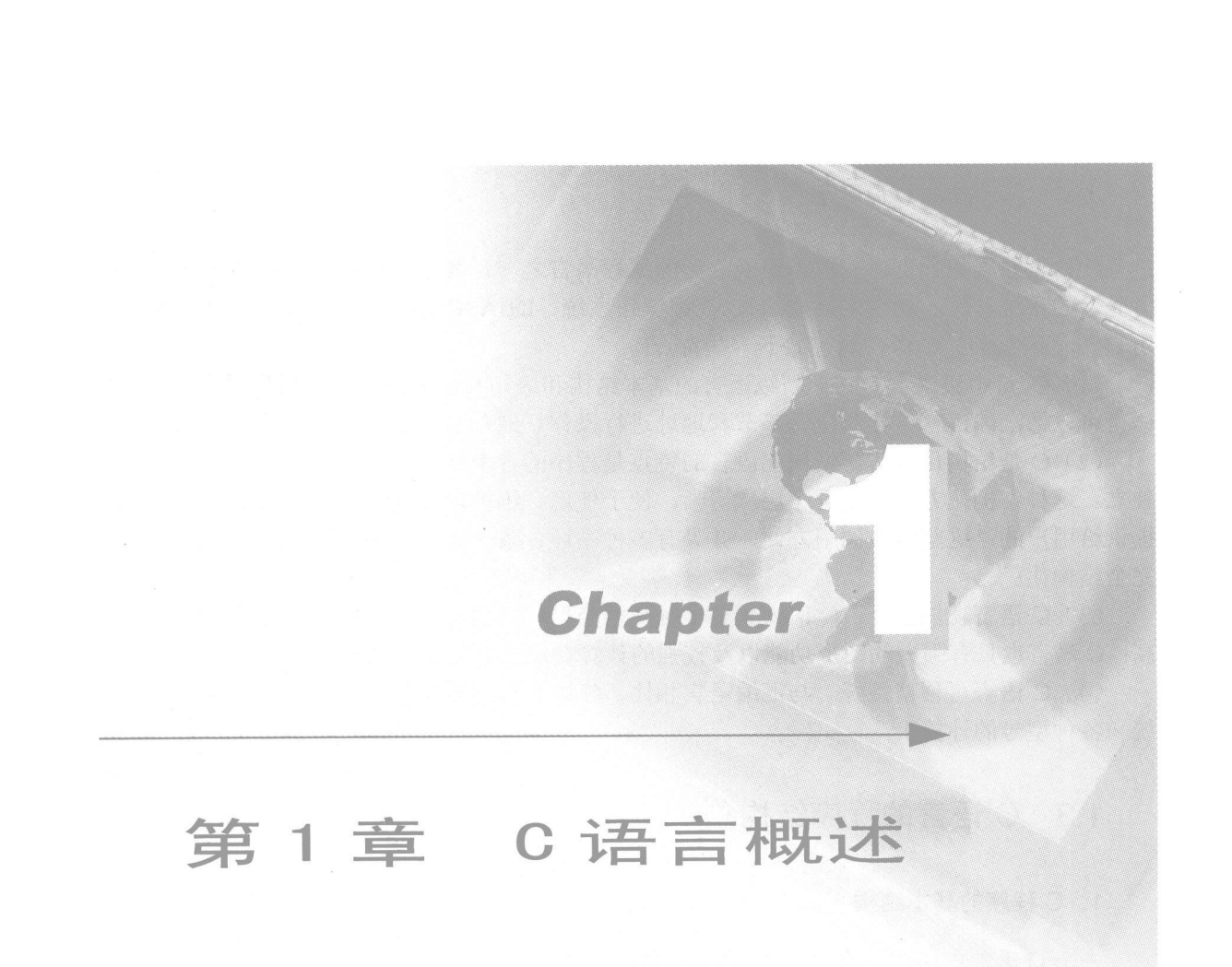
本章小结 (272)

附录 A 常用字符与标准 ASCII 码对照表..... (273)

附录 B 运算符和结合性..... (275)

附录 C Turbo C 标准函数库 (276)

参考文献..... (279)



Chapter

第 1 章 C 语言概述

1.1 C 语言产生的背景

C 语言是 1972 年由美国贝尔实验室的 Dennis Ritchie 设计发明的，它由早期的编程语言发展演变而来，可用于编写系统软件和应用软件，是广泛流行的高级编程语言。

C 语言诞生以前，系统软件主要是用汇编语言编写的，汇编语言可以实现对计算机硬件的直接操作，但是它依赖于计算机硬件，其可读性和可移植性都很差。而一般的高级语言却难以实现对计算机硬件的直接操作，所以人们希望有一种计算机语言既能具有高级语言的优点，又能具有低级语言的功能，C 语言就是在这种背景下产生的。

1970 年，美国贝尔实验室的 Ken Thompson 根据 BCPL (Basic Combined Programming Language) 语言，设计出了较先进的 B 语言，而 B 语言则导致了 C 语言的问世。随着微型计算机的普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准研究所 (ANSI) 为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准。目前，在微型计算机上广泛使用的 C 语言编译系统有 Microsoft C, Turbo C, Borland C 等。虽然它们的基本部分都是相同的，但还是有一些差异，所以还需注意自己所使用的 C 语言编译系统的特点和规定 (可以参阅有关手册)。

本书使用的 C 语言编译系统是 Turbo C 2.0。

1.2 C 语言的特点

C 语言发展如此迅速，成为最受欢迎的编程语言之一，其主要原因是它的功能强大。许多著名的系统软件，如大名鼎鼎的 UNIX 操作系统、DBASE III PLUS、DBASE IV 等都是用 C 语言编写的。C 语言具有如下一些特点：

(1) C 是中级语言。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，C 语言可以像汇编语言一样对位、字节和地址进行操作。通常也将 C 语言称为高级语言。

(2) C 是结构化语言。结构化语言的特点是程序的各个部分除了必要的数据交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护以及调试。C 语言是以函数形式提供给用户的，这些函数调用方便，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。

(3) C 语言功能齐全。C 语言具有多种数据类型，并引入了指针概念，可使程序效率更高；C 语言也具有强大的图形功能以及较强的计算功能和逻辑判断功能。

(4) C 语言可移植性好。与汇编语言相比，C 语言程序适用范围大，可用于各种操作系统和各种型号的计算机。

1.3 C 语言的程序结构

1. C 程序的基本结构

一个 C 语言程序，或者由一个 main() 函数（又称主函数）和若干个其他用户自定义函数结合而成，或者仅由一个 main() 函数构成。

为了说明 C 语言程序结构的特点，先看几个程序。这几个程序由简到难，表现了 C 语言程序组成结构上的特点。虽然有关内容还未学习，但可以从这些例子中了解到 C 语言程序的基本架构和书写格式。

【例 1.1】 仅由 main() 函数构成的 C 语言程序示例。

```
main()
{
    printf("How do you do! \n");
}
```

程序运行结果：

How do you do !

说明：

(1) C 程序的基本结构是函数，函数也叫模块，是完成一个程序功能的最小功能单位。

(2) C 函数从左花括号“{”开始，到对应的右花括号“}”结束。

【例 1.2】 编写一个求两数之和的程序。

```
main()                                     /*求两数之和*/
```

```

{
    int a,b,sum;
    a=123;b=321;
    sum=a+b;
    printf("sum=%d\n",sum);
}

```

程序运行结果：

```
sum=444
```

说明：

(1) C 语句是组成 C 程序的基本单位，具有独立程序功能，以分号结尾。

(2) 注释语句不是 C 语句，只是对程序的说明，以 “/*……*/” 为标记，不被系统编译和执行，目的是对程序进行注释和说明，帮助阅读程序，增强程序的可读性。

(3) 书写风格。

① 书写位置自由，一行中的语句数量无限制，注释语句可出现在任何位置。但推荐同学们采用本书的书写格式，以利于程序的阅读和修改。

② 采用缩进格式，不同结构层次的语句从不同位置开始，使程序清晰易读。

【例 1.3】 由 main()函数和一个用户自定义函数 max()构成的 C 语言程序示例。

```

int max(int x, int y)                                /* 定义函数 max */
{
    int z;
    if (x>y) z=x;
    else z=y;
    return(z);
}
main()
{
    int n1, n2, z ;
    printf(" Input two numbers: ");
    scanf("%d,%d", &n1,%n2 );
    z= max(n1, n2);                                    /* 函数调用*/
    printf("max=%d\n",z );                             /* 输出函数的返回值 */
}

```

程序运行情况：

```

Input two numbers: 3,8 ↵                               /* 符号 “↵” 代表回车键*/
max = 8

```

说明：

(1) 调用的功能函数 max()是自定义函数。

(2) printf 函数是一个由系统定义的标准函数，在程序中可直接调用。

(3) main() 函数可以放在程序的任何位置，C 程序执行时总是从 main()开始。

2. 关键字

从上面的程序可以看到有 `int, if, return` 等字符串的出现, 它们被称为关键字。关键字是由 C 语言规定的具有特定作用的字符串, 通过以后各章的学习, 要了解它们的含义, 正确地使用它们。C 语言的关键字共有 32 个, 根据关键字的作用, 可分为数据类型关键字、控制语句关键字、存储类型关键字和其他关键字四类。

(1) 数据类型关键字 (12 个): `int, long, short, char, float, double, enum, signed, struct, union, unsigned, void`;

(2) 控制语句关键字 (12 个): `break, case, continue, default, do, else, for, goto, if, return, switch, while`;

(3) 存储类型关键字 (4 个): `auto, extern, register, static`;

(4) 其他关键字 (4 个): `const, sizeof, typedef, volatile`。

3. 小结

(1) 函数是 C 语言程序的基本单位。`main()` 函数的作用, 相当于其他高级语言中的主程序; 其他函数的作用, 相当于子程序。

(2) C 语言程序总是从 `main()` 函数开始执行。一个 C 语言程序, 总是从 `main()` 函数开始执行, 而不论其在程序中的位置。当主函数执行完毕时, 亦即程序执行完毕。习惯上, 将主函数 `main()` 放在程序的最前头。

(3) 任何函数 (包括主函数 `main()`) 都由函数说明和函数体两部分组成, 其一般结构如下:

```
[ 函数类型 ] 函数名( 函数参数表 )           /*函数说明部分*/
{ 说明语句部分
  执行语句部分                               /*函数体部分*/
}
```

说明:

① 本书使用的语法符号约定。

● 方括号 “[]” 表示可选 (即可以指定, 也可以默认);

● 省略号 “……” 表示前面的项可以重复;

● 竖线 “|” 表示多 (含 2) 中选 1。

② 函数说明部分由函数类型 (可默认)、函数名和函数参数表等 3 部分组成, 其中函数参数表的格式为:

数据类型 形参 1[, 数据类型 形参 2, ……]

③ 函数体部分一般由说明语句和可执行语句两部分构成。

● 说明语句部分可由变量定义、类型定义、函数说明、外部变量说明等语句组成。函数体中的变量定义语句, 必须放在所有可执行语句之前。

● 可执行语句一般由若干条可执行语句构成。

(4) 源程序书写格式。

- ① 所有语句都必须以分号“;”结束,函数的最后一个语句也不能省略“;”。
- ② 程序行的书写格式自由,既允许1行内写几条语句,也允许1条语句分写在几行上。
- ③ 注释语句的位置自由,可以单独占1行,也可以跟在语句的后面;如果1行写不下,可另起1行继续写;注释中允许使用汉字。在非中文操作系统下,看到的是乱码,但不影响程序运行。

注意: 为避免遗漏须配对使用的符号,例如注释符号、花括号、圆括号等,在输入时,可成对地连续输入这些配对使用的标识符,然后再在其中插入要编辑的内容。

1.4 C 程序上机步骤

1. 运行 C 语言程序的一般过程

Turbo C 是一个集源程序编辑、编译、调试、运行于一体,利用菜单驱动的集成化程序开发环境。它是 DOS 上的一个 C 语言系统,也能以全屏幕方式或窗口方式运行在各种 Windows 环境中。

运行 C 语言程序的一般过程如下:

- (1) 启动 TC, 进入 TC 集成环境。
- (2) 输入与编辑源程序。编辑并保存后文件的扩展名为.c。
- (3) 对源程序进行编译。如果编译成功,则可进行下一步操作;否则,返回第(2)步骤修改源程序,再重新编译,直至编译成功。编译成功后文件的扩展名为.obj。
- (4) 与库函数进行连接。如果连接成功,则可进行下一步操作;否则,根据系统的错误提示,进行相应修改,再重新连接,直至连接成功。连接成功后生成的可执行文件的扩展名为.exe。
- (5) 运行可执行的目标程序。通过观察程序运行结果,验证程序的正确性。如果出现逻辑错误,则必须返回第(2)步骤修改源程序,再重新编译、连接和运行,直至程序正确。

2. TC 的启动、退出与命令菜单

(1) 启动 Turbo C。启动 Turbo C 有两种方法:

① 在 Windows 系统中,找到 TC 2.0 的目录,双击该目录中的 TC.EXE 文件,系统将自动启动一个 MS-DOS 窗口,TC 在其中运行。

② 启动 MS-DOS 命令窗口,在 TC 子目录下输入 TC 并按回车键即可进入 Turbo C 2.0 集成开发环境。

启动 Turbo C 2.0 后,其集成开发环境如图 1.1 所示。其主菜单条横向排列在屏幕顶端,并被激活,其中 File 菜单项成为当前项。主菜单的下面,是 Edit (编辑)窗口和 Message (消息)窗口。最下面一行为参考行。这 4 个部分构成了 Turbo C 2.0 的主屏幕,以后的编程、编译、调试以及运行等都将在这个主屏幕中进行。

主菜单在 Turbo C 2.0 主屏幕顶上一行,除 Edit 菜单外,其他各项均有子菜单,只要同时按【Alt】键和某菜单项的第一个字母,就可进入该项的子菜单中。

编辑窗口的顶端为状态行,其中

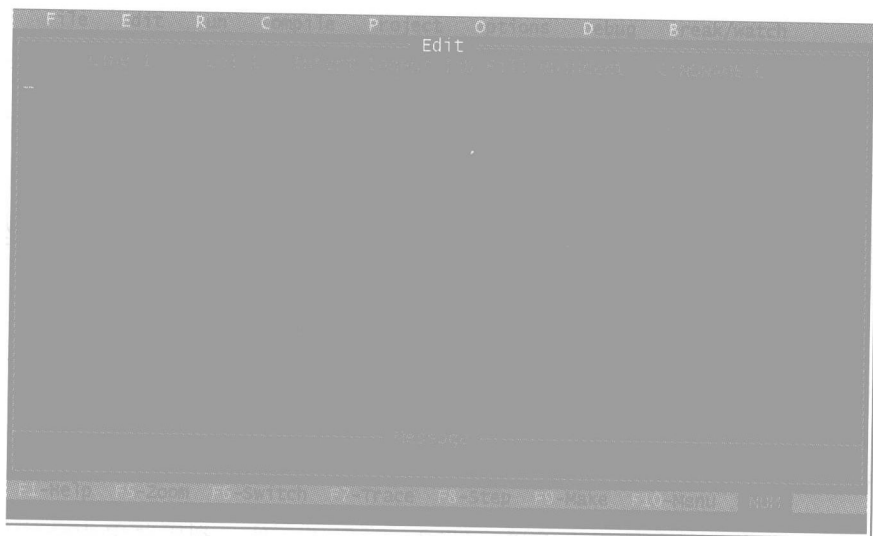


图 1.1 Turbo C 2.0 集成开发环境

- “Line 1 Col 1”显示光标所在的行号和列号，即光标位置。
- “Insert”表示编辑状态处于“插入”状态。当处于“改写”状态时，此处为空白。
- “C: NONAME.C”显示当前正在编辑的文件名。显示为“NONAME.C”时，表示用户尚未给文件命名。

屏幕底端是 7 个功能键的说明，以及 Num Lock 键的状态（显示“NUM”时，表示处于“数字键”状态；若为空白，表示处于“控制键”状态）。

（2）命令菜单的使用。

- ① 按下功能键【F10】，激活主菜单。如果主菜单已经被激活，则直接转下一步。
- ② 用左、右方向键移动光带，定位于需要的主菜单项上，然后再按【Enter】键，打开其子菜单（纵向排列）。
- ③ 用上、下方向键移动光带，定位于需要的子项上，按【Enter】键即可。执行完选定的功能后，系统自动关闭菜单。

注意：菜单激活后，若不使用，可再按【F10/Esc】键关闭，返回原来状态。

（3）退出 Turbo C。退出 TC 有两种方法：

- 执行 File\Quit 菜单命令（先选择 File 主菜单，再选择并执行 Quit 子菜单）。
- 按下快捷键【Alt+X】（同时按下【Alt】键和【X】键，然后同时放开）。

3. 编辑并保存 C 语言源程序

按【Alt+F】组合键进入 File 菜单，如图 1.2 所示。File 菜单的子菜单共有 9 项，下面分别进行说明。

（1）Load：载入一个文件，可用类似 DOS 的通配符（如*.C）来进行列表的选择；也可装入其他扩展名的文件，只要给出文件名（或路径）即可。该项的快捷键为【F3】，即只要按【F3】键即可执行该菜单的功能，而不需要先进入 File 菜单再选择此项。

（2）Pick：将最近装入编辑窗口的 8 个文件列成一个表让用户选择，选择后将该程序载入编辑区，并将光标置在上次修改过的地方。其组合快捷键为【Alt+F3】。

（3）New：新建文件，默认文件名为 NONAME.C，存盘时可改名。



图 1.2 File 菜单

(4) **Save**: 相当于“保存”命令，将编辑区中的文件存盘，若是第一次存盘，将询问保存文件的路径及文件名，其快捷键为【F2】。

(5) **Write to**: 相当于“另存为”命令，可由用户给出文件名将编辑区中的文件另外存盘，若该文件已存在，则会询问是否要覆盖掉原文件。

(6) **Directory**: 显示目录及目录中的文件，并可由用户选择。

(7) **Change dir**: 显示当前默认目录，用户可以改变默认目录。

(8) **Os shell**: 暂时退出 Turbo C 2.0 到 DOS 提示符下，此时可以运行 DOS 命令，若想再回到 Turbo C 2.0 中，只要在 DOS 状态下键入“exit”即可。

(9) **Quit**: 退出 Turbo C 2.0，返回到 DOS 操作系统中，其组合快捷键为【Alt+X】。

以上各项可用光标键移动光带进行选择，按【Enter】键则执行。也可用每一项的第一个大写字母直接选择。若要退到主菜单或从它的下一级菜单返回均可按【Esc】键，Turbo C 2.0 所有菜单均采用这种方法进行操作。

4. 常用编辑操作

在编辑源程序过程中，随时都可以按【F2】键（或 File/Save 菜单命令），将当前编辑的文件存盘，然后继续编辑。这是一个良好的编程习惯。

按下述各键可实现相应的编辑操作：【PageUp】向前翻页；【PageDown】向后翻页；【Home】将光标移到所在行的开始；【End】将光标移到所在行的末尾；【Ctrl+Y】删除光标所在的一行；【Ctrl+T】删除光标所在的一个词。其他操作不再赘述。

5. 编译、连接单个源程序文件

按【Alt+C】组合键可进入 Compile 菜单，该菜单有以下几项内容，如图 1.3 所示。

(1) **Compile to OBJ**: 将一个 C 源程序编译生成.obj 目标文件，同时显示生成的文件名。其组合快捷键为【Alt+F9】。

(2) **Make EXE file**: 此命令生成一个.exe 的文件，并显示生成的.exe 文件名。其中.exe 文件名是下面几项之一：