中等职业技术教育计算机教材

C语言
程序设计与上机指导

吕凤翥 韩 联 编著

清华大学出版社
http://www.tup.tsinghua.edu.cn

中等职业技术教育计算机教材

C语言程序设计与上机指导

吕凤翥　韩联　编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书较全面、准确地讲述了 C 语言的基本知识和语法规则，由浅入深地讲述了 C 语言简单程序的编写方法。本书包含 C 语言的内容有：常量、变量、运算符和表达式、语句、函数和存储类、指针、结构和联合以及读写函数和文件操作。每章后面备有大量思考题、练习题和上机指导。

全书语言通俗易懂，例题丰富，讲解详细，重点突出，概念准确，适于自学。本书可作为高中、中专和职业高中的 C 语言课的教材和参考书，也可以作为电脑爱好者自学 C 语言的指导用书。

图书在版编目(CIP)数据

C 语言程序设计与上机指导/吕凤翥，韩联编著. —北京：清华大学出版社，1999
中等职业技术教育计算机教材
ISBN 7-302-03590-3

Ⅰ. C… Ⅱ. ①吕… ②韩… Ⅲ. C 语言-程序设计-技术教育-教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字(1999)第 21030 号

出版者：清华大学出版社(北京清华大学学研楼，邮编 100084)
http：// www. tup. tsinghua. edu. cn
印刷者：北京昌平环球印刷厂
发行者：新华书店总店北京发行所
开 本：787×1092 1/16 印张：17.75 字数：412 千字
版 次：1999 年 8 月第 1 版 2000 年 3 月第 2 次印刷
书 号：ISBN 7-302-03590-3/TP・1978
印 数：10001～15000
定 价：21.00 元

中等职业技术教育计算机教材编写委员会

序　言

从第一台电子计算机问世到今天，短短五十年，人类从生产到生活发生了巨大的变化，以计算机为核心的信息技术作为一种崭新的生产力，正在向社会的各个领域渗透。过去说：没有电将寸步难行；现在要说：没有计算机就没有现代化。

计算机科学与技术的划时代的意义是为人类提供了“通用智力工具”。著名的计算机科学家、图灵奖的获得者G·伏赛斯曾预言：计算机将是继自然语言、数学之后而成为第三位的、对人的一生都有大用处的“通用智力工具”，用还是不用这个智力工具，对人的智能的发挥和发展肯定大不一样。十年前有识之士在《中国计算机工业概览》中写道：“我们往往欣赏中国人的聪明才智。我国有丰富的智力资源和脑力劳力的优势，这当然是事实，但我们是否考虑过，社会发展到今天如果不同时有效地利用‘电脑’，这个‘人脑’的优势是会丧失的。”机遇和挑战并存，将有关信息科学的知识和应用能力纳入到学生的知识结构中，是提高人才素质的需要，是落实“科教兴国”战略的一项重要内容。

在中等职业技术教育中计算机应该是一门新的主修课。这套教材面向的是职业高中、中等专科学校的各类非计算机专业的学生，其特点是：以应用为主，突出实用性和操作性。

以应用为主，不等于不需要讲一些必要的原理。从打好基础的角度看，懂一点计算机的基本原理，对于消除计算机的神秘感，使用和驾驭计算机是大有好处的。这套教材的作者都是具有多年第一线教学经验的资深教师，在书的写法上，充分考虑职业高中和中专学生的工作需要和认知规律，精心选择内容，采用循序渐进的教学方法，将重点放在基本概念和基本操作方法上。书中特别安排了上机指导，这是十分必要的，也是这套书的特色之一。计算机的课程实践性极强，不上机，不动手，是学不会的。因此，我建议同学们一定要理论联系实际地学，既动手又动脑，才能学得从容，学得深入，才能掌握真才实学。越动手，你就越能找到成功的感觉；越动手，你就越爱用计算机为你服务；越动手，你就越会感到：计算机入门不难，深造也是完全办得到的。

中国计算机学会普及委员会主任
国际信息学奥林匹克中国队总教练
全国高等学校计算机基础教育研究会副理事长
清华大学计算机科学与技术系教授

吴文虎

前　言

当今世界，计算机、通信、微电子和软件技术的发展和应用已成为衡量一个国家现代化程度的主要标志之一。

随着我国改革开放的进一步深入，目前全国各地职业高中及各类中专的各非计算机专业相继都开设了计算机课，它标志着我国职业高中、中专的计算机教育、教学已进入一个新的发展阶段。

学习计算机，一要学什么是计算机，二要学计算机的操作，学习内容包括理论和实践操作。计算机是一门应用型学科，操作性强。随着计算机在社会各个领域的应用越来越广泛，对计算机操作能力的要求也越来越高。所以，职业高中、中专非计算机专业都在开设计算机课。计算机课的教学要面向社会、面向市场，既要让学生学习计算机知识，又要对学生进行计算机操作技能的训练，重点是侧重操作和技能性方面的训练。

近几年社会上普通中学及职业高中、中专计算机专业的教材、资料比较齐全，而适合职业高中、中专非计算机专业的教材却比较少。在教学对象、教学要求、教学内容和教学方法上，职业高中、中专非计算机专业和计算机专业的教学有着较大的差别。选好教材、用好教材是搞好计算机教学的重要保证。出版一套适合各类职业高中、中专非计算机专业适用的系列教材，就是我们编写这套教材的初衷。

根据职业高中、中专非计算机专业计算机教学的特点，这套教材在注重系统性、科学性的基础上重点突出了实用性和操作性，将重点讲述计算机的基本概念和基本操作方法。按照由浅入深的教学原则，把各册教材的内容分割成若干个模块，采取循序渐进的教学方法，力求通俗而不肤浅，深入而不玄奥。各部分都采用举实例的方法讲述操作技术；对重点概念、重要的操作技能，力争讲深讲透。

侧重上机操作，将上机指导作为主要内容之一是本教材的又一特色。每章后的上机指导内容通俗易懂，操作循序渐进。每个上机指导包括目的与要求、软硬件环境和操作步骤三部分。有些操作练习有详细的参考步骤，其目的是为了举一反三；有些操作练习没有参考步骤，其目的是为了使学生进一步巩固所学知识和掌握操作方法。每章的上机指导配合小结、习题，使学生在动脑、动手的过程中牢固地掌握计算机实用技术。

本套教材的作者均为从事计算机教育10年以上的计算机高级教师，来自北京市部分职业高中计算机专业及非计算机专业计算机教学的第一线，有丰富的计算机教育、教学经验，并出版过多本计算机教育的书籍。本套教材第一批共10册，均为中等职业教育中急需的计算机教材。通过本套教材的学习，学生可以掌握计算机专业基础知识和技术，较熟练地掌握计算机的使用和维护技能，并具有初步的程序设计能力。对教材内容中不妥或需要改进之处，殷切希望广大师生向我们指出，以便再版时修改和补充。来信请寄：北京清华大学出版社编辑部(100084)。

这套教材编写的内容对社会上人事部门、劳动部门的技术等级考试也具有指导作用。

编者的话

这是一本供高中、中专和职业高中学生学习C语言的教材或参考书，也是一本广大电脑爱好者自学C语言的辅导用书。该书的两位作者分别是大学和中学长期讲授C语言课程的教师，具有较丰富的教学经验和上机实习经验。本书是在总结教学经验的基础上，针对中学层次的学生的基础和接受能力编写的。

本书较全面准确地讲述了C语言的基本知识和语法规则，详细介绍和分析了简单C语言程序的编写方法。本书从一个具体的C语言程序例子讲起，读者一开始就看到了C语言程序的结构特点和所包含的词法及语法现象，本书就该程序中出现的词法和语法问题由浅入深地进行讲解。本书共分8章，第1章从C语言的一个例子入手，讲解C语言程序结构、书写格式以及C语言程序如何实现；第2章讲述C语言的词法、常量和变量，这些都是C语言的基本知识；第3章进一步讲述C语言的运算符、表达式以及类型转换，这些内容是编程的基础；第4章讲述了C语言中所有语句形式、规则、功能和使用方法，这些内容也是为编写C语言程序打下基础；第5章讲述了函数的定义和说明，函数的参数及返回值，函数的调用方式，作用域的规则和存储类的概念及应用，这些内容指出了C语言的编程方法；第6章讲述指针的概念和应用，特别是指针在数组和函数方面的具体应用，指针是C语言的重要特点，也是学习C语言的难点之一；第7章讲述了两种重要的数据类型：结构和联合，特别是结构在C语言程序经常出现，对结构的定义和应用应该掌握；第8章讲述了C语言的文件操作，着重介绍了一般文件的打开、关闭操作，对已打开文件的读写操作以及定位读写指针的操作等，文件操作也是C语言程序一个重要方面。本书用较短的篇幅讲述C语言中所有常用的内容，语言简练，概念准确，例题中基本没有重复。

本书在编写方面具有如下特点：在内容上较为系统和全面，包括了C语言中主要的基础知识、基本技能和常用的语法规则；在写法上突出重点、详述难点、揭示疑点。由于作者都是第一线的讲课教师，因而能够抓住学生心理，及时回答学生可能发问的问题；在讲解上先讲清概念，指出方法和规则，再提示在应用中应注意的事项，最后通过列举例题加深理解和学会应用，对每个例题都做了详尽的分析说明，并指出编程的技巧和方法；在安排上每章后面备有相当丰富的思考题，提出了本章应掌握的主要内容，是用来帮助复习和检查学习情况的；也有形式多样的练习题，通过这些题目可以练习概念，训练操作方法，分析程序输出结果和训练编程能力；还有上机指导，通过上机实践帮助读者进一步理解所讲过的内容，培养动手能力和分析解决问题的能力。

根据多年教学经验，我们认为学习C语言应先了解有关概念、规则和方法，再通过例题进一步掌握有关规则和方法，然后通过做题或看程序进一步消化和理解，最后再通过上机实践，可以学到一些实践中的知识和训练动手能力及培养编程方法。要学会C语言的编程方法，先要掌握C语言的基本语法规则，在此基础上多看程序多上机实践，多吸取别

人的编程技巧，多学习别人的分析解决问题的方法，这样就会提高你的编程能力。你不妨按照我们告诉你的方法试一试。

本书所有的例题和习题中的程序都在 Turbo C 2.0 版本的 C 语言编译系统下调试过。

由于编写时间较紧，难免会有错误，敬请读者指教。

谢谢喜欢这本书的读者朋友。

编　者

1999 年 4 月于北京

目　录

第1章　C语言程序的一个例子

1.1　从C语言程序的一个例子讲起

让我们先看一个C语言程序的例子，从这个例子讲述C语言程序的结构、书写格式，C语言中的单词、语句和函数。

[例1.1]　从键盘上输入一个数，求出它与已知数的积。

程序内容如下：

```
#include <stdio.h>
#define N 5
main( )
{
    int x,y;
    gets("Input a number:");
    scanf("%d",&x);
    y=N * x;
    printf("N * x=%d\n",y);
}
```

这是一个比较简单的C语言程序，程序开始的两行是两条预处理命令，第一条预处理命令称为文件包含命令，该命令将它所包含的文件stdio.h的内容复制到该程序的开头，供后面程序使用；第二条预处理命令称为宏定义命令，它用来定义一个常量N，其值为5。关于预处理命令将在本章后面预处理功能中讲述。

接着，程序中定义了一个函数，其名字是main，称为主函数，本例中该函数没有参数，因此在函数名后面的一对圆括号中是空的；如果有参数，该参数应该放在圆括号内。下面是由一对花括号({})括起的5条语句，称为函数体。在本书的第5章中将讲述函数的定义，函数的定义由两部分组成，其中一部分称为函数头，另一部分称为函数体。本例中，main( )便是函数头，而一对花括号括起的5条语句构成了函数体。

在函数体中，第一条语句说明或是定义了两个变量x和y，它们都是整型(int)变量；第二条语句是调用一个系统所提供的函数gets( )，该函数被包含在stdio.h文件中，这便是程序开头要包含stdio.h的原因。这条语句的功能是在屏幕上显示如下的字符串：

Input a number:

第三条语句也是调用一个系统所提供的函数scanf( )，该函数的功能是用来从键盘上接收输入的数据，然后将它赋给指定的变量。本例中，scanf( )函数用来将键盘上键入的数据给变量x赋值，即使变量x从键盘上获取数值；第四条语句是一条赋值语句，它使用了赋值运算符(=)将表达式N * x的值赋给变量y，这里，N * x是用N常量乘以x变量中的值得到一个积值，再赋给变量y；第五条语句是调用一个系统所提供的函数

printf( ),该函数的功能是用来将所指定的表达式的值按一定格式显示在屏幕上。本例中就是将变量 y 的值按照指定的十进制格式(%d)显示在屏幕上。

以上是对[例 1.1]程序的一些解释。看了程序,读了解释后,一定会产生很多问题,或许会感到C 语言程序好难啊! 也许你会问下面的一些问题:

(1) 这个程序[例 1.1]到底是什么意思?

(2) 解释中有那么多名词何时才能搞明白?

(3) C 语言程序在结构上到底有什么特点?

(4) C 语言程序在书写上应注意哪些问题?

(5) 如何执行这个程序获得输出结果?

在这一节中,先回答前两个问题,后三个问题将在本章其他节中回答。

这个程序的功能概括起来是这样的:在执行这个程序时,屏幕上会显示如下提示信息:

Input a number:

这表明让执行者(用户)从键盘中输入一个数(例如输入 8),然后回车,这时程序中的变量 x 将获得数值为 8。经过 N * x 的运算后,其积为 40,因为常量 N 为 5,这是在程序中设定好的。将 40 赋给 y,y 获取值为 40,再将 y 的值按规定的格式输出到屏幕上,这时屏幕上显示如下结果:

N * x=40

这是该程序的功能,也是该程序执行后的输出结果。

在该程序的解释中,可以看到确实有许多陌生的名词和概念,它们包括:预处理命令和输入输出函数。include 和 define 以及输入函数 scnaf( )和输出函数 gets( )、printf( )都将在本章后面的章节中讲述。还有常量、变量、赋值、语句、函数、字符串、运算符、表达式、数据类型、输入、输出等名词和概念将在本书后面章节中逐一讲述,这里只要求有些了解和留下一些印象,因为后面还要详细讲解。

1.2 C 语言程序结构和书写格式

1.2.1 C 语言程序结构

C 语言程序结构特点归结起来有如下几点:

1. C 语言程序是由一个或多个文件组成的,文件由一个或多个函数组成,函数又由若干条语句组成,而语句又由一个或多个单词组成,单词由一个或多个字符组成。字符是组成C 语言程序的最小元素,函数是组成C 语言程序的最小模块,上述关系可简单表述如下:程序—文件—函数—语句—单词—字符。

本书将按相反顺序讲述C 语言程序的基本结构,在第 2、3 章中讲述字符和单词,第 4 章中讲述语句,第 5 章中讲述函数,每章中都贯穿着C 语言的程序或文件。在前面讲述的[例 1.1]就是一个简单的C 语言程序,该程序是由一个文件组成的,该文件在存储时对应一个文件名,它的扩展名要求是.c。该文件只由一个函数组成,该函数名字是 main。该函

数的函数体由5条语句组成，这5条语句中除第1条是说明语句外，其余4条都是执行语句。执行语句多是表达式语句，表达式语句是由一种表达式后面加分号(;)组成的，每条语句又由若干个单词组成，例如，函数体内第一条语句是由3个单词组成的，它们分别是关键字int和标识符x和y，它们之间用分隔符分隔，这里所用的分隔符有逗号或空格。有关单词问题在下一章讲述，这里提到的关键字、标识符和分隔符都是单词。单词是由字符组成的这一点很清楚。

2. C语言是结构化程序设计语言。结构化程序设计的特点是模块化，即结构化程序是由若干个模块组成的，模块之间通过输入输出接口连接。结构化程序设计的优点是便于编程、调试、维护、扩充和移植等。在C语言程序中，函数是最小的模块，实际上C语言程序是由若干个函数组成的，这些函数可以集中在一个文件中，也可以分布在多个文件中。组成C语言程序的若干个函数其放置的前后顺序无关，函数与main()的顺序有关。函数之间是调用的关系，即一个函数中可以调用另一个函数或另几个函数。函数中不能定义函数。因此，可以认为C语言程序是一个函数串，这些函数是并行放置的，它们之间是调用关系。

3. 在一个C语言程序中只能有一个主函数，该函数名为main。C语言程序中必须有一个并且只能有一个主函数main( )。如果C语言程序是由多个文件组成的，那么只能有一个文件中带有一个主函数main( )，该文件称为主文件。在C语言程序中，没有主函数或者有一个以上主函数时，都会出现错误信息。主函数放置在程序中的位置无关，将主函数放在文件头或文件尾或文件中间都可以。

4. 在执行一个C语言程序时，总是从程序中的主函数main( )开始的，如果找不到主函数就无法执行。程序中的其他函数由主函数调用，或者用主函数所调用的函数来调用。函数调用是靠名字的，因此每个函数都要有一个确定的名字，并且在同一个程序中的函数名不得相同。[例1.1]的程序中只有一个函数，它必须是主函数。下列举一个具有两个函数的程序。

[例1.2] [例1.1]问题的另一种编程方法。

程序内容如下：

```
#include <stdio.h>
#define N 5
main( )
{
    int x,y;
    gets("Input a number:");
    scanf("%d",&x);
    y=fun(x);
    printf("N * x=%d\n",y);
}
fun(a)
int a;
{
    int b;
```

```
    b=N*a;
    return b;
}
```

该程序的功能与例[1.1]完全相同，但在程序的编写上将原来在主函数体中进行的求积运算改为在另外一个函数中进行。主函数中调用这个求积运算的函数来完成求积运算，具体地讲，在 main( )中，调用 fun( )函数，并将 fun( )函数的返回值赋给变量 y。有关函数的调用和返回值问题将在第 5 章中详述。

该程序与[例 1.1]一样都是一个文件的程序，一个文件由两个函数组成：主函数 main( )和一个被主函数调用的函数 fun( )。这两个函数放置的顺序无关，可以将主函数 main( )放在前面，也可以将它放在后面。执行该程序时，先执行主函数 main( )，在主函数体中再调用 fun( )函数，fun( )函数结束后将返回到主函数，继续执行主函数中调用函数后面的语句直到主函数结束为止。

1.2.2 C 语言程序的书写格式

一般说来，C 语言的语句比较精练、短小，一个语句或一个表达式将包含许多内容，因此，C 语言程序的可读性较差，需要有良好的书写习惯，以便提高程序的可读性。

C 语言程序的书写格式习惯上有如下要求：

1. C 语言程序中每行可写一条语句，也可写多条语句，一般一行写一条语句。C 语言程序中也可以一条语句写在多行上，一般不用续行符(\)。有时一条语句写成多行时，需要在上一行末尾加续行符。

2. C 语言程序中每条语句的末尾必须加一个分号(;)，而不是一条语句的末尾则不应加分号。分号用来表示一条语句的结束。C 语言程序中语句不需要加行号，只有在 goto 语句转向到的某条语句前需要加语句标号，具体用法详见第 4 章语句。

3. C 语言程序中经常出现花括号({})，使用成对的花括号来表示不同用法。花括号具体书写格式有三种，本书采用其中的一种，其格式规定：每个花括号独占一行，并且左花括号(又称开括号)和右花括号(又称闭括号)都与使用它们的语句对齐，而花括号中的语句都向右缩进两个字符，本书中都采用这种规定书写程序，前面讲过的[例 1.1]和[例 1.2]中函数体的花括号就是按此规定书写的。

在说明语句中，有时用花括号给出初始值表和在枚举类型中用花括号作为枚举表定界符时，花括号不必占一行，这些规定后面会讲到。

4. C 语言程序的书写中要注意适当的缩进。使用缩进的方法使程序更加清晰易读。一般地，循环语句中的循环体和条件语句(if 语句)中的 if 体、else 体等都应缩进，这些内容将在第 4 章中详述。

5. 在书写程序中，要习惯使用/*…*/对程序中某些部分进行注释，所注释的内容只能阅读，而不参与编译和运行。使用注释的方法也可增加程序的可读性。

为了强调书写格式对可读性的影响，下面举两个例子——[例 1.3]和[例 1.4]。由于 C 语言程序书写比较灵活，一条语句中只要不把单词分开，其他可随意分隔。[例 1.3]是一个求两个数之和，并把其和输出显示在屏幕上的程序，由于没有按照习惯的书写格式书

写，虽然可以编译执行，但是读起来很困难。[例 1.4]是将[例 1.3]的程序按习惯的书写规则重新书写一遍，显然这样书写好读多了。

[**例 1.3**] 编写程序求两个数之和。

```
add(x,
y)
int
x,y;{
return
x+y; } main( ){
int a,
b;
a=10
;b=5;
printf("%d\n"
,add(a,b))
;}
```

该程序由于没按习惯的格式书写，因此很难读懂。习惯的书写格式是每行一个语句，单词之间用空格符分隔，说明语句中多个变量之间和函数参数之间用逗号分隔，每条语句结束时用一个分号。将[例 1.3]程序按规定格式书写为[例 1.4]格式。

[**例 1.4**] [例 1.3]程序的习惯书写格式。

```
add(x,y)
    int x,y;
    {
        return x+y;
    }
    main( )
    {
        int a,b;
        a=10;
        b=5;
        printf("%d\n",add(a,b));
    }
```

执行该程序后，会在屏幕上显示出如下结果：

15

执行[例 1.3]也会获得相同结果，可见这两个程序都是正确的，只是书写格式不同，[例 1.3]的书写太随便了，读起来很困难，[例 1.4]的书写格式比较规范，可读性比较好。

1.3 C 语言的预处理功能

在前面讲过的[例 1.1]程序的开头，有如下两个命令：

#include <stdio.h>

#define N 5

仔细观察会发现，这两个命令有如下特点：它们都以(#)开头，并且行的末尾没有分号作为结束符，因此它们不是语句，而称为预处理命令。

C 语言编译系统提供一种预处理功能。预处理功能是指具有这种功能的若干条命令在编译时先被执行，执行后再做正常的编译工作。可见，具有预处理功能的命令是在编译之前预先处理的，故称为“预处理”，具有预处理功能的命令称为“预处理命令”。预处理功能主要提供三种预处理命令，它们分别是宏定义、文件包含和条件编译等，本书中主要讲述前两种，即宏定义和文件包含。

预处理命令写在程序中时要求在命令字前面加“#”号，用来标识其后为预处理命令。因此，程序在编译时，先将标有“#”号的预处理命令处理完，再进行正常的编译工作。

归结一下，预处理命令具有如下特点：

1. 多数预处理命令具有一种替代的功能，这种替代是简单的替换，而不进行语法检查。如果存在语法错，则在编译时指出。

2. 预处理命令在正常的编译之前执行，预处理命令被执行后再进行编译，因此，编译是对预处理后的结果进行的。

3. 标识预处理命令的符号是“#”号，凡是预处理命令都以“#”号开头。

4. 预处理命令后面不加分号，这也是在形式上与语句的区别。

5. 预处理命令可以放置在程序的任何位置，即可放在程序开头，也可放在中间和末尾。但是，多数预处理命令被放置在文件或程序的开头。

1.3.1 宏定义命令

宏定义命令是一种常用的预处理命令，宏定义命令又分两种：一种是简单的宏定义，它不带有参数；另一种是带参数的宏定义。无论哪一种宏定义，它都是将一个标识符(或者称为宏名)定义为一个字符串(或者称为宏体)。本书只讲述前一种简单的宏定义，而带参数的宏定义本书不再详述。

1. 简单宏定义格式和功能

简单宏定义格式如下：

#define <标识符> <字符串>

其中，define 是命令字(也称关键字)，它表明该命令是宏定义；<标识符>是宏名，宏名的写法同标识符；<字符串>用来表示<标识符>所代替的串。

简单宏定义的功能是：把一个标识符定义为一个串。在程序中书写的是标识符，编译前将标识符用它被定义的串来替换，称为“宏替换”。例如，在前面讲过的[例 1.1]中

#indefine N 5

就是一个简单的宏定义，将标识符 N 定义为 5。在程序中出现了带有标识符 N 的语句

y=N * x;

该语句被宏替换后，则变成：

y=5 * x;

这是在编译时首先要做的事情。