21世纪高职高专系列规划教材·计算机类

21SHIJI GAOZHI GAOZHUAN XILIEGUIHUA JIAOCAI·JISUANJILEI

C语言程序设计教程

主编 李培金 丁春莉

C YU YAN CHENG XU SHE JI JIAO CHENG

西北大学出版社
NORTHWEST UNIVERSITY PRESS

C语言程序设计教程

21世纪高职高专系列规划教材·计算机类

C语言程序设计教程

主　编　李培金　丁春莉
副主编　段宏斌　张　克

西北大学出版社
中国·西安

内容简介

全书共分十三章，分别介绍了C语言的基本概念、语法规则、TC运行环境、数据类型、运算符与表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、编译预处理、指针、结构型、位运算、文件和TC综合应用等内容。每章课后附有大量的习题与实训供读者练习。本书是学习C语言程序设计的基础教材，教材从组织形式上采用实例与启发式的教学方法，通俗易懂，层次分明，结构清晰。

本书可作为高职高专计算机应用、信息管理、电子类各专业的学习教材，也可供C语言程序设计的爱好者自学使用。

图书在版编目（CIP）数据

C语言程序设计教程/李培金等编 .—西安：西北大学出版社，2004．8

ISBN 7－5604－1969－0

Ⅰ.C...　Ⅱ.李...　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088107号

C语言程序设计教程

出版发行	西北大学出版社	社　　址	西安市太白北路229号
电　　话	029－88302590	邮政编码	710069
经　　销	新华书店经销	印　　刷	西安市商标印刷厂
版　　次	2004年8月第1版	印　　次	2004年8月第1次印刷
开　　本	787×1092　1/16	印　　张	17.25
字　　数	420千字		
书　　号	ISBN 7－5604－1969－0/TP·26	定　　价	25．00元

前　言

程序设计是计算机及相关专业的一门专业基础课，也是计算机应用、信息管理、软件、电子类各专业的学生必备的基本技能之一。C 语言程序设计目前已广泛应用于软件设计与开发中，与其他程序设计语言相比，C 语言具有功能强，运算符丰富，使用灵活，执行效率高，源程序简洁清晰等特点，C 语言的函数式结构能方便实现模块化程序设计。

通过多年的 C 语言教学实践，笔者体会到，对目前的高职高专学生而言，C 语言之所以较难掌握，主要原因是 C 语言本身语法细节较多，运算符（如 + +/- -等）丰富且组织性强，特别是指针的概念及使用较抽象，从而使学生难以很快掌握。

根据这一普遍现象，本教材在编写时，结合高职高专学生的学习特点，并做了认真的分析与调研，制定了本书的编写原则：即内容体系要完整但要突出重点，理论与实践并重但要突出实践训练，以实例分析为重点但要突出设计方法。基与这一基本原则，本教材体现以下几个方面的特点：

1．引入实例教学和启发式教学方法，理论讲述结合实例，通过实例展现理论知识。内容深入浅出，通俗易懂，从而使学生在潜移默化中学到知识及方法。

2．内容突出实用，语言叙述言简意赅，对必需的知识体系全面介绍，对常用的内容辅以实例，对生僻的内容和细节则概要介绍，体现“内容体系完整但突出重点”的原则。

3．良好的源程序书写风格，程序设计风格和算法设计思想，贯穿整个教材。使学生逐步养成良好的程序组织和书写习惯。

4．本书选用 TC V2．0 作为教学上机环境，并做了较详细的介绍，凡与本版本关系密切的内容，均以 TC 为例予以介绍。

本书共有十三章，分别介绍了 C 语言的基本概念、基本数据类型、运算符与表达式、三种基本结构（顺序、选择、循环）、函数、构造型数据类型（数组、指针、结构体、文件）及使用方法、编译预处理、位运算及综合应用等内容。

本书由李培金、丁春莉任主编。李培金编写第十一、十二章和附录，丁春莉编写第八、九章，贾步忠编写第五、六章，张克编写第三、十、十三章，夏东盛编写第四、七章，段宏斌编写第一、二章。

本书在出版之际，主编代表全体作者，感谢西北大学出版社的大力支持及为本书出版所做的一切工作。

由于作者水平有限，书中错误在所难免，敬请专家和广大读者批评指正，以便我们再版时不断修正与完善。

编　者

2004 年 8 月

目　录

第一章　C语言概述

C语言是一种面向过程的并且很灵活的程序设计语言，它把许多程序设计技巧交给了编程人员。C语言的语法源自ANSI C标准（ANSI：美国国家标准化协会）。掌握了C语言后，再学C++、VC++、Java等其他语言就比较容易了。

在C语言诞生以前，系统软件（例如操作系统）主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差；但一般的高级语言又难以实现对计算机硬件的直接操作，于是人们盼望有一种兼有汇编语言和高级语言特性的新语言。C语言就是在这种背景下应运而生的。本章要点如下：

·语言的发展和特点

·C语言程序结构和书写规则

·C语言的编译环境

第一节　C语言的发展和特点

在上图中向左的横箭头表示即使在同一种数据类型间进行运算时，C语言也要进行转换，用于提高计算精度。向上的纵向箭头表示当运算对象类型不同时的转换方向。其转换过程如表2-4所示。

一、程序设计语言的发展

语言是人们交流思想、传达信息的工具。人们进行交流所用的语言（英语、汉语、日语、法语等）称之为自然语言（Natural Language）。人们为了有效地实现人与计算机之间的通信，人们设计出多种词汇少、语法简单、意义明确的适合于计算机使用的语言，这样的语言被称为计算机语言，也称为程序设计语言（Programming Language）。程序设计语言是人与计算机进行沟通的工具。随着计算机技术的飞速发展，程序设计语言也在不断地发展变化，从初期的机器语言发展到现时的面向对象的高级语言，经历四个时代：

第一代为“机器语言”：由一组0和1组成的二进制指令码的集合，是计算机硬件惟一可以直接识别和执行的语言。用机器语言编写计算机程序，必须熟悉该机器的指令系统中的各条指令的功能的二进制代码。这些代码难记，难理解，难纠错，即使是专业人员，要编写一个正确无误的程序也不是一件轻而易举的事情，而且效率低，劳动强度大。因此，在计算机诞生后的早期，只有少数专业部门有能力使用计算机。

第二代为“汇编语言”：为了提高编程效率相减轻劳动强度，50年代中期开始用“助记符”代替用0和1表示的机器指令来编程，用这种助记符描述的指令系统，称为汇编语言。用汇编语言编写的程序称为汇编语言程序。但是计算机并不能直接识别和执行汇编语言源程序，必须在计算机上配置一个称之为汇编系统的软件，通过这个汇编系统，把汇编

语言源程序翻译为机器语言指令序列，后者称为目标程序、对于目标程序计算机就能够识别和执行。用汇编语言编程，虽然比用机器语言编程省事多了。但是它与机器语言一样，都依赖于所用计算机的指令系统，同一个程序不能在指令系统不同的计算机上运行，即移植性差。另一方面，用上述这两种语言编程时，除了要有正确的解题步骤之外，还要熟悉计算机的内部结构。这两种语言都称为是面向问题的机器语言，或称之为低级语言。

第三代为“面向过程的高级语言”：它是由一些接近于自然语言和数学语言的语句组成的，因此，更接近于要解决问题的表示方法，并在一定程度上与机器无关，用高级语言编写的程序，接近自然语言与数学语言，易学、易用、易维护。C语言就是一种面向过程的高级语言。下面是一个计算园面积的C语言程序段：（其中：/* ... */称为注释行，/*与*/之间的内容称为注释内容）

```
main ()                          /* 告诉编译器C程序由此开始执行 */
{                                /* 程序片段执行开始 */
float r, area;                   /* 定义园半径r与面积s为实型 */
r = 5. 356;                      /* 给半径r赋值 */
area = 3. 14159 * r * r;         /* 计算面积 */
printf ("%f \ n", area);         /* 输出面积的值 */
}                                /* 程序执行结束 */
```

但由于高级语言编写的程序，计算机不能直接执行，该程序必须先翻译成计算机能识别的代码。这项翻译工作是由称之为“翻译程序”的专门软件来完成的。各种高级语言都有相应的一套编译程序。其编辑处理与语言处理过程如图1-1所示。

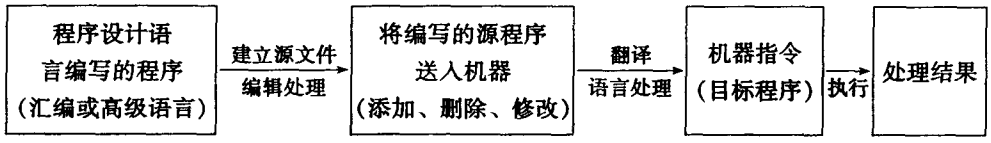

图1-1 编辑处理与语言处理过程示意图

由高级语言编辑的源程序，经过翻译程序的“翻译”，其结果称为目标程序，翻译程序有两种典型的实现途径，分别称为解释过程与编译过程，如图1-2和图1-3所示。

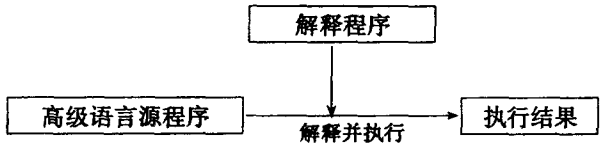

图1-2 解释过程示意图

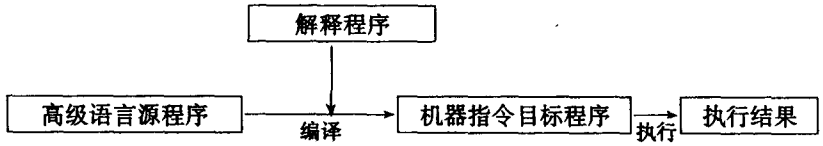

图1-3 编译过程示意图

第四代“面向对象的高级语言”：用面向过程的高级语言编写程序比用汇编语言等容易多了，但程序设计者必须具体写出每一步如何进行的全过程，每个环节都必须考虑得十分周全。面向对象程序设计者处理的是对象，只要告诉计算机“做什么”，不必指出“怎样做”，由计算机自己利用这些语言系统提供的软件功能去解决“怎样做”的问题。如由C语言派生出的C++语言，它是一种多范型程序设计语言，用它既可以编写面向对象的程序，也可以编写面向过程的程序。

二、C语言的历史及发展

C语言是目前世界上最为流行的计算机高级程序设计语言之一，它既适合于编写应用软件，又特别适合于编写系统软件。C语言是在B语言的基础上发展起来的，它的根源可以追溯到高级语言ALGOL60。它离硬件比较远，不宜用来编写系统程序。1963年英国剑桥大学推出了CPL（Combined Programming Language）语言。CPL语言在ALGOL60的基础上更靠近硬件，但规模较大，难于实现。1967年英国剑桥大学的Matin Richards对CPL语言作了简化，推出了BCPL语言。1970年美国贝尔实验室的Ken Thompson对BCPL语言又作了进一步的简化，设计出了更简单又很接近硬件的B语言，并用B语言写出了第一个UNIX操作系统。但由于B语言过于简单，功能有限，1972～1973年贝尔实验室的D. M. Ritchie在B语言的基础上设计出了C语言（取BCPL语言名中的第二个字母）。C语言既保持了B语言精练、接近硬件的优点，又克服了过于简单、数据无类型等缺点。随后C语言经多次改进，到1978年以后已能移植到大、中、小、微型机上。现在C语言已风靡世界，成为应用最广泛的计算机语言之一。

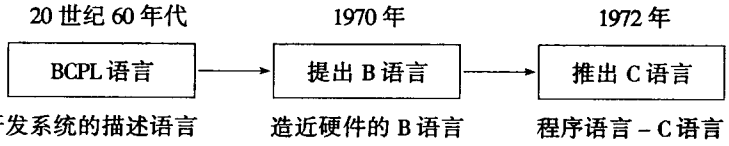

图1－4　C语言的发展

目前广泛流行的各种版本的C语言编译系统虽然基本相同，但它们之间也有一些差异。在微机上使用的Microsoft C、Turbo C、Quick C、Borland C等，它们的不同版本也略有差异，因此读者需要了解所用的计算机系统的C编译的特点和规定。

三、C语言的特点

C语言之所以能存在和发展，并且有很强的生命力，是因为它有如下的特点：

1．语言简洁、紧凑，使用方便、灵活

C语言一共有32个关键字，9个控制语句，压缩了一些不必要的成分，程序书写形式自由，语句简练。

2．运算丰富，适用的范围广泛

C语言一共有34种运算符，它把括号、赋值符号、强制类型转换符号等都作为运算符处理，从而使C语言的运算类型极为丰富，表达式类型多样合理灵活使用各种运算，

可以实现在其他高级语言中难以实现的运算和操作。

3．数据结构丰富，具有现代语言的各种数据结构

C语言的数据类型有：整型、实型、字符型、数组类型、指针类型、共用体类型等。用这些类型能实现各种复杂的数据结构（如链、表、树、栈等）的运算。尤其是指针类型的数据，使用起来灵活多样。

4．具有结构化的控制语句

在C语言中用if…else语句、while语句、switch语句、for语句等可以实现程序中所有的控制结构。另外，函数是C语言程序的基本单位，用它作为程序块的基本单元，以实现程序的模块化。C语言是结构化的理想语言。

5．编程限制少，程序设计自由度大

一般的高级语言语法规定和检查比较严格，能查出几乎所有的语法错误。而C语言允许程序的编写者有较大的自由度，因此放宽了语法检查。例如，对数组下标越界不作检查。变量的类型使用也比较灵活，例如，整型量与字符型数据可以通用，使某些运算变得更加简单、直接。编写者应当仔细检查源程序，保证其正确性。

6．可直接对硬件操作

C语言是一种高级语言，同时提供了类似汇编语言的大部分功能，如能直接访问物理地址，能进行位操作。这就使C语言既具有高级语言的功能，又兼备低级语言的许多功能，可以用来编写系统程序。

7．目标代码的质量高，程序执行的效率高

C语言生成的目标代码一般只比汇编语言生成的目标代码的效率低10%～20%。

8．程序的可移植性好

C语言程序可以基本上不作修改就能用各种型号的计算机和操作系统，使它具备了良好移植性。

9．C语言的弱点

C语言本身也有其弱点：①运算符及优先级多，不容易记忆。②C语言的语法限制不太严格，这在增加程序设计的灵活性的同时，在一定程度上降低了某些安全性，这对程序设计人员提出了更高的要求。

总之，由于C语言的上述特点，使C语言越来越受到程序设计人员的重视，并且已经在广泛的领域里得到应用。因而学习和使用的人也越来越多。

第二节　C语言程序的结构与书写规则

一、C程序的总体结构

下面通过几个简单的示例，介绍C语言程序的基本构成和书写格式，使读者对C语言程序有一个基本的了解。在此基础上，再进一步了解C语言程序的语法和书写规则。一个完整的C语言程序结构有以下两种表现形式：

仅由一个main（）函数（又称主函数）构成。

例1．1　求三个数的平均值的C语言程序。

```
/*功能：求三个数的平均值*/
main ()                              /* main () 称为主函数*/
{float a, b, c, ave;                 /*定义a, b, c, ave为实型数据*/
a=17;
b=19;
c=14;
ave= (a+b+c) /3;                     /*计算平均值*/
printf ("ave=%f\n", ave);            /*在屏幕上输出ave的值 */
}
```

程序运行结果：

```
ave=16．666666
```

由一个且只能有一个main () 函数和若干个其他函数结合而成。

其中：若干个其他函数由用户自己设计。

例1．2　输出两个数中的较大值的C语言程序。

```
/*功能：输出两个数中的较大值 */
main ()                                  /*主函数*/
{int num1, num2, max;                    /*定义num1、num2、max为整型变量 */
scanf ("%d,%d", &num1, &num2);           /*由键盘输入num1、num2的值 */
printf ("max=%d\n", max (num1, num2));
/*在屏幕上输出调用max函数的返回值 */
}
/*用户设计的函数max () */
int max (int x, int y)                   /* x和y分别取num1和num2传递的值*/
{if (x>y) return x;                      /*如果x>y，将x的值返回给max */
else return y;                           /*如果x>y不成立，将y的值返回给max */
}
```

程序运行情况：

```
25，18↙ ("↙"表示按回车键，以下相同)
max=25
```

在以上两个示例中，例1．1所示的C语言程序，仅由一个main () 函数构成，它相当于其他高级语言中的主程序；例1．2所示的C语言程序，由一个main () 和一个其他函数max () 构成，函数max () 相当于其他高级语言中的子程序。

3．几点说明

在编写C语言程序时有以下几方面的其本要求：

(1) 在C语言程序中，大小写字母有着严格的区分，并以小写为主。对于每一行的起始位置，C语言并不要求。

例1．3　在计算机屏幕上显示下列字符串

```
Hello, students.
This your first C program..
```

C 源程序如下：

```
main ( )
{
printf ("Hello, students. \ n") ;   /* 显示（输出）第一行字符串，\ n 表示换行 */
printf ("This your first C program.. \ n") ;   /* 显示（输出）第二行字符串 */
}
```

程序运行结果：

```
    Hello, students.
    This your first C program..
```

(2) 对于每个经编译后生成可执行文件的 C 程序，必须且仅有一个 main ( )，我们称之为主函数。main ( ) 函数中所指定的功能，称为函数体。main ( ) 中第一个左花括号和最后一个右花括号则分别表示函数体的开始和结束。

例 1. 4 通过键盘输入两个整数，计算其平均数并在屏幕上显示

源程序如下：

```
main ( )
{
int x , y;                                  /* 定义两个整型变量 */
float z ;                                   /* 定义一个实型变量 */
scanf ("d% d %", &x , &y ) ;                /* 从键盘输入 x，y 值 */
z = ( x + y) /2. 0 ;                        /* 计算 x，y 的平均值 */
printf ("The average value is %f", z );     /* 输出平均值 z */
}
```

程序运行情况：

```
10 20↙
The average value is 15. 000000
```

(3) C 语言程序的基本程序单位是函数。构成函数体的单位是语句，每个语句以分号 (;) 作为语句的结尾。一个完整的 C 语言程序，除了必须有一个主函数 main ( ) 外，一般还要编写大量的自定义函数，供主函数或其他函数调用，也还要调用大量的由 C 语言系统提供的标准库函数，如上例中 scanf、printf 都是标准函数。

(4) 为了增加 C 程序的易读性，以便进行程序的交流。程序中的/ * …… * /为插入的注释。必须以 / * 开头，以 * /结束。这种注释不能在一个语句行中间或前面。一般说来，当注释加在函数开头时，用于说明函数的功能；注释加在变量后面时，用于表示变量的含义；注释加在语句后面时，表示语句的作用。一个易读的程序，除了应具备良好的结构或算法，还应有必要的注释。

例 1. 5 将例 1. 2 按以下结构组织并完成输出两个数中的较大值的 C 语言程序。

```
/* ____________________________________
```

程序名 EXAMPLE．C 功能为：输入 a，b 两个数，输出其中最大数。

```
_____*/
#include <stdio.h>                          /*包含输入输出头文件*/
float max(float x, float y)                 /*定义求 x，y 的最大值的函数 max*/
{ float z;                                  /* 定义实数变量 z */
if(x>y) z=x;                                /* 如果 x 大于 y，则 x=>z */
else z=y;                                   /* 否则 y=>z */
return z;                                   /* 返回 z，则 max 函数值为 z 的值 */
}
main()                                      /* 定义主函数 */
{ float a, b, c;                            /*定义实数变量 a，b，c */
printf("Please input two numbers (a, b):"); /*屏幕出现提示信息*/
scanf("%f, %f", &a, &b);                    /*从键盘输入 a，b*/
c=max(a, b); /* c 等于 a，b 中的最大数 */
printf("%f,%f, the max is %f\n", a, b, c);  /*输出 a，b 和最大数 c 的值*/
}
```

请读者上机调试程序运行情况，并比较例 1．5 和例 1．2 在程序组织结构上有什么区别。

二、函数的一般结构

1．无参函数

无参函数是指函数被调用时，主调函数无须传送数据给被调函数。无参数函数一般被用来执行指定的一组操作，可以带回或不带回函数值。其一般结构为：

[类型标识符]　函数名（[void]）
{[说明部分]
语句序列
}

本书符号约定如下：

[……]——加方括号时，表示其内容既可以指定，也可以缺省。

……——表示前面的项可以重复。

注意：未加任何符号的表示该项为必选项。

“用类型标识符”指定函数的类型，即函数带回来的值的类型。对于无参函数一般不需要带回函数值，因此可以不写类型标识符。

例 1．6　定义一个无参函数

```
print-message()
{
printf("How do you do?  \n");
}
```

2．有参函数

在调用函数时，主调函数可以将数据传给被调函数使用，被调函数中的数据也可以带回来供主调函数使用。其一般结构为：

[类型标识符] 函数名（形式参数表列）

{[说明部分]

语句序列

}

例 1．7 定义一个有参数函数

```
int max (int x, int y)                    /*形式参数说明*/
{ int z ;                                 /*函数体中的说明部分*/
z = x > y ? x : y ;                       /*将 x, y 中较大的数赋给 z*/
return (z) ;
}
```

3．几点说明

(1) 函数说明部分

函数说明部分由函数类型（可缺省）、函数名和函数形式参数表列（简称形参表）三部分组成。其中：函数形参表的一般格式为：

([数据类型参数 1,][数据类型参数 2,] ……)

例如，上例中的函数说明部分 int max (int x, int y)，其各部分含义对应如下：

int——函数类型

max——函数名

int x, int y——函数参数表

(2) 函数体部分

函数体部分由函数说明部分以下的一对大括号“{}”内的若干条语句构成。函数体一般又由数据说明部分和函数执行部分两部分构成，如果一个函数内有多对大括号，则最外面的一对大括号是函数体的范围。

·说明部分

说明部分由变量定义、自定义函数声明、外部变量说明等部分组成，其中变量定义是主要的。例如，例 1．2 中 main () 函数体里的“int num1, num2;”语句，定义了 2 个整形变量 num1 和 num2。

·函数执行部分

函数执行部分一般由若干条可执行语句构成。例如，在例 1．2 的 main () 函数体中，除变量定义语句“int num1, num2;”外，其余 5 条语句构成该函数可执行语句部分。

有关函数的详细内容，将在后续章节介绍。

三、C 源程序书写的一般规则

1．C 程序书写格式自由，一行内可以写入几个语句，一个语句也可以分开写在多行上。

2．各语句之间用分号分隔。分号（语句的结束标志）是语句的必要组成部分，分号

是不可缺少。即使是程序的最后一个语句也必须有分号。

3．C语言本没有输入输出语句。其输入输出功能是由系统提供的库函数 scanf 和 printf 等函数来实现的，即C对输入输出实行“函数化”。

例如，例1．5中的程序也可以写为：

```
# include < stdio. h >
float max (float x, float y)
{ float z; if (x > y) z = x ; else z = y ; return z ;}
```

为了使程序的书写格式更能适应于人们的阅读习惯，程序设计者书写程序时通常采用“缩排”方式。即属于较内层的语句从行首缩进若干列，并使属于同一结构层次的语句对齐。在这一点上应养成一种良好的习惯。

第三节　C语言的语句

高级语言源程序的基本组成单位是语句。C语言的语句分为基本语句和复合语句；基本语句按其功能可分为两类：如图1－5所示。

1．描述计算机要进行的操作和运算（如赋值语句），这类语句称运算型语句。

2．控制操作和运算的执行顺序（如循环控制语句），这类语句称流程控制语句。

3．变量定义和声明语句统称为数据描述语句。

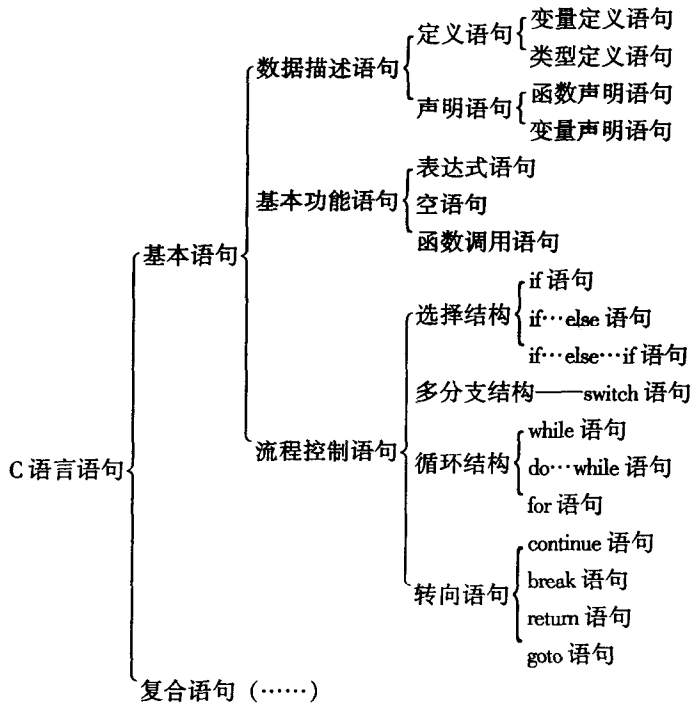

图1－5　语句分类示意图

第四节 Turbo C 的基本操作

一、源程序的输入到执行

源程序：用高级语言所提供的语句和函数写出的语句序列称为源程序。程序从输入到运行一般需要如下几步：

1．编辑

编辑源程序的作用是建立与修改源程序。源程序是经 ASC I I 码的方式输入到内存中。DOS 中 EDIT、WINDOWS 中的记事本及 Turbo C 的集成环境中均带有编辑功能，C 语言源程序一般扩展名为．c。

2．编译

编译是将编写好的源程序翻译为二进制的目标代码，由相应的编译软件来完成。不同的高级语言需要不同的编译程序。编译以后产生的二进制的目标代码一般扩展名为 ．obj，它不能直接运行。

3．连接

编译产生的目标二进制代码是对每一个模块直接翻译产生的，需要将各个模块与系统模块相连，才可以产生扩展名为 ．exe 的可执行文件。

4．执行

运行可执行的 exe 文件就可以获得所需要的结果。

二、在 Turbo C 下运行 C 程序的步骤

Turbo C 是一个自带编辑器、编译器、连接器和一些实用程序的综合软件，把 C 程序从输入到执行的几个过程集成在一个窗口环境，在集成环境下可直接建立和运行 C 程序，具有简便易学好用的特点，大大方便了用户的程序开发。假设 Turbo C 系统已装入机器。下面以一个具体的程序为例来看 Turbo C 的运行步骤：

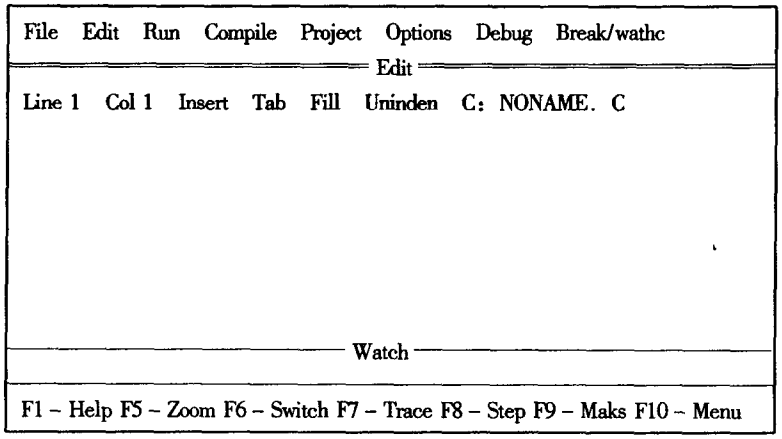

图 1－6 Turbo C 的主屏幕