贵州省高等院校非计算机专业

计算机基础课程教材

第二册

C语言教程

主编 李祥 副主编 谢晓尧

贵州人民出版社

贵州省高等院校非计算机专业计算机基础课程教材

C语言教程

主 编 李 祥

副主编 谢晓尧

贵 州 人 民 出 版 社

贵州省高等院校非计算机专业计算机基础课程教材

C 语言教程

李祥　主编　　谢晓尧　副主编

责任编辑　李立朴　孟筑敏　徐一
封面设计　张世申
版式设计　徐一

出版发行　贵州人民出版社
社　　址　贵阳市中华北路 289 号
邮政编码　550001

印　　刷　贵阳科海印务有限公司
开　　本　787×1092 毫米　1/16
字　　数　35 万字
印　　张　14.5
印　　数　1～4000 册
版　　次　2002 年 10 月第 1 版
印　　次　2002 年 10 月第 1 次印刷

书　　号　ISBN7—221—05980—2/G·2213
定　　价　20.00 元

内容提要

本书是按照教育部计算机基础教学三层次要求由贵州省教育厅组织有关专家编写的教材。全书共分 15 章，主要内容包括：C 语言的发展史、特点及构成；各种数据类型、各类运算和基本语句；结构化程序设计及流程控制；指针、数组和各种函数；结构体与共享体；文件和位运算；编译预处理。书中最后介绍了用 C 语言实现的常用算法和 C 语言的上机指南，并给出了大量实用程序例子、习题和上机实习题。

本书主要是理工院校非计算机专业计算机基础第二层次教学用书，同时也是“计算机等级考试”二级的指定教材。

目 录

第1章 C语言概论

1.1 C语言的发展史

随着电子计算机的迅速发展和广泛应用，C程序设计语言（以下简称C语言）在计算机软件开发中的作用日益重要，已成为了国际上流行的、很有发展前途的计算机程序设计语言。如果以层次结构划分语言的话，C语言是介于汇编和高级语言之间的（见图1.1），所以，他适合作为系统描述语言，既可用来写系统软件，也可用来写应用软件。

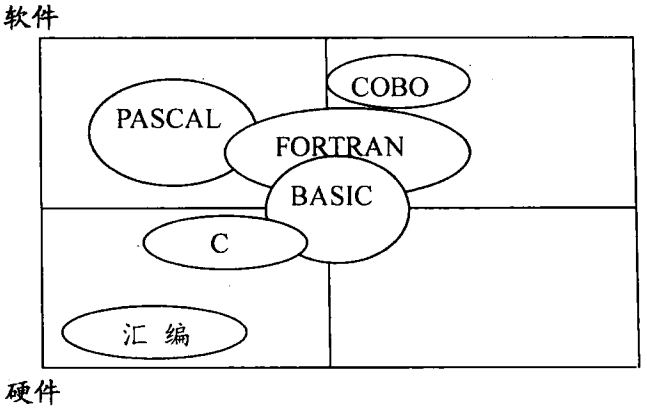


图1.1 计算机程序语言的层次结构

C语言的研究和诞生起源于系统程序设计的深入研究和发展，他作为书写UNIX操作系统的语言，伴随着UNIX的发展、流行而得到普及。C语言的根源可以追溯到ALGOL 60高级语言。

1960年，来自丹麦、英国、法国、德国、荷兰、瑞士和美国的十三名代表自1月11日至16日在巴黎举行了会议，共同提出了ALGOL 60高级程序设计语言的正式报告。这是一种面向问题的高级语言，他离硬件较远，不宜用来编写系统程序。

1963年，英国的剑桥大学推出了CPL（Combined Programming Language）语言。

CPL语言在ALGOL 60的基础上接近硬件一些，但规模比较大，难以实现。

1967年，英国剑桥大学的马丁·理查德（Martin Richards）在CPL语言的基础上，实现并推出了BCPL（Basic CPL）语言。BCPL语言是计算机软件人员在开发系统软件时，作为记述语言使用的一种程序语言。BCPL语言的突出特点是：

他是结构化的程序设计语言；

能够直接处理与机器本身数据类型相近的数据；

具有与内存地址对应的指针处理方式。

1970年，美国贝尔实验室的肯·汤普森（Ken·Thompson）以BCPL语言为基础，设计了一种类似于BCPL的语言，称为B语言（取BCPL的第一个字母）。他用B语言在PDP-7机上实现了第一个实验性的UNIX操作系统。

1972年，贝尔实验室的戴尼斯·M·利奇（Dennis·M·Ritchie）和布朗·W·卡尼汉（Brian·W·Kernighan）为克服B语言的诸多不足，在B语言的基础上重新设计了一种语言，由于是B的后继，故称为C语言（取BCPL的第二个字母）。

1973年，贝尔实验的汤普森和利奇合作，用C语言重新改写了UNIX操作系统，此后，随着UNIX操作系统的发展，C语言的应用越来越多，影响越来越大。

1977年，出现了独立于具体机器的C语言编译版本。C语言的独立和推广也推动了UNIX操作系统在各种机器上的迅速实现。

1978年，卡尼汉和利奇正式出版了《The C Programming Language》一书，此书即成为现在广泛使用的C语言版本基础，亦被称为标准C语言。

1983年，美国国家标准化协会（ANSI）颁布了C语言的新的标准版本“ANSI C”。“ANSI C”比标准C有很大的扩充和发展。

1987年，ANSI在综合各种C语言版本的基础上，又推出了美国标准C语言版本，他是目前功能最完善、性能最优良的C语言新版本，称之为“87 ANSI C”。现在流行的C编译系统都是以他为基础的。

C语言和其他高级语言的发展关系如图1.2所示。

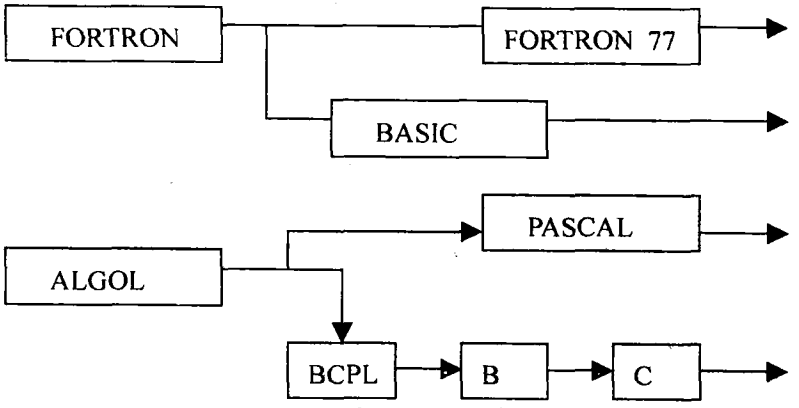


图1.2 C语言的发展历史

1.2 C语言的特点

一种程序设计语言之所以能存在和发展，并具有生命力，总是有其自身与众不同的特点。C语言的主要特点如下：

1. C语言是处于汇编语言和高级语言之间的一种记述性程序语言。C语言在程序语言世界中所处的位置如图1.1所示。可以看出，C语言是比较靠近硬件与系统的，即与汇编语言比较接近。C语言既有面向硬件和系统、像汇编语言那样可以直接访问硬件的功能，又有高级语言面向用户、容易记忆、便于阅读和书写的优点。

2. C语言是一种结构化程序设计语言，即程序的逻辑结构可以用顺序、选择和循环三种基本结构组成。C语言具有诸如 if～else、for、do～while、while、switch～case 等结构化语句，十分便于采用由顶向下、逐步细化的结构化程序设计技术。因此，用C语言编制的程序，具有容易理解，便于维护的优点。

3. C语言是便于模块化软件设计的程序语言。C语言程序的函数结构，十分利于把整体程序分割成若干相对独立的功能模块；并且为程序模块间的相互调用以及数据传递提供了便利。这一特点也为把大型软件模块化，由多人同时进行集体性开发的软件工程技术方法提供了强有力的支持。

4. C语言备有种类丰富的运算符。除一般高级语言使用的+、-、*、/ 四则运算及与（AND）、或（OR）、非（NOT）等逻辑运算功能外，还可以实现以二进制位（bit）为单位的位与（&）、位或（|）、位非（～）、位异或（^）以及移位操作（>>、<<）等位运算。并且具有如 a++、b--等单向运算和 +=、-=、*=、/=、等复合运算功能。

5. C语言程序中可以使用如#define、#include 等编译预处理语句，能够进行字符串或特定参数的宏定义，以及实现对外部文本文件的读取和合并。同时还具有#if、#else 等条件编译预处理语句。这些功能的使用提高了软件开发的工作效率，并为程序的组织和编译提供了便利。

6. 数据结构丰富，具有现代化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、结构体类型、共享体类型等。能用来实现各种复杂数据结构（如链表、树、栈等）的运算。尤其是指针类型数据，使用起来比 PASCAL 更为灵活、多样。

7. C语言程序具有较高的移值性。C语言的语句中，没有依存于硬件的输入/输出语句。程序的输入/输出功能是通过调用输入/输出函数实现的。而这些函数是由系统提供的独立于C语言的程序模块库，因此，C语言程序本身并不依存于机器硬件系统，从而便于

在硬件结构不同的机种间实现程序的移植。

8. 生成目标代码质量高，程序执行效率高。一般只比汇编程序生成的目标代码效率低10%～20%。

9. 语法限制不太严格，程序设计自由度大。例如，对数组下标越界不作检查，由程序编写者自己保证程序的正确。对变量的类型使用比较灵活，例如，整型量与字符型数据以及逻辑型数据可以通用。一般的高级语言语法检查比较严，能检查出几乎所有的语法错误。而C语言允许程序编写者有较大的自由度，因此放宽了语法检查。程序员应当仔细检查程序，保证其正确，而不要过分依赖C编译程序去查错。“限制”与“灵活”是一对矛盾。限制严格，就失去灵活性；而强调灵活，就必须放松限制。一个不熟练的人员，编一个正确的C程序可能会比编一个其他高级语言程序难一些。也就是说，对用C语言的人，要求对程序设计更熟练一些。

由以上的分析可见，C语言把高级语言的基本结构与低级语言的高效实用性很好地结合起来，使其成为一个迅速普及和应用的现代通用程序设计语言。

第2章 C语言的构成

2.1 C语言的构成

下面通过两个简单的C语言源程序，介绍C语言程序的基本组成和结构。

例2.1 一个打印一行文本的C语言程序。

```
/* 一个打印一行文本的C语言程序 */
main ( )
    {
        printf( "Welcome to c!\n" );
    }
```

输出结果：

```
Welcome to c!
```

说明：

1. 程序的第一行“ /* 一个打印一行文本的C语言程序 */ ”，是注释行，程序员向程序代码中加入注释是为了提高程序的可读性，注释行不参加编译，C语言编译器会忽略掉注释行，注释内容放在“/*”和“*/”之间。

2. 第二行“ main ()”是每一个C语言程序必须写的，他是一个函数。函数是C语言程序的基本单位，C语言程序是由一个或多个函数组成的。其中必须有一个函数是main。他后面的“()”内可有函数的参数，也可为空，但括号不能省。函数由“{”开始，到“}”结束。

3. 第四行“printf(“Welcome to c!\n”);”是一条C语言语句，每条C语言语句都必须以分号结束，C语言中分号也可称为语句结束符。“printf”是系统提供的库函数。printf函数的功能是把要输出的内容送到显示器去显示。

例 2.2　求两个数的和。

```
#include <stdio.h>                          /*“文件包含”命令*/
  int add(x, y)                            /*定义求和函数，有两个参数*/
  int x;                                   /*参数 x 是整型*/
  int y;                                   /*参数 y 是整型*/
{
    int sum;                               /*定义变量 sum*/
    sum = x + y;                           /*求出 x+y 的值赋给变量 sum*/
    return sum;                            /*返回 sum 的值*/
}
main( )                                    /*主函数*/
{
    int a;                                 /*定义变量 a*/
    int b;                                 /*定义变量 b*/
    scanf (“%d%d”, &a, &b);                /*从键盘输入两个数分别赋给 a、b*/
    printf (“%d\n”, add (a, b));           /*求出 a，b 的和，并显示*/
}
```

输入：

50　60

输出结果：

110

说明：

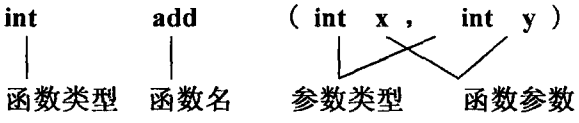
本程序有两个函数，函数编写的先后顺序可以任意，但程序是从主函数开始执行，并在主函数结束。在主函数中可以调用其他函数。

使用的函数可以是系统提供的库函数，也可以是程序员自己定义的，如果是系统提供的库函数，则要用“#include”包含该函数的库文件。

C 语言规定变量在使用之前必须定义，局部变量的定义应该放在函数内的前面。

函数由两部分组成：

1）函数的首部。



2）函数体。即函数首部下面的大括号“｛｝”内的部分。

由以上两例可看出 C 语言程序的一般组成形式如下：

包含文件

子函数类型说明

全程变量定义

```
main( )
{
    局部变量定义
    <程序体>
}
函数返回值类型  f1 ( 函数参数表 )
{
    局部变量定义
    <程序体>
}
函数返回值类型  f2 ( 函数参数表 )
{
    局部变量定义
    <程序体>
}
.
.
函数返回值类型  fn ( 函数参数表 )
{
    局部变量定义
    <程序体>
}
```

2.2 源程序的书写格式和编程风格

1. C 程序书写格式自由，一行内可以写几个语句，一个语句也可以分写在多行上。为清晰起见，一般在一行内写一个语句。

2. 为了能清楚地体现出程序结构，程序员应采用锯齿形程序格式，即将 if 语句的内嵌语句和循环结构中的循环体在书写时向右缩进几列。如果是多层嵌套则多层缩进。

3. 对变量和函数的命名应与变量或函数的功用相关，以便于程序阅读。

4. 在程序中应加入适当注释，以便于程序阅读。

5. 注意控制字母的大小写，除宏名以外的名字一般应用小写字母。

第3章 基本数据类型及运算

3.1 C语言的标识符

和其他高级语言一样，C语言用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符。简单地说，标识符就是一个名字。在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。C语言规定，标识符只能是字母(A～Z，a～z)、数字(0～9)、下划线(_)组成的字符串，并且其第一个字符必须是字母或下划线。

以下标识符是合法的：

a，　　x，　　x3
BOOK1　，　　sum5

以下标识符是非法的：

3s　　以数字开头
s*T　　出现非法字符*
-3x　　以减号开头
bowy-1　出现非法字符 -(减号)

在使用标识符时还必须注意以下几点：

1．标准C不限制标识符的长度，但他受各种版本的C语言编译系统限制，同时也受到具体机器的限制。例如在某版本C语言中规定标识符前八位有效，当两个标识符前八位相同时，则被认为是同一个标识符。

2．在标识符中，大小写是有区别的。例如BOOK和book是两个不同的标识符。

3．标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便于阅读理解，做到“顾名思义”。

4. 标识符不能与C语言系统的关键字重复，下表是C的关键字。

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	static	sizeof	struct	switch	typedef	union
unsigned	void	volatile	while			

3.2 常量与变量

3.2.1 常量

在程序执行过程中，其值不发生改变的量称为常量。

直接常量(字面常量)：

整型常量：如12、0、-3；

实型常量：如4.6、-1.23；

字符常量：如‘a’、‘b’。

符号常量：用标示符代表一个常量。在C语言中，可以用一个标识符来表示一个常量，称之为符号常量。

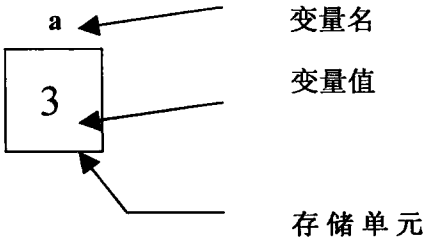
符号常量在使用之前必须先定义，一般形式为：#define 标识符 常量

#define 是一条预处理命令（预处理命令都以“#”开头），称为宏定义命令（在后面预处理程序中将进一步介绍），其功能是把该标识符定义为其后的常量值。一经定义，以后在程序中所有出现该标识符的地方均代之以该常量值。

习惯上符号常量的标识符用大写字母，变量标识符用小写字母，以示区别。

3.2.2 变量

其值可以改变的量称为变量。一个变量应该有一个名字，在内存中占据一定的存储单元。变量定义必须放在变量使用之前。一般放在函数体的开头部分。要区分变量名和变量值是两个不同的概念。



习惯上变量名用小写字母表示，以增加可读性。

在选择变量名和其他标识符时，应注意做到“见名知意”，即选有含意的英文单词（或其缩写）作标识符，如 name、day、city、country 等，除了数值计算程序外，一般不要用代数符号（如 b、c、xl 等）作变量名，以增加程序的可读性。这是结构化程序的一个特征。本书在一些简单的举例中，为简单起见，仍用单字元的变量名（如 a、b、C 等）请读者注意不要在其他所有程序中都如此用。

在 C 语言中，要求对所有用到的变量作强制定义，也就是“先定义，后使用”，这样做的目的是：

1．凡未被事先定义的，不作为变量名，这就能保证程序中变量名使用的正确。例如在定义部分写了：

```
int  Student;
```

而在执行语句中错写成：Starenr

如： Starenr =30

在编译时检查出 Starenr 未经定义不作为变量名；因此输出“变量 Starenr 未经说明”的信息，便于用户发现错误，避免变量名使用时出错。

2．每一个变量被指定为一确定类型，在编译时就能为其分配相应的存储单元。

如指定 a、b 为 int 型，如果所用的是 IBM PC 的 MS C，则为 a 和 b 各分配两个字节，并按整数方式存储数据。

3．每一变量属于一个类型，就便于在编译时据此检查该变量所进行的运算是否合法。例如，整型变量 a 和 b，可以进行求余运算：

```
a%b
```

%是“求余” 运算符，得到 a%b 的整余数。如果将 a、b 指定为实型变量，则不允许进行“求余”运算，在编译时会指出有关“出错信息”。

下面各节分别介绍整型、实型（浮点型）、字符型数据。

3.3 C 的基本数据类型的定义、初始化及使用

在本章中，我们将介绍数据类型的说明。所谓数据类型是按被定义变量的性质、表示形式、占据存储空间的多少、构造特点来划分的。在 C 语言中，数据类型可分为：基本数据类型，构造数据类型，指针类型，空类型四大类。

1．基本数据类型：基本数据类型最主要的特点是，其值不可以再分解为其他类型。也就是说，基本数据类型是自我说明的。