HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

C语言基础教程

C YU YAN JI CHU JIAO CHENG

陈宝贤　主编

华中理工大学出版社

前　言

C 语言是由美国贝尔实验室 D. M. Ritchie 等人为描述和实现 UNIX 操作系统而研制的一种程序设计语言。由于 C 语言具有结构清晰、紧凑，语言表达能力强，处理灵活等特点，既能与其它高级语言一样可以用来编制应用程序，又具有"低级语言"的功能可编制汇编语言所能编制的绝大部分程序，因而得到不断发展、完善。特别是进入 80 年代后，C 语言越来越受到重视。近年来，在国内外得到迅速推广使用。目前，我国大学、专科学校的计算机专业已普遍开设 C 语言课程，其它相关专业也陆续增开 C 语言课程。掌握 C 程序设计基本知识和方法，已成为计算机应用人员开发应用软件的基本要求。

我们认为，大专学校 C 语言程序设计课程的教学应达到如下要求：

(1) 能正确、合理地使用 C 语言的各种数据类型，掌握各种类型数据的定义及构造方法；

(2) 熟练掌握 C 语言的基本形式、语法、语义；

(3) 掌握各种运算符的运算规则及相应表达式的使用；

(4) 掌握 C 语言程序设计基本方法，掌握函数、函数调用等基本概念和基本用法等；

(5) 掌握 C 语言指针概念及其应用；

(6) 掌握 C 语言文件的基本概念及其使用；

(7) 熟练掌握 Turbo C 上机操作。

根据上述基本教学要求，在编写中努力做到：

(1) 基本概念叙述准确，分解难点，通俗易懂；

(2) 内容安排，做到由浅入深，程序实例，难易适中，有启发性，便于教学和自学；

(3) 注重理论与实践相结合的原则，使读者能较好地掌握 C 程序编制的初步技能。

《C 语言基础教程》全书共 12 章，书中程序以 Turbo C 2.1 系统为运行环境。采用本书作 C 语言课程教材时，根据不同专业、不同层次学生的情况，可安排教学课时 40～55 学时，上机时数应至少达到 26 机时。

本书可作为大专学校、中专学校 C 语言程序设计课程的教材，亦可作为计算机 C 语言培训教材，也可作为 C 语言初学者的参考书籍。

在本书编写过程中，得到了湖南省计算机专科学校校、系有关领导和老师的热情支持，在此向他们表示衷心的感谢。

本书由湖南省计算机专科学校计算机系陈宝贤任主编，湖南省计算机专科学校计算机系李明钧、李根强和湖南财经学院姚靠华任副主编。参编人员还有：湖南省计算机专科学校周学毛，湖南大学曾云，湖南财经专科学校刘红冰。担任各章编写任务的是：陈宝贤（第三、四、五、六章），李明钧、李 素（第八、十一章）。李根强（第九、十章），周学毛（第一章），曾 云（第十二章），姚靠华（第二章），刘红冰（第七章）。湖南计算机专科学校实验中心郭林超参加了部分章节的程序调试工作。

由于编者水平有限，疏漏之处在所难免，欢迎使用本教材的教师、学生以及其他读者批评指正，万分感谢。

编著者

1996 年 5 月

目　录

第一章　C语言基础

程序设计语言是计算机系统中最重要的软工具，是提供描述计算过程而设计的一种具有语法语义描述的符号集。从计算机诞生以来，至今已问世的各种程序设计语言有成千上万种。程序设计语言的发展大致可分为如下几个时期：

(1) 低级语言时期（机器语言、汇编语言）；

(2) 高级语言初创时期（FORTRAN、ALGOL、COBOL 等语言）；

(3) 高级语言发展初期（LISP、APL、PL/1、BASIC 等语言）；

(4) 结构程序设计语言时期（PASCAL、Ada、C 等语言）；

(5) 第四代语言（ADF、IDEAL、MAPPER 等语言）。

在结构程序设计语言时期推出的绝大多数语言都强调“抽象”，基本上都不适合于开发系统软件。而同时期推出的C语言，由于具有许多类汇编语言特征，有丰富的运算符（包括位运算符），有现代的程序控制结构与数据结构，使它在系统软件的开发应用领域中获得成功，目前已成为最流行的计算机语言之一。从PC机、工作站到大型、巨型计算机上都已配有C语言的编译器。

1.1　C语言的由来与特点

1.1.1　C语言的由来

C语言是广泛流行的程序设计语言，在系统软件开发，以及事务处理、科学研究、模拟等各个领域的应用软件开发中，均获得较大成功。

C语言是由美国 AT&T(American Telephone & Telegram)贝尔实验室 D. M. Ritchie 在B语言的基础上研制成功的。最初的C语言只是为描述和实现 UNIX 操作系统提供一种工作语言而设计的。在C语言的产生、使用推广过程中，贝尔实验室的 K. Thompson 亦起到了重要作用。

C语言的起源可以追溯到 ALGOL 60。其形成演变过程为：ALGOL 语言(1960 年)——CPL 语言(1963 年)——BCPL 语言(1969 年)——B 语言(1970 年)——C 语言(1973 年)。ALGOL 60 是一种面向问题，用于科学计算的语言，与计算机硬件距离较远，不宜于用来编写系统程序。但 ALGOL 60 具有精确而完备的文本，采用了精确而形式化的语法描述体系，孕育了许多很有用的程序设计与程序设计语言思想，这些方面都影响着以后程序设计语言的设计。由英国剑桥大学推出的 CPL(Combined Programming Language)语言，企图在 ALGOL 基础上和计算机硬件更接近一些，但其规模大，难以实现。因而英国剑桥大学又推出简化 CPL 语言——BCPL (Basic Combined Programming Language)语言。1970 年贝尔实验室的 Ken Thompson(UNIX 操作系统的创始人)在实施 UNIX 系统时对 BCPL 作了进一步简化，设计了非常简单而又接近硬件的 B 语言，但 B 语言过于简单，功能有限，适用范围小。为此，

D. M. Ritchie 在 B 语言的基础上设计出 C 语言。C 语言保持了 BCPL 和 B 语言精练、接近硬件的优点，克服了 B 语言“无类型”的数据结构等缺点，系统地引进了多种基本数据类型，并导出了其它组合类型和函数。

1973 年 K. Thompson 和 D. M. Ritchie 用 C 语言重新改写了 UNIX 的内核，即 UNIX 的第 5 版本。1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们普遍注意。与此同时，C 语言的编译器也被移植到 IBM 360/ 370、VAX-11/780 等计算机。1977 年出现了不依赖于具体机器的 C 语言编译文本，推动了 UNIX 迅速在各种机器上实现，随着 UNIX 系统在各大学和公司的普及，C 语言也迅速得到了推广和应用。

为了给出 C 语言文本的详细的定义，Brian W. Kernighan 和 Ritchie 于 1978 年出版了称为 C 语言白皮书的“The C Programming Language”。1983 年，美国国家标准化协会(ANSI)根据 C 语言各种版本对 C 的发展和扩充，制订了新的标准，称为 ANSI C。此后的许多 C 语言的新版本都参照了此标准，这样就给 C 语言程序的移植创造了更有利的环境。在 80 年代，C 语言变得更为普及，几乎在每一种体系结构和操作系统上都有 C 语言的编译器，并成为 PC 机上的通用编程工具。在微型机上常见使用的 C 语言版本有 Microsoft C、Turbo C、Quik C。

1.1.2 C 语言的特点

C 语言是一种既适合系统程序设计，又适合应用程序设计的程序设计语言。C 语言已有国际标准 ISO/IEC 9899：1990，C 语言国家标准也已报批。C 语言的主要特点有以下几点。

1. C 语言表达能力强，通用性好

C 语言是面向结构程序设计的语言，通用性好，不局限于某种机器。可以完成通常由硬件实现的算术和逻辑运算，能有效地取代汇编语言来编写各种系统软件和应用软件。也可以实现数值计算、数据处理、CAD、人工智能等功能，所以它又是一种高效通用语言。

2. 具有丰富的数据类型和结构化的控制语句

C 语言具有在常规的基本数据类型的基础上，按层次构造各种结构类型。如数组、指针、结构体和联合体等。尤其是指针类型，能非常方便地对任意复杂的数据结构，进行数据处理。同时，它的各种结构化的控制语句功能强，足以描述和编写具有良好程序风格的程序。

3. 程序简洁清晰、模块化

一个 C 语言的程序，由若干个函数定义的集合构成。用函数作为程序模块，以实现程序的模块化。C 语言中的函数，提供了编制结构化程序的手段。使得程序结构清晰，易于阅读和维护。C 语言提供了丰富的运算符，使用灵活，使程序书写简洁，执行效率高。

4. C 语言生成的代码质量高，可移植性好

C 语言有些运算符直接反映了现代机器指令，可生成较短的机器代码，其生成代码效率仅比用汇编程序生成的目标代码效率低 10%～20%。

C 语言的移植性好，主要体现在下面两个方面：

(1) I/O 功能通过调用 I/O 函数库来实现，而这些函数是系统提供的独立于 C 语言的程序模块库，这样 C 语言本身可不依附于机器硬件系统。

(2) 一种机型的 C 语言核心编译器可以很小，系统的实用性部分和预处理部分均与机器无关，使它从一个系统到另一系统的移植改写很容易实现。

5. C 语言在编程上提供了较大的自由度

C 语言语法限制不太严格，对变量的类型使用比较灵活，类型检验较弱，安全性较差。C 语

言允许程序员有较大的自由度，也给开发者增加了出错的可能性。因而，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。

C语言的优点很多，但也有缺点。主要缺点是繁多的语言规则，难记易混；语言不严格，容易出错。

1.2 C语言的基本概念

1.2.1 基本符号

(1) 数字0～9共10个。

(2) 英文字母A～Z、a～z共52个。

(3) 特殊符号，通常由1～2个符号组成，主要用来表示运算符。例如：

+	−	*	/	%	<	<=	>	>=	→
==	! =	&&	‖	!	&	¦	~	=	.
++	−−	?:	<<	>>	()	[]	{}	,	;

1.2.2 标识符

C语言中所使用的符号常量、变量、函数、类型、文件都应有唯一的名称，通常用一串字符来命名，称为标识符。

标识符的命名规则：

(1) 必须以字母或下划线(下划线也起一个字母作用)开头，例如：_ ABC Lev5；

(2) 必须由字母、数字或下划线组成，长度一般不超过8个字符，Turbo C中规定长度不超过32个字符；

(3) 大小写字母含义不同，例如MAX，max表示不相同的标识符；

(4) 不允许使用保留字(或叫关键字)作标识符。

C语言的标识符可分成三类：

1. 保留字

保留字用来说明某一固定含义的字串，不能作它用。C语言新标准保留字为32个。名称如下：

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

volatile　　while

在 Turbo C 中还扩充了 asm pascal far huge near 等作为保留字，具体使用请参阅有关资料。

2. 特定字

具有特定含义的若干标识符，叫做特定字。如：define undef include ifdef ifndef endif line 等。这些特定标识符主要用在 C 语言的编译预处理中，不要作为一般标识符随意使用。

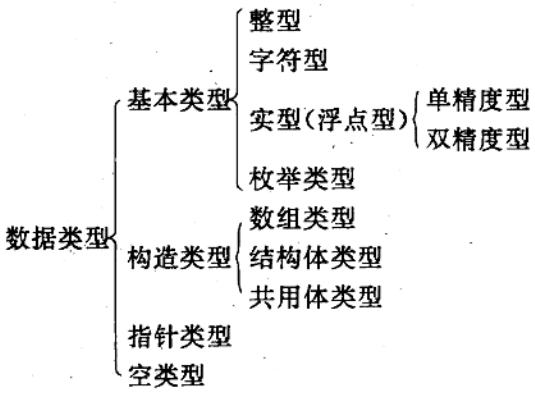
3. 一般标识符

由用户定义的标识符。例如：a sum day name sex L12 -a12 是合法**一般标识符**。

在用户定义标识符时，其名称应尽量做到见名知意，以便增加程序的可读性。

1.2.3 数据类型

C 语言提供了丰富的数据类型。C 语言的数据类型分类如下：



C 语言对数据类型的规定，说明了各种数据类型的可能取值和能在其上进行哪种运算。

C 语言没有提供逻辑数据类型，在逻辑运算表达式中，以非零值表示真(True)，以零值表示假(False)。

1.2.4 常量与变量

1. 常量

C 语言的数据有常量和变量之分。常量是指在程序运行过程中，其数值不能被改变的量称为**常量**。常量在使用中亦区分不同的数据类型。例如，12、－34、0 为**整型常量**，5.38、－3.14159 为**实型常量**，'a'、'A' 为**字符常量**。

常量也可以定义一个标识符来代表，称为**符号常量**。

2. 变量

变量是在程序运算过程中其值可以改变的量。每个变量都应有一个唯一的名称，即**变量名**。每个变量根据其类型在存储器中占据一定的存储单元，在该存储单元中存放变量的值。

C 语言中，使用变量时必须“先定义，后使用”。定义变量时，给出变量名称，并对每一变量指定确定的数据类型；凡未事先定义的变量名称，不能作为变量名。

1.2.5 语句

语句是程序的主要成分，**语句**的作用是向计算机系统发出操作指令。一个C语言语句经编译后可产生若干条机器指令。C语言的语句按大类分大致可分为简单语句和构造语句两大类。**简单语句**包括：表达式语句、变量说明语句、空语句等。**构造语句**包括：复合语句、控制语句等。在C语言中只有可执行语句，没有非执行语句。

C语言语句的**书写规则**：

(1) 每个简单语句的结束符是分号“;”；

(2) 复合语句由花括号“{ }”内若干个有序语句构成；

(3) 一个语句可以拆开写在几行上，不必加续行符；

(4) C语言允许一行上写几条语句；

(5) 语句行前空格数多少不影响语句，通常按一定规律缩进书写，以便增加程序可读性。

1. 变量说明语句

变量说明语句的作用是用来定义变量数据类型。一般格式为：

类型说明符 变量名1 [,变量名2,…];

```
例如： int a,b,c;                /*指定a、b、c变量均为整型变量 */
       float d;                  /*指定d变量为实型变量 */
       char e1,e2;               /*指定e1、e2变量为字符变量*/
```

2. 表达式语句

表达式语句的作用是通过赋值表达式语句改变变量的值，另外可进行函数调用。其一般格式为：

表达式;

```
例如： f=1.23;                   /*给变量f赋值*/
       printf("%f",y);           /*调用格式输出函数 */
```

由于C程序中大多数语句是表达式语句，因而有人把C语言称作**表达式语言**，使用表达式语句是C语言的一个特色。

3. 空语句

空语句在语法上占据一个简单语句的位置，执行空语句不执行任何操作。在程序中用来作为空循环的循环体，有时也用来作被转向点，提供标号出口。空语句仅由分号“;”组成，其格式为：

;

4. 控制语句

控制语句的作用是在程序中完成特定的控制功能。C语言的控制语句有：条件语句、多分支选择语句、转向语句、for循环语句、while循环语句、do—while循环语句、结束本次循环语句、中止执行switch或循环语句、从函数返回语句9种控制语句。

5. 复合语句

复合语句的作用是代替多个语句。用来表示一组相关的语句，便于程序的阅读，其次用于只允许出现一个语句的有关控制语句中，例如循环语句中的循环体、条件语句的内嵌语句等。

复合语句的格式为：

{[内部数据说明;]

语句序列;}

在复合语句中出现内部数据说明的复合语句,称为分程序。

复合语句的语句序列可以是简单语句,也可以是另一个复合语句,此时,形成了语句的嵌套层次结构。使用复合语句是C语言的特征之一,注意在复合语句的"}"之后不能加分号;

1.2.6 函数与程序

一个C语言程序可以由若干个源程序文件组成。一个源程序文件可以由若干个函数和预编译命令组成,源程序是C语言编译的文件模块。C语言对一个一个的源文件进行编译,然后连接装配成一个可运行的程序。

C语言程序的基本构成单元是函数。一个函数包含数据定义部分和执行部分。函数是程序的功能模块。

C程序总是由一个主函数main( )和若干被调用的函数组成。C语言程序运行从主函数main( )开始执行,一直到main( )函数运行终止结束。其它函数由主函数调用后执行,也可以由别的函数或自身调用后执行。同一个函数可以被一个或多个函数多次调用。函数构成C语言程序是C语言基本特征。

C语言系统将一些常用的功能模块编写成标准函数,提供用户使用,这类函数称为**库函数**。一般的C编译系统都有数百个库函数。

1.3 整型数据

1.3.1 整型常量

整型常量通常用十进制整数表示,也允许用八进制或十六进制整数表示,整型常数不能带小数点或逗号分隔。

十进制整数以数码直接开头。例如,124、-36、0。

八进制整数以前导零开头,在C新标准中不允许用8和9两个数码。例如。0124,相当于十进制数$1\times8^2+2\times8+4=84$。

十六进制整数以前导0x开头,在十六进制中,除数码0~9外,使用a~f(或A~F)数码,相应于十进制的10~15数值。例如,0x2C5,相当于十进制数$2\times16^2+12\times16+5=709$。

1.3.2 整型数据分类

C语言的整型数据按取值范围,即数据的二进制位数的多少分为短整型、基本整型、长整型3种。

C标准没有具体规定各类整型数据所占内存字节数,其二进制位数与各种机器的硬件有关。一般以一个机器字存放一个基本整型数据,长整型数据的字节数不少于基本整型,而短整型数据的字节数不长于基本整型,也允许3类数据的字节数相同。表1.1列出了IBM-PC及其兼容机所支持的整型数据类型,以及各类数据所占字节数、取值范围。

表 1.1 Turb C 所支持的整型数据

关键字	所占位数	取值范围
short[int]	8	−128～127
unsigned short [int]	8	0～255
signed short int	8	−128～127
int	16	−32768～32767
unsigned [int]	16	0～65535
signed int	16	−32768～32767
long [int]	32	−2147483648～2147483647
unsigned long [int]	32	0～4294967295
signed long int	32	−2147483648～2147483647

1.3.3 整型变量定义及其初始化

C 语言规定所有的变量在使用前都必须先定义，指定变量的数据类型。变量定义语句，一般放在一个函数的开头部分。对变量定义的格式为

类型名 变量名 1 [,变量名 2,…];

例如：
```
int x,n;          */指定变量 x、n 为整型变量*/
long a,b,c;       */指定变量 a、b、c 为长整型变量*/
unsigned v;       */指定变量 v 为无符号整型变量*/
```

变量的初始化是在定义变量的同时给变量赋初值，使这些变量在程序开始执行时就具有确定的值。变量初始化格式为：

类型名 变量名 1=常量 1 [,变量名 2=常量 2,…];

例如：
```
int  i=1,j=2;
```

等效于
```
int  i,j;
i=1;
j=2;
```

如果对 n 个变量赋同一个初值，如对 i,j 都赋初值 1，可以写成
```
int  i=j=1;
```

在一条变量初始化语句中，允许对有的变量赋值，有的变量不赋值。例如：
```
int  i, j=2;
```

等效于
```
int  i;
int  j=2;
```

在使用整型常量时，请注意以下两点：

(1) 对于超过−32768～32767 取值范围的整数，可以在整型常量后面加字母 L 或 l，表示该整数为长整型常量。

例如 1239L　　/* 表示整数 1239 是长整型数 */

(2) 只要整型常量的数值不超过变量的取值范围，这个整数就可以赋值给该变量。

例如，整数 52767 已超 int 型变量的取值范围，它不能赋值给 int 型变量，但可以赋值给

long int 型变量。

1.4 实型数据

1.4.1 实型常数

C 语言实型常数又称**浮点数**。它可由整数、小数、指数三部分组成。表示形式有以下两种：

(1) 小数形式。表示形式为：±n.n，±n.，±.n。其中，n 为 0～9 数码组成的数值。

例如 0.0，－12.3，.23，20.45 都是小数形式的浮点型常数。

(2) 指数形式。表示形式为：±nE±m，±n.nE±m，±.nE±m，±n.E±m。其中，n 为 0～9 数码组成的数值；m 为 0～9 数码组成的不超过 3 位的整常数；表示指数的阶码，E 为**指数符号**，E 后面的正负号为**阶符**。指数符号也可以写成 e。

例如：
－12.3E2　　表示 -12.3×10^{2}
2E－3　　表示 2×10^{-3}
.23E6　　表示 0.23×10^{6}
16.E7　　表示 16×10^{7}
16.e7　　表示 16×10^{7}

指数形式的浮点数在书写时要注意符合规定形式。又如 e4，2e3.5，.e－5，.e4 都不是合法的浮点数。

1.4.2 实型数的分类

C 语言的实型变量分**单精度**(float 型)和**双精度**(double 型)两类，均采用 8087 浮点数表示格式，双精度变量所占的字节数是单精度变量的 2 倍。表 1.2 列出 Turbo C 系统的各类浮点数所占的字节数、取值范围以及数值有效精度。

表 1.2 Turbo C 编译系统的浮点数

关键字	字节数	取值范围	精度(位)
float	4	3.4e－38～3.4e＋38	7
double	8	1.7e－308～1.7e＋308	15
long double	8	1.7e－308～1.7e＋308	15

1.4.3 浮点型变量的定义及其初始化

浮点型变量的定义及其初始化与上节介绍的整型变量的定义及其初始化方法相同。例如，

```
float  a,b;              /* 指定变量 a,b 为 float 型变量 */
doule  x ;               /* 指定变量 x 为 doule 型变量 */
float a=16.235;          /* 指定 a 变量为 float 型变量，赋初值 16.235 */
double b=7.8413456e－7; /* 指定 b 变量为 double 型变量，赋初值 7.8413456e－7 */
```

在使用浮点数时，请注意以下几点：

(1) 实型常量不区分 float 型和 double 型。一个实型常量可以赋给一个 float 型或 double