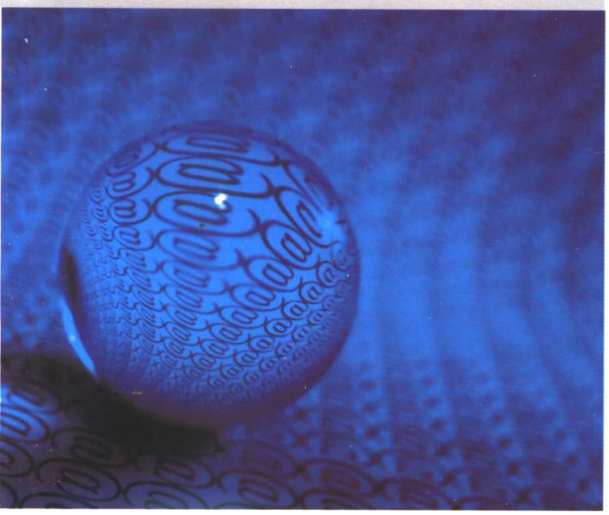
21世纪

高等院校计算机科学与技术规划教材



C/C++

程序设计教程

苏长龄　黄　岚　主　编
张　力　戴银飞　副主编

中国水利水电出版社
www.waterpub.com.cn

21世纪高等院校计算机科学与技术规划教材

C/C++程序设计教程

苏长龄　黄　岚　主编

张　力　戴银飞　副主编

中国水利水电出版社

内 容 提 要

本书是将C语言及C++语言二者合二为一的教材，突出特点是：概念准确、内容简洁、由浅入深、循序渐进、繁简适当。书中将C及C++的精华全部概括其中，书中程序都在计算机上调试通过。

全书共分11章，内容包括：C语言概述、数据类型、运算符和表达式、三种基本结构的程序设计、数组、函数、构造数据类型、指针、文件、类和对象、继承、多态性及两个附录。每章后都附有适量的习题，读者可通过习题巩固已学的知识。

本书可作为高等院校非计算机专业学生学习C语言程序设计的教材，也可作为其他人员学习C语言程序设计的参考书。

图书在版编目（CIP）数据

C/C++程序设计教程 / 苏长龄，黄岚主编. —北京：中国水利水电出版社，2004

（21世纪高等院校计算机科学与技术规划教材）

ISBN 7-5084-2133-7

Ⅰ. C… Ⅱ. ①苏…②黄… Ⅲ. C语言—程序设计—高等学校—教材 Ⅳ. TP312

中国版本图书馆CIP数据核字（2004）第076506号

书　　名	C/C++程序设计教程
作　　者	苏长龄　黄　岚　主编　张　力　戴银飞　副主编
出版　发行	中国水利水电出版社（北京市三里河路6号　100044）
	网址：www.waterpub.com.cn
	E-mail：mchannel@263.net（万水）
	sales@waterpub.com.cn
	电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787mm×1092mm　16开本　21.25印张　479千字
版　　次	2004年8月第1版　2004年8月第1次印刷
印　　数	0001—5000册
定　　价	30.00元

前　言

随着计算机的普及和社会信息化程度的提高，掌握一门计算机语言已经成为计算机用户必备的技能之一。目前，不论是从事计算机专业工作的人员，还是非计算机专业的人员，都将C语言作为学习程序设计语言的入门语言。因为C语言功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有高级语言的优点，又具有低级语言的许多特点。而C++是在C语言的基础上开发的一种高级程序设计语言，其包括C的全部功能，既支持面向过程的程序设计方法，又支持面向对象的程序设计方法。随着面向对象程序设计方法的不断普及与应用，学习和掌握C++语言已成为许多计算机专业工作者和广大计算机应用人员的迫切需要。

目前，单独讲述C语言和C++语言的教材很多，而将二者结合讲述的还很少，尤其是C++教材，很多内容与C语言的内容重复，真正面向对象程序设计的内容很少，而且从内容来看，偏重概念、理论，实例少而无系统性，导致学习者难以理解面向对象程序设计的思想和方法。

本教材包含C语言和C++语言两部分内容。在C++部分，以一个实例贯穿教材始终，使读者由浅入深、循序渐进地掌握面向对象程序设计的思想和方法。全书在编写过程中，力求做到概念准确、内容简洁、由浅入深、循序渐进、繁简适当。书中全部实例和习题都经过上机调试通过。

本书既可作为高等院校本专科学生的教材，也可作为其他计算机应用人员学习高级语言程序设计的参考书。本书由苏长龄、黄岚主编，张力、戴银飞任副主编，参加本书编写的还有宋雅娟、杜威、边晶等。

由于时间紧迫，加之作者水平有限，书中难免有不足之处，恳请读者提出宝贵意见和建议。

编　者

2004年5月

目　　录

第 1 章　概述

1.1　C 语言概述

1.1.1　C 语言及其特点

1. C 语言简介

随着第一台计算机 ENIAC 的诞生，计算机语言随即产生，人们利用计算机语言编制程序以便更好地使用计算机。计算机语言经历了机器语言、汇编语言、高级语言的不同发展阶段。机器语言（又称机器指令）是计算机能够识别和直接执行的二进制代码，用机器语言编写的程序难懂、易出错；为了克服机器语言的不足，人们又发明了汇编语言，汇编语言使用助记符来表示机器语言，但计算机不能直接执行，需通过汇编系统将汇编语言翻译成机器语言。机器语言与汇编语言都是面向机器的语言，一般称为低级语言。随着计算机的发展，人们发明了接近自然语言的计算机语言，这些语言能直接表达计算式和逻辑式，如 FORTRAN、Pascal、Basic、COBOL、C 以及 C++等，均称为高级语言。

C 语言是一种得到广泛重视并普遍应用的计算机程序设计语言，也是国际公认的最重要的几种通用程序设计语言之一，它既可用来编写系统软件也可用来编写应用软件。

1972 年，C 语言由贝尔实验室的 Dennis Ritchie 和 Brian Kernighan 根据 Thompson 的 B 语言创建，B 语言是由一种早期的编程语言 BCPL（Basic Combined Programming Language）发展演变而来的。BCPL 的根源可以追溯到 1960 年的 ALGOL 60（Algol Programming Language），ALGOL 60 是一种面向问题的高级语言，离硬件较远。1963 年，英国剑桥大学推出 CPL（Combined Programming Language）语言，CPL 修改了 ALGOL 60，使其能够直接作较低层次的操作。1967 年英国剑桥大学的 Martin Richards 对 CPL 做了改进，推出了 BCPL 语言。

最初的 C 语言是为描述和实现 UNIX 操作系统而提供的一种工具语言。但 C 语言并没有被束缚在任何特定的硬件或操作系统上，它具有良好的可移植性。1977 年出现了不依赖于具体机器的 C 语言编译文本——《可移植 C 语言编译程序》，用该程序编写的 UNIX 系统迅速在各种机器上运行，UNIX 系统支持的 C 语言也被移植到相应的计算机上。C 语言和 UNIX 系统在发展过程中相辅相成，得到了广泛应用，使它先后被移植到各种大、中、小、微型计算机上。

以 1978 年发表的第七版本 UNIX 系统中的 C 语言编译程序为基础，B.W.Kernighan 和 D.M.Ritchie 合著了 *The C Programming Language*。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，被称为标准 C 语言。1983 年，美国国家标准化协会（ANSI）根

据C语言问世以来的各种版本对C语言的发展和扩充制定了新的标准，称为ANSI C。1990年，C语言成为国际标准化组织（ISO）通过的标准语言。本书介绍的C语言部分以ANSI C为基础。书中的程序全部在Turbo C 2.0系统中调试通过。

2. C语言的特点

一种语言之所以能存在和发展，并具有生命力，总是有不同于其他语言的特点。C语言也是如此，它的特点是多方面的，人们从不同的角度可总结出众多特点，但从全面考虑可归纳为：

（1）C语言是比较低级的语言。有人把C语言称为高级语言中的低级语言，也有人称它是中级语言。它具有许多通常只有像汇编语言才具备的功能，如位操作、直接访问物理地址等等，这使C语言在进行系统程序设计时显得非常有效，而过去系统软件通常只能用汇编语言编写。事实上，C语言的许多应用场合是汇编语言的传统领地，现在用C语言代替汇编语言，使程序员得以减轻负担，提高效率，且写出的程序具有更好的可移植性。

C语言具有更多的接近硬件操作的功能，而不直接处理复合对象。如作为整体看待的字符串、数组等操作，这些较高级的功能必须通过调用函数来完成。这看起来是个缺陷，但这种语言规模小，更容易说明，学习起来也快。比如说，C语言只有32个关键字，而一些微机上的BASIC语言，关键字多达100个以上。

（2）C语言是结构化的程序设计语言。C语言的主要结构成分是函数，函数允许一个程序中的各任务分别定义和编码，使程序模块化。C语言还提供了多种结构化的控制语句，如用于循环的for、while、do-while语句，用于判定的if-else、switch语句等，十分便于采用自顶向下、逐步细化的结构化程序设计技术。因此，用C语言编制的程序容易理解、便于维护。

（3）C语言具有丰富的运算能力。在C语言中除了一般高级语言使用的算术运算及逻辑运算功能外，还具有独特的以二进制位（bit）为单位的位与、位或、位非以及移位操作等运算。并且C语言具有如a++、b--等单项运算和+=、-=等复合运算功能。

（4）C语言数据类型丰富，具有现代语言的各种数据类型。C语言的基本数据类型有整型（int）、浮点型（float）、字符型（char）。在此基础上按层次可产生各种构造类型，如数组、指针、结构体、共用体等。同时还提供了用户自定义数据类型。用这些数据类型可以实现复杂的数据结构，如栈、链表、树等。因此，C语言具有较强的数据处理能力。

（5）C语言具有预处理能力。在C语言中提供了#include和#define两个预处理命令实现对外部文件的包含以及对字符串的宏定义。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）C语言的可移植性好。目前，C语言在许多机器上实现，大部分是由C语言编译移植得到的。C编译程序的可移植性，也就使C语言程序便于移植。

C语言的优点很多，但也有一些不足。如语法限制不太严格、类型检验太弱、不同类型数据转换比较随便，这就要求程序员对程序设计的方法和技巧更熟练，以保证程序的正确性。

总之，C语言已成为国内外广泛使用的一种计算机语言，并且基于C语言新的语言版

本和配套工具软件在不断涌现，如后面将要介绍的 C++等。

1.1.2　C 语言程序的结构特点

C 语言是函数型语言，函数是构成 C 语言程序的基本单位。下面通过一个例子来分析 C 语言的程序结构。

例 1-1　C 语言的程序结构。

```
main()                      /*主函数*/
{
  int a,b,sum;              /*定义三个变量*/
  a=3; b=4;                 /*为 a，b 赋值*/
  sum=add(a,b);             /*调用函数 add，将得到的值赋给变量 sum*/
  printf("sum=a+b=%d\n",sum);
}
int add(int x,int y)        /*定义 add 函数和形式参数 x，y */
{
  int z;
  z=x+y;
  return(z);                /*返回 z 的值*/
}
```

运行结果：

```
sum=a+b=7
```

从程序可以看到，C 语言程序有如下特点：

1. *C 程序是由函数组成的*

C 语言源程序由若干个函数组成，组成程序的若干函数中必须有且只有一个名为 main 的函数。函数是 C 程序的基本单位。例 1-1 中包含两个函数：main()和 add()。因为在 main()函数中调用 add()函数，所以 main()为主函数，add()为被调用函数。被调用函数可以是系统提供的库函数（例 1-1 中的 printf()函数），也可以是用户根据需要自己定义的函数（例 1-1 中的 add()函数）。

2. *C 语言函数由两部分组成*

（1）函数的说明部分。这部分包括函数名、函数类型、参数名、参数类型。

如例 1-1 中 add()函数的说明部分：

```
int        add        ( int      x ,      int      y )
 ↓          ↓          ↓         ↓         ↓         ↓
函数类型   函数名      参数类型   参数名    参数类型   参数名
```

函数名后必须有一对圆括号()，这是函数的标志，参数可有可无，如 main()函数无参数。如果有参数，放在圆括号中，如 int add(int x,int y)。

参数类型的说明也可以放在圆括号外，这是传统的函数说明形式，如：

```
add(x,y)
 int x,y;
```

（2）函数体。函数说明以下的大括号内的部分。如果一个函数内有多对大括号，则最外层的一对为函数体的范围。

函数体一般包含变量定义和执行语句两部分。如例 1-1 中 main()函数中的 int a,b,sum;是变量定义部分，其余是语句执行部分。

3. main()函数

C 程序必须有 main()函数，习惯上称其为主函数。C 语言程序运行时，总是从 main()函数开始，并且在 main()函数中结束。main()函数可以放在整个程序的任意位置，通常我们总是把 main()函数放在程序中其他函数的前面。

4. C 程序书写格式自由

C 程序没有行号，书写格式自由，一行内可写多条语句，且语句中的空格和回车符均可忽略不计。一个语句也可以写在多行上，用“\”作续行符。

5. C 程序中的每个语句后必须有一个分号

分号“;”是 C 语句的一部分。例如：sum=a+b;分号必不可少，即便是程序的最后一个语句也应包含分号。

6. C 语言本身没有输入输出语句

输入和输出的操作是由库函数 scanf()和 printf()等函数来完成的。C 语言对输入输出实行“函数化”。

7. 可以在 C 程序的任何部分加注释，以提高程序的可读性

注释使程序变得清晰，能帮助我们阅读和理解程序。给程序加注释是一个良好的编程习惯。C 语言注释部分由“/*”开始，至“*/”结束。注释应括在/* …… */之间，/和*之间不允许留有空格。注释部分允许出现在程序中的任何位置上。注释可为若干行，但不允许嵌套。

下面我们介绍几个简单的 C 语言程序，以便对 C 程序结构有一个深入的了解。

例 1-2 最小的 C 程序例。

```
main()
{ }
```

这是一个最小的 C 程序，它什么也不做，但这是合法的 C 语言程序。

例 1-3 函数 C 程序例。

```
main()                  /*主函数*/
{
  printf("This is a C program.");
}
```

运行结果：

```
        This is a C program.
```

例 1-4 求两个数中的较小数。

```
main()                          /*主函数*/
{
  int a,b,c;                    /*变量说明*/
  scanf("%d,%d",&a,&b);         /*输入 a，b 值*/
  c=min(a,b);                   /*调用 min 函数，将结果赋给 c*/
  printf("min=%d",c);           /*输出 c 的值*/
 }
```

```
int min(int x,int y)          /*定义 min 函数，x,y 为形参*/
{
     int z;
     if(x<y) z=x;
     else   z=y;
     return(z);               /*将 z 的值返回，通过 min 带回调用处*/
}
```

键盘输入：

```
3,5
```

运行结果：

```
min=3
```

本程序包括主函数 main()和被调用的函数 min()。min()函数的作用是将 x 和 y 的较小值赋给变量 z。return 语句将 z 的值返回给主函数 main()。返回值是通过函数 min()带回到 main()函数的调用处。

1.2　C 语言的基本符号

1.2.1　基本符号集

C 语言的基本符号集是 ASCII 字符集，由以下几部分组成：

（1）阿拉伯数字 10 个：0、1、2、3、… 、9

（2）大小写英文字母各 26 个：　A、B、C、… 、Z、a、b、c、… 、z

（3）下划线：_

（4）特殊符号，主要是指运算符和操作符，它通常由 1～2 个特殊符号组成：

+	-	*	/	%	<	<=	>	>=	==	!=	&&	||	!
,	&	|	～	=	++	--	?:	<<	>>	()	[	]	.
->	+=	-=	*=	/=	%=	&=	^=	|=	^	#	sizeof		

1.2.2　标识符

标识符是一个字符序列。C 语言的标识符可分为用户标识符、保留字和预定义标识符三类，有些教材称保留字为“关键字”。标识符的作用是作为常量、变量、函数和类型的名称。

1. 用户标识符

用户可以根据需要对 C 程序中用到的变量、符号常量、用户函数或文件指针进行命名，形成用户标识符，这类标识符的构成规则如下：

（1）由英文字母、数字、下划线组成，且第一个字符不能是数字，必须是字母或下划线。

例如：a、_A、aBc、x1z、y_3 都是合法的标识符，而 123、3_ab、#abc、!45、 a*bc 都是非法标识符。

（2）大、小写英文字母的含义不同。比如 SUM、Sum 和 sum 代表 3 个不同的标识符，这一点一定要注意。

（3）C 语言本身并没有要求标识符的长度，不同的 C 编译系统允许包含的字符个数有所不同。通常可以识别前面 8 个字符，但在任何机器上，所能识别的标识符的长度总是有限的，有些系统可以识别长达 31 个字符的标识符（如 VAX-11 VMSC），而有些系统只能识别 8 个字符长度的标识符。这意味着即使第 9 个字符不同，只要前 8 个字符一样，系统也认为是同一个标识符，如： Category1 和 Category2 表示同一个标识符。因此，为了避免出错和增加可移植性，标识符最好前 8 个字符有所区别。

（4）用户定义标识符时，应当尽量遵循“简洁明了”和“见名知意”的原则。一个写得好的程序，标识符的选择应尽量反映出所代表对象的实际意思。如表示“年”可以用 year，表示“长度”可以用 length，表示加数的“和”可以用 sum 等，这样增加了标识符的可读性，使程序更加清晰。

2. 保留字

保留字是 C 语言编译系统固有的、用作语句名、类型名的标识符。C 语言共有 32 个保留字，每个保留字在 C 程序中都代表着某一固定含义，所有保留字都要用小写英文字母表示，且这些保留字都不允许作为用户标识符使用。C 语言的保留字如表 1-1 所示。

表 1-1 C 语言保留字

描述数据类型定义	描述存储类型	描述数据类型	描述语句
typedef	auto	char	break
void	extern	const	case
	register	double	continue
	static	float	default
	volatile	int	do
		long	else
		short	for
		signed	goto
		struct	if
		union	return
		unsigned	sizeof
		enum	switch
			while

注意：

（1）所有保留字都用小写字母表示。

（2）保留字不能用作常量名、变量名、函数名和类型名。

3. 预定义标识符

这些标识符在 C 语言中都具有特定含义，如 C 语言提供的编译处理预命令#define 和

#include。C 语言语法允许用户把这类标识符作其他用途，但这将使这些预定义标识符失去系统规定的原意。鉴于目前各种计算机系统的 C 语言已经把这类标识符作为统一的编译预处理中的专用命令名使用，因此为了避免误解，建议用户不要把这些预定义标识符另作它用或将它们重新定义。

1.3　程序设计方法简介

1.3.1　算法

1. 算法的概念

在日常生活中，我们每做一件事，都要遵循一定的方法。如你要到某一个地方，你要按照某一条路线去，乘什么车，在什么地方下车，这就是解决问题的方法，也就是算法。程序设计也是如此，编写一个程序，首先要设计算法，依据此算法进行编程。那么，什么是算法呢？著名计算机科学家沃思（N.Wirth）对程序有如下的描述：

程序=数据结构+算法

它说明一个程序由两部分组成：

（1）对数据的描述和组织形式，即数据结构（Data Structure）。

（2）对操作或行为的描述，即操作步骤，也就是算法（Algorithm）。

数据是操作的对象，操作的目的是对数据进行加工处理。编写一个程序的关键就是合理地组织数据和设计算法。所谓算法，就是一个有穷规则的集合，其中规则确定了一个解决某一特定类型问题的运算序列。简单地说，算法就是为解决一个具体问题而采取的确定的有限操作步骤。这里的算法指的是计算机算法。

2. 算法的特性

算法须具备如下五个特性：

（1）有限性：一个算法必须总是在执行有限步之后结束。

（2）确定性：算法中每条指令的含义必须明确，不允许有二义性。

（3）可行性：算法中的操作都是可以通过已经实现的基本运算执行有限次来完成。

（4）输入：一个算法有零个或多个输入，即执行算法时需要从外界取得要处理的信息。

（5）输出：一个算法有一个或多个输出。算法的目的就是为了求“解”，输出就是将所求得的“解”，输出是与输入存在某些特定关系的量。

3. 算法的组成要素

算法含有如下两大要素：

一是操作。每个操作的确定不仅取决于问题的需求，还取决于它们取自哪个操作集，它与使用的工具系统有关。如算盘上的操作集由进、退、上、下、去等组成；做菜的操作集包括煎、炒、炸、煮等。计算机算法要由计算机实现，组成它的操作集是计算机所能进行的操作。在高级语言中所描述的操作主要包括各种运算，如：算术运算、关系运算、逻辑运算、函数运算、位运算、I/O 操作等。计算机算法是由这些操作组成的。