普通高等教育“十一五”规划教材
21世纪大学计算机基础分级教学丛书

C/C++
语言程序设计

张颖江 主审
李振立 程 玉 主编

科学出版社
www.sciencep.com

高等教育“十一五”规划教材
21世纪大学计算机基础分级教学丛书

C/C++语言程序设计

张颖江 主审
李振立 程 玉 主编

科学出版社
北京

内 容 简 介

本书为C语言程序设计或C++程序设计课程教材，在充分考虑教学层次和类型、学生层次及其复杂性等重要因素的基础上，为积极处理好学习掌握知识体系与“面向应用”的关系编写而成。与传统的C语言及C++教材相比，本书更符合计算机程序设计课程的教学需要，以及计算机科学和技术的发展趋势，在体系结构、重点、难点、详细安排等方面更加合理，内容更加新颖适用。

本书为高等学校非计算机专业C/C++语言程序设计课程教材，也可作为各类成教学院、网络学院和计算机培训班的教材，或相关教师的教学参考使用。

图书在版编目(CIP)数据

C/C++语言程序设计/李振立，程玉主编. —北京：科学出版社，2009
普通高等教育“十一五”规划教材　21世纪大学计算机基础分级教学丛书
ISBN 978-7-03-026353-7

Ⅰ.C…　Ⅱ.①李…②程…　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(2010)第003471号

责任编辑：梅　莹/责任校对：闫　陶
责任印制：彭　超/封面设计：苏　波

科学出版社　出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com

武汉市新华印刷有限责任公司印刷
科学出版社发行　各地新华书店经销
*
2009年12月第　一　版　开本：787×1092　1/16
2009年12月第一次印刷　印张：16 3/4
印数：1—4 000　　字数：412 000

定价：28.00元

（如有印装质量问题，我社负责调换）

前　言

根据理工类计算机基础课程教学指导分委员会《计算机基础课程教学基本要求》，结合多年的教学实践，我们编写了这本《C/C++语言程序设计》。

本书将两种同源的程序设计语言有机地结合在一起，为学生提供了较为全面的程序设计思想。C语言具有高级语言的诸多特点，又具有汇编语言的特点，既适合用于开发操作系统和系统软件，又适用于编写嵌入式系统等硬件系统的开发程序。C语言应用范围广，数据处理能力强、操作简单、易读性好，是最实用的入门级计算机高级语言。C++包括面向过程和面向对象两种编程方式，面向对象程序设计把类与对象相关的数据和代码结合成一个有机的整体，形成数据成员和行为操作的封装体，实现对数据的访问控制和信息隐蔽。类与对象具有抽象性、封装性、传递性、可见性、安全性、继承性、派生性、多态性等诸多特性。

按照教学基本要求，本书从“算法基础与程序设计”领域中选择如下的知识单元，包括程序与程序设计语言、数据类型基础、基本控制结构、基本算法概念、程序设计过程、过程与函数、构造类型与指针、文件、面向对象编程等知识单元组织教学内容。全书分为10章：第1章C与C++概述，第2章C/C++程序设计语言，第3章程序的基本结构，第4章数组，第5章函数与函数的重载，第6章指针，第7章结构体与共用体，第8章类与对象，第9章C++程序的面向对象特性，第10章文件与输入输出流。本书涉及C语言的全部教学内容和C++程序设计语言的主要教学内容。

本书由长期工作在教学一线的教师编写，各知识单元编排顺序得当，结构合理严谨，内容丰富，由浅入深、循序渐进，详略度把握得体，书中配置了大量运行在VC环境下的例题，是一套理想的C/C++程序设计教材。本书既可作为各类高等学校本、专科非计算机专业的C语言程序设计或C++程序设计课程的教材，也可以作为独立学院、高职高专、网络学院的计算机语言课程教材。对于计算机技术爱好者，尤其是具有一定计算机基础而又欲获得提高的广大爱好者，本书无疑是一本极好的自学读物。

本书由李振立、程玉主编，张颖江主审，其他编委还有钮焱、李军、沈海波、张群、熊英、杜江毅、明喆、吕瑾文、陈荆亮、李珺。在全书的策划、编写、出版过程中，王春枝、湛俊三、马丹、冯昭昭等同志给予了大力支持，在此深表谢意。

编　者

2009年11月

目　录

第1章 C与C++概述

1.1 C及C++的发展史

1.1.1 C语言的起源

C语言是国际上广泛使用的计算机程序设计语言。C语言具有一般高级语言的特性，程序不依赖计算机硬件，可读性和可移植性好，接近于自然语言或数学语言；又具有低级语言的特性，可以内嵌汇编指令，将汇编指令作为C语言的指令，可以直接对计算机硬件进行操作，如对内存地址的操作、位操作、I/O操作等。C语言集高级语言和低级语言的优点于一身，适用于作为系统描述语言，用于编写大型的操作系统、编译系统和应用软件，也可以作为单片机、DSP、EDA、ARM等嵌入式系统的开发语言。

C语言属于面向过程的程序设计语言。面向过程是一种以事件为中心的编程思想，将事件的产生、发展、变化和结果等事件运作过程作为研究的重点，采用模块化的方法设计源程序，由主控模块分级调用各子模块，各个模块依照事件运作的逻辑次序组织程序流程，用框图或程序流程图描述程序的算法。C语言将要处理的信息数字化，表示成各种类型的数据。数据的类型、数据的组织和数据的传递合称为程序的数据结构，数据结构也是程序设计的重要内容。因此，可以说，面向过程的程序是由算法和数据结构共同组成的，用公式表示为

面向过程的程序＝算法＋数据结构

C语言是在B语言的基础上发展起来的。1970年，美国贝尔实验室的学者肯·汤普森(K. Thompson)以BCPL(Basic Combined Programming Language)语言为基础，经过简化，设计出简单而且接近于硬件的B语言，并用B语言写了第一个UNIX操作系统，在PDP-7上实现。1972年，贝尔实验室的学者丹尼斯·利奇(D. Ritchie)在B语言的基础上设计了C语言。1973年，肯·汤普森和丹尼斯·利奇两人合作改写UNIX(第5版)，其中90%是用C语言改写的。1977年，出现不依赖于具体计算机的C语言编译文本《可移植C语言编译程序》，使C语言迅速移植到各类计算机上，从而推动了UNIX操作系统在各类计算机上的开发。UNIX的日益推广，推动了C语言的广泛应用。1978年，B. Kernighan和D. Ritchie两人以UNIX第7版使用的C语言为基础，合著发表了*The C Programming Language*这部名著，这本书介绍的C语言称为标准C。1983年，美国国家标准协会(ANSI)对C语言进行扩充和规范制定了新的标准，称为ANSI C。1987年公布了C语言的美国国家标准，1989年ISO/IEC提出了国际标准草案，1990年公布了C语言的正式标准。

1.1.2 C语言的特点

C语言的主要特点有以下几个。

(1) 语言简洁、紧凑，使用灵活、方便。语法限制不太严格，语言描述简单、明晰，没有复杂的控制结构，程序设计自由度大。

(2) 运算符丰富，表达能力强。C 语言将括号、赋值、强制类型转换等都作为运行符处理，使运算符的范围更广泛，表达式的类型多样化，运算更灵活方便，增强了语言的表达能力。

(3) 数据结构丰富，结构化程度高。C 语言具有结构化程序的典型特征，程序代码与数据分离，具有计算机语言的各种数据结构。其数据类型有整型、实型、字符型、数组类型、指针类型、结构类型、共用体类型等。它能实现链表、树、栈等复杂数据结构，具有结构化的流程控制语句，编制的程序具有模块化和结构化的特点。

(4) 兼有高级语言和低级语言的特点，允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能。

(5) 生成目标代码质量高，程序执行效率高，一般高级语言程序执行效率低，C 语言的代码效率仅次于汇编语言，代码效率达到汇编语言的 80%～90%。

(6) 比汇编程序的可移植性好。程序基本上不作修改就能用于各种型号的计算机或各种操作系统。

1.1.3　从 C 到 C++

C++是在不断增强 C 语言功能的过程中诞生的一种计算机编程语言。受面向对象程序设计语言的影响，AT&T 贝尔实验室的布贾尼·斯特劳斯特卢普(B. Stroustrup)发明了“带类的 C”(C with Classes)，将面向对象程序设计的基本概念，包括类与对象、派生、多重继承、多态性、虚函数、运算符重载、模板、异常、名字空间、访问控制等概念逐渐引入 C 语言，形成了 C++的雏形。1998 年，国际标准化组织(ISO)颁布了 C++程序设计语言的国际标准 ISO/IEC 14882-1998。

C++是 C 语言的超集，很多用 C 语言编写的应用程序和库函数可以用于 C++之中。因此，C++既支持面向过程程序设计，又支持面向对象程序设计。C++既包含了 C 的所有优点，又对数据类型进行了扩充和加强，通过类和对象的概念把数据和对数据的操作封装在一起，通过派生、继承、重载、虚函数和多态性等特征实现了软件重用和程序自动生成，使得大型复杂软件的构造更加严谨，软件的维护变得更加有效和容易。

1.1.4　C 与 C++的集成开发环境

C 与 C++都是 AT&T 贝尔实验室发明的，贝尔实验室发布的集成开发环境 Dev-C++是一款自由软件，遵守 GPL 协议，用户在遵守 GNU 协议的前提下可以从 devpak.org 上取得最新版本的各种工具包，获取源代码，并拥有对这一切工具自由使用的权利。Dev-C++集合了 GCC，MinGW32 等众多自由软件，是一款非常实用的 C/C++开发工具，很多高水平的软件开发人员用它编写出许多著名软件。在全世界自由软件爱好者的研究和开发下，Dev-C++仍在不断地发展和进步。

常用的 C 与 C++集成开发环境有 Borland 公司的 Turbo C/C++，Borland C++，C++Builder；微软的 Microsoft C，Visual C++，Visual studio.NET；多平台的 C++集成开发环境有 gcc，QT，eclipse+CDT 等软件。在 2008 年以前，全国计算机等级考试二级 C 语言使用的集成开发环境是 Turbo C(简称 TC)，初学者入门时使用简单的 TC2.0 集成开发环境，有助于对 C 语言的操作和理解。2008 年以后，全国计算机等级考试使用的集成开发环境是 Visual C++(简称 VC)。在等级考试的带动下，Visual C++成为最常用的 C 语言集成开发

环境。掌握了Visual C++集成开发环境后可进一步朝纵深发展，再学习Visual studio. NET软件和QT等软件。

1.2 C语言的程序架构

1.2.1 C语言程序的基本架构

C语言是一种函数型语言，其源程序是由一系列函数构成的。下面通过几个简单的例子讨论C语言程序的架构。

例1.1 建立一个文件名为ex1_1的C源程序，在屏幕上输出“How do you do?”后回车换行。(文件名ex1_1.c)

```
#include<stdio.h>                    /*包含标准输入/输出 stdio.h头文件*/
main( )                              /*主函数由主函数名和参数组成*/
{                                    /*函数体开始*/
printf("How do you do?\n");          /*用格式输出函数 printf输出字符串*/
}                                    /*函数体结束*/
```

在Turbo C集成环境中编辑该程序，按快捷键**Ctrl+F9**，集成环境编译、连接、运行该程序，在输出屏幕上显示字符串“How do you do?”。

例1.2 输入两个整数，并输出较大的一个数。(文件名ex1_2.c)

```
/*注释:从两个数中找出较大的数*/
#include<stdio.h>                    /*包含标准输入/输出头文件*/
main(void)                           /*主函数参数无返回值*/
{                                    /*函数体开始*/
int x,y,z;                           /*定义函数中使用的变量 x,y,z*/
scanf("%d %d",&x,&y);                /*用格式输入函数输入两个整数*/
if(x>y) z=x;                         /*如果 x>y那么把 x的值赋给 z*/
else z=y;                            /*否则把 y的值赋给 z*/
printf("max=%d\n",z);                /*用格式输出函数 printf输出最大值*/
}                                    /*函数体结束*/
```

在Turbo C集成环境中编辑该程序，按快捷键**Ctrl+F9**，集成环境编译、连接、运行该程序，执行到“scanf("%d%d",&x,&y);”语句，计算机等用户输入数据，用户输入“8 12”两个整数后回车，计算机继续执行该程序，在输出屏幕上输出较大的数12。

例1.3 先定义一个求最大值子函数，然后由主函数调用该子函数，试编写源程序，当输入两个整数时，输出较大的一个数。(文件名ex1_3.c)

```
#include<stdio.h>                    /*包含标准输入/输出头文件*/
int max(int x,int y)                 /*用户定义函数 max*/
{                                    /*函数体开始*/
int z;                               /*定义函数中使用的变量 x,y,z*/
if(x>y)z=x;                          /*如果 x>y那么把 x的值赋给 z*/
else z=y;                            /*否则把 y的值赋给 z*/
return(z);                           /*返回计算结果*/
}                                    /*函数体结束*/
```

```
void main(void)                    /*主函数函数无返回值,参数无返回值*/
{                                  /*函数体开始*/
int a,b,c;                         /*定义函数中使用的变量 x,y,z*/
scanf("%d%d",&a,&b);               /*用格式输入函数输入两个整数*/
c=max(a,b);                        /*调用函数 max*/
printf("max=%d\n",c);              /*用格式输出函数 printf 输出最大值*/
}                                  /*函数体结束*/
```

该程序的功能与例 1.2 中程序相同,只是采用了函数调用来计算两个数中的较大值。

对照以上三个简单的 C 程序,可以看出 C 程序的基本架构由编译预处理命令、子函数、主函数、函数及参数说明语句部分等组成。

1. 编译预处理

编译预处理命令有文件包含(.h 头文件)、宏定义和条件编译三种。

在 C 语言中经常使用的包含文件有标准输入/输出头文件“stdio.h”、字符串头文件“string.h”、控制台输入/输出头文件“conio.h”、数学函数库头文件“math.h”等。其格式为

```
#include<filename.h>
```

例如,

```
#include<stdio.h>
```

或

```
#include "filename.h"
```

例如,

```
#include "stdio.h"
```

用 #include <filename.h>格式来引用标准库的头文件,编译器从标准库目录开始搜索。用 #include "filename.h"格式来引用非标准库的头文件,编译器从用户工作目录开始搜索。

宏定义的作用是定义符号常量,用一个短的名字代表一个长的字符串。定义格式为

```
#define 标识符 字符串
```

例如,

```
#define PI 3.1415926
```

2. C 语言中的函数

C 语言是函数型语言,程序模块都是函数型模块,一个 C 程序可以由一个或多个具有相对独立功能的函数构成,其中有且仅有一个以 main 命名的主函数。程序从 main 函数开始执行,直到执行完函数体程序才运行结束。其他子函数由函数调用语句引用,运行完子函数返回到调用处继续执行后续语句。函数与函数之间用参数传递信息。子函数只能被主函数或其他子函数调用,不能单独运行。

3. 函数的组成

一个函数由函数头部和函数体两部分组成。函数的头部包括函数的类型、函数名和形式参数表。形式参数表可以无类型,参数表为无类型时,用类型名 void 表示。函数体用一对花括号“{ }”括起来,包括局部说明部分和语句部分。一般形式为

函数类型 函数名(参数表列)	int max(int x,int y)
{	{
局部说明部分;	int z;

语句部分;	if(x>y)z=x;
	else z=y;
	return(z);

}	}

4. 函数的调用

C函数分为标准函数和用户定义函数两类。标准函数是由编译程序提供的,其定义是以编译后的目标代码形式存放在系统的函数库中,称为C函数库。用户编程时直接调用标准函数,例1.2中的格式输入函数scanf和格式输出函数printf都是标准函数。

格式输入函数scanf的格式为

```
scanf("输入格式",输入项地址表列);
```

其中,"输入格式"是用双引号括起来的字符串,包括格式说明和普通字符。格式说明是由%字符开头后接格式字符,如%d表示十进制整型数,%f表示浮点型数,%c表示单个字符等;普通字符是照原样输出的字符,包括可打印的字符和转义字符两种。"输入项地址表列"是由若干个地址所组成的表列,可以是变量的地址,如&a,&b,或字符串的首地址str1。

格式输出函数printf的格式为

```
printf("输出格式",输出表列);
```

其中,"输出格式"也是用双引号括起来的字符串,同样包括格式说明和普通字符。"输出表列"是输出的常量、变量或表达式,用逗号分隔。

用户定义函数必须由用户在源程序中编写函数定义,根据模块功能,设计和编写用户定义的函数语句,供其他函数调用。用户定义函数与调用函数的函数名必须一致,两者参数的个数与参数的类型必须按位置相同放置,函数定义中的参数为虚参,函数调用中的参数为实参,参数的传递按位置虚实对应。在例1.3中调用函数max的语句是赋值语句

```
c=max(a,b);        /*调用max函数,a,b为实参已有确定的数值*/
```

执行该语句将函数的值赋给变量c,保存在变量c指向的内存单元。式中,"="为赋值号。

5. 书写格式

C语言的书写格式自由。C语句用分号";"作结束符,一条语句可以写成多行,多条语句也可以写成一行。C语言的书写支持缩进格式,将同一层次的语句,语句的开头对齐在同一列,每下一个层次,语句开头通过加空格缩进若干列,从而形成层层缩进对齐的书写格式。

6. 注释

程序中可加必要的注释,使用一对符号"/* */"作程序中注释的定界符,表示"/*"和"*/"之间的内容是注释。注释不影响程序的编译和执行,能增加程序的可读性,是一种良好的程序设计风格。

1.2.2 C语言程序逻辑顺序

C语言程序模块中,语句块的逻辑顺序是按局部说明、数据输入、数据处理和数据输出的

次序排列的。

例 1.4　已知圆的半径为 r,编写求圆的面积与周长的程序。(文件名 ex1_4.c)

```
#include<stdio.h>                    /*包含标准输入/输出头文件*/
#define PI 3.1415926                 /*宏定义*/
main( )                              /*主函数*/
{                                    /*函数体开始*/
float r;                             /*定义函数中使用的局部变量 r*/
double s,w;                          /*双精度变量 s,w*/
scanf("%f",&r);                      /*用格式输入函数 scanf 输入半径 r(数据输入)*/
s=PI *r *r;                          /*求圆的面积(数据处理)*/
w=2 *PI *r;                          /*求圆的周长(数据处理)*/
printf("s=%f,w=%f\n",s,w);           /*用格式输出函数 printf 输出面积和周长*/
}                                    /*函数体结束*/
```

说明:

(1) 程序开头的编译预处理命令由文件包含“#include <stdio.h>”和宏定义“#define PI 3.1415926”两部分组成。文件包含和宏定义是命令,不是语句,结束没有分号。宏定义不是给 PI 赋值,而是定义符号常量,C 程序在编译时会将非双引号内的所有 PI 都置换为 3.1415926。

(2) 函数体内的局部说明“float r,s,w;”说明变量 r,s,w 是浮点型变量,在内存中为每个变量分配 4 个字节的存储单元。

(3) 在 C 语言中没有专门的输入语句,用格式输入函数“scanf("%f",&r);”输入数据。

(4) 求圆的面积“s=PI*r*r;”与求圆的周长“w=2*PI*r;”是处理数据的语句块。

(5) 在 C 语言中没有专门的输出语句,用标准函数“printf("s=%f,w=%f\n",s,w);”输出数据。

1.2.3　C 语言的风格

C 语言的风格指 C 程序的风范格局,是程序内容与书写形式相互统一的表现形式,是在程序员的个性特征与其所受的教育、使用教材、编程经历、团队的规定等诸多条件影响下形成的编程习惯。编程思想的开放性、程序开发的统一性、程序结构的层次性和程序内容的易读性,以及程序员对程序书写形式的一贯性都是 C 语言风格的表现形式。每一个软件开发团队都可以自行规定表征 C 语言风格的书写形式,内容主要包括大括号的放置位置、缩进格式、空格的使用、注释方法、命名系统或函数等。

1. 大括号的放置位置

编写 C 语言源程序,一般建议主函数中的大括号“{”表示函数开始,应该放在 main( )的下一行的第 1 列,其他语句可以另起一行;反大括号“}”单独放在一行。例如,

```
void main( )                         /*主函数*/
{                                    /*函数体开始,标准 C 提倡单独占一行*/
printf("Hello!");                    /*用标准输出函数 printf 输出字符串*/
}                                    /*函数体结束,单独放在一行*/
```

子函数中的大括号也应该这样处理。例如,

```
int fun1(int x,int y)                /*子函数*/
{                                    /*函数体开始,单独占一行*/
```

```
x=x+y;y=x-y;x=x-y;          /*函数处理多条语句可以写在一行*/
printf("x=%d,y=%d",x,y);    /*用标准输出函数 printf 输出整数 x,y*/
}                           /*函数体结束,单独放在一行*/
```

习惯上,很多程序员将函数体开始与变量类型说明写在一行。例如,

```
int main( )
{ int a;
```

2. 缩进格式

在C/C++的集成开发环境中,按一次**Tab**键默认跳过8个字符,即C语言默认每层控制结构缩进深度为8个字符。虽然有的用户设置的缩进深度为4个字符或2个字符,但这不是一个好的风格。因为控制结构的每一层构成一个语句块,块的开始和结束是缩进对齐的,缩进量设置为8个更加醒目。当用户连续编程十几个小时后,缩进深度的大小就非常重要了,能帮助用户更清楚地划分结构层次、理解程序内容。同时,当用户程序中的控制结构层次太多,程序代码缩进到右边,看起来不舒服时,也能起到提醒程序员修改程序的作用,因为嵌套这样多层的控制结构理解起来已经很困难了。

3. 空格

在C语言程序中使用空格能形成良好的程序风格,有些空格是作为语句的分隔符在语句中必须存在的,例如,在const,virtual,inline,case等关键字之后至少要留一个或一个以上的空格,否则无法辨析关键字。另一类关键字如if,for,while之后应留一个空格再跟左括号"(",以突出关键字,这类关键字添加的空格不是必须的。有些语言的集成开发环境会自动地给程序中的单词加空格,形成良好的程序风格。良好的程序风格使用的空格有如下规律:

(1) 函数名之后不要留空格,紧跟左括号"(",关键字与"("之间加空格,以体现二者的区别;在"("之后紧跟数据。

(2) 符号")",",","; "向前紧跟数据,紧跟处不留空格,","之后要留空格;如果";"不是一行的结束符号,其后要留空格。

(3) 单目运算符(一元操作符)如"!","~","++","--","&"(取地址运算符)等前后不加空格。

(4) 双目运算符包括赋值与复合赋值运算符如"=","+=","-=","*=","/=";比较操作符如">","<",">=","<=","==","!=";算术操作符如"+","-","*","/","%";逻辑操作符如"&&","||";位域操作符如"&","|"等,其前后应当加空格。

(5) 数组运算符"[]",域运算符".",指向运算符"->"的前后不加空格。

4. 注释

注释通常用于版本或版权声明、函数功能说明、函数的接口说明、结构提示以及重要代码行的说明,有助于理解代码,但过多地使用注释会使程序繁复,影响阅读程序代码。另外,注释不能够嵌套。C语言使用的注释符为"/*…*/"。C++语言中,程序块的注释常采用"/*…*/",行注释一般采用"//…"。保持良好程序风格时使用的注释要注意以下几点:

(1) 程序中的注释起提示功能,不能太多,不可喧宾夺主;注释的花样要少。

(2) 可加也可不加的注释一律不加;注释应当准确、易懂,防止有二义性的注释。

(3) 注释与代码之间保持一致性,修改代码的同时要修改相应的注释;删除不用的注释。

(4) 注释应简短,但要尽量避免在注释中使用缩写。

(5) 注释的位置应与被描述的代码相邻，可以放在代码的上方或右方，不可放在下方。

(6) 当代码比较长或有多重嵌套时，应当在一些段落的结束处加注释，以便于阅读。

5. 命名系统

C是一种简洁而强大的语言，用户对标识符的命名应该简洁、明晰，望文知意，命名要基于容易记忆、容易理解的原则。Microsoft公司使用的命名规则为匈牙利命名法，该命名规则是在变量和函数名中加入特定的前缀，以帮助用户理解变量的属性和类型，增进对程序的理解。匈牙利命名法的基本原则是

变量名＝属性＋类型＋对象描述＋对象名

用属性、类型和对象描述作为前缀，对象名要求有明确含义，可以取对象名字全称或名字的一部分。可以根据需要选择常用的前缀组织成用户特有的匈牙利命名法，常用的前缀见表1.1。

表1.1　常用匈牙利命名法前缀

前缀	类型	说明	举例
ch	char	字符型	chA1,chB2
N	short	短整型	nNu1,nC2
i	int	整型	iK,iJ,iN,iM
l	long	长整型	lAB1
u	unsigned int	无符号整型	uLong1
f	float	浮点型	fFirst,fSe
d	double	双精度型	dDou1,dE2
ld	long double	长双精度型	ldA1,ldB2

用匈牙利命名法时便于识记变量的属性和类型，使变量名清晰易懂，增强了代码的可读性，方便不同的程序员之间相互交流代码。但对于简单的一个字母的变量，加上前缀后，变量字母增多，书写程序加大语句长度。例如，用匈牙利命名法为变量I,J,K命名，则为

```
int iI,iJ,iK;          //前缀 i 表示 int 类型
double dX,dY;          //前缀 d 表示 double 类型
```

有些语言如Fortran语言，使用了变量的隐含说明，凡是以I,J,K,L,M,N这些字母开头的变量名隐含规定为整型，以其他字母开头的隐含规定为实型。根据这种思路，扩展匈牙利命名法，可以制定一个方案，将26个字母划分一下，指定几个字母开头的名称表示一种数据类型。例如，用c,s开头表示字符型；用i,j,k开头表示整型；用n开头表示短整型；用l,m开头表示长整型；用u,v开头表示无符号整形；用e,f开头表示浮点型；用d,x,y开头表示双精确度型；用z开头表示长双精确度型。用这种方案为C源程序命名，可以简洁地表示为

```
int i,j,k;
double d,x,y;
```

在命名系统中存在着共性规则，共性规则被大多数程序员采纳，用户应当在遵循以下这些共性规则的前提下为标识符命名：

(1) 标识符应当直观易懂，容易拼读，可望文知意。标识符最好采用英文单词命名，尽量不用汉语拼音命名。

(2) 标识符的长度应当符合小长度并且大信息的原则。

(3) 命名规则尽量与所采用的操作系统或开发工具的风格保持一致。如 Windows 应用程序的标识符通常采用“大小写混排”的方式，如 AddChild；而 Unix 或 Linux 的应用程序的标识符通常采用“小写加下划线”的方式，如 add_child。

(4) 程序中避免相似的标识符，不要出现仅靠大小写区分的标识符，如“int x,X;”。

(5) 程序中避免出现标识符完全相同的局部变量和全局变量，容易造成误解。

(6) 变量的名字应当使用“名词”或“形容词＋名词”；全局函数的名字应当使用“动词”或者“动词＋名词”。

6. 函数

函数是 C 语言的基本功能单元，标准函数的调用要注意格式符与附加格式符的功能、变量与变量的地址的区别，避免产生错误。自定义函数应该短小精干、功能单一，不要设计多用途的函数。函数体内的语句规模要小，应尽量控制在 50 行代码之内。函数名称应该表述函数的功能，帮助用户理解函数的语句功能和描述的整体功能。

函数设计要确保函数的功能正确，同时要注意函数的接口。函数接口的两个要素是参数和返回值。C 语言中，函数的参数和返回值的传递方式分为值传递和地址传递(或称指针传递)两种。应使用现代的函数定义格式，不要使用早期的函数定义格式。早期的函数定义将参数的类型说明放在函数名的下一行。例如，

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

使用函数时应注意参数规则：

(1) 参数要书写完整，不能只写参数的类型不写参数名字。当函数没有参数时，应该用 void 填充。例如，

```
int GetValue(void);
```

(2) 参数命名要望文知意，参数放置的位置顺序按先目标，后源的习惯。例如，编写字符串拷贝函数 StrCopy 的函数名与参数为

```
void StrCopy(char *strDes,char *strSour);
```

(3) 函数和参数个数应控制在 5 个以内，避免使用太多参数。

(4) 不使用类型和数目不确定的参数。

函数返回值的规则如下：

(1) 不要省略返回值的类型。如果函数没有返回值，那么应声明为 void 类型。

(2) 函数名字与返回值类型不发生冲突。

(3) 将正常值和错误标志区别开，不要混在一起返回。正常值用输出参数返回，错误标志用 return 语句返回。

养成良好的习惯，编写风格良好的程序是程序员的基本功，也是一个团队共同工作的基础。以上主要介绍 C 语言程序风格的一些要素，在后面的章节中还会涉及更多的内容。

1.3 C/C++语言的单词

C/C++语言用一整套带有严格规定的符号体系来描述 C/C++语言的词法、语法、语义和语用。符号体系的基础是基本字符集，基本字符集中的若干字符组合成一个具有独立意义