HZ BOOKS

高等院校计算机教材系列

# C++语言程序设计

管建和 编著

为教师配有教学课件

机械工业出版社
China Machine Press

高等院校计算机教材系列

# C++语言程序设计

管建和 编著

机械工业出版社
China Machine Press

C++语言是国内外广泛流行的程序设计语言，由于C++具有面向对象程序设计语言的特征，因此对该语言的掌握程度已成为衡量软件开发人员技术水平的重要指标。本书假定读者没有编程基础，完整讲解C++语言特性及编程方法，有C语言基础的读者也可以通过本书进一步地学习，从而掌握C++语言的编程技术。

该书适合用作计算机专业或非计算机专业的程序设计基础教材，也可以供初学计算机编程的人员自学使用。



**图书在版编目(CIP)数据**

C++语言程序设计/管建和编著. —北京：机械工业出版社，2007. 5
(高等院校计算机教材系列)

ISBN 978-7-111-21211-9

Ⅰ. C…　Ⅱ. 管…　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(2007)第041318号

机械工业出版社(北京市西城区百万庄大街22号　邮政编码　100037)
策划编辑：朱　劼
责任编辑：杨庆燕
北京慧美印刷有限公司印刷·新华书店北京发行所发行
2007年5月第1版第1次印刷
184mm×260mm · 18.25印张
定价：29.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：(010)68326294

# 前　言

C++语言是在C语言基础上发展起来的编程语言，它是带类的C语言。C语言是自20世纪80年代以来迅速推广和广泛使用的一种程序设计语言。它既具有高级语言表达能力丰富、可移植性好等特点，又具有低级语言能够方便地实现汇编语言级的操作、目标程序效率高的特点。发明C++程序设计语言的主要目标是在保留C语言的原有精华的基础上，提供全面的面向对象的编程支持，使编写出的程序结构更加清晰、更容易维护和扩充，同时又不丧失其高效性。C++对C作了大量的扩充，如增加了常值(const)数据、显式类型转换、语句中的变量声明、内联函数、引用(reference)、函数参数缺省、函数及操作符重载等新的表现手段以及更严格的类型机制，同时还完全支持面向对象的概念，如对象、类、属性、方法、派生类与继承、多态等。C++和C一样，现在已被广泛地应用。

本书全面系统地讲述了C++语言的基本概念、数据类型、指针、语句和面向对象的程序设计方法，并对C++面向对象语言的抽象性、封装性、继承性与多态性进行了全面介绍，内容包括C++语言的数据与运算、基本语句、指针、函数、自定义数据类型、类与对象、继承与派生、多态性、模板、异常处理机制以及I/O流库等知识。

本书可以作为计算机及相关专业第一门语言课的教材，不要求读者具有其他语言的基础。本书共有13章，下面简单介绍一下各章的内容。

第1章主要介绍C++语言的发展简况及其特点，C++语言的程序组成及其结构，C++语言程序的格式特点，关键字和标识符，三个特殊的C++语句(空语句、复合语句和注释语句)，简单输入/输出和数制与编码方面的知识。

第2章主要介绍C++语言的基本数据类型，整型常量、实型常量、字符型常量、字符串型常量和换码序列常量，变量的声明和使用，数组的数据结构，各种运算符及其表达式，运算符的优先级等。

第3章主要介绍C++语言的基本程序结构，各种分支语句，for语句、while语句和do-while语句的循环控制过程，以及break语句、continue语句、return语句和goto语句辅助控制编程等内容。

第4章主要介绍C++语言的指针概念，指针的声明与初始化，指针的运算，指针与数组的关系，字符指针与字符串的应用，指针数组与多级指针，以及new与delete运算符，引用，const型指针和void型指针等内容。

第5章主要介绍C++语言中函数的概念，函数的定义及其声明，函数之间的参数传递，数组在函数间的传递，指针函数和函数指针，函数指针数组，函数重载，内联函数，递归函数，标准函数，宏和其他预编译语句等内容。

第6章主要介绍结构体的定义及其声明，结构体数组与结构体指针，结构体与函数，结构体嵌套，位字段结构体简介，联合体，枚举类型以及typedef语句等内容。

第7章主要介绍面向对象技术与C++类类型，类的定义，已有类的对象的声明和使用，对象数组和对象指针，构造函数和析构函数，对象在函数之间的传递，this指针，以及常对象与常成员函数等内容。

第8章主要介绍类的程序结构，基类与派生类，单继承、多继承与继承链，派生类与基类之间的内在关系，二义性与虚基类，类层次中的访问规则等内容。

第9章主要介绍友元的特性、编程方法和友元使用的局限性，静态成员的特性以及如何使用静态成员函数等内容。

第10章主要介绍静态绑定和动态绑定，实现多态性的虚拟函数，虚拟函数的技术内幕，纯虚拟函数及抽象类等内容。

第11章主要介绍重载和模板实现多态性的基本概念，运算符重载的实现，函数模板的定义及其用法，类模板的定义及其用法，模板的有关特性等内容。

第12章主要介绍异常的概念，异常处理编程方法，异常处理类等内容。

第13章主要介绍C++语言运行库中提供的流输入输出，磁盘文件的输入/输出等有关内容。

本书根据作者多年来的教学实践的讲稿并查阅了国内外相关书籍和资料写成，在此向书后列举的参考文献的作者表示衷心的感谢。另外，在本书编写过程中，张帆、程汤培、邵昊和李求实等同志对本书的部分章节进行了整理，在此致以感谢。

由于作者水平有限，书中难免会有缺点和错误之处，敬请广大读者批评指正，并给予宝贵的修改意见。

管建和

2006年底于北京

# 目　录

# 第1章　C++概述

程序设计语言是一组用来定义计算机程序的语法规则。它是一种被标准化的交流语言，用来向计算机发出指令。一种计算机语言能够让程序员准确地定义计算机需要使用的数据，并精确地定义在不同情况下应当采取的动作。本章主要介绍C++语言的发展简况及特点，C++语言的程序组成及其结构，关键字和标识符，C++的几个特殊语句，简单的输入输出，以及数制的基本知识等内容。

## 1.1　C++语言的发展简况及其特点

C++语言是在C语言的基础上发展而来的，是带类的C语言。由于C++增加了一种抽象数据类型——class类类型，从而改变了C语言的局限性，能够支持面向对象的程序设计，它是当前计算机科学的主流语言，深受程序员的欢迎。

1. C++语言的发展简况

由于C++语言是从C语言发展而来的，要介绍C++语言的发展简况，首先要讲C语言的发展历史。实际上，C语言是一种编译型面向过程的程序设计语言，它是从英国剑桥大学一个名叫Martin Richards的学者在20世纪60年代开发的BCPL(Basic Combind Programming Language)语言的基础上发展而来的。

BCPL语言是Martin Richards在开发系统软件时，作为描述性语言而使用的一种程序设计语言。利用该语言具有能够进行结构化程序设计、能够直接处理与机器本身数据相近的数据类型、能够处理与内存地址相对应的指针计算方式等特点。

1970年，美国Bell实验室的Ken·Thompson在软件开发工作中，继承和发展了BCPL语言的许多特点，进一步提出了一种叫“B语言”的语言，并且使用B语言描述和开发了当时美国DEC公司最新型的PDP-7型小型机上的UNIX操作系统。

在美国Bell实验室实现的更新型的小型机PDP-11的UNIX操作系统的研发工作中，Dennis·M·Ritchie和Brian·W·Kernighan对B语言做了进一步充实和完善，于1972年推出了一种新型的程序语言——C语言。该语言一经推出就在国际上广泛流行。

20世纪80年代，由于软件工程的需要，面向对象程序设计(Object-Oriented Programming，OOP)方法在软件设计领域引起了普遍的重视(实际上，关于面向对象的程序设计的基本概念已经提出了20多年)。AT&T Bell实验室的计算机科学家Bjarne Stroustrup结合流行的C语言的所有成分，开发出了面向对象的程序设计语言C++，因此许多人都认为C++是对C的改进或扩充。然而，C++本身确实是一种完备的程序设计语言。

图1-1说明了C++面向对象程序设计语言的发展简况。

2. C++语言的特点

C++语言发展迅速，目前已经成为最受欢迎的程序设计语言之一，究其原因，主要是

C++程序设计语言具有强大的编程功能以及语言本身具有诸多特点的缘故。目前，许多操作系统平台及其工具软件和应用软件系统均是用C++程序设计语言编写的。

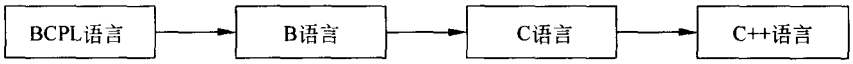


图1-1 C++面向对象程序设计语言的发展简况

**C++语言具有以下主要特点：**

1)C++语言是一种中级程序设计语言。

C++语言继承了C语言的诸多特性，它能把高级语言的基本结构和语句与低级语言的实用性结合起来。和C语言一样，C++语言能够处理汇编语言中的位、字节和地址等数据，而上述这三种数据是计算机系统运行所涉及的最基本的工作单元数据。

2)C++是一种结构化程序设计语言。

结构化程序设计语言的最大特点是代码和数据能够分离，即程序的各个部分除了必要的信息交流外彼此独立。C++语言结构化方式可使编写的程序层次清晰，便于调试、使用以及维护。C++语言的每个功能模块都是以函数形式给出的，这些函数可方便地通过类对象来调用；该语言具有多种条件分支语句、循环语句以控制程序流向，从而使功能模块程序完全结构化。

3)C++是一种面向对象的程序设计语言。

C++语言是带类的C语言，C++语言中的类类型是集抽象性、封装性、继承性和多态性于一体的数据类型，可以帮助人们开发出具有模块化、数据抽象程度高、体现信息隐蔽、可复用、易修改、易扩充等特性的应用程序。

4)C++语言是一种功能齐全的编程语言。

C++语言具有各种各样的数据类型，并继承了C语言中指针的概念，可提高编程效率。C++语言也具有访问操作系统的功能及强大的I/O功能。另外，该语言的计算功能和逻辑判断功能也比较强大，完全可以实现智能决策的目的。

5)C++语言是一种移植性很好的编程语言。

C++语言适合多种类型的操作系统，如Windows 98、Windows NT、UNIX等，因此，C++语言也适用于多种机型。一种系统平台上的C++程序代码可以不加修改或稍加修改就能在另一种系统平台上运行。

当然，C++语言还有许多其他特点，这里不再一一列举。

## 1.2 C++语言程序组成及其结构

和其他语言一样，C++语言具有自身的基本程序组成元素和结构，用户可按其规定的格式和提供的语句来编写应用程序。

下面给出一个简单的C++语言程序例子。

**【例1.1】** 编写一个求圆柱体体积的应用程序。

```
//计算公式:V=s*l=3.1415926*r*r*l
#include "iostream.h"
```

```
#define PI 3.1415926
double Volume(double,double);  //函数声明
void main(void)
{
    double radius,high; /*声明存放半径和高度的两个变量*/
    cout << "Please enter radius and high:";  //输入提示
    cin >> radius >> high;                    //输入语句
    cout << "\nThe volume of cylinder is:" << Volume(radius,high) << endl;
}
double Volume(double r,double h)  //求体积函数
{
    double v;
    v = PI * r * r * h;
    return v;
}
```

运行结果：

```
Please enter radius and high:3 5 ↙
The volume of cylinder is:141.372 ↙
```

该例是一个完整的 C++ 语言的程序。下面就利用这个例子来说明 C++ 语言的程序组成及其源程序的特点。

1. C++ 语言程序的组成

C++ 语言的程序主要有以下几个基本组成部分：

1）预处理命令：C++ 语言提供了三类预处理命令：宏定义命令（define）、文件包含命令（include）和条件编译命令（if-else-endif）。预处理命令为我们编写复杂应用程序带来了极大方便。

2）输入/输出语句（Input/Output 语句，简称 I/O 语句）：C++ 语言的程序中总少不了输入和输出的语句，以实现与程序内部的信息交流。特别是屏幕输出的功能，几乎每个程序都要用到，我们使用该功能可以把计算机的结果显示在屏幕上。

3）函数：C++ 的程序是由若干个文件组成的，每个文件又由若干个函数组成。因此，可以认为 C++ 的程序就是由若干个函数组成，函数与函数之间是相对的，并且是并行的，函数之间可以调用。

**注意**：在组成一个应用程序的若干个函数中，必须有一个也只能有一个函数名称为 main( ) 的函数。

4）语句：语句是组成程序的基本单元。每个函数都是由若干条语句组成的。但是，空函数是没有语句的函数。语句由单词组成，单词间用空格符分隔，语句以分号结束。分号既是语句的结束符，也是语句之间的分割符。C++ 中除了有说明语句、表达式语句和空语句之外，还有复合语句、分支语句、循环语句和转向语句等。

5）变量：变量是通过声明语句来定义的，大部分程序都需要声明变量和使用变量。在面向对象程序设计中，变量也称为对象。对象可看作是某类类型的实例，而实例就是某个类的对象。C++ 语言本身对变量和对象不加区分。

6）其他：除了上述 5 个部分以外，C++ 程序还有其他组成部分。例如，符号常量和注释信息也是程序的一部分。在 C++ 程序中，尽量把某些常量定义为符号常量，那么使用符号常量时，该符号常量代表某个确定的常量值。

2. 书写 C++ 程序的基本原则

书写 C++ 程序时通常要遵从以下基本原则：编写程序时，一般一行只写一条语句；短语句可以一行写多个；长语句可以分为多行来写；分行时，原则上不能将一个单词分开，用双引号引用的一个字符串也最好不要分开，如果一定要分开，有的编译系统要求在行尾加续行符(“\”)，有的编译系统则无此要求。

从例 1.1 的程序可以看出，用 C++ 语言编写的程序在格式上与许多语言不一样。

3. C++ 语言源程序的特点

C++ 语言编写的源代码程序(简称源程序)的格式具有以下特点：

1)C++ 语言的关键字是由小写字符构成的，习惯上也使用小写字母书写程序。

2)大多数语句在结尾必须用“;”作为终止符，否则 C++ 不认为该语句结束。

3)每个程序必须有一个且只能有一个主函数，即名称为 main( )的函数。

4)每个函数的函数体(包括主函数和每个子函数，如例 1.1 中的 main( )函数和 Volume( )函数)必须用一对花括号“{”和“}”括起来。

5)一个较完整的程序文件大致含有文件程序段(一组#include < *.h > 语句)、函数声明部分、全局变量声明、主函数和若干个子函数。在主函数和子函数中又包括局部变量定义、库函数调用、流程控制语句、用户函数的调用语句等。

6)注释部分包含在“/*”和“*/”之间或“//”之后，在编译时它将被 C++ 编译器忽略。

7)像其他一些语言一样，C++ 的变量在使用之前必须先声明其数据类型，未经声明的变量不能使用。变量类型的声明应放在可执行语句前面，如例 1.1 的 main( )函数中的第一条语句就是变量声明语句，它必须放在所用的执行语句前面。

8)在 C++ 语言中，大小写字母是有区别的，相同字母的大小写代表不同的变量。

9)在 C++ 语言中，程序的书写格式非常灵活，没有严格限制。例 1.1 的程序可写成下面的等价形式：

```
//计算公式:V=s*l=PI*r*r*l
#include  "iostream.h"
#define   PI 3.1415926
double Volume(double,double); //函数声明
void main(){double radius,high; cout << "Please enter radius and high:";cin >> radius >>
high; cout << " \nThe volume of cylinder is:" << Volume(radius,high) << endl;} double
Volume(double r,double h){ double v;v=PI*r*r*h;return v; }
```

上面这种写法虽然在语法上没有错误，但阅读起来很困难，层次不清，甚至有些混乱，如有错误很难查找和修改。建议读者在编程时遵从前面介绍的程序书写格式的基本原则。

为了使程序结构更加清楚、易于阅读、维护和修改，请记住以下几点：

1)一般一行只写一条语句。

2)一条复杂语句(如分支语句和循环语句)应放在多行上。

3)遇到嵌套语句应向后缩进，必要时为程序增加适当的注释行。

总结 C++ 语言的格式特点，C++ 源程序的一般格式如下：

```
<头文件包含>
<其他预编译语句>
<函数声明语句>
<全局变量声明语句>
<自定义数据类型的定义>
```

```
<数据类型> main(<参数声明列表>)
{
    <语句序列>
}
<数据类型1> sub1(<参数声明列表1>)
{
    <语句序列>
}
<数据类型2> sub2(<<参数声明列表2>>)
{
    <语句序列>
}
    ⋮
<数据类型n> subN(<<参数声明列表n>>)
{
    <语句序列>
}
```

其中：

1)sub1()，...，subN()表示用户自定义的子函数。

2)<语句序列>是由声明语句、函数调用语句、表达式语句、控制流程语句或其他语句等构成的；其他成分后面会逐个介绍。

## 1.3 关键字和标识符

任何语言都要有词汇，表达程序的词汇是由关键字和标识符等构成的。

1. 关键字

关键字也叫保留字，**是指已被 C++ 语言本身使用，不能作为其他用途使用的单词**。关键字不能用作变量名、函数名等。

C++ 语言常用的关键字如下：

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

2. 标识符

符号常量名、变量名和函数名等不能使用关键字，需要用其他符号串，即标识符，来表示。**所谓标识符是用户为程序中各种需要命名的“元素”所起的名称。**

C++ 标识符的定义必须满足以下规则：

1)所有标识符必须由字母(a~z，A~Z)或下划线(_)开头，其他部分可以是字母、下划线或数字(0~9)组成的字符串。

2)标识符是区分大小写字母的，即大小写不同，代表不同的标识符。

3)如果标识符太长，通常只有前 32 个字符有效。

4)标识符不能使用 C++ 语言的关键字。

表 1-1 中列举出了几个正确和不正确的标识符。

**表 1-1 正确标识符与不正确标识符**

| 正确 | 不正确 |
|---|---|
| i | My result |
| m_ Total | text $ |
| Filename2 | 5Filename |
| _ Debug | IsEmpty? |
| Key_ board | key. board |
| FLOAT | float |

## 1.4 几个特殊的C++语句

在C++语言程序设计中，有几个语句比较特殊，虽然简单，但很重要。下面分别介绍这几个特殊的语句。

1. 空语句

只有分号“;”组成的语句称为空语句。其一般形式如下：

```
;
```

空语句是什么也不执行的语句。在程序中，空语句通常作为空循环体。

例如：

```
while ( getchar( )! = '\n');
```

该语句的功能是，只要从键盘输入的字符不是回车则重新输入。这里的循环体为空语句。

2. 复合语句

把若干个语句用大括号({ })括起来组成的一个语句称复合语句。其一般形式如下：

```
{
   <声明1>;
   ……
   <计算1>;
   <声明2>;
   <计算2>;
   ……
}
```

在程序中应该把复合语句看成单条语句，而不是多条语句。例如，

```
{
   int x(110),y=76,z;
   z = x + y;
   cout <<"z=" <<endl ;
}
```

是一条复合语句。复合语句内的各条语句都必须以“;”结尾，在右大括号“}”的外侧可以不加分号。下列语句为一个空复合语句(相当于空语句)：

```
{
}
```

下列语句为只含有一个语句的复合语句(可以不加大括号)：

```
{
      z = x + y;
}
```

3. 注释语句

注释语句是用来向程序员提示或解释有关程序意义的语句。

C++语言的注释语句有两种形式。第一种形式如下：

```
//<注释字符串>
```

该语句为单行注释语句，如存在多行注释内容，则要使用多个这样的注释语句。

例如：

```
//loop:为循环控制变量
//sum: 用于存放输入数据之和的变量
```

第二种形式如下：

```
/*
      <注释字符串 1 >
      <注释字符串 2 >
      ……
*/
```

以“/ * ”开头并以“ */”结尾之间通常有多行注释语句，当然也可以只有单行注释语句。在“/ * ”和“ */”之间的即为注释内容。

例如：

```
/* 下面是实现求平方根的程序 */
/* 程序中变量意义如下:
     s:代表圆的面积
     r:代表圆的半径
*/
```

**注意：** 程序编译时，不对注释作任何处理。注释仅仅用来向程序员提示或解释有关程序的意义，它可以出现在程序中的任何位置。在调试程序时，可以对暂不使用的语句用注释符括起来，使翻译跳过这些语句不作处理，待调试结束后再去掉注释符。

## 1.5 简单的输入/输出

本节介绍运行库中两个常见的 I/O 函数以及流式对象的简单输入和输出。

1. 格式化输入/输出函数

C++ 语言标准运行库(runtime library)提供了两个控制台格式化输入/输出函数 printf( )和 scanf( )，这两个函数可以在标准输入/输出设备上以不同的格式读写数据。printf( )函数用来向标准输出设备(通常为计算机屏幕)写数据；scanf( )函数用来从标准输入设备(通常为键盘)上读数据。要运用这两个函数，需要在程序开始处写上预编译的包括语句：

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

(1)格式化输出函数 printf( )

printf( )函数是格式化输出函数，一般用于向标准输出设备按规定格式输出信息。这里所说的标准输出设备通常为显示器。

调用 printf( )函数的一般格式为：

```
printf( <格式化字符串 >, <参量表 > );
```

其中：

1) <格式化字符串 > 主要包括两部分：一部分是正常字符，这些字符将按原样输出；另一部分是格式化规定字符，以“%”开始，后跟一个或几个规定字符，用来确定输出内容的格式。

2）<参量表>是需要输出的一系列参数数据，其个数必须与<格式化字符串>所声明的格式化规定字符指定的输出参数个数相同，各参数之间用“,”分开，且顺序要一一对应，否则将会出现意想不到的错误。

例如：

```
printf("% d",9);
printf("x = % f",x);
printf("x = % d ,y = % d \n",x,y);
```

C++语言运行库的I/O函数提供的格式化规定符及其作用如表1-2所示。

表1-2 C++语言运行库的I/O函数的格式化规定符及其作用

| 符 号 | 作 用 |
|---|---|
| %d | 十进制有符号整数 |
| %u | 十进制无符号整数 |
| %f | 浮点数 |
| %s | 字符串 |
| %c | 单个字符 |
| %p | 指针的值 |
| %e | 指数形式的浮点数 |
| %x,%X | 无符号以十六进制表示的整数 |
| %o | 无符号以八进制表示的整数 |
| %g | 自动选择合适的表示法 |

说明：

①可以在"%"和字母之间插进数字，以表示最大场宽，下面举例说明。

例如，“%3d”表示输出3位整型数，不够3位右对齐。

```
int i =9;
printf(" i = % 3d",i);
```

输出为：i = 9

```
int i =10
printf(" i = % 3d sss",i);
```

输出为：i = 10sss

```
printf("s = % s p = % d","张三",26);
```

输出为：s = 张三 p = 26 //输出正确

```
printf("s = % s p = ","张三",26);
```

输出为：s = 张三 p = //输出错误

插进浮点数表示场宽的形式为：

```
% m.n f
```

其中，m为场宽，n为小数位数。如%9.2f表示输出场宽为9的浮点数，其中小数位为2，整数位为6，小数点占一位，不够9位右对齐。

插进字符串表示场宽的形式为：

```
% m s
```

其中m为场宽。如%8s表示输出8个字符的字符串，不够8个字符则右对齐。

如果字符串的长度或整型数的位数超过声明的场宽，则按其实际长度输出。但对浮点数，若整数部分位数超过声明的宽度，将按实际整数位输出；若小数部分位数超过了声明的宽度，则按声明的宽度进行四舍五入后输出。

另外，若想在输出值前加一些0，就应在场宽项前加0。

例如,%04d表示在输出一个小于4位的数值时，将在前面补0使其总宽度为4位。

如果用浮点数表示字符串或整型数的输出格式，小数点后的数字代表最大宽度，小数点

前的数字代表最小宽度。

例如，%6.9s 表示显示一个长度不小于 6 且不大于 9 的字符串。若长度大于 9，则第 9 个字符以后的内容将被删除。

②可以在"%"和格式字母之间加小写字母 l，表示输出的是长型数。

例如：　%ld　表示输出 long 型整数。

　　%lf　表示输出 double 型浮点数。

③可以控制输出左对齐或右对齐，即在"%"和格式字母之间加入一个"-"号说明输出为左对齐，否则为右对齐。

例如：　%-7d　表示输出 7 位整数左对齐。

　　%-10s　表示输出 10 个字符左对齐。

**【例 1.2】** 一个简单的输出小程序。

```
#include "stdio.h"
void main(void)
{
    int k=16 ; float j=98.3f ; char str[10]= "Hello! ";
    printf("k= %3d,j= %9.3f,%-10s \n",k,j,str);
}
```

运行结果：

```
k=  16,j=     98.300,Hello!↙
```

**注意：**↙为回车换行符号，以后的例子都如此。

表 1-3 是几个特殊的控制格式的规定字符。

**表 1-3　几个特殊的控制格式的规定字符**

| 字　符 | 作　用 |
|---|---|
| \ n | 回车换行 |
| \ f | 清屏并换页 |
| \ r | 回车 |
| \ t | Tab 符 |
| \ xhh | 表示一个 ASCII 码用 16 进制表示，其中 hh 是 1 ~ 2 个 16 进制数 |

(2)格式化输入函数 scanf( )

scanf( )函数是格式化输入函数，它从标准输入设备(键盘)读取输入的信息。scanf( )函数调用的一般格式为：

```
scanf(<格式化字符串>,<地址表>);
```

其中：

1)<格式化字符串>包括以下三类字符：

- 格式化说明符：格式化说明符与 printf( )函数中的格式说明符基本相同。
- 空白字符：空白字符会使 scanf( )函数在读操作中略去输入中的一个或多个空白字符。
- 非空白字符：一个非空白字符会使 scanf( )函数在读入时剔除与这个非空白字符相同的字符。

2）<地址表>：地址表是需要读入的所有变量的地址，而不是变量本身。这与 printf( ) 函数完全不同，要特别注意。各个变量的地址之间用“,”分开。

【例 1.3】 scanf( ) 函数的使用形式举例。

```
#include "stdio.h"
void main(void)
 {
       int i,j;
       printf("i =?");
       scanf("% d ",&i);
       printf("j =?");
       scanf("% d ",&j);
       printf(" i = % d,j = % d \n",i,j);
 }
```

运行结果：

```
i =?15↙
j = ?78↙
i =15 ,j =78↙
```

对于字符串数组或字符串指针变量，由于数组名和指针变量名本身就是地址，因此使用 scanf( ) 函数时，不需要在它们前面加上"&"操作符。

【例 1.4】 scanf( ) 函数的使用形式。

```
#include "stdio.h"
void main(void )
 {
     char str[20] ,strl[10], * p = strl;
     scanf("% s",p); /* 从键盘输入字符串 */
     scanf("% s",str);
     printf("% s% s \n",p); /* 向屏幕输出字符串 */
 }
```

运行结果：

```
Hello,↙
John!↙
Hello,John!↙
```

可以在格式化字符串中的"%"各格式化规定符之间加入一个整数，表示任何读操作中的最大位数。

例如在例 1.4 中，若规定只能给字符串指针 p 输入 10 个字符，则第一条 scanf( ) 函数语句变为

```
scanf ("% 10s",p);
```

程序运行时，一旦输入字符的个数大于 10，p 就不再继续读入，而后面的一个读入函数即 scanf("%s", str)就会从第 11 个字符开始读入。

2. I/O 流

C++ 语言为实现数据的输入/输出定义了许多复杂的类，这些类都以 ios 为基类，从 ios 直接或间接派生出来的，这些类形成 I/O 流库。输入和输出操作都是由“流”来处理的。所谓流是指数据从一个位置流向另一个位置。