

# C语言第三课

## 第三节 程序的结构和基本要素

赵冯平



# 目录

- ◆一、C、C++程序结构
- ◆二、数据的存储和数据的输入
- ◆三、关键字、常数、常量、变量、标识符、运算符、表达式、数据类型
- ◆四、数值计算，整型及浮点型——int、double
- ◆五、文字处理，字符型及字符串型——char、string
- ◆六、逻辑判断及逻辑运算，布尔型——bool
- ◆七、逐步求精设计过程



# 一、C、C++ 程序结构——格式

- C/C++ 程序由一个或多个函数构成，其中一个函数必须是main()主函数，它是程序执行的入口点，其它的函数是一段完成特定功能的独立程序，也称为程序（或函数）模块。本节仅介绍含有一个主函数的程序。
- 函数定义（函数模块程序结构）的一般形式如下：

```
<返回值类型> <函数名>(<类型><形式参数1>,<类型><形式参数2>, ...)  
{  
    <语句序列>  
}
```

```
#include <iostream>    //把程序库文件iostream中的功能引入（包含进入）本程序  
using namespace std;    //如果不用using namespace std;语句， 使用标准库的命令时就得写上名字空间的全名.  
int main()              //程序执行入口  
{  
    cout << "hello" << endl; //不用using namespace std;时，使用cout要写成std::cout << "hello" << std::endl;  
    return 0;  
}
```



# 一、C、C++程序结构——说明

- 例1-3的几个程序仅含有主（main）函数，它们的返回值的类型都是int，函数名main后的小括号内的参数列表为空，大括号{}内的内容是函数体。函数体内写出本函数执行的**命令**（也称为**语句**）。
- “**#include<iostream>**” 指令把程序库文件iostream引入（包含于）本程序；
- 对于指令 “**using namespace std;**” ，我们暂时这样理解：如果不用**它**，使用标准库时就得**写上**名字空间的全名，例如**cout** << "hello" << **endl**；要写成**std::cout** << "hello" << **std::endl**；
- 当执行程序时，计算机按照函数体内指令书写的顺序，依次执行指令，直至**return** 命令（或执行完最后一条指令后）才结束程序的执行。



## 二、数据的存储和数据的输入——变量、cin说明

- 使用cout命令，可输出字符串或算式的值。程序运行时用**变量**来存储数据，使用cin可以在程序运行时给程序中的**变量**输入数据，程序运行时，程序代码位于计算机内存中，同样，变量也位于计算机的内存中；
- 变量必须先定义后使用，定义就是说明变量的名字和它存放的数据类型，如整数型int或浮点型double（带小数点的数或称为实数），我们可以把变量想象成存放数据的盒子(见**例1-4**程序中的图示)。**例1-4**演示如何定义变量、输入变量值。

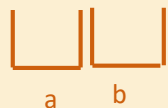


## 二、数据的存储和数据的输入——变量、输入示例

- 例1-4 编写几个程序，分别完成： A) 输入长方形的长和宽，计算并输出它的周长和面积； B) 银行存款年利率为r%，输入r及本金数额，计算并输出存一年后的本息总额； C) 火车0时0分从北京出发，开行t分钟后到达目的地，输入t，计算并输出到达目的地的时间（时:分）。

```
//A
#include<iostream>
using namespace std;
int main()
{
    int a,b;    //定义两个整型变量
    cin>>a>>b;  //输入a、b的值
    cout<<(a+b)*2<<" "<<a*b;
    return 0;
}
```

运行程序A，从键盘输入（按回车键结束）： 3 4  
//注：cin命令的作用：把3 4分别装入变量a,b  
输出结果为：  
14 12



```
//B
#include<iostream>
using namespace std;
int main()
{
    double je,r; //定义浮点型变量
    cin>>r>>je;
    cout<<je*(1+r/100)<<endl;
    return 0;
}
```

```
//C
#include<iostream>
using namespace std;
int main()
{
    int t; //定义整型变量
    cin>>t;
    cout<<t/60<<':'<<t%60<<endl;
    return 0;
}
```



## 二、数据的存储和数据的输入

- 请同学们试运行程序B和C，可参照例1-3的数据作为输入或自行设计输入数据，分析输出结果。

有了定义和使用**变量**的机制，在程序运行过程中，能通过**cin**命令输入变量值，这样程序就更灵活了。

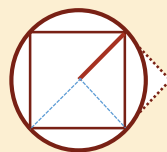
- 例1-4程序和例1-3的程序相比，功能是不是变得更强了？是不是更抽象了？



### 三、关键字、常数、常量、变量、标识符、运算符、表达式、数据类型\_1

- 例1-5: 编写程序, 输入一个整数, 代表圆的半径, 计算并输出圆的面积和内接正方形的面积。

```
1  #include<iostream>
2  using namespace std;
3  #define pi 3.14159 //定义常量pi
4  int main()
5  { //每行在双斜线之后的内容是注释, 它们的目的是为了更方便阅读程序, 但不会影响程序的运行
6    int r,s_z;           //声明2个存放整数的变量, 变量是程序运行时存放数据的场所
7    double s_y;          //声明1个存放实数的变量, 变量可被想象成存放数据的盒子
8    cin>>r;              //输入命令, 程序运行时, 输入的整数存入r。可用c函数scanf("%d",&r);
9    s_z=2*r*r;           //读出r的值, 计算内接正方形面积, 赋值给变量s_z。 =是赋值运算符
10   s_y=pi*r*r;          //读取pi及r的值, 计算后装入(赋值给)变量s_y。 =是赋值运算符
11   cout<<"Circle="<<s_y<<endl; //输出结果, 也可用命令: printf("Circle=%lf\n", s_y);
12   cout<<"Square="<<s_z<<endl; //输出Square= 及变量s_z存放的值
13   system("pause");
14   return 0;
15 }
```



➤ 输入、保存、编译并运行程序。输入 2  
按回车键, 则输出结果如下:

Circle=12.5664

Square=8

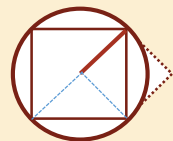




## 三、关键字、常数、常量、变量、标识符、运算符、表达式、数据类型\_2

- 下面通过几个例子介绍程序中一些要素和基本概念，以及程序设计、逐步求精技术等内容。
- 例1-5：**编写程序，输入一个整数，代表圆的半径，计算并输出圆的面积和内接正方形的面积。

```
1  #include<iostream>
2  using namespace std;
3  #define pi 3.14159 //定义常量pi
4  int main()
5  { //每行在双斜线之后的内容是注释，它们的目的是为了更方便阅读程序。
6      int r,s_z;          //声明2个存放整数的变量，变量是程序运行时存放数据的场所
7      double s_y;         //声明1个存放实数的变量，变量可被想象成存放数据的盒子
8      cin>>r; //输入命令，程序运行时，输入的整数存入r。可用c函数scanf("%d",&r);
9      s_z=2*r*r; //读出r的值，计算内接正方形面积，赋值给变量s_z。 =是赋值运算符
10     s_y=pi*r*r; //读取pi及r的值，计算后装入（赋值给）变量s_y。 =是赋值运算符
11     cout<<"Circle="<<s_y<<endl; //输出，可用命令：printf("Circle=%lf\n", s_y);
12     cout<<"Square="<<s_z<<endl; //输出Square= 及变量s_z存放的值
13     system("pause");
14     return 0;
15 }
```



➤ 关键字 常数 常量  
变量

标识符：给常量、变量对象等起的名字

运算符：+ - \* / % = >  
< >= != ==

表达式：s\_z=2\*r\*r 2\*r\*r

数据类型： int double



### 三、关键字、常数、常量、变量、标识符、运算符、表达式、数据类型\_3

我们已经了解程序的结构，下面我们结合例1-5介绍程序的一些基本概念及要素。  
敬请强烈关注。

- **关键字**：C、C++语言专用的用以表示特定语法结构的单词，又称为保留字，如int、double、return。

- **常数与常量**：第3行的3.14159为常数，常数有4种类型：

整型（整数常数，例如35）

浮点型（3.14）

字符型（如'A'，见下面例1-7介绍）

字符串常数（"Square="、" Welcome to c++!"）

第三行#define定义了符号常量pi，它的值为3.14159，编译预处理时，程序中所有的pi，均以3.14159替代。



### 三、关键字、常数、常量、变量、标识符、运算符、表达式、数据类型\_4

- **变量**：6、7行是程序中函数体的说明部分，定义了r、s\_z两个整型变量及s\_y浮点型（实型）变量，变量也称为内存变量，它是分配给运行程序的用来存放数据的带有名字的且指定了存放特定类型数据的**内存区域**。为方便理解，可以把变量想象成存放数据的盒子。

- **标识符 (ID)**：指我们给程序中的对象（函数、常量、变量等）起的名字，标识符是以字母或下划线开头，后面跟随字母、数字、下划线的字符串。

r、s\_y、PrintStr、pi、\_A、qw123、first\_name是正确的标识符；

2abc、int（是保留字）、d#是错误的标识符。给对象起名字，最好能做到“见字识义”。



### 三、关键字、常数、常量、变量、标识符、运算符、表达式、数据类型\_5

- **运算符与表达式**：算术运算符有：+、-、\*、/、%，C语言中“=”为赋值运算符，运算符把操作对象连接起来表示特定的运算就构成了表达式（例如：`pi*r*r`、`s_z=r*r`），表达式更为专业的定义是：常量、变量是表达式，运算符连接起来的表达式是表达式.....。

编程者在程序中写表达式，程序被执行时计算机进行表达式计算。对于**赋值表达式**

`(s_y=pi*r*r;)`，计算机执行时，先计算“=”右边的值，然后把结果转换成“=”左边变量类型的值，并赋值（装入）给该变量。

- **运算符** C++提供了丰富的运算符，如关系运算符“>、>=（大于等于）、<、<=、==（相等比较）、!=（不等于）”，计算机执行关系运算时，比较两个对象的关系，结果为真或假，1（非零）表示真，0（零）表示假。

例如“`4!=3`”的值为真，“`(3+2) ==5`”为真，“`3>=4`”为假。

命令“`cout<<(4!=3)`”输出的值是1；命令“`cout<<(4<=3)`”输出的值是0。



### 三、关键字、常数、常量、变量、标识符、运算符、表达式、数据类型\_6

- **赋值 “=” 运算**：有 “=” 算符的表达式称为赋值表达式（例： $s = \pi * r * r$ ，假设  $s$  是整型变量），计算机执行赋值表达式时，先计算 “=” 号右边的值（ $\pi * r * r$ ，结果是浮点型），然后把结果转换成 “=” 左边变量（整型变量  $s$ ）的类型的值（这里要去掉小数点后的数），并赋值（装入）给该变量（ $s$ ）。
- **数据类型**：常量、变量都是有特定类型的数据，类型规定了数据占用内存的大小，规定能对它们进行的运算。例如不能对浮点型进行求余 $\%$ 运算。



## 四、数值计算，整型及浮点型——int、double \_1

- **例1-6：**编写程序，输入三个整数r、h1、h2，计算图1-5所示图形的体积（下圆柱上圆锥），r为半径，h1为圆柱高，h2为圆锥高。 注意学习逐步求精的分析、设计及实现过程。
- 分析：程序一般有定义变量、输入、计算及输出4部分，据此，我们得到下面的程序框架，框架给出4个任务。题目的要求就是通过顺序执行这4个任务来完成的。输入、保存并编译程序，正确后继续.....

```
#include<iostream>
using namespace std;
int main()
{
    //1.定义需要存放数据的变量?
    //2.输入数据到指定变量?
    //3.计算?
    //4.输出体积?
    return 0;
}
```

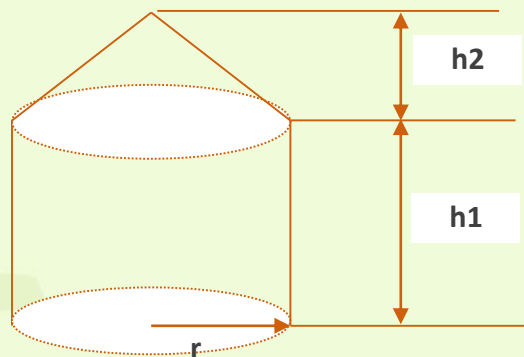


图1-5

$$V = \frac{1}{3}sh$$

$$V = \pi r^2 h = sh$$





## 四、数值计算，整型及浮点型——int、double \_2

- 例1-6：编写程序，输入三个整数r、h1、h2，计算图1-5所示图形的体积（下圆柱上圆锥），r为半径，h1为圆柱高，h2为圆锥高。
- 进一步精细化上面程序，完成任务1.及任务2.后得下面程序，编译正确后继续.....

```
#include<iostream>
using namespace std;
#define pi 3.14159 //模仿例1-4定义pi
int main()
{
    //1.定义需要存放数据的变量
    int r,h1,h2;
    double v,v1,v2,s;
    //2.输入数据到指定变量
    cin>>r>>h1>>h2;           //输入命令
    //3.计算?
    //4.输出体积?
    return 0;
}
```

把变量想象成存放数据的盒子，整型变量占4个字节，double型变量占8个字节（详见第二章）

部分变量的示意图

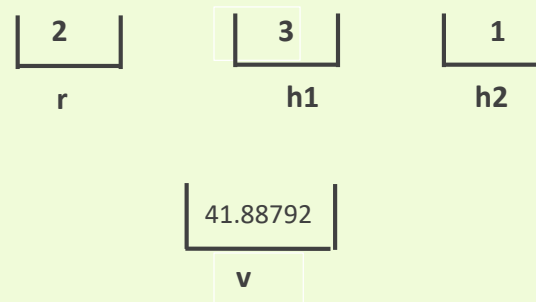


## 四、数值计算，整型及浮点型——int、double \_3

- 例1-6：编写程序，输入三个整数r、h1、h2，计算图1-5所示图形的体积（下圆柱上圆锥），r为半径，h1为圆柱高，h2为圆锥高
- 继续求精，完成任务3.及任务4.后，最终得到编译正确的版本如下

```
#include<iostream>
using namespace std;
#define pi 3.14159 //模仿例1-5定义pi
int main()
{
    int r,h1,h2;
    double v,v1,v2,s;
    cin>>r>>h1>>h2; //输入命令
    s=pi*r*r; //计算圆柱底面积，结果赋值给s
    v1=s*h1; //读取s、h1的值，计算后装入v1
    v2=s*h2/3; //取出s、h2的值,计算圆锥体积
    v=v1+v2; //取出v1、v2的值,计算后赋值给v
    cout<<v<<endl; //可用printf("%.5lf",v);
    return 0;
}
```

程序运行时假设输入了 2 3 1



若程序运行时输入  
**2 3 1**  
则输出结果如下  
**41.8879**



## 四、数值计算，整型及浮点型——int、double \_4

- 例1-6: 编写程序，输入三个整数r、h1、h2，计算图1-5所示图形的体积（下圆柱上圆锥），r为半径，h1为圆柱高，h2为圆锥高。

```
#include<iostream>
using namespace std;
#define pi 3.14159 //模仿例1-5定义pi
int main()
{
    int r,h1,h2;
    double v,v1,v2,s;
    cin>>r>>h1>>h2; //输入命令
    s=pi*r*r; //计算圆柱底面积，结果赋值给s
    v1=s*h1; //读取s、h1的值，计算后装入v1
    v2=s*h2/3; //取出s、h2的值,计算圆锥体积
    v=v1+v2; //取出v1、v2的值,计算后赋值给v
    cout<<v<<endl;//取v的值并输出，可用
    printf("%.5lf",v);
    system("pause");
    return 0;
}
```

- 运行程序，给r、h1、h2输入不同的初始数值，我们发现，cout命令自动输出（最多）六位数字。
- 我们可以用printf("%.5lf",v);命令指定输出5位小数，%.5lf称为格式控制符，它使变量v按5位小数输出。

把“例1-6”中输出命令cout改成使用printf，使程序的输出结果保留4位小数。运行并测试程序。



## 四、数值计算，整型及浮点型——int、double \_5

- **整除、商、余数** 对于C++语言的算符 “/” ，如 $m/n$ ，当 $m$ 、 $n$ 都是整数时，是求 $m$ 除以 $n$ 的商，算符 “%” （如 $m\%n$ ）是求余数，除运算的本质是平分， $q$ （商）为分得的， $r$ （余数）为分剩的。5只猴子平分17个核桃（猴子们无法破开坚硬的核桃），每只猴分得  $17/5=3$ 个，分(了之后)剩 $17\%5=2$ 个。

请特别注意： $5/2$ 的结果是2，而 $5.0/2$ 及  $5/2.0$ 的结果是2.5。下面是应用 “/” 、 “%” 运算的例子。

- 假设现在时间是下午2:23，我们很容易计算：从午夜0:0到现在逝去的分钟数 $14*60+23$ 。反过来问题是，从午夜0:0到现在时间过了863分钟，问现在的时间是几时几分？

要回答这个问题，整数的 “/” 、 “%” 运算就有了用武之地。

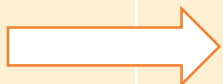


## 四、数值计算，整型及浮点型——int、double \_6

### 整除、商、余数

- **例1-7：**神舟X号飞船于某天0:0发射，共飞行了t分钟后返回地球，问返回时刻是几时几分。

```
//程序的初步设计如下
#include <iostream>
using namespace std;
int main()
{
    //分析题目要求，定义变量
    //输入
    //计算
    //输出
    return 0;
}
```



```
#include <iostream>
using namespace std;
int main()
{
    int HEnd,MEnd,t;
    cin>>t;
    HEnd=t/ 60%24;
    MEnd=t%60;
    cout<<HEnd<<':'<<MEnd<<endl;
    system("pause");
    return 0;
}
```



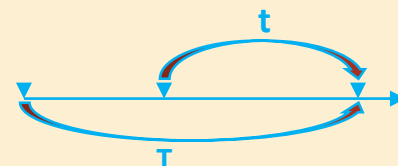
## 四、数值计算，整型及浮点型——int、double \_7

### 整除、商、余数

- 例1-8：神舟X号飞船于某天19:05发射，共飞行了t分钟后返回地球，问返回时刻是几时几分。

```
#include <iostream> //方法1 请通过具体例子理解抽象代码
using namespace std; //时间的加法运算，分钟计数是60进制
int main()           //小时计数是24进制
{
    //mjl存放积累的分钟数
    int HStart,MStart, HEnd,MEnd,t,mjl;
    HStart=19; MStart=5;
    cin>>t;           // t=1000以t=1000为例，演示计算过程
    mjl=MStart+t;      // 19:05经过1000分钟后变为19: 1005
    HEnd=HStart+mjl/60; //19+1005/60=35 35:??
    MEnd=mjl%60;        //1005%60 35:45
    HEnd=HEnd%24;       //35%24 11:45
    printf("%d:%02d",HEnd,MEnd); // MEnd输出占2位数，
    return 0;           //不足2位时前面补0
}
```

```
#include <iostream> //方法2 同例1-7思想
using namespace std;
int main()
{
    int HStart,MStart, HEnd,MEnd,t,T;
    HStart=19; MStart=5;
    cin>>t;
    T=HStart*60+MStart+t;
    HEnd=T/ 60%24;
    MEnd=T%60;
    cout<<HEnd<<':'<<MEnd<<endl;
    return 0;
}
```



- ✓ 运行程序，
- ✓ 若输入：1000
- ✓ 则输出为：11:45



## 四、数值计算，整型及浮点型——int、double \_8

### 整除、商、余数，不同类型变量之间赋值

- 整除和求余数 (%) 运算应用广泛。再次强调运算  $5/2$  与  $5.0/2$ 、 $5/2.0$  的区别，切记!!!
- 整型变量  $m=5$ ,  $n=2$ ;  $m/n$  的值是 2;  $1.0*m/n$  的值是 2.5;  $\text{double}(m)/n$  的值是 2.5,  $\text{double}(m)$  的作用是把  $m$  转换成 double 型数据。int 和 double 型数据在计算机中的存储方式、占用内存大小是不同的。

```
double x; x=3.7;
```

```
int n;
```

```
n=x; //n不能存放小数, x的值先被去掉小数点部分, 然后赋给n, n的结果是3
```

- double 值赋给 int 型变量时，要去掉小数；整型数值赋给 double 型变量时，要转换存储方式。



# 五、文字处理，字符型及字符串型——char、string \_1

恺撒密码 (Caesar cipher), 是一种最简单且最广为人知的加密技术。明文字母与密文字母对应关系表:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- **例1-9: 加密** 某部门需对长度是4个字母的单词进行加密, 使用字母替代加密法, 加密规则: 一个字母用编码 (ASCII编码表见附录) 位于其后面的第三个字母代替 (例如用 'D' 代替 'A' ), 下面程序输入4个字母的单词, 输出加密后的单词。

```
#include <iostream>
using namespace std;
int main()
{ char ch1,ch2,ch3,ch4,c1,c2,c3,c4; //定义4个变量存放输入字符数据, 4个变量存放结果
  cin>>ch1>>ch2>>ch3>>ch4;      //输入
  c1=ch1+3;                        //读取ch1的值, 进行加3运算后把结果赋值给c1
  c2=ch2+3;
  c3=ch3+3;
  c4=ch4+3;
  cout<<c1<<c2<<c3<<c4<<endl; //输出, 也可用: printf("%c%c%c%c",c1,c2,c3,c4);
  system("pause");
  return 0;
}
```

运行程序, 若输入 **good**, 则输出是: **jrrg**



C:\WINDOWS\system32\cmd.exe

LOVE  
ORYH

请按任意键继续.

## 五、文字处理，字符型及字符串型——char、string \_2

- **char 字符型** 用单引号括起来的一个符号，就是一个字符数据，char可用来定义字符型的变量。英文中的每个符号（字母、数字、标点符号等），在ASCII（American Standard Code for Information Interchange）码表中有一个对应的8位(即一个字节Byte)二进制编码，见**附录B**。下面列出一些字符的编码：

- 'A'的编码为 “01000001” ，对应的十进制数是65、十六进制数是0x41，参见第二章；
- 'a'的编码为 “01100001” ，对应的十进制数是97、十六进制数是0x61；
- 'b'的编码为 “01100010” ，对应的十进制数是98、十六进制数是0x62；
- '0'的编码为 “00110000” ，对应的十进制数是48、十六进制数是0x30；
- ' ' 空格的编码为 “00100000” ，对应的十进制数是32、十六进制数是0x20；



.....

机器中，“位”用具有两种稳定物理状态的器件表示。位、字节的详细介绍参见第二章。

1 个字节 Byte:

--	--	--	--	--	--	--	--

1 个二进制位 bit:

--



## 五、文字处理，字符型及字符串型——char、string \_3

### • 例1-9：加密

```
#include <iostream>
using namespace std;
int main()
{ char ch1,ch2,ch3,ch4,c1,c2,c3,c4; //定义4个变量存放输入字符数据，4个变量存放结果
  cin>>ch1>>ch2>>ch3>>ch4;      //输入
  c1=ch1+3;                        //读取ch1的值，进行加3运算后把结果赋值给c1
  c2=ch2+3;
  c3=ch3+3;
  c4=ch4+3;
  cout<<c1<<c2<<c3<<c4<<endl; //输出，也可用：printf("%c%c%c%c",c1,c2,c3,c4);
  system("pause");
  return 0;
}
```

C:\WINDOWS\system32\cmd.exe

LOVE  
ORYH

请按任意键继续. .

- ✓ 变量说明语句“char ch1,ch2,ch3,ch4,c1,c2,c3,c4;”定义了8个字符型变量，每个变量可存放一个字符。执行赋值表达式语句“c1=ch1+3;”时，首先把字符变量ch1存放的字符（ASCII码值）转换成整数类型，然后进行加法运算，再把结果转换成字符类型数据(8位ASCII码值)，并赋值给变量c1。类型转换详见第二章第四节。



## 五、文字处理，字符型及字符串型——char、string \_4

1. 字符常量（单个字符）要用单引号括起来，例'A'、'6'、'\*'。字符串常量（指一个以上的字符）要用双引号括起来，例"Welcome"、"A"、"^&\*!23djkf"。
2. 字符'A'占用一个字节的存储空间，字符串"A"占用二个字节的存储空间。
3. 注意字符串"2"、字符'2'和数值2的区别。
4. 8个二进制位（bit）构成一个字节（Byte），在机器中，“位”用具有两种稳定物理状态的器件表示。位、字节的详细介绍参见第二章。

1个字节Byte:

--	--	--	--	--	--	--	--

1个二进制位bit:

--

- 对于字符型数据，内部存储为整数（即ASCII编码），外部表现为文字符号。查ASCII码表（附录B）可以知道某字符的ASCII码值，也可用下面程序求出字符的ASCII码值。

机器中，“位”用具有两种稳定物理状态的器件表示。位、字节的详细介绍参见第二章。

1 个字节 Byte:

--	--	--	--	--	--	--	--

1 个二进制位 bit:

--

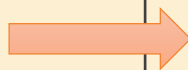




## 五、文字处理，字符型及字符串型——char、string \_5

- 例1-10：编写程序，输入一个字符，输出它的ASCII码值。

```
//初步设计的程序框架
#include <iostream>
using namespace std;
int main()
{
    //定义变量
    //输入一个字符
    //计算
    //输出对应的码值
    return 0;
}
```



```
#include <iostream>
using namespace std;
int main()
{
    int n;    //n占内存4个字节
    char ch;  //ch占内存1个字节
    cin>>ch;  //scanf("%c",&ch); 使用scanf命令，程序运行时可以输入空格符' '。
    n=ch;      //读取ch的值，自动转换为整数后赋值给(装入)变量n
    cout<<n<<endl; //取出n的值，再输出到屏幕，也可用：printf("%d",n);
    return 0;
}
```



首先输入上面左边列的程序框架，编译正确后，再一边扩展程序一边编译，.....。  
运行程序，若输入

A

则输出为

65

cin输入命令以空格为输入对象的分隔符，要输入  
“空格”字符本身，要使用scanf("%c",&ch);命令。





## 五、文字处理，字符型及字符串型——char、string\_6

- **string类型** c++提供了字符串类型（c语言没有此类型，它是c++通过类的定义机制扩展的），在程序开头增加“#include <string>”指令，即可使用string类型变量。
- **例1-11：向某人问好。**

```
1  #include<iostream> //
2  #include<string>
3  using namespace std;
4  int main()
5  {
6      string firstName,lastName,name;
7      cin>>firstName>>lastName;
8      name=firstName+" "+lastName;
9      cout<<"Hello, "<<name+'!'<<endl;
10     system("pause");
11     return 0;
12 }
```

字符串变量可以存放任意长度的字符串

Zhao

Fengping

firstName

lastName

Zhao Fengping

name

// + 是字符串的连接运算

运行程序，输入：

Zhao Fengping

输出为：

Hello, Zhao Fengping!



## 五、文字处理，字符型及字符串型——char、string \_7

### ●例1-11：向某人问好。

```
1  #include<iostream> //字符串变量可以存放任意长度的字符串
2  #include<string>
3  using namespace std;
4  int main()
5  {
6      string firstName,lastName,name;
7      cin>>firstName>>lastName;
8      name=firstName+" "+lastName;           // + 是字符串的连接运算
9      cout<<"Hello, "<<name+!"<<endl;
10     system("pause");
11     return 0;
12 }
```

- 再测试运行例1-11程序，按1次回车键、空格键及（或）Tab键，再输入Zhao，然后按1次回车键及空格键或Tab键，再输入Fengping，最后按回车键。输出结果是否和原来的一样？ **结论：跳过空白，输入字符串**



## 五、文字处理，字符型及字符串型——char、string\_9

- **例1-6：**编写程序，输入三个整数r、h1、h2，计算图1-5所示图形的体积（下圆柱上圆锥），r为半径，h1为圆柱高，h2为圆锥高。 注意学习逐步求精的分析、设计及实现过程。

```
#include<iostream>
using namespace std;
#define pi 3.14159 //模仿例1-5定义pi
int main()
{
    int r,h1,h2;
    double v,v1,v2,s;
    cin>>r>>h1>>h2; //输入命令
    s=pi*r*r; //计算圆柱底面积，结果赋值给s
    v1=s*h1; //读取s、h1的值，计算后装入v1
    v2=s*h2/3; //取出s、h2的值,计算圆锥体积
    v=v1+v2; //取出v1、v2的值,计算后赋值给v
    cout<<v<<endl; //取v的值并输出，可用
    printf("%.5lf",v);
    system("pause");
    return 0;
}
```

- 再测试运行例**1-6**程序，在输入的3个数据2、3、1之前或中间按任意次数的空白符（指空格、Tab、回车符Enter），输入数值1之后的回车表示输入结束。结果输出与原来一样。

- **cin是如何“吃进”数据的？**

空白符（空格、Tab、回车符Enter）被略过，直到有效输入数据



## 五、文字处理，字符型及字符串型——char、string\_10

### ●例1-11：向某人问好。

```
1  #include<iostream> // 字符串变量可以存放任意长度的字符串
2  #include<string>
3  using namespace std;
4  int main()
5  {
6      string firstName,lastName,name;
7      cin>>firstName>>lastName;
8      name=firstName+" "+lastName; // + 是字符串的连接运算
9      cout<<"Hello, "<<name+"!"<<endl;
10     system("pause");
11     return 0;
12 }
```

- 例1-11表明，**string**类型的字符串可以进行赋值(=)及连接运算(+); 对于连接运算+，**+号两边的运算对象有一个必须是string类型的字符串变量**。例如 "abc"+"xy"或"abc"+"!"是错误的用法。



## 五、文字处理，字符型及字符串型——char、string \_11

- 字符串的比较操作：

```
string s,t;
```

```
s="as";t="AS";
```

```
cout<<(s<=t)<<endl;    //a的ASCII码是97， A的ASCII码是65， 结果是0
```

```
s="as";t="asd";
```

```
cout<<(s<t)<<endl;      //结果是1
```

```
s="^&*%";t="^&%$@";    // *的ASCII码是42  %的ASCII码是37
```

```
cout<<(s<t)<<endl;      //结果是0
```



## 五、文字处理，字符型及字符串型——char、string\_12

- 字符串是一个整体，但是也可以存取(访问)字符串中指定的第几个字符，例：

char ch; int k; string s="abcde"; //定义变量s，同时设置初值为"abcde"

s[1]= 'A'; //存 (赋值) 字符串s的第二个字符设置成字符'A'

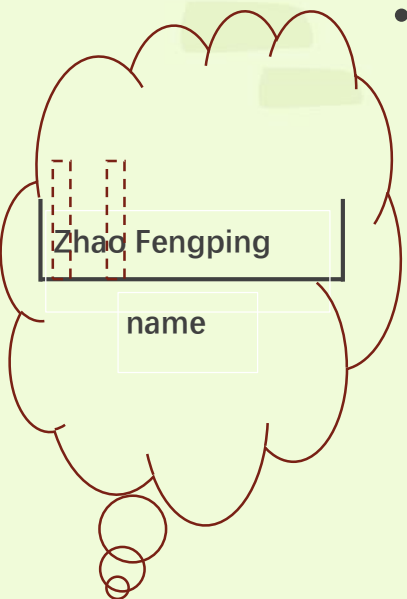
cout<<s[0]<<s<<endl; //取 输出s的第一个字符及s，即：aaAcde

ch=s[2]; //取 取字符串s的第三个字符'c'存入字符变量ch

k=2; cout<< s[k]<<endl; //取 输出字符串s的第k+1 (3) 个字符

cout<< s[1+2]<<endl; //取 思考：输出字符串s的几个字符？输出是什么？

k=3; cout<< s[k-1]<<endl; //取 思考：输出字符串s的第几个字符？输出是什么？





## 五、文字处理，字符型及字符串型——char、string\_13

- 用函数**size()**或**length()**求出字符串长度，从而知道字符串s中有几个字符，例：

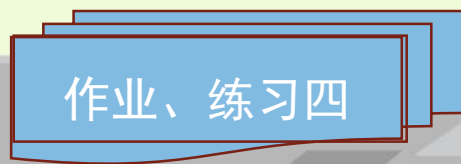
```
string s="abcde"; int len; //定义变量s，同时设置初值为"abcde"
```

```
cout<< s.size()<<endl; //显示字符串s的长度。s.size()即字符串s的长度，此处输出5
```

```
len= s.size();
```

```
cout<< s[len-1]<<endl; //输出字符串s的最后一个字符。
```

```
cout<< s[s.size()/2]<<endl; //思考：输出字符串s的哪一个字符？
```



• int 整型

• double 浮点型

• char 字符型

• string 字符串型

• bool 逻辑型

运算	bool	char	int	double	string
赋值	=	=	=	=	=
+			+	+	
连接					+
减			-	-	
乘			*	*	
除			/	/	
求余			%		
&&    !	适用				
!= < >=等所有比较	适用	适用	适用	适用	适用



## 六、逻辑判断及逻辑运算，布尔型——bool \_1

- 除了进行数值计算、文字信息处理，电脑也能完成逻辑判断类任务，下面我们介绍逻辑型数据、逻辑型变量及逻辑运算。布尔型（bool）是c++语言特有的类型。
- 例1-12：逻辑运算** 输入两个整数，判断其中一个是否为另一个的因数，输出1表示“是”，0表示“否”。

```
#include <iostream>
using namespace std;
int main()
{ int n1,n2;
  bool f;           //定义布尔（逻辑）型变量f
  cin>>n1>>n2;
  f=(n1%n2==0)||(n2%n1==0); //取n1、n2的值,进行逻辑运算后赋值给f。
                           //n1除以n2的余数等于0或者n2除以n1的余数等于0

  cout<<f<<endl;
  system("pause");
  return 0;
}
```

- bool** 布尔类型
- “bool f;”语句定义了布尔类型变量f
- 布尔类型数据只有两个值  
“1”（真）  
“0”（假）；



C:\WINDOWS\system32\cmd.exe

```
2 4
1
请按任意键继续.
```

C:\WINDOWS\system32\cmd.exe

```
4 2
1
请按任意键继续.
```

C:\WINDOWS\system32\cmd.exe

```
5 3
0
请按任意键继续.
```



## 六、逻辑判断及逻辑运算，布尔型——bool\_2

- 逻辑指的是思维的规律和规则，是对思维过程的抽象。“||”、“&&”、“!” 逻辑运算是“或者”、“而且（并）”、“否定”的数学抽象。

例如，老师说：“同学们明天带足球或者带篮球来学校”，家长说：“你每天要学习而且要锻炼身体”，若能准确理解上述两句话要你遵守的规则，那么简单的逻辑运算不难理解。令a表示“某（明）天带了足球来学校”，b表示“某（明）天带了篮球来学校”，那么： $a \& \& b$  及  $a || b$  所表示的意义就是……。

要准确表达复杂的逻辑，需要学习并自觉运用“逐步求精”方法；学习逻辑代数知识，效果会更好。



## 六、逻辑判断及逻辑运算，布尔型——bool\_3

- **bool** 布尔类型 “bool f;” 语句定义了布尔类型变量f，布尔类型数据只有两个值 “1”（真）和 “0”（假）；布尔运算也称为逻辑运算，布尔运算有 “与（&&）、或（||）、非（!）” 三种，定义如下：

a	b	a&&b	a  b	!a	!b
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0



## 六、逻辑判断及逻辑运算，布尔型——bool\_4

- $|x| \leq 10$  (x的绝对值小于等于10, 即  $-10 \leq x \leq 10$ ), c/c++表达式表示为: **`(x >= -10 && x <= 10)`**
- a、b不同时为零, c/c++表达式表示为: **`!(a == 0 && b == 0)`** 或者 **`(a != 0 || b != 0)`**
- 字符变量ch的值是大写英文字母, c/c++表达式表示为: **`(ch >= 'A' && ch <= 'Z')`**
- &&算符的优先级高于|| (就像乘法\*的优先级高于加法+), 式子 `a || b && c`, 机器先计算 `b && c`;
- 输入一个字符, 判断其是否为大写英文字母, 程序片段如下:

```
char ch; bool f;  
cin >> ch;  
f = (ch >= 'A' && ch <= 'Z');  
cout << f << endl;
```





## 七、逐步求精设计过程 \_1

- **例1-13：落实逐步求精** 编写程序，输入一个三位整数，求其各位数字之和及平均值，保留2位小数。
- 在Dev-c++集成开发环境窗口中单击菜单“**文件[F]**”，选择子菜单“**新建[N]**”、“**源代码[S]**”，并输入以下代码（代码包含返回值为0的程序基本框架，同时描述了解决的问题，本例要求完成的大题目，按分步思想，被分成了4个要完成的小题目），保存为文件名li13Sum.cpp，编译li13Sum.cpp。

```
#include<iostream>
using namespace std;
int main()
{
    //1.分析题目要求，确定需要定义的变量，然后说明变量？
    //2.输入一个三位整数？
    //3.计算？
    //4.输出和、平均值？
    return 0;
}
```



## 七、逐步求精设计\_2

- **例1-13：落实逐步求精** 编写程序，输入一个三位整数，求其各位数字之和及平均值，保留2位小数。
- 进一步对各个问题求精（解决各个小题目），得到下面的程序版本(1、2小问题已解决)，编译，检查程序是否正确，如果有错，检查修改后重新编译.....。

```
#include<iostream>
using namespace std;
int main()
{
    //1.分析题目要求，确定需要定义的变量，然后说明变量
    int n,ge,shi,bai,sum;           //个、十、百位数字用变量ge,shi,bai存放
    double avg;
    //2.输入一个三位整数
    cin>>n;
    //3.计算ge,shi,bai, sum  avg?

    //4.输出和、平均值 sum  avg?

    return 0;
}
```



## 七、逐步求精设计\_3

- 例1-13：落实逐步求精 编写程序，输入一个三位整数，求其各位数字之和及平均值，保留2位小数。
- 编译正确之后继续进行求精细化。最后得到下面版本：

```
#include<iostream>
using namespace std;
int main()
{
    //1.分析题目要求，确定需要定义的变量，然后说明变量
    int n,ge,shi,bai,sum;
    double avg;
    //2.输入一个三位整数
    cin>>n;
    //3.计算ge,shi,bai, sum avg
    ge=n%10;           //123%10 的结果是3
    shi=n/10%10;       //123/10%10=12%10=2
    bai=n/100;         //123/100=1
    sum=ge+shi+bai;
    avg=sum/3.0;
    //4.输出和、平均值：sumavg
    cout<<"sum="<<sum<<endl;    //printf("sum=%d\n",sum);
    printf("average=%.2lf\n",avg); // average=原样输出，avg的值在%.2lf位置按指定格式输出
    system("pause");           //程序暂停命令
    return 0;
}
```

➤ 编译并运行程序，  
输入317，按“回车键”，输出结果如下：

sum=11

average=3.67



## 七、逐步求精设计\_4

- **例1-13**的另一个重点，是要大家学会整数各位数字的拆分， $n/10$ 的结果是 $n$ 去掉个位后的值， $n/100$ 的结果是 $n$ 丢去了低二位后的值，例如算式“ $1234/100$ ”的值为12， $n\%10$ 的结果是 $n$ 的个位数， $n\%100$ 的结果是 $n$ 的最低二位数。所以两种运算 $n/10\%10$ 、 $n\%100/10$ 都能求出整数 $n$ 的十位数字。
- 程序中的语句“ $\text{avg}=\text{sum}/3.0;$ ”，可改为“ $\text{avg}=\text{double}(\text{sum})/3;$ ”， $\text{double}(\text{sum})$ 把整型变量 $\text{sum}$ 的值转换为 $\text{double}$ (浮点)型，  
或者改为“ $\text{avg}=1.0*\text{sum}/3;$ ”。



# 练习——测试 / 运算

- 练习：把例1-13语句改为“`avg=sum/3`”，然后运行程序，输入317，结果会怎样？请作出说明。

```
#include<iostream> //例1-13
using namespace std;
int main()
{
    //1.分析题目要求，确定需要定义的变量，然后说明变量
    int n,ge,shi,bai,sum;
    double avg;
    //2.输入一个三位整数
    cin>>n;
    //3.计算ge,shi,bai, sum avg
    ge=n%10;           //123%10 的结果是3
    shi=n/10%10;       //123/10%10=12%10=2
    bai=n/100;         //123/100=1
    sum=ge+shi+bai;
    avg=sum/3.0; //修改这里
    //4.输出和、平均值：sumavg
    cout<<"sum="<<sum<<endl;    //printf("sum=%d\n",sum);
    printf("average=%.2lf\n",avg); // average=原样输出，avg的值在%.2lf位置按指定格式输出
    system("pause");             //程序暂停命令
    return 0;
}
```



## 练习——比较：秒转换成时:分:秒，三位整数拆分。

- 练习：比较例1-7，由时间总分钟数计算时、分的值，与例1-13求一个三位数的百、十及个位数字，发现都是使用/、%运算，它们有共同点吗？实际上，时间的时:分:秒计数方法是：秒是60进制、分是60进制、小时是24进制。

例1-7：神舟X号飞船于某天0:0发射，共飞行了t分钟后返回地球，问返回时刻是几时几分。

```
#include <iostream>
using namespace std;
int main()
{
    int HEnd,MEnd,t;
    cin>>t;
    HEnd=t/ 60%24;
    MEnd=t%60;
    cout<<HEnd<<':'<<MEnd<<endl;
    system("pause");
    return 0;分
}
```





# 习题

## • 练习一

1. 输入2个整数，表示三角形的底和高，计算并输出三角型的面积。
2. 输入正方形的边长 $a$ (有小数点)，计算它的周长 $c$ 和面积 $s$ 。提示： 测试数据： 输入 2.15 输出 8.600 4.622

## • 练习二

1. 编程：输入三个整数 $a$ 、 $b$ 、 $c$ ，求它们的平均值 $avg$ 和乘积 $product$ 。你能找到编译后产生的扩展名为".exe"的文件吗？
2. 摄氏温度 $c$ 与华氏温度 $f$  的转换公式： $f=c*9/5+32$ 。输入实数 $c$ ，输出华氏温度 $f$ 。 测试数据： 输入 26.8 输出 80.24
3. 神州X号飞船飞行了 $H$ 小时 $M$ 分钟 $S$ 秒，问飞船飞行的总秒数 $t$ 是多少？编程输入三个整数，分别表示 $H$   $M$   $S$ ，输出以秒计数的飞行总时间 $t$ 。 测试数据： 输入 1 30 10 输出 5410
4. 输入一个整数 $n$ ，输出它的个位数字。 测试数据： 输入 253 输出 3 输入 127 输出 7



# 习题

- 练习四

1. 定义两个字符串变量s1,s2，输入两个字符串，比较s1,s2的大小，即输出s1<s2的值，分析结果。
2. 输入一个单词（长度是奇数的字符串），输出它的长度及居中的一个字符。输入abcde 输出5 c

- 练习五

1. 输入实数X，判断绝对值 $|X| > 10$  ( $x > 10$ 或者 $x < -10$ ) 是否为真。测试数据：输入9.2 输出0
2. 奇偶数判断，输入一个整数n，当n是偶数时，输出1，否则输出0。

- 综合练习



再 见！





# ASCII 码表（附录B）

ASCII表																										
( American Standard Code for Information Interchange 美国标准信息交换代码 )																										
高四位	ASCII控制字符												ASCII打印字符													
	0000						0001						0010	0011	0100	0101	0100	0111								
	0						1						2	3	4	5	6	7								
低四位	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl	
0000	0	0		^@	NUL	\0	空字符	16	▶	^P	DLE	数据链路转义	32		48	0	64	@	80	P	96	`	112	p		
0001	1	1	☺	^A	SOH		标题开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q		
0010	2	2	☹	^B	STX		正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r		
0011	3	3	♥	^C	ETX		正文结束	19	!!	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s		
0100	4	4	♦	^D	EOT		传输结束	20	⏏	^T	DC4	设备控制 4	36	\$	52	4	68	D	84	T	100	d	116	t		
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK	否定应答	37	%	53	5	69	E	85	U	101	e	117	u		
0110	6	6	♠	^F	ACK		肯定应答	22	—	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v		
0111	7	7	•	^G	BEL	\a	响铃	23	↕	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w		
1000	8	8	☐	^H	BS	\b	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x		
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM	介质结束	41	)	57	9	73	I	89	Y	105	i	121	y		
1010	A	10	☐	^J	LF	\n	换行	26	→	^Z	SUB	替代	42	*	58	:	74	J	90	Z	106	j	122	z		
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	\f	换页	28	└	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124			
1101	D	13	🎵	^M	CR	\r	回车	29	↔	^]	GS	组分分隔符	45	-	61	=	77	M	93	]	109	m	125	}		
1110	E	14	🎵	^N	SO		移出	30	▲	^^	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~		
1111	F	15	🎵	^O	SI		移入	31	▼	^_	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	␣	^Backspace 代码: DEL	
注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。																										

