

简单信息的表达和运算

算术运算和自增自减运算

赵英良



引言

- ▶ 表示运算的符号称为**运算符**。
 - 算术运算
 - 比较运算
 - 逻辑运算
 - 位运算

1. 算术运算

- ▶ **算术运算**包括加、减、乘、除、乘方、指数、对数、三角函数、求余数等，这些都是算术运算。
- ▶ C++中用`+`，`-`，`*`，`/`，`%`分别表示加、减、乘、除和求余运算，例如
`a+b`，`a-b`，`a*b`，`a/b`
- ▶ 若`a`,`b`都是整数，则`a%b`表示`a`除以`b`的余数，如
`int a=26; //声明整型变量a并初始化为26`
`int b=10; //整型变量b并赋初值10`
`cout<<a%b<<endl;`
得到的结果就是6

运算符作用的对象称为**操作数**



%运算的两个操作数
必须都是整数

注意事项

- ▶ (1)乘法运算符不能省略
- ▶ 数学中写 b^2-4ac ，程序中应写为
 $b*b-4.0*a*c$
- ▶ (2)除数是表达式时注意加括号，例如
- ▶ 应写为
 $x/2.0/a$ 或 $x/(2.0*a)$
- ▶ 不能写成
 $x/2*a$ $x/2a$

$$\frac{x}{2a}$$

其他算术运算的实现

- ▶ C++中**没有乘方运算符**，也没有指数、对数、三角函数的运算符，它们是通过函数的调用实现的。如

`pow(x,n)` x^n

`exp(x)` e^x

`log(x)` x的自然对数的对数

`log10(x)` x的常用对数

`sin(x)` x的正弦（弧度）

`cos(x)` x的余弦（弧度）

`tan(x)` x的正切（弧度） $1/\tan(x)$ 余切

`asin(x)` 反正弦函数 $x \in [-1, 1]$

- ▶ 它们都属于数学函数，使用时需要在程序的开头写

```
#include<cmath>
```

- ▶ 例如

```
#include<iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int main()
```

```
{ double x;
```

```
  cin>>x;
```

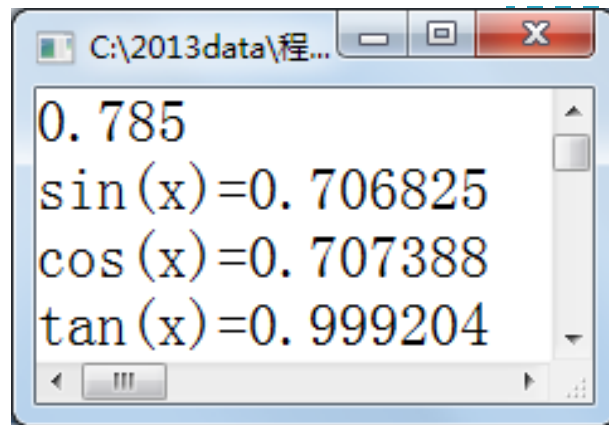
```
  cout<<"sin(x)="<<sin(x)<<endl;
```

```
  cout<<"cos(x)="<<cos(x)<<endl;
```

```
  cout<<"tan(x)="<<tan(x)<<endl;
```

```
  return 0;
```

```
}
```



对指数运算， x^n ，如果 n 是2,3,4较小的整数时，一般使用连乘更简单

$x*x$ $x*x*x$ $x*x*x*x$

2. 自增自减运算符

- $i = i + 1;$
- $i = i - 1;$
- 自增运算符 $++$
- 自减运算符 $--$

双目运算（二元运算）：

运算符需要两个操作数

单目运算（一元运算）：

运算符需要一个操作数

- $i++$, $i--$, $++i$, $--i$
- 前置 $++$: $++i$, 前置 $--$: $--i$
- 后置 $++$: $i++$, 后置 $--$: $i--$

作用

(1)无论前置、后置，++使变量自身加1，--使变量自身减1

```
int n=5,m=10;
```

```
n++; //相当于n=n+1;
```

```
m--; //相当于m=m-1
```

```
cout<<n<<endl; //结果6
```

```
cout<<m<<endl; //结果9
```


作用

- (1)无论前置、后置，++使变量自身加1，--使变量自身减1
- (2)作为表达式时，前置运算符的结果是加1或减1之后的结果；后置运算符的结果是加1或减1之前的结果

```
int n=5,m=10;
```

```
int a,b,c,d; //声明整型变量
```

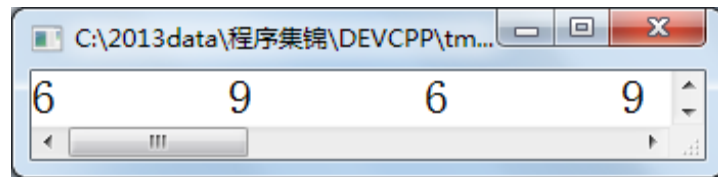
```
a=n++; b=m--;
```

```
cout<<a<<" \t"<<b<<"\t"<<n<<'\t'<<m<<endl;
```

```
n=5,m=10;
```

```
a=++n; b=--m;
```

```
cout<<a<<" \t"<<b<<"\t"<<n<<'\t'<<m<<endl;
```



建议

- ▶ 使用单独的表达式语句

```
int n=20;
```

```
i++;
```

```
i--;
```

```
++i;
```

```
--i;
```

- ▶ 尽量不将其放入表达式中

```
int a,n=10;
```

```
a=i+++i++; //不推荐
```

```
cout<<i++<<'\\t'<<i++<<'\\t'<<i++<<endl; //绝对不用
```

3. 算术运算符的优先级

- ▶ 先乘除求余,后加减, 级别相同的从左向右逐个计算

$1+2*3$ $2+3+4$ $6*2/3$

- ▶ 圆括号可以改变优先级

$(1+2)*3$

- ▶ 圆括号可以嵌套, 越靠里, 圆括号的优先级越高。

$((1+2)*3+(2+3)*2)*7$

- ▶ $\%$ 和 $*$, $/$ 优先级别相同

建议使用圆括号
标明计算的优先顺序

- ▶ 后置 $++$ 、 $--$ 的优先级 $>$ 前置 $++$ 、 $--$ 的优先级

- ▶ 前置 $++$ 、 $--$ 的优先级 $>$ 乘除的优先级

4.运算的数据类型

- ▶ (1)相同类型的数据进行算术运算，结果还是这种数据类型
 - $1+2$, $1-2$, $1*2$, $1/2$
 - $1.0+2.0$ $1.0-2.0$ $1.0*2.0$ $1.0/2.0$
- ▶ (2)不同类型的数据运算，默认向级别高的类型转换
(低) $\text{short, char} \rightarrow \text{int} \rightarrow \text{unsigned} \rightarrow \text{long} \rightarrow \text{double}$ (高)
 $1/2.0$ $1.0/2$

▶ (3)强制类型转换

- (`<类型>`)`x` 或 `<类型>`(`x`)
- 将`x`转换为`<类型>`的数据
- `x`可以是表达式
- `(double)a/b`
- `int(3.0/2.0)`
- `double(1/2)`

5. 复合运算符

- ▶ 为简化书写，C++ 提供复合算术运算符

$+=$, $-=$, $*=$, $/=$, $\%=$

- ▶ 双目运算

$a+=b;$ 等价于 $a=a+b;$

$a-=b;$ $a=a-b;$

$a*=b;$ $a=a*b;$

$a/=b;$ $a=a/b;$

$a\%=b;$ $a=a\%b;$

总结

▶ 算术运算符

- +, -, *, /, %; 没有乘方运算

▶ 自增自减运算

▶ 优先级

- 先乘除、求余； 后加减

▶ 运算的数据类型

- 整型和整型运算结果是整型(下取整)

▶ 强制类型转换

- 在变量或表达式前写 (<类型>)



1 / 2 结果为0

1.0 / 2 结果为0.5

```
int a=7,b=3;
```

```
double c;
```

若想得到a除b的实数结果

```
c=(double)a/b;
```