简单信息的表达和运算

混合运算的类型转换

赵英良

1.混合运算和隐式转换

- 2+5=7
- 整型2+浮点5　行吗？成吗？中不中？
- C++中允许相关的数据类型进行混合运算。
- 相关类型
  - 尽管在程序中的数据类型不同，但逻辑上进行这种运算是合理的
- 相关类型在混合运算时会自动进行类型转换，再计算
- 隐式转换

2.算术转换

- 转换原则：
  - 确保计算值的精度。转换后尽可能不丢失有效数字。
  - short，2字节，能表示的数小
  - int，4字节，能表示的数大
  - int 转short，要丢掉2字节的信息，不可取
  - short 转 int，增加存储空间，但不丢失信息，可行

- 转换方式
  - 字节少的向字节多的转换
    - char, unsigned char,short,unsigned short ,bool → int

      cout<<('a'+5)<<endl;//'a'转换为int再加5，结果为整型
  - 精度低的向精度高的转换
    - int →float →double

      cout<<(5+3.5)<<endl; // 5转换为double型再相加
  - 有符号向无符号转换　　// 结果为double
    - int → unsigned int

      unsigned int ui=32678;

      cout<<(5+ui)<<endl; //5转换为unsigned int

3.隐式转换时机

```
int a=256+97;
double b=256+97.625;
char c1,c2;
c1=a;  //不推荐
c2=b; //不推荐
```

- 混合类型的表达式

```
int n=2;
char c='a';
cout<<(c+n)<<endl;  //'a'的ASCII为97，加2得整型99
```

- 赋值
  - 向左值类型转换

```
doube a=8.12;
int k;
k=a;//a转换为int，舍去小数，k为8
```

向低精度转换 会 损失有效数字

- 在需要关系、逻辑表达式的地方

非0转true；0转false

4.显式转换

- 程序中明确标记转换的类型，就是显式转换，也就是强制类型转换
- 格式

  <类型>(<表达式>)

  (<类型>)<表达式>

举例

```
double a=128,b=30;
int n=3;
a=(double)n/2;//强制将n转换为double
n=int(a/b);      //强制将a/b的值转换为int
```

优先级：高于乘法和除法，
和逻辑非、按位取反同级
结合顺序：从右向左

总结

- (1)相同类型的数据运算，结果的类型不变
- (2)不同类型的数据运算，自动向精度高的类型转换
- (3)赋值运算，向左值类型转换
- (4)强制类型转换
  - （<类型>）变量或常量
  - <类型>（<表达式>）
- (5)精度高向精度低的类型转换，会损失精度
- (6) 整型相除，结果为整型，小数会被舍去。★

①整型/整型=整型，3/2=1,而不是1.5
②double赋值给int,会舍去小数
③char和int相加，是字符的ASCII和整数相加