

# 小马加编信息学教案(五)

## 关系运算符与逻辑结构

- 一. 课程内容
- 二. 知识讲解
  - 1. 关系运算符
    - \* 1.1 关系运算符种类
    - \* 1.2 关系运算符使用注意
    - \* 1.3 关系运算符应用举例
  - 2. 逻辑运算符
    - \* 2.1 逻辑运算符种类
    - \* 2.2 逻辑运算符的优先级
    - \* 2.3 逻辑运算符使用注意
    - \* 2.4 逻辑运算符应用举例
- 三. 经典例题
- 四. 提高巩固

### 一. 课程内容

1. C++关系运算符
2. C++逻辑结构

### 二. 知识讲解

#### 1. 关系运算符

##### 1.1 关系运算符种类

C++ 提供了六种关系运算符

关系运算符	含义	应用举例	表达式结果
==	等于	1==2	0
!=	不等于	1!=2	1
<	小于	1<2	1
>	大于	1>2	0
<=	小于或等于	1<=2	1
>=	大于或等于	1>=2	0

关系表达式的结果为逻辑值真（非0值）或假（0）。

同时注意，非零值即为真，零值即为假

## 1.2 关系运算符使用注意

- “等于”与“赋值”的区别；

注意等于是"=="，而赋值是"="

如果误用了"="，则得到的是赋值语句的值，赋值语句的值非零即为真

- 实型数据（浮点数）的关系运算；

浮点数在计算机中存储会产生一定的误差(极小但是存在)

所以不能直接判断两个浮点数是否相等

如果要判断两个数是否相等，可以选择判断两个浮点数差值的绝对是否小于一个很小的数(可取  $10^{-6}$ )

ex:

```
fabs(a - b) < 1e-6
```

- 运算符的优先级，如下表：

算术运算符	*、/、%	由高到低 ↓
	+、-	
关系运算符	<、>、<=、>=	
	==、!=	
赋值运算符	=	

## 1.3 关系运算符应用举例

假设变量定义为 `int x = 6, y = 4, z = 5;` 请分别写出下列表达式的值。

- (1) `x + y > y + z`      (2) `'x' < 'z'`  
 (3) `x == x >= y`      (4) `(x < y) > (y <= z)`

## 2. 逻辑运算符

### 2.1 逻辑运算符种类

C++ 提供了 &&（逻辑与）、||（逻辑或）和！（逻辑非）三种逻辑运算符  
其中，&& 和 || 是双目运算符，而！是单目运算符。

- 逻辑非运算规则

A	!A	A	!A
0	1	1	0

- 逻辑与运算规则

A	B	A&&B	A	B	A&&B
0	0	0	1	0	0
0	1	0	1	1	1

- 逻辑或运算规则

A	B	A  B	A	B	A  B
0	0	0	1	0	1
0	1	1	1	1	1

三种逻辑运算是分支语句的重要基础，必须牢牢理解并且掌握

- 逻辑非即是简单的取反
- 逻辑或当且仅当两者都为假时为假，其他均为真
- 逻辑与当且仅当两者都为真时为真，其他均为假

### 2.2 逻辑运算符的优先级

优先级	运算符	结合性	优先级	运算符	结合性
1	( )	从左到右	6	==,!=	从左向右
2	!,(正),(负),++,--	从右到左	7	&&	从左向右
3	*,/,%	从左向右	8		从左向右
4	+(加),-(减)	从左向右	9	==,+=,-=,*=,/=,%=	从左向右
5	<,<=,>,>=	从左向右			

### 2.3 逻辑运算符使用注意

- 整型数据在参与逻辑运算时，根据其值是 0 或非 0，分别当作逻辑值“假”或“真”进行处理

- 逻辑表达式是“短路计算”(重要)

在逻辑表达式的计算过程中，一旦能够确定整个表达式的值，就会立刻停止计算。

## 2.4 逻辑运算符应用举例

写出下列语义的 c++ 表达式

- (1) 一个 0~100 之间（不包括 0 和 100）的数  $a$ 。
- (2)  $x$  大于  $y$  且  $x$  大于  $z$ 。
- (3)  $x$  是英文字母。

# 三. 经典例题

1. 阅读下列程序，写出程序结果，并上机编程检验答案

```
#include<iostream>
using namespace std;

int main(){
    int n1 = 4, n2 = 5, n3;
    n3 = (n1 > n2);
    cout << n3 << " , " ;
    n3 = (n1 < n2);
    cout << n3 << " , " ;
    n3 = (n1 == 4);
    cout << n3 << " , " ;
    n3 = (n1 == 5);
    cout << n3 << endl;
    return 0;
}
```

```
#include<iostream>
using namespace std;

int main(){
    char x,y;
    x = ' m ' ;
    y = ' n ' ;
    int n;
    n = x < y;
    cout << n << endl;
    n = x == y - 1;
    cout << n << endl;
    n = ( ' y ' != ' Y ' ) + ( 5 > 3 ) + ( y - x == 1 );
    cout << n << endl;
    return 0;
}
```

```
#include<iostream>
using namespace std;

int main(){
    int a, b, c, d;
    a = 2; b = 3; c = 0;
    d = (a > b) && (c = 4);
    cout << d << " " << c << endl;
    return 0;
}
```

2. 编写一个程序，输入两个字符  $a$  与  $b$ ，如果  $a$  是  $b$  的小写字母，则输出 1，否则输出 0

提示:输出一个判断  $a$  是否是  $b$  的小写字母的关系表达式的值即可(成立该值为 1,否则为 0)

样例输入	样例输出
a A	1
b P	0

3. 编写一个程序，输入两个正整数  $a$  与  $b$ ，如果  $a$  和  $b$  有一个非零，则输出 1，否则输出 0

提示:输出一个判断  $a$  与  $b$  是否有一个为非零的逻辑表达式的值即可(成立该值为 1,否则为 0)

样例输入	样例输出
4 0	1
0 0	0

## 四. 提高巩固

1. 编写一个程序，输入两个正整数  $a$  与  $b$  ,如果  $a > b$  ，则输出 1 ，否则输出 0

提示:输出一个判断 $a$ 是否大于 $b$ 的关系表达式的值即可(成立该值为 1 ,否则为 0 )

样例输入	样例输出
14 15	0
90 55	1

2. 编写一个程序，输入一个正整数，如果该年份是 3 的倍数，输出 1 ，否则输出 0

提示:输出一个判断能否被 3 整除的关系表达式的值即可(成立该值为 1 ,否则为 0 )

样例输入	样例输出
15	1
55	0

3. 编写一个程序，输入一个年份，如果该年份是闰年，则输出 1 ，否则输出 0 .

提示:输出一个判断闰年的关系、逻辑复合表达式的值即可(成立该值为 1 ,否则为 0 )

样例输入	样例输出
2016	1
2100	0