Writing Conditions (Class Notes)

16/05/2024

Author: Siddharth Jha

Conditions are statements which return true or false, Conditions are written using relational and logical operators, Sometimes, Arithmetical Operators are also used. Conditions look like : a>b , a<b , a>b && a<b , a!=b, a==b, a+b==1

Arithmetic Operators: + , - , * , / (division) , % (modulo)

Precedence of Arithmetic Operators: * / and % have same priority, after them: + and -

relational Operators : > (greater than)
< (less than)
>= (greater than or equal to)
<= (less than or equal to)
== (equals to)
!= (not equal to)

Logical Operators: ! (Logical NOT) ** Reverses meaning of a condition
&& (Logical AND) ** Returns true if both conditions are true
|| (Logical OR) ** Returns true if at least one condition is true

condtion1 && condition2 **:** this combined condition returns true if condition1 and condition2 both are true

condtion1 || condition2 *:* this combined condition returns true if at least one of the conditions is true or both are true.

Precedence of logical operators: ! (NOT) && (AND) || (OR)
**precedence means order of execution (like BODMAS in Mathematics)

Lets see a few conditions….

To check whether a number A is greater than B or not.
A > B : if this condition returns true, we can say A is greater than B

To check whether a number A is smaller than B or not.
A < B : if this condition returns true, we can say A is smaller than B

To Check if A and B are equal
A==B : if this condition returns true, we can say A and B are equal

To check if A and B are unequal
A != B : if this condition is true, we can say A is not equal to B

To check if A is greater than or equal to B
A>=B: if this condition is true, we can say A is greater than or equal to B

To check if A is smaller than or equal to B
A<=B: if this condition is true, we can say A is smaller than or equal to B

Examples:

To check if A is greater than 5:	A>5
To check if B is less than 1 :	B<1
To check if C is equal to 2:	C==2
To check if sum of a and b is equal to 10:	a+b == 10
to check if k is not equal to 3:	k != 3
to check if n is greater than or equal to 10:	n >= 10
to check if p is smaller than or equal to k:	p<=k

Checking Multiple Conditions:

***** We need logical operators to check multiple conditions:***

To check if a number N is between 10 and 20 (excluding 10 and 20):
N>10 && N<20

To check if a number N is between 10 and 20 (including 10 and 20)
N>=10 && N<=20

To check if N is 10 or 20 (here N needs to be exactly 10 or 20)
N== 10 || N==20

To check if N is not 10 and not 20
N != 10 && N != 20

to check if N is anything except between 10 and 20
! (N>=10 && N<=20) can also be written as N<10 || N>20

Divisibility Check:

*****A is divisible by B if we get 0 as remainder after dividing A by B. to get remainder we can use modulo (%)***

To check if A is divisible by 2: A%2 == 0 (it means that we are getting 0 as remainder after dividing A by 2)

To check if N is divisible by 5: N % 5 ==0

to check if N is divisible by i : N %i==0

To check if A is divisible by B: A%B ==0

to check if N is a multiple of 3: N%3==0

*****The statements N is multiple of 3 and N is divisible by 3 are same***

** Getting the last digit of a number

To get the last digit of a number N , we need to divide N with 10 and take the remainder , the remainder is the last digit. k= N% 10; here k stores the last digit of N

Examples 12 %10 is 2
34 % 10 is 4
100 % 10 is 0
45% 10 is 5
78 % 10 is 8
99 % 10 is 9

If we write **k=123 % 10;** k will store 3 as 3 is the last digit of 123

To check if a number N ends with 5 : N % 10 == 5
To check if a number N ends with 7: N%10 ==7
To check if a number N ends with P : N%10 ==P

***** we use these conditional statements in conditional constructs (if -else if -else) and in conditional operator (ternary)***

A few examples of how to use conditional statements (**Although we will learn conditional construct in next chapter**)

Ternary Syntax: *(condition) ? if true : if false;*

double k= (5>4) ? 1 : 0; here k will store 1 as the condition is false

String s= (7> 2) ? "HI" : "Hello"; here s will store "HI" as the condition is true

int p= (2==4) ? 5 : 25; here p will store 25 as the condition is false

If-else syntax:

if(condition) {

Codes inside this block execute if condition is true

}**else**{

Codes inside this block execute if condition is false
}

Example:

```
if(n > 0){
System.out.println("Positive Number");
}else{
System.out.println("Negative Number");
}
```

*** here, if n is greater than 0 “Positive Number” is printed otherwise it prints “Negative Number”*

Conversion from if else to ternary:

if-else:

```
int p;

if(a>b){
p=1;
}else{
p=2;
}
```

Using ternary:

```
int p= (a>b) ? 1 : 2 ;
```