Mr G's Java Jive

#14: Complex Comparisons

Sometimes, in an **if** statement, a simple comparison of one condition isn't good enough. Sometimes you need to compare two or more. This quick sheet explains how to do that.

Simple vs Complex

Just about all of our previous comparisons were simple ones. Things like **if(age<0)** or **if(choice==1)**. These only checked for one condition. But if you've taken the time to "look under the hood" in **gatling.java**, you might have found some times when I've had to compare two or more conditions. And we definitely had to do this in our last program when we said **if(choice<1 || choice>4)**. When we did that last one, I gave you just enough explanation to get you through it. Now it's time for a little more.

And and Or

If you remember any logic from your math classes (provided, of course, that they even went over logic in your math classes), you'll remember the difference between a logical **and** and a logical **or**. Just in case you don't remember (or never learned it), I'll go over it here.

A logical **and** says that **both** sides of the condition have to be true in order for the whole condition to be true. For example, you have to be 16 **and** have a valid license to drive a car. If you're 16, but have no license, you're out of luck. If you have a valid license, but aren't 16…well wait a minute…you'd have to be 16 to get the license anyway. But you see what I mean. With an **and** comparison, if only one part of the comparison is true, then the whole thing is **false**.

A logical **or**, on the other hand, says that only **one** side of the condition has to be true in order for the whole thing to be true. For example, if you won $1,000,000 **or** your family owns the dealership, you can have a new car.

Logical Symbolism

The symbols for **and** and **or** are very simple. For **and** it's **&&**. Pay careful attention to the fact that that's **two** ampersands (the real name for the **and** sign) in a row, with **no space** between them. For **or** it's **||** (that's **shift-backslash** twice), once again, with **no space** between them.

Now that you know this, you can write such sophisticated statements as:

if(num>0 && num<53)
if the number is greater than 0 **and** less than 53
if(num<1 || num>52)
if the number is less than 1 **or** greater than 52
if(year%400==0 || (year%100!=0 && year%4==0))
now there's a tricky one, it's an **or** that includes an **and** as part of the second condition
also notice that the second set of conditions is inside of parentheses so it's treated as one

In all cases you have to compare two **completely written out** conditions. You can't say **if(num>0 && <53)**.

Another Promise Kept

I told you this would be another short one. Handout #15 will deal with loops and counting.

This page intentionally left **almost** blank