Transcript: Statements

>> In this segment, we'll look at if statements using logical operators. So our first example, if the temperature, we'll call that temp, is greater than 80, and the humidity is greater than or equal to 60, then we wanna tell the user to stay indoors. So an if statement like that would look like so.

If (temp greater 80, and, notice this is the logical operator, and, humidity greater than or equal 60. If both of those are true, then we wanna print out, hot, stay indoors. Now, when this statement is processed, temp is compared to 80. If that's true, then we go on and see if humidity is greater than or equal to 60, and if it's true we print out the the statement there.

If when processing this statement temp is not greater than 80, if that's false, then we're done, and so we just skip that statement. If it is true we go on and check humidity if it's false then we're done and we skip to the end. So that's just a simple if, and, and we combined, both the relational operators and, and the logical operator and.

So here's a next example, now this is an an if-else statement. Notice up at the top, this is really just an if with an optional else. All these if, if-else, and else-if-else statements are really just if statements with a, with a optional else or else ifs, and we'll see that here.

So, suppose we wanna add, an otherwise, that the weather is good if those previous conditions are false, then we would have something like this. If temp greater than 80 and humidity greater than or equal to 60, we print out hot, stay indoors, else we print out, the weather is good.

So, when this statement is executed exactly one of these print statements is gonna gonna actually be executed. So if the conditions there are true we print the first one, otherwise we print the second one. We'll never print both of them, just one or the other. So, here's another one and now, now we're adding an else if right before the else, and, and again these are just optional.

This really is just an if statement with optional else if and else. So here's another condition here. Suppose we say if the temperature is greater than 80 and humidity, greater than or equal to 60, tell the user it's hot, so that's gonna be the if part. Otherwise, if the temperature is less than 40, let's tell the user it's cold, that's gonna be in our else if.

And then finally, for any other condition, let's tell, our user the weather is good, so that, that's in the else. So this is what the statement would look like. If temp greater than or equal rather, temp greater than 80, and humidity greater than or equal to 60, gonna stay indoors.

Else if the temperature is less than 40 we're gonna say it's cold, stay indoors. And then finally an else, this is all other cases we just say the weather is good there. So, just looking at these ,um, keep in mind that an if statement can have any number of else if blocks, and then you can follow that with else if you like, and there's two ways to write this.

The first an and technically correct, this is really just a if statement, followed by an else. And then inside the else, we've got another if, and it's got an else, and inside it, we've got another if statement, and it's got an else. You know, if we kept writing it this way we would cascade sort of across the page, if we had many different options there and we could have 15 or 20 or more.

So what usually happens is we line all of these up and we put the if on the same line with the else, which is what we've been doing in all of our examples. That way, when reading this, you can just sort of scan down, and see all of the different possibilities, and if none of those work, you've got the else at the end there.

Now just keep in mind though the else or else if clause is optional, all of these are if statements. So, let's look at a couple of examples in, in, in the ID, in j graphs. We'll look at, essentially this example we've just shown here and we'll add the control structure diagram.

And then we'll also look at a, the triangle clause. So, here's the first one and this is just in the examples for conditionals and loops. So you go to conditionals and loops, and then the examples folder. And here, I've, I've double-clicked on if-else example, and, and we have it here, added line numbers.

Those are optional, of course, we can take those off. Now keep in mind, I'm gonna put them back in here just to sort of see them. So here, this is the way we normally write, we've got if, condition one, and then whatever we wanna do. If that's true we do this, otherwise we go down here and look at, the else if and see if the condition true is, condition two is true.

If it's true we do that block. And so on all the way down till we catch that, that else at the end. And then this is the cascading version, as it cascades across. Now, we can remove the control structure diagram. The button up here on the menu does that.

And so this is what it would look like without the diagram, a little bit harder to tell where we're going. For instance, after this condition here, if we happen to do this one say condition true is true. It may not be clear where we go next, but where we're going is to this if statement.

So if we've got the diagram, we can sort of see that. So we, we, this is the true path we're zipping on down there are two main statements here. You see the main stem for this first if statement and then the second one. So we go in here and check to see if this condition is true.

If it's true, we do this block, and then we ascape through, the dotted lines, if you will, and go look for the next stem, and we're down here at that second if statement. Same sort of thing here, wherever we find something true, we keep looking. And whenever it's true, we go down in here we would, we would just end.

So that's that's the way the control structure's diagram is read with if statements. It should be sort of obvious, but in case you didn't realize your sort of flowing through the dotted lines, that's what we've got. Now, keep in mind when you come in here the the when we look a the horizontal stem, here, on the if statement, we go in and we follow the true, this is the true path, the solid bar.

And, if that's false, then we go the dotted line, which is the false path down to the next, if else. And so, the dotted lines are false, and {UNKNOWN] condition is false, we follow it, and so that's what we've got. Okay, so let's look at a triangle this is a real example.

And we will be looking at the triangle clause in a number of examples following here, and so let's try to understand what's going on here. So we got client's triangle, and it has three fields. And these are all doubles, S1, S2, and S3, and notice they are declared private, so private double.

And these represent the size of a triangle. We're going to create triangles here, if you will. And these are the three sides, side 1, side 2, and side 3, if you will. And then here is our constructor, public triangle. Takes in side 1, side 2, and side 3, and then just assigns these to S1, S2, and S3, just like you'd expect.

And then it's the get classification method that we're gonna concentrate on here. This, combines our, logical operators with also the relational operators and, and checks for equality. So, to be a triangle, we're classifying this triangle as either isosceles, may not be a triangle, or equilateral, or scalene. So those are the three ways we are gonna, classify this.

Recall an isosceles has two sides equal, and an equilateral has all sides equal, and scalene, none of the sides are equal. So this is the way our if statements look here. We're gonna just say it's isosceles up front, and then we'll change it if it turns out to be something else here.

So, going into the if statement, here, we wanna make sure, all the sides are greater than or equal to 0. Rather greater than 0. So we're saying here if side 1 is less than or equal to 0, then its not a triangle. Or if side 2 is less than or equal to 0, its not a triangle.

And if side or if side 3 is less than or equal to zero, its not a triangle. So we will say result is not a triangle, and then result would get returned there. So, assuming all this is false, and, and,and, and the sides are all greater than 0.

That's the only way we would get to this next else if here. So if that's the case, now we've got another condition to check. Turns out we've got three sides, and if one of the sides is greater than the sum of the other two, it also can't be a triangle.

You can just sort of imagine this, you've got two sides, say you've got the long side. Imagine the long side, and then the two other sides you've added, if they can't reach each other because the long, the side, the other side is greater than their sum, then you're not gonna have a triangle either.

So here we're looking at, if side 1 is greater or equal to side 2 plus side 3, or if side 2 is greater than S1 plus S2, or side 3 is greater than or equal S1 plus S2. If any of those are the case, it's also not a triangle, OK?

In that case we would set results not a triangle and go down here and return. Well if we get through all of that, then we just check to see if S1 equal equal S2 and S2 equals S3. If these are both the same, these are added, and so if these are both true, then all three sides, by the transitive property, S1 = S2 = S3.

So we have an equilateral triangle, and we would set result of that, and go down and return the result. The last if-else, is checking to see if none of the sides are equal. So here we're seeing S1 is not equal to side 2, S1 is not equal not side 3, and S2 is not equal to side 3.

That actually takes care of all the cases there. And so none of the sides are equal, we'll set the result to scalene out and return that. This also contains an equals method, where we're gonna decide that two triangles are equal if all their sides are equal. So we pass in triangle t here, and then we check to see if s1 == t.s1, and s2 == t.s2.

And and so on here, same with S3 equals side 3 of the triangle passed in. If all those are true then we return, our fact that would evaluate to true and that's what we get returned. If this evaluates to false, in other words they're not all equal respectively, then we would return false there.

Then here's a two string for a triangle. We would just print out the string here, side 1, equals side 1 units. So this might be, suppose the sides were 3, 4 and 5, we would say side 1 equals 3 units, side 2 equals 4 units, side 3 equals 5 units.

And then we would get the classification, and then we would say that that's a, since they're all different, that's gonna be a scalene. So, we'll look at this example, in more detail later. But I just wanted to go through and show you some, some, use, nice use of, of combining the, relational operators, inequality operators, with logical operators, or and and specifically there.

And also the not with respect to not equal. That's actually not a logical operator, that's that's a the equality operator. Okay, next we'll take up, loops.