AUBURN UNIVERSITY

Transcript: Conditional Operators

>> Now let's take a look at more conditionals and loops. We'll begin with a switch statement, which can be used in place of if statements, with special restrictions. Then we'll look at the conditional, or ternary operator. This is like a mini if-else statement, but since it's an operator, it can actually be placed inside other statements.

And then we'll consider three looping statements, the do-while, the for and the for-each. Let's begin with the switch. Consider the following if statement, where input is a char value, and notice we've designed answer to be of type string. And we say, if input == t, the char t, then we're gonna assign, an answer, to the string value, true.

Else if, input == f, we'll assign an answer to the string value, false. Otherwise, in all other cases, as in else, an answer, will be assigned invalid. So, a corresponding switch statement would look like this. Here we have, switch on input. And input is an expression, in this case, it's the single variable of type char.

And so, it evaluates to a char, and then we're gonna jump directly to the case label for that particular char. And so if it evaluates to t, we would jump to case t and assign an answer, to the string, true. And then we would break out of the case, if it's an f, and so on there.

All the way down to a default, which is everything else. So if it's not a, a t or an f, it would jump directly to default, and the answer would then assign the string invalid. So let's look at this a little closer now and again. So input is, evaluated, and then suppose it's f, it jumps directly to case f.

Where the statements in, and beginning in case f there, are executed down to a break. And then if a break, is encountered, it actually breaks out of the switch. And and just keep in mind in here that case t, case f, and default in our statement here are labels, not executable statements.

So we'll see in a moment that we could leave out the break, sometimes we want to do that, but in most cases, we don't. So let's take a look at an example, true or false, it's this, this same code, but we'll see how that jump takes place. So here we've got the switch.

And down here is the, corresponding, if, if statement. And so, I'm gonna run this in debug mode. And I'll key in f to begin with. And then as I step, the switch is gonna get, is gonna evaluate input here, and then it's gonna jump directly to case f.

And then we execute the two statements there, the break statement being the last one we actually break out. We go down there and print. And the corresponding if statement, we would actually have a booming expression, input == t. And that's gonna be false, and then we would check to see if it's f, and that would be true in this case.

So we'll go in, and an answer would get assigned to the string false, and then we'll jump out and print, OK? Let's run it one more time in debug mode, and this time I'm gonna key in a z. Now if the switch input will be evaluated, notice over in the debug tab here, it's a z, and it's not gonna match any of the case labels.

So it's gonna jump directly into the default to the first statement, past the default label, if you will, and that gets executed. In the corresponding if statement, and think in terms of a very long, if else, if else, if else, if. And so on, if we were checking for a lot of characters, how inefficient this would be.

Here we're checking to see if it's true, that's false. We check to see if it's f, that's false, and finally we get, get to the invalid. So the difference between the switch and the if statement is, is that the switch evaluates this expression and jumps directly to the corresponding label.

And bypasses all these other, Boolean expressions that you'd find in the if statement. So next, we'll take a look at, the situation where, break statements are left out.