>> Now let's see what happens when there's no break statement in the switch. Suppose in our example, input is t. We would expect it to jump to case t. And then because of the absence of break statements, it's gonna continue and right on through all of the statements there.

And it's gonna actually end up with answer equal to invalid, regardless of what input you keyed in. So this is probably not what we meant here. Let's go over here and see how the CSV changes. Notice here with the break statements, it looks pretty similar to an if statement.

And that if you go into case t and that works out, you end up breaking out and you go down here and the next thing you do is print the answer and so on with case f. If we take the break statement out, I'll just comment it out here.

And we'll recompile it. Now notice you can iterate any case but once you get in, you're staying in since there was no break statements. And that's, that's probably not, not what we want to happen. Again if we run this in debug mode, key and a t. And then as we step here, we're gonna step in at case t, so we get true and notice we'll just match right on though there, okay.

Probably not what we want to do here. Uncomment these guys. And we'll let that recompile. So again, we probably didn't mean to leave out the breaks here. We actually need them to make the logic work correctly. But let's consider an example where we want to print the remaining days in the week.

Using what we call a fall through switch statement. This is gonna be one with one or more missing break statements. So here's our fall through. In fact, notice we don't have I think we have a, yeah, we have a break statement way down at the end here. But the idea is here, the user is gonna key in, a day of the week and an abbreviation for it, for instance, SU or Sun for Sunday.

It would be FR or FRI for Friday down here. And what happens day of week gets evaluated, we drove it to uppercase, of course, and then we've got all these uppercase so they'll match. We're using strings in this case instead of chars, and whatever day of week was keyed in, it'll jump to, to that label.

Suppose TU was keyed in. It'll jump to the TU label and then begin executing the statements below that until it reaches a break statement. And so it's gonna go ahead and print out if TU is keyed in, Tuesday, Wednesday, Thursday, Friday, Saturday and then break out. And if it doesn't match anything here it would actually go to the, default.

So let's, let's run this in debug mode. It's waiting for day of week and let's key in, we'll key in Tuesday, tu here. So we're about to switch on the day of week. We expect it to jump directly to TU, or actually the statement, inside that label. The statement following the label, even though we have multiple labels here.

And then we just start marching through this, since there's no break statement, all the way down until we get to a break statement, which is after we print Saturday. Okay, so this is the use of a fall through switch where we've got some missing breaks. And a couple things to notice here.

We can enter in any of the labels, notice, and we would enter here and check both of these labels of course. And depending on where we enter, we stay in just like the control structure diagram indicates and also notice that we can double up the labels. Here I've listed them separately and sometimes for readability, it's better to list them on separate lines but we can also double them up like so.

It's also permissable. Although I didn't do it here, to actually list the case all on the same line. So we could've, could've done this as well. Do this for a couple of them. This is gonna work as well. Just run a debug mode again. This time I'll key in, let's key in the year, TH.

So I expect this to jump directly to Thursday. There's the label, Thursday, and it'll print out Thursday, Friday, and Saturday. So lots of flexibility with the, with the switch statement. And if you need to break out after a particular case, you definitely need to put the break statement.

But sometimes it's useful like in this example to omit break statements.