



>> Now, let's consider when to use a switch statement. Two things come to mind, when when, if you're checking to see if a result is equal to several different values. That is, you've got a lot of `==` logic. And, when you have alternatives based on an acceptable switch expression type.

Now we've looked at two cases there, one where we were using a `care` switch type and the other a `string` switch type. But we can actually use others, in Java 6 and earlier, the switch statement worked on `char`, `byte`, `short`, and `int`. And beginning in Java 7 and continuing, the switch statement also works on the wrapper classes for these listed above here.

As well as `string` and `enum` types, and we, we used it on a `string` in our examples of, example of the fall through, switch. So, let's look at an example here. We're gonna look at a little tax program, with `if else`. And, and then also with the switch statement, to sort, sort of compare how that would work.

So here, we're gonna have the user enter their income, and we're going to create a tax bracket for them, figure out what that is. And we're going to divide income by 25,000. So no matter what, what the income, we're gonna end up with the tax bracket in our program here.

In our, and we've got `if else` for, the first four tax brackets. And then, above that, we would just print that it's greater than whatever the tax bracket happened to be there, and, and so on. So let's, let's try this, let's run it in debug mode, it's wanting an income.

Let's give it an income of let's say, \$80,000. So, that's gonna be a `taxBracket = 3`, which we see here is, is, at least 75 and less than 100,000. And so, in our `if` statement, we're actually gonna go through each one of these conditions. Each one of these Boolean expressions and evaluate them, and eventually we get to one that's true.

A tax bracket that does equal 3, and so we print that out, and, and, away we go there. Let's run one more example, let's suppose we've got an income of \$1 million. We think big here. And let's step here, here again we go through all of them. Now suppose we had, you know, 30 or 40 tax brackets, we would keep going and going.



But eventually we'd get to the else there, and we'd find out that this is tax bracket 40, and so on. Now, if we look at the, at the same program using a switch. Since we cleverly have tax brackets that are based on an int, it's the same code up here.

We can use a switch on the tax bracket, and just depending on which one it is, we'd jump straight down to it, and so on. So let's, let's run this one in debug mode. Waiting on income, let's give it that income of \$80,000 again. We've got an income of \$80,000, now, we're about to switch on, on the tax bracket, which is 3.

And those, we're gonna jump directly to case 3, we're not gonna go through the other cases. We're gonna get there and sorta be done with it, and so on. Now, suppose we had a, a million dollars, that million dollars we had before. That's tax bracket 40, so we could actually, in this program, and it would be pretty clear logic here, have 39, 40 cases.

In this case, since we, in this situation, since we only have four cases. This is gonna go all the way to the default when it gets evaluated, just like we'd expect. So we should jump all the way to default, and it did there. So, just to review here, we talked about, when to use one and then here's why.

Depending on the circumstances, the switch statement can reduce the code's visual complexity, and possibly even the logic. When you think about the remaining days of the week example we looked at, with that fall through switch. An if statement version of that would have replicated print statements, and you know, been a little bit messier there.

And then the other reason, and perhaps this is the main reason. Is that, a switch statement does jump directly to, to the, correct case label. Whereas an if else is going to evaluate every Boolean expression, until it gets to one that's true, or all of them are false.

And so, depending on what you're doing in the program, a switch can make your program, much more efficient. In fact, think of the case, how the operating system handles character input from the keyboard. We've got, up to 255, combinations of keys there, 0 to 255, and this is actually not handled as an if statement.



If you keyed in a z, you would have to go through a through z, or beginning at, code 0 there. But instead it's handled with a switch type statement. So whatever key you press, it goes directly there, and you'd go on.