Welcome back! In this video we'll discuss another process to control the flow of a program.

The switch statement, unlike an if-else statement which only evaluates a condition to either be true or false, can evaluate and execute one of many outcomes or code blocks.

After this video you'll know how to write a C# switch statement and understand how it controls program flow based on a condition or variable.

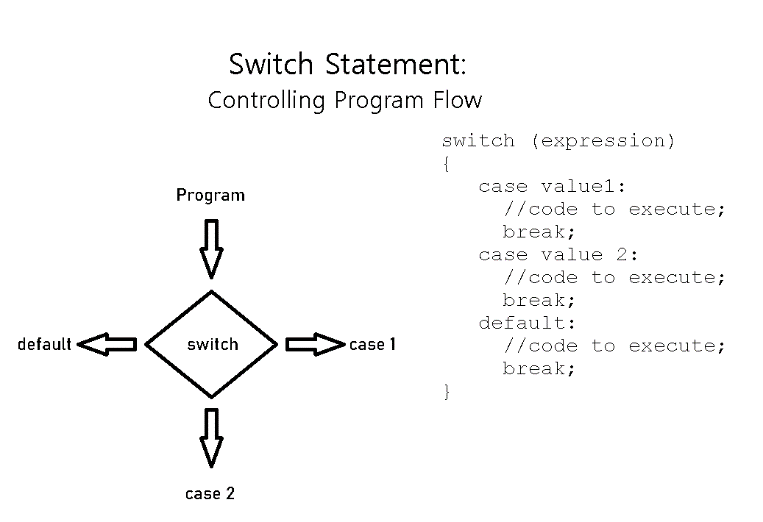
If you don't have prior programming experience with an object oriented programming language like C++ or Java, I recommend watching the previous videos especially the one about writing a C# if-else statement.

By now the Visual Studio IDE should be a familiar platform for developing C# programs.

The topics for this video are introduction to the switch statement and a sample code template, then we'll open up Visual Studio to write and execute a basic switch statement, followed by re-writing the basic if-else statement with altered logic to provide different functionality for the program.

AGENDA

* Introduction to a basic code template for a Switch statement
* Demonstrate a basic Switch statement in action
* Altering the logic of a Switch statement
* Summary

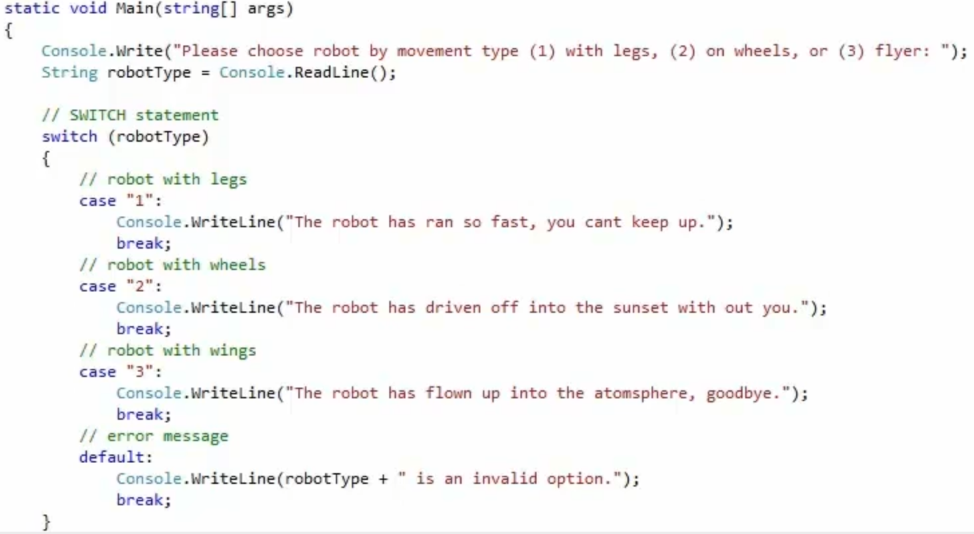
Let's being with an overview of a switch statement and a sample code template.

On the left is a state diagram showing the program flow for a switch statement. The code on the right is a template for a switch statement that takes a condition or variable of type character, integer, double, or string.

Each case is specific to one value and it includes a code block to execute and a break statement which skips to the closing curly bracket and continues the program execution as normal.

The default block is like an else clause and contains the default code for execution or displays an error message.

Everyday examples include choosing to pay with cash, credit, credit card, debit card, or check. Or selecting the shipping method for an online order, or choosing from the three types of Robots from the example code from the previous video.

Next we'll write a C# statement based on the previous code template. So here we are back in Visual Studio, with a code example of a switch statement starting at line 17.

This code behaves just like the chained if-else code from the previous videos. Each case value is associated to a Robot choice of 1, 2, or 3 just like in the if statement. The program will print out a menu of the three Robot choices, read the user's input, and assign the value to the variable robotType, then pass the value to the switch.

The switch statement will then go to the correct case clause and print out the associated code.

If the value is not represented by a case, then the program execution goes to the default clause and prints out an error message.

Let's press F5 to run the program in debug mode and I'll pull up the console application.

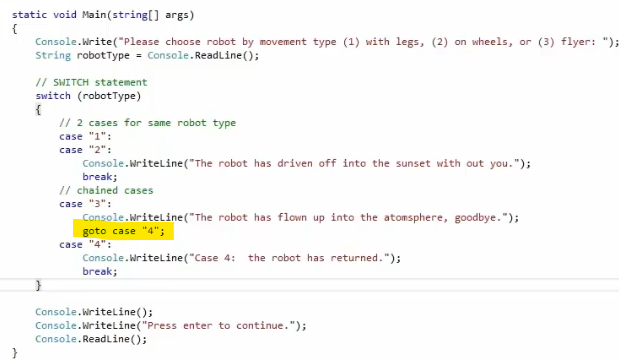
So far the program is functioning the same as the previous if-else program with the three menu options. So I'm going to choose the flying robot by entering the value 3 the same message appears that the robot has flown away.

Now I'm going to run the program again but this time I'll choose 1 for the walking robot and as you can see the robot has run away.

Let's quickly run the program one more time with an invalid input of 90. As you can see, the default clause has been triggered and the error message has been printed out.

Returning back to the code, as you can see the switch statement is easier to manage and maintain than multiple if-else statements – especially when there are several outcomes or values to check for. For example if you're choosing a way to ship you're online order, or when you go to the store and you need to select a payment option.

At this point I've discussed the basic structure to write a switch statement.

But the logic can be altered to provide different functionality. For instance, executing the same code for two or more cases, chaining multiple case clauses that will execute one after another, or not even having a default clause at the end.

At this point I'm going to replace the current code that you see here with code that I've written previously.

I'm going to cut & paste and remove the code.

In this example, option 1 and 2 will print out the same message whereas option 3 will print out the robot that has flown away and also execute case number 4.

If you notice I've omitted the default clause at this point. So let's run the program this time and I'll select option 1 for robot with legs, but if you noticed it's also printed out the message that it drove off.

Please choose robot by movement type <1> with legs, <2> on wheels, or <3> flyer:

1

The robot has driven off into the sunset without you.

Press enter to continue.

This can be useful when the same code should be executed for multiple variables or conditions such as a signature box which appears when you're paying with either a credit card or debit card, or a different type of credit card.

Let’s run the program one more time, selecting option 3 which will print out the robot flying away but it will also print the message for case number 4 that the robot has returned.

Please choose robot by movement type <1> with legs, <2> on wheels, or <3> flyer:

3

The robot has flown up into the atmosphere. Good bye.

Case 4: the robot has returned.

Press enter to continue.

To recap, the if-else and switch statements can both control the program flow with logical decisions based on a condition, user's input, or simply a value.

When the state has two outcomes, true or false, the if statement is an ideal scenario, whereas the switch is best used for multiple values or choices where you have more than two options.