

Switches

C# Programming
The switch construction

©Rob Miles

Making Choices

- A program can use the C# conditional statement to make a decision when it runs
- This allows a logical value to decide which of two statements are performed
- However, we often need to make a choice between multiple options
- That is what the switch construction is for

Making a Choice

```

Enter the type of window:
1 = casement
2 = standard
3 = patio door
  
```

- The above code lets a user select the type of window as part of a more advanced double glazing program

C# Handler Methods

```
static void handleCasement ()
{
    Console.WriteLine("Handle Casement");
}
```

- It makes sense to create a method to handle each window type
- The program must then select the correct method based on the user input

Chapter 9 : Switches 9-Nov-12 ©Rob Miles 4

Selecting Methods using if

```
if ( selection == 1 )
{
    handleCasement();
}
else
{
    if ( selection == 2 )
    {
        handleStandard();
    }
}
```

- The program could use if conditions to select the required method

Chapter 9 : Switches 9-Nov-12 ©Rob Miles 5

Using a switch Construction


```
switch (selection)
{
    case 1 : handleCasement ();
            break ;
    case 2 : handleStandard () ;
            break ;
    case 3 : handlePatio () ;
            break ;
    default : Console.WriteLine ( "Invalid number" ) ;
            break ;
}
```

- Switch makes it easier to select statements

Chapter 9 : Switches 9-Nov-12 ©Rob Miles 6

The selection value


```
switch (selection)
{
    case 1 : handleCasement ();
             break ;
    case 2 : handleStandard ();
             break ;
    case 3 : handlePatio ();
             break ;
    default : Console.WriteLine ( "Invalid number" );
             break ;
}
```



- This value selects the case to be obeyed

The case keyword


```
switch (selection)
{
    case 1 : handleCasement ();
             break ;
    case 2 : handleStandard ();
             break ;
    case 3 : handlePatio ();
             break ;
    default : Console.WriteLine ( "Invalid number" );
             break ;
}
```



- At run time the matching case is obeyed

The break keyword


```
switch (selection)
{
    case 1 : handleCasement ();
             break ;
    case 2 : handleStandard ();
             break ;
    case 3 : handlePatio ();
             break ;
    default : Console.WriteLine ( "Invalid number" );
             break ;
}
```



- The break keyword marks the end of the selected code

The default keyword

```
switch (selection)
{
    case 1 : handleCasement ();
            break ;
    case 2 : handleStandard () ;
            break ;
    case 3 : handlePatio () ;
            break ;
    default :
        Console.WriteLine ( "Invalid number" ) ;
            break ;
}
```



- We can provide a default behaviour if the selection value doesn't match any cases

Chapter 9 : Switches 9-Nov-12 ©Rob Miles 10

Selecting with strings

```
switch (command)
{
    case "casement" :
        handleCasement ();
        break ;
}
```

- A program can select on strings or characters

Chapter 9 : Switches 9-Nov-12 ©Rob Miles 11

Multiple Case Options

```
switch (command)
{
    case "casement" :
    case "c" :
        handleCasement ();
        break ;
}
```

- This version of the code allows the selection to be made for "c" or "casement"

Chapter 9 : Switches 9-Nov-12 ©Rob Miles 12

Summary

- Switches provide a quick way to select one option from many
- They do not make anything possible we couldn't do before, but they do make it easier
