Program flow

Agenda

Execution flow

If statement

Switch statement

Do while statement

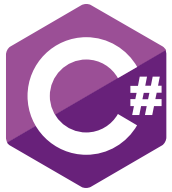
While statement

For statement

Jump statements

Debug code on VS Code

Execution flow



Flow

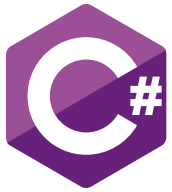
Statements executed from top to bottom

Branching, looping and jumping statements change the execution flow

Demo

Execution flow

if statement



If...else statement

```
if (expression)
{
    // block of code, runs if expression is evaluated to true
}
else
{
    // block of code, runs if expression is evaluated to false
}
```

If...else statement without curly braces

```
if (expression)
     // one line statement runs if expression is evaluated to true
else
     // one line statement runs if expression is evaluated to false
```

Simple If statement

```
if (expression)
{
    //  block of code, runs if expression is evaluated to true
}
```

If...else if statement

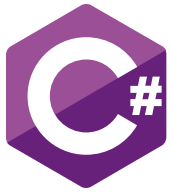
```
if (expression1)
{
    //  block of code, runs if expression1 is evaluated to true
}
else if (expression2)
{
    //  block of code, runs if expression2 is evaluated to true
}
else
{
    //  block of code, runs if expression1 and expression2 are
    //  evaluated to false
}
```

Demo

Create if statements

Use the conditional operator

switch statement



Switch statement

```
switch (code)
{
    case "Code1":
        // Code to call if use case 1
        break;
    case "Code2":
        // Code to call if use case 2
        break;
    case "Code3":
        // Code to call if use case 3
        break;
    default:
        // Code to call if use case 4
        break;
}
```

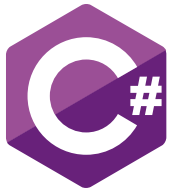
Switch statement

```
int value = 100;
switch (value)
{
    case <= 5:
        Console.WriteLine($"{nameof(value)} is lower or equal to 5");
        break;
    case > 5 and <= 10:
        Console.WriteLine($"{nameof(value)} is greater than 5 and lower or equal to 10");
        break;
    case > 10 and < 100:
        Console.WriteLine($"{nameof(value)} is greater than 10");
        break;
    case 100:
        Console.WriteLine($"{nameof(value)} is equal to 100");
        break;
    case > 100:
        throw new Exception($"The value is not valid : {value}");
```

Demo

Create switch statements

do...while statement



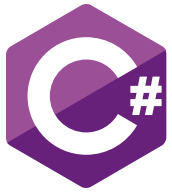
do...while statement

```
int counter = 0;
do
{
    Console.WriteLine(counter);
    counter++;
} while (counter < 100);
```

Demo

Create do while statements

while statement



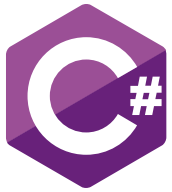
while statement

```
int counter = 0;
while (counter < 100)
{
    Console.WriteLine(counter);
    counter++;
}
```

Demo

Create while statements

for statement



for statement

```
for (var i = 0; i < 10; i++)
{
    Console.WriteLine(i);
}
```

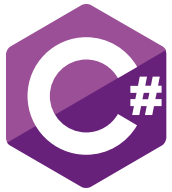
for statement

```
for (var i = 0; i < 10; i = i * 2)
{
    Console.WriteLine(i);
}
```

Demo

Create for statements

Jump statements



Break statement

Terminates an enclosing loop or switch statement

Continue statement

Stops the current iteration and starts the next one in the enclosing loop

Goto statement

Transfers the program control to a labeled statement

Return statement

Ends the execution flow in a method

Can also return a value

Demo

Create jump statements

Demo

Code debug on VS Code

Challenge

Display all the multiples of 7 and 13 between 0 and 1000

Summary

A C# program is executed from top to bottom...

... unless a branching, looping or jump statement is executed

An if statement executes one branch of the multiple branches depending on some condition expressions

A switch statement executes one of the case label blocks

The while statement executes a block of code while a condition expression is true (zero or more)

The do while statement execute a block of code while a condition expression is true (one or more)

A for statement executes a block of code while a condition expression on the loop counter is true

Jump statements are the return, break, continue and goto statements