Lesson 12 - Loops and Logic I

Loop

A `loop` in programming refers to a code block that executes some lines of code repeatedly, so think of 5 lines labelled `2`, `3`, `4`, `5`, `6`. Now we want to execute these 5 lines, 3 times each. so this is how the code will be executed:

```
1, 2, 3, 4, 5, 6, 2, 3, 4, 5, 6, 2, 3, 4, 5, 6, 7, 8, 9 ... and so on.
```

So line 1, then 2-6, again 2-6, and one last time 2-6 then we move on to 7, 8 and more. Each repetition is called an `iteration`, so one iteration in this case would be `2`, `3`, `4`, `5`, `6`.

Now lets take a look at the types of loops and how to implement them.

While loop

The `while` loop has the following basic structure:

```
// some code

while (condition)
{
        // code to be executed repeatedly while condition is true
}

// more code
```

The `while` loop will keep repeating itself as long as the provided `condition` is true, and as soon as the `condition` becomes `false`, the `while` loop will end and move on to execute code outside the `while` block.

Steps

1. Start loop

2. Check condition
3. Execute while block if true or go to step 5 if false
4. Go back to step 2
5. Out of while block.

Example

```
int num = 6;

while (num <= 10)
        Console.WriteLine(num++);
```

Output

```
6
7
8
9
10
```

Example

```
int age = 15;
int year = 2022;

while (age < 18)
{
        Console.Write($"Year: {year}\tAge: {age}");
        Console.WriteLine("\t\tYou are a child, come back later.");
        // 1 year passes
        year++;
        age++;
}

Console.WriteLine($"\nCongratulations! you are an adult in {year}.");
```

Output

```
Year: 2022    Age: 15    You are a child, come back later.
Year: 2023    Age: 16    You are a child, come back later.
Year: 2024    Age: 17    You are a child, come back later.
```

```
Congratulations! you are an adult in 2025.
```

Do while loop

This is similar to a while loop, but it executes atleast once, because the condition is checked *after* the loop block executes. So we can say it is guaranteed to execute atleast once.

The `do - while` loop has the following basic structure:

```
// some code

do
{
        // code to be executed repeatedly
} while (condition);     // note the semicolon at the end

// more code
```

Example

```
int age = 10;
bool condition = false;

do
{
        Console.WriteLine($"You are {age++} years old.");
} while (condition);
```

Output

```
You are 10 years old.
```

Break statement

Previously we used the `break` statement to exit out of a `switch` statement. We can also use breaks in loops to exit out a loop.

Example

```
int age = 10;

while (age > 0)
{
    Console.WriteLine($"You are {age++} years old.");
    if (age > 100)
        break;
}
```

Continue Statement

This is similar to `break`, however it only skips the current iteration and moves on to the next.

Example

```
int num = 0;

while (++num <=10)
{
        if (num == 5)
                continue;

        Console.WriteLine(num);
}
```

Output

```
1
2
3
4
6
7
8
9
10
```

Example

Example of how to print even numbers.

```
int num = 0;

while (++num <= 20)
{
    if (num % 2 != 0)
        continue;

    Console.WriteLine(num);
}
```

Output

```
2
4
6
8
10
12
14
16
18
20
```

Guard Clause

A guard clause is a special type of if condition in while we convert nested if conditions into seperate if conditions, in which the condition is reversed and you exit the current code if the condition becomes true, and if all guard clauses do not exit, then you execute the code from the inner blocks.

Without Guard Clause

```
int age = 20;
double height = 6;
double weight = 65;

if (age > 18)
{
    if (age < 30)
    {
        if (height > 5.5)
        {
```

```
            if (weight > 60)
            {
                if (weight < 80)
                {
                    Console.WriteLine("You pass");
                }
            }
        }
    }
}
```

With Guard Clause

```
int age = 20;
double height = 6;
double weight = 65;

if (age < 18) return;
if (age > 30) return;
if (height < 5.5) return;
if (weight < 60) return;
if (weight > 80) return;

Console.WriteLine("You pass");
```