

C# .NET Loops

What will you be learning?

- For loops
- While loops
- Do loops

What is a loop?

- A loop is a way to execute a piece of code repeatedly
- Go round and round until an end condition is met

Why use loops?

- Eg: you need to add from 0 to 10

Solution not using loop:

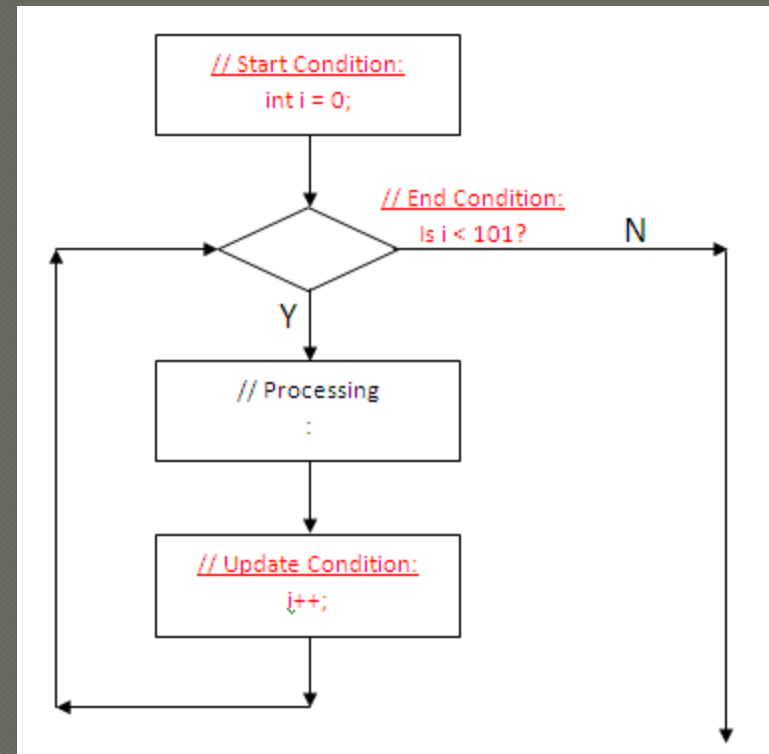
```
int answer;
```

```
answer = 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10;
```

- The above solution not suitable for very large numbers (says from 0 to 10000) – programmers are lazy, what would be a faster way to solve?

Flowchart “for” loop

- Start with $i = 0$
- End when i is not < 101
- Update i by adding 1

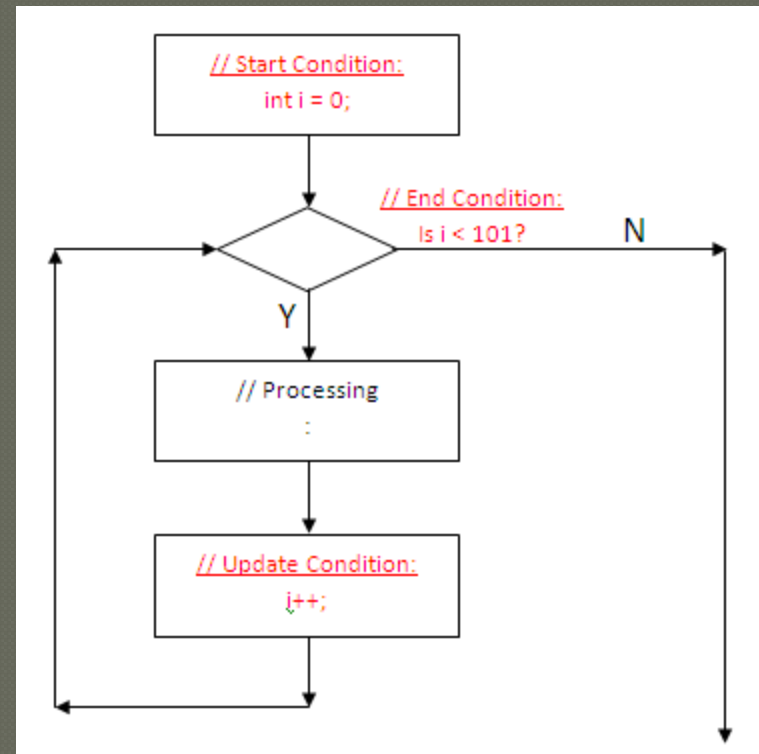


for loop

```
for (    ) // start, end, update conditions
{
    // Processing
}
```

for loop

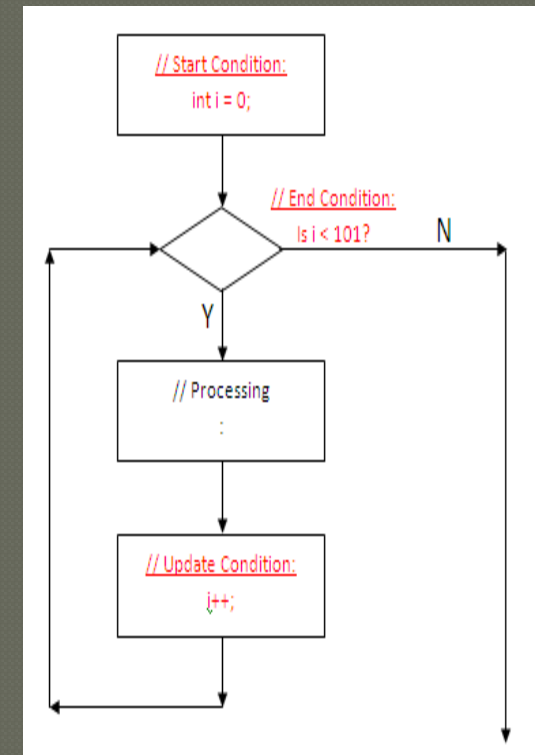
```
for (int i=0; i < 101; i++ )  
{  
    // Processing  
}
```



for (int i=0; i < 101; i++)

Write syntax of the following for loops

- From $j = 10$ until $j = 1000$ (inclusive), update j by 1 for each loop
- From $t = 2$ until $t = 1000$ (inclusive), update t by 2 for each loop
- From $m = 5$ until $m = 0$ (inclusive), update m by -1 for each loop



Exercise Part 1 – For Loops

- Add a new project to “Part 1 For Loops”
- Put one button and add the following codes for its click method (pg 112):

Exercise Part 1 – For Loops

```
private void button1_Click(object sender, EventArgs e)
{
    int answer = 0;

    for (int i = 1; i < 101; i++)
    {
        answer = answer + i;
    }

    MessageBox.Show(answer.ToString());
}
```

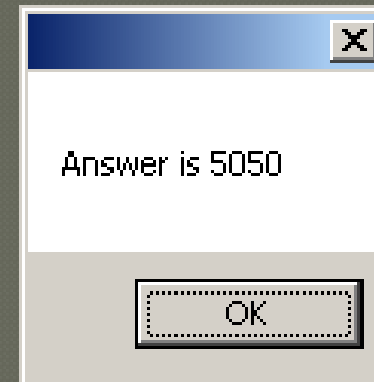
Exercise Part 2

Loop Start Values and Loop End Values

- Add a project “Part 2” to the solution



A screenshot of a Windows application window titled "Form1". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is light beige. It contains two labels, "Loop Start" and "Loop End", each followed by a white text input box. To the right of these inputs is a button labeled "For Loops".



Do it yourself - TimeTable

- Modify the previous project

A screenshot of a Windows application window titled "Form1". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is light beige. It contains three input fields with labels: "Start Number" with the value "1", "End Number" with the value "5", and "Multiply By?" with the value "7". To the right of the "End Number" field is a button labeled "For Loops". Below these inputs is a list box containing five lines of text: "1 times 7 = 7", "2 times 7 = 14", "3 times 7 = 21", "4 times 7 = 28", and "5 times 7 = 35". A yellow arrow points from the list box to the code snippet on the right.

To add item to the listBox:

```
listBox1.Items.Add("xxxx")
```

To clear the listBox:

```
listBox1.Items.Clear();
```

break continue

- Add new project to SpfChapter6 solution: “Extra – break and continue”
- Add a button and add codes into its Click method:

```
for (int x=0; x < 11; x++)  
{  
    if (x==5) break;  
    MessageBox.Show(“x =”+x);  
}
```

break continue

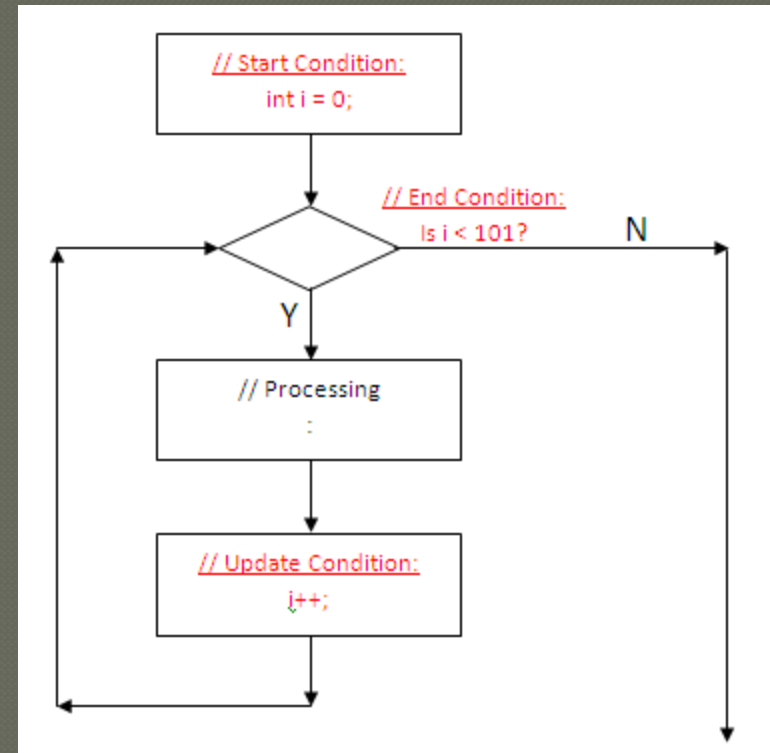
```
for (int x=0; x < 11; x++)  
{  
    if (x==5) continue;  
    MessageBox.Show("x =" + x);  
}
```

while loop

```
while ( ) // End condition  
{  
  
}
```

while loop

```
int i = 0;  
while (i < 101)  
{  
    // Processing  
    :  
    i++; // at the end  
}
```



do loop

```
do  
{
```

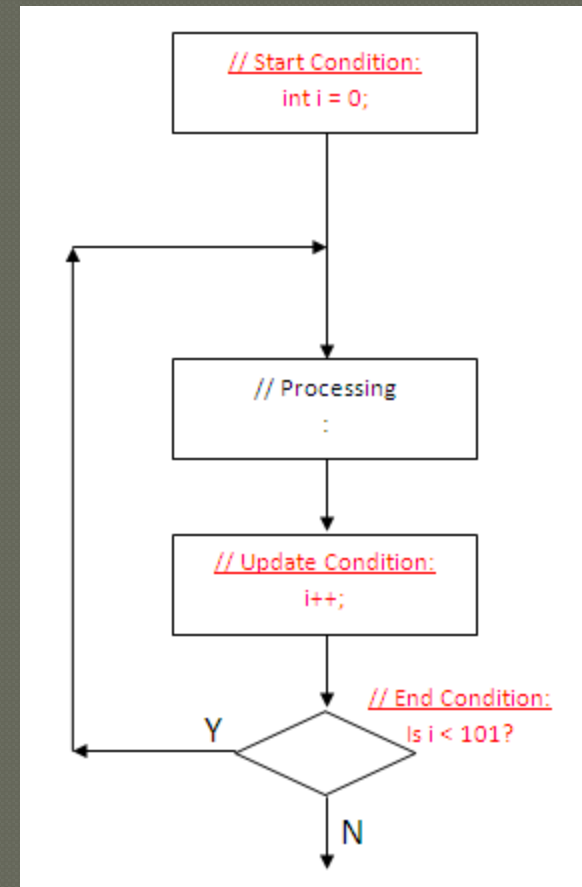
```
} while ( ); // End condition
```



// The difference between the while(){..} and Do {..} while(); loops is that the code in a Do loop will get executed at least once, because its while part is at the end.

do loop

```
int i = 0;  
do  
{  
  // Processing  
  :  
  i++; // at the end  
} while (i < 101);
```



Part 3

- Modify the previous project
- Change from "for" loop to "while" and then to "do while" loop

Summary

- A loop is a way to execute a piece of code repeatedly
- We shall be using loops again when we cover **Arrays**
- There is also "**foreach**" loop that applies for string, array and objects