University of Hull

Using Loops

C# Programming

What we can do so far

- We know there are three flavours of loops

  `do - while` – put the test at the end

  `while` – put the test at the start

  `for` – create and update a control variable
- We also know that we can `continue` (go round again) and `break` (leave the loop)

Pick a Loop

- I want to ask a user for a film number and reject values outside the range of 1-6
- What kind of loop?

Pick a Loop

- I want to ask a user for a film number and reject values outside the range of 1-6
- What kind of loop?
- A do – while loop will work best

Pick a Loop

- I want to ask a user for a film number and reject values outside the range of 1-6
- What kind of loop?
- A do – while loop will work best
- Why?

Pick a Loop

- I want to ask a user for a film number and reject values outside the range of 1-6
- What kind of loop?
- A do – while loop will work best
- Why?
- Because this loop always performs the statements in the loop at least once

Pick another Loop

- I want the user to enter a number, and make my program print out the times table for that number, up to 12
- What kind of loop?

Pick another Loop

- I want the user to enter a number, and make my program print out the times table for that number, up to 12
- What kind of loop?
- A For loop would work best for this

Pick another Loop

- I want the user to enter a number, and make my program print out the times table for that number, up to 12
- What kind of loop?
- A For loop would work best for this
- What would it look like?

Pick another Loop

- I want the user to enter a number, and make my program print out the times table for that number, up to 12
- What kind of loop?
- A For loop would work best for this
- What would it look like?

```
for ( i = 1; i < 13 ; i = i+1)
```

Final Loop

- I want to print out a square of * characters on the screen
- The square should be 40 across and 10 down
- What kind of loop?

Final Loop

- I want to print out a square of * characters on the screen
- The square should be 40 across and 10 down
- What kind of loop?
- This is a tricky one
- We need to put one loop inside another

Nesting Loops

- It is perfectly OK to put one loop inside another
- Many programs do this
- In this case it might help if we rephrase the problem

“Print out 10 lines, each containing 40 characters”

Two control variables

- If we have two loops we need two counters

```
int charNo;
int lineNo;
```

- One counts the lines, the other counts the characters

The Outer Loop

- The Outer Loop repeats for each line

```
for (lineNo = 0; lineNo < 10; lineNo = lineNo + 1)
{
}
```

- We want 10 lines, and so the loop goes round 10 times

The Inner Loop

- The Inner Loop repeats for each character

```
for (charNo = 0; charNo < 40; charNo = charNo + 1)
{
}
```

- We want 40 characters and so the loop goes round 40 times

The Final Program

```
int charNo;
int lineNo;

for (lineNo = 0; lineNo < 10; lineNo = lineNo + 1)
{
    for (charNo = 0; charNo < 40; charNo = charNo + 1)
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
```

- This is the final program

The Final Program

```
int charNo;
int lineNo;

for (lineNo = 0; lineNo < 10; lineNo = lineNo + 1)
{
    for (charNo = 0; charNo < 40; charNo = charNo + 1)
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
```

- Create the counter variables

The Final Program

```
int charNo;
int lineNo;

for (lineNo = 0; lineNo < 10; lineNo = lineNo + 1)
{
    for (charNo = 0; charNo < 40; charNo = charNo + 1)
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
```

- This is the outer loop, it goes round once for each line on the screen

The Final Program

```
int charNo;
int lineNo;

for (lineNo = 0; lineNo < 10; lineNo = lineNo + 1)
{
    for (charNo = 0; charNo < 40; charNo = charNo + 1)
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
```

- This is the inner loop
- It goes round once for each character

The Final Program

```
int charNo;
int lineNo;

for (lineNo = 0; lineNo < 10; lineNo = lineNo + 1)
{
    for (charNo = 0; charNo < 40; charNo = charNo + 1)
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
```

- This writes a single *
- It uses Write, so it doesn’t take a new line

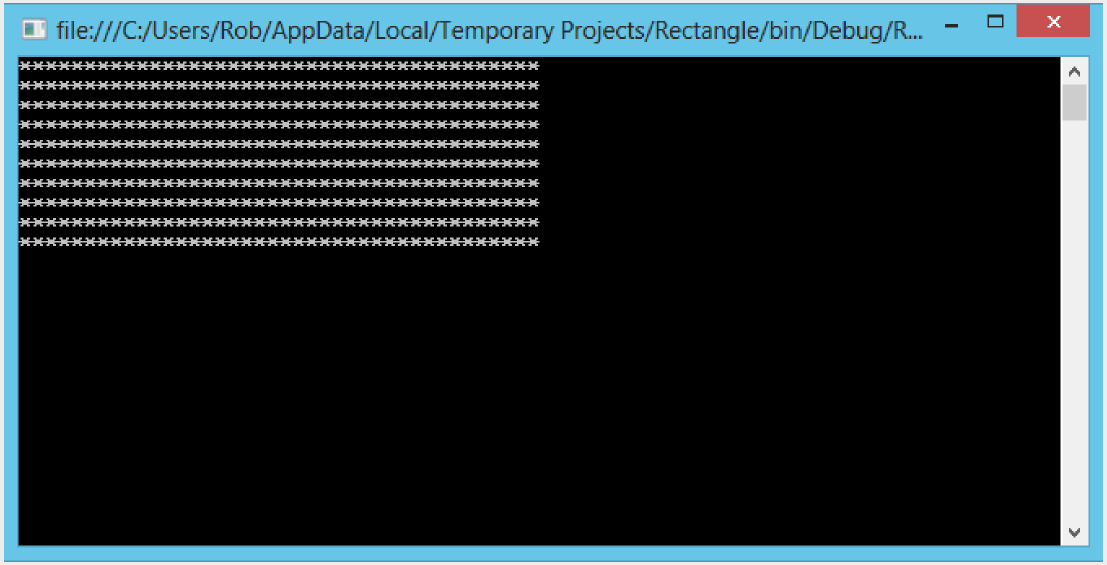
The Final Program

```
int charNo;
int lineNo;

for (lineNo = 0; lineNo < 10; lineNo = lineNo + 1)
{
    for (charNo = 0; charNo < 40; charNo = charNo + 1)
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
```

- Once we have printed the * characters the program must move on to the next line

Final Output



file:///C:/Users/Rob/AppData/Local/Temporary Projects/Rectangle/bin/Debug/R...

```
****************************************
****************************************
****************************************
****************************************
****************************************
****************************************
****************************************
****************************************
****************************************
****************************************
```

- This is what the program produces

Changing the code

- To print more lines we need to change the value 10 to a different one
- To print more characters we need to change the value 40 to a different one

Summary

- You need to pick the loop that is appropriate for the task in hand
  - Decide where the test needs to go
  - Decide if you are repeating something a particular number of times
- You can put one loop inside another