TUTORIAL: LOOP

BME 121, Fall 2016

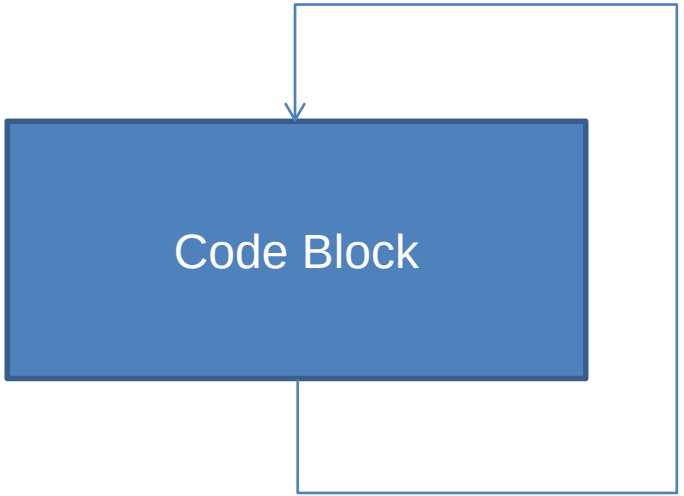
Rasoul Nasiri

Topics

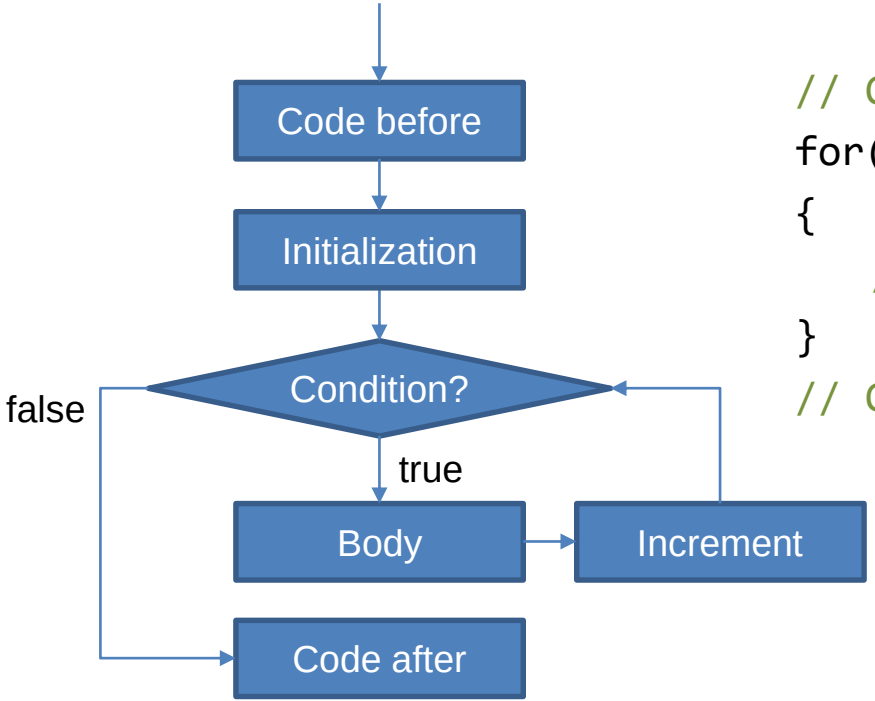
- For loop
- Bart Simpson and iteration
- Loop and drawing geometric shapes
- Nested loop
- Coin flip example
- Rolling a dice
- Some statistic in loop

Programming Loops

- A language feature that allows a programmer to tell the computer to **perform a certain (set of) instruction(s) over and over again**
- Machines are very efficient at repetitive labour!



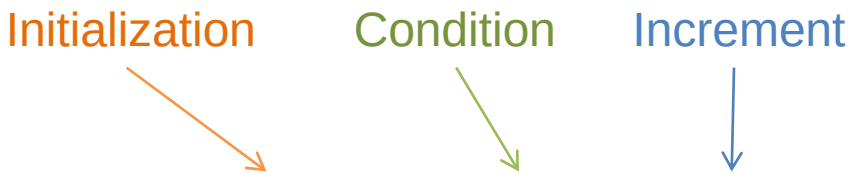
For Loop




```
// Code before for loop
for(initialization; condition; increment)
{
    // Body
}
// Code after for loop
```

For Loop

- Like a While Loop, but incorporates a counter control mechanism

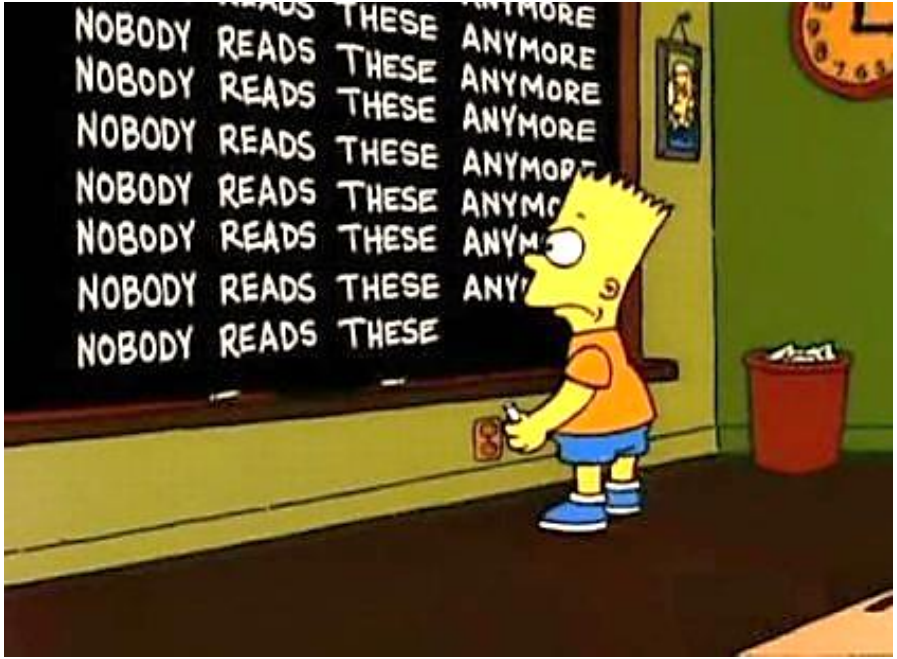



```
for(int n = 0; n < 10; n++)
{
    // Body
    Console.WriteLine(n);
}
```

- “For n = 0 to 9, n increasing 1 at a time, display value of n”

Practice Problem 1 – Bart Simpsons

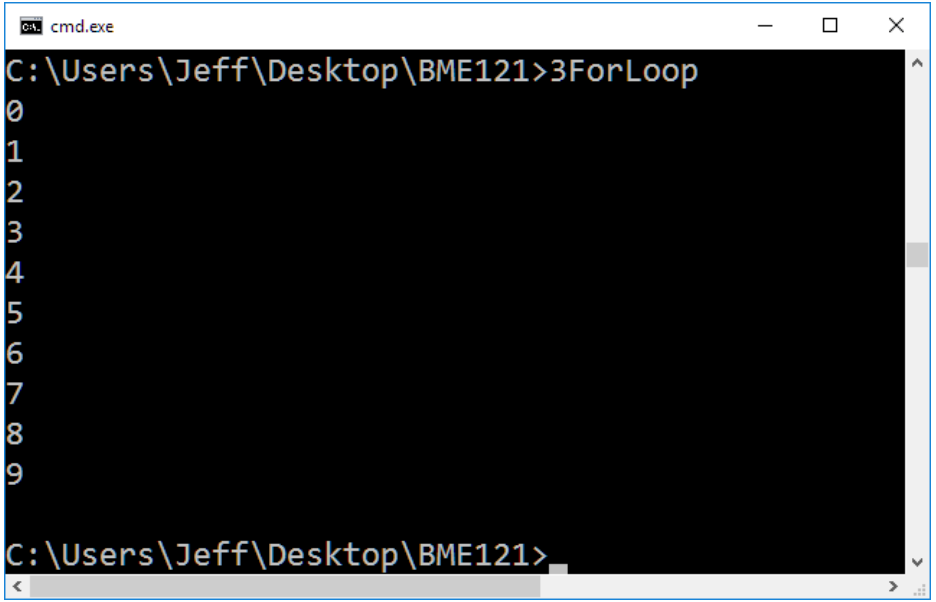
- Make a program which helps Bart Simpsons write this message on the board (console):
  A. 10 times
  B. As many times as Bart wants
- **Hint**: Ask Bart (the user) how many times he wants the message written for B.



Practice For Loop – Example: Count from 0 to 9

```
for(int n = 0; n < 10; n++)
{
    // Body
    Console.WriteLine(n);
}
```

- Also known as a Count-controlled Loop: repeats a specific number of times.

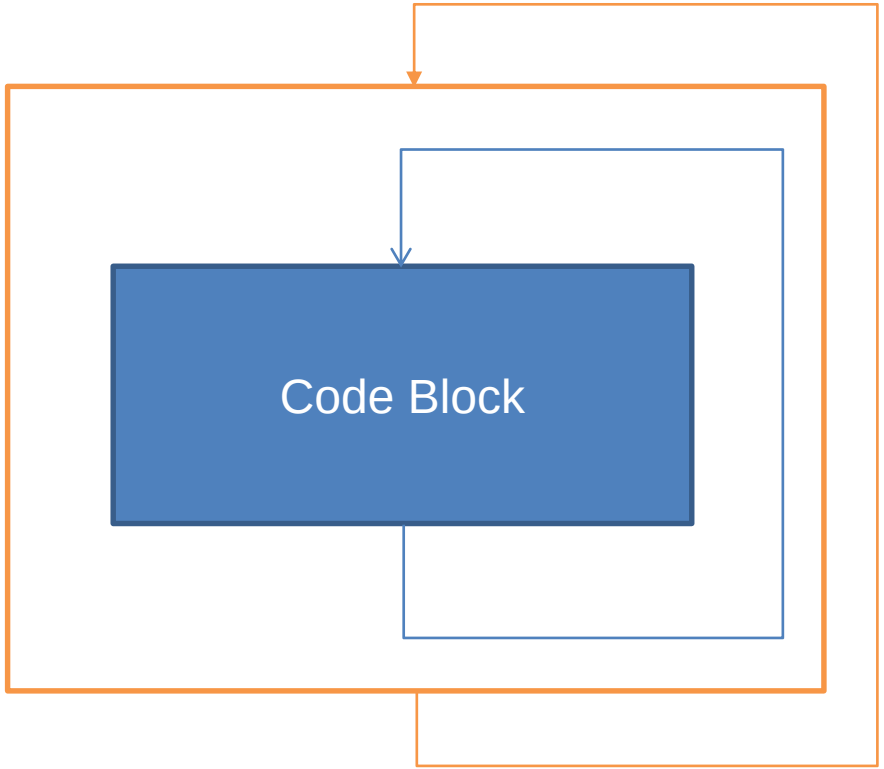


Practice Problem 2 – Count the sheeps

- Using a For loop:
  A. Count from 0 to 99
  B. Count from 99 to 0 by only changing the initialization, condition, and increment from A
  C. Count from 1372 to 1472
  D. Sum the numbers 1 to 100, then display the sum
  E. Multiply the numbers 1 to 10, display the product of the numbers along the way

Nested Loops

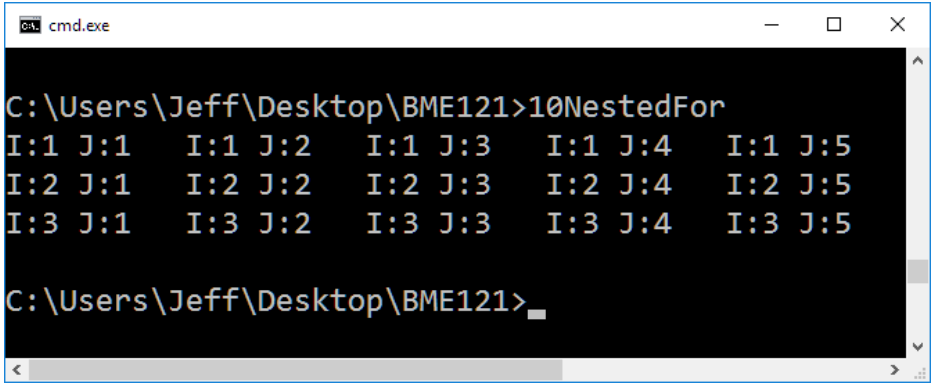
- A loop inside another loop
  - Inner loop is nested within the outer loop
- The entire code body of the outer loop consists of the inner loop, and potentially other code
- Inner loop has to finish repeating (however many times it needs) before outer loop can continue onto its next repetition



Nested For Loops

- More complex diagram, same principle

```
for(int i = 1; i <= 3; i++)
{
    for(int j = 1; j <= 5; j++)
    {
        Console.Write(
            "I:{0} J:{1}   ",
            i, j);
    }
    // Escape each line
    Console.Write("\n");
}
```



Nested For Loops – drawing shape

- Using for loop to draw the rectangular shape using stars

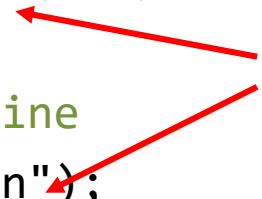
```
for(int i = 1; i <= 3; i++)
{
    Console.WriteLine(“*****");
}
```

What if we want to define width and length?

Nested For Loops – drawing shape

- Using for loop to draw the rectangular shape

```
for(int i = 1; i <= length; i++)
{
    for(int j = 1; j <= width; j++)
    {
        Console.Write(“*");
    }
    // Escape each line
    Console.Write("\n");
}
```

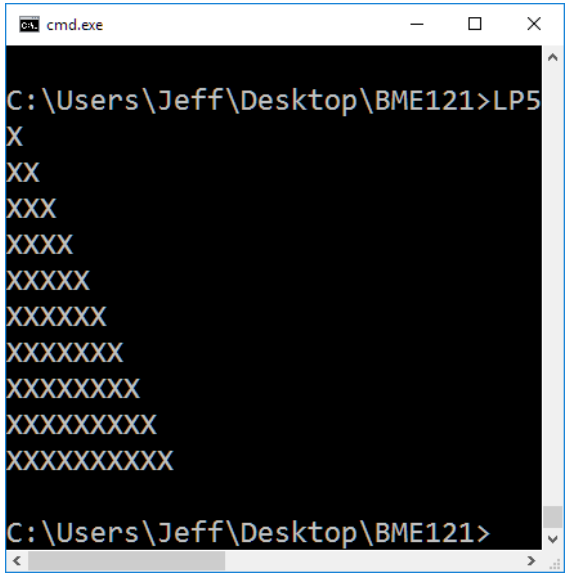


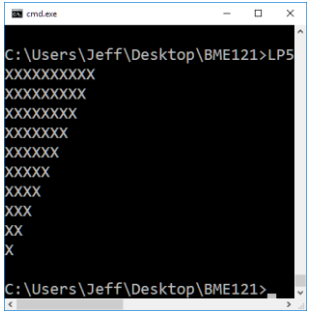
Nested For Loops – drawing shape

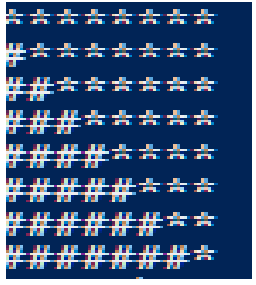
- Using for loop to draw the following triangle shape

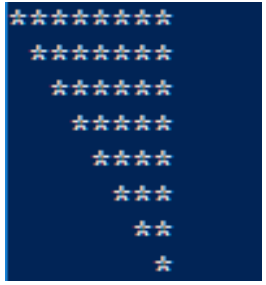
```
for(int i = 1; i <= height; i++)
{
    for(int j = 1; j <= i; j++)
    {
        Console.Write(“*”);
    }
    // Escape each line
    Console.Write("\n");
}
```

What about these rectangles?









For Loop – Flipping a coin

- Flipping a coin has two possible results: head/tail (1/0)
- Simulate flipping a coin with random class in C#

```
random.Next(0,2); //return 0 or 1
```

- Do the flipping100 times and count the heads(success) and tails
- Repeat this as an experiment 8 times
- What is the number of heads in each experiment?
- Average over all 8 experiments?

See https://www.freeonlinedice.com/#coin

1D Arrays

- Arrays are a way to have an organized collection of variables of the same type:

```
// make a 1D array of 10 integers
int[] x = new int[10];
// make a 1D array of 20 doubles
double[] y = new double[20];
```

- first element of an array is always at index 0

```
x[0] = 1; // store value of 1 in the first element of the array
```

- *array*.Length gives us the number of variables in the array

```
Console.Write(x.Length); // shows 10
```

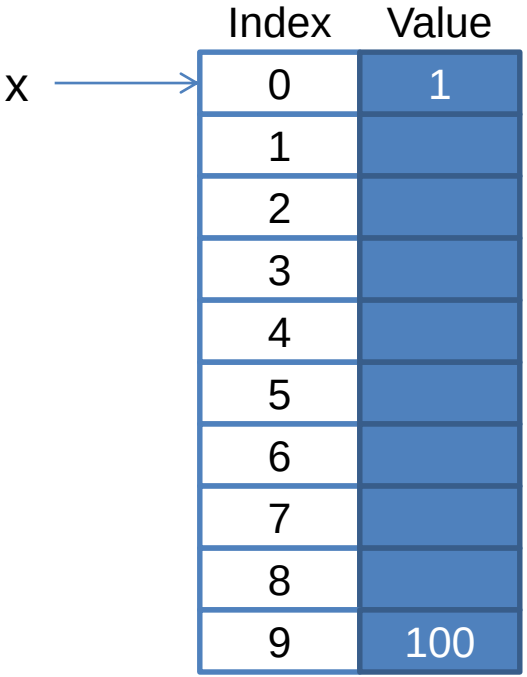
- last element of an array is always index (Length – 1), 9 in this case

```
x[9] = 100;
```

1D Arrays

```
int x = 5;
```


```
int[] x = new int[10];
x[0] = 1;
x[9] = 100;
```



Note:

x refers to the entire array

x[index] refers to one particular element in the array

For Loop – rolling a dice

- Simulate rolling a dice
- The output of rolling could be between 1 and 6
- Let’s roll 100 times and count the number of times you see each number in a 1D array of length 6.
- Calculate the percentage of observations for each number
- What would happen if we increase 100 to 1000, or 1000000?

THE END …