

Technology Introduction Class

Lesson 5

11/11/2017





Class Overview

Roughly 8 weeks. Will adjust based on student needs.

1. Technology Overview – concepts, ingredients, how do they work? (HW)
2. Technology trend – internet, automation, robots, artificial intelligence (SW)
3. Programming introduction – how to write and run a basic program
4. C# fundamentals (I)
5. C# fundamentals (II)
- 6. C# fundamentals (III)**
7. how to solve real problems with programs
8. Project



Today's Agenda

- Review previous homework
- Topics:
 - Foreach loop
 - More on string operations (compare, contains, case sensitive, case insensitive ...)
 - Passing parameters to Functions/Methods

foreach

- foreach loop can do the same thing for loop does, but often simple.

```
static void foreachEx1()
{
    int[] numbers = new int[] { 100000, 9, -1, 2, 3, 11, 22, 999999 };

    //use for loop first
    for (int i = 0; i < numbers.Length; i++)
    {
        Console.WriteLine(numbers[i]);
    }

    //use foreach loop
    foreach (int x in numbers)
    {
        Console.WriteLine(x);
    }
}
```

- For each iteration, the variable x is assigned to the next value in the array

<https://code.sololearn.com/c96B4bx953jj/#cs>



foreach

- ▶ Exercise: print the names in the array students:

```
var students = new string[]{"Tom", "Mike", "Dan", "Jason", "Mike", "Keith", "Curios George"};
```

<https://code.sololearn.com/c96B4bx953jj/#cs>

foreach

- ▶ You can also exit foreach loop with "break"

```
static void foreachEx2()
{
    var students = new string[]{"Tom", "Mike", "Dan", "Jason", "Mike", "Keith", "Curios George"};

    //find if there's any Jason in the names; if found - exit loop
    string tobefound = "Jason";
    foreach (var s in students)
    {
        Console.WriteLine(s);
        if (s == tobefound)
        {
            Console.WriteLine(tobefound + " is found");
            break;
        }
    }
}
```

<https://code.sololearn.com/c96B4bx953jj/#cs>

foreach

- ▶ Exercise: Find how many "Mike" in the array

```
var students = new string[]{"Tom", "Mike", "Dan", "Jason", "Mike", "Keith", "Curios George"};

//count how many "Mike" in the array
string tobefound = "Mike";
int numberOfMike = 0;
foreach (...)
{
    ...
}
Console.WriteLine(numberofMike);
```

<https://code.sololearn.com/c96B4bx953jj/#cs>

For loop is still better in some cases

- Example: get the difference between elements

```
var fb = new int[]{0, 1, 1, 2, 3, 5, 8, 13, 21, 34};
for (int i=0; i<fb.Length-1; i++)
{
    var diff = fb[i+1] - fb[i];
    Console.WriteLine(diff);
}
```

Exercise:

How do you print out sum of every 2 numbers, like {1, 3, 5, 21, 55}?

<https://code.sololearn.com/c96B4bx953jj/#cs>

String operations

- Numbers and text are two most common data type a program need to process
- To process text, we need to use some common string functions
- String comparison

```
string x = "Mike";  
if ( x == "mike")  
{  
    Console.WriteLine("equal");  
}  
else  
{  
    Console.WriteLine("not equal");  
}
```

NOT Good

Look at this function - String.Compare(String, String, Boolean)

[https://msdn.microsoft.com/en-us/library/zkcaxw5y\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/zkcaxw5y(v=vs.110).aspx)

<https://code.sololearn.com/c7AaAzamDyZD/#cs>

String operations

String.Compare(String, String, true) is a case-insensitive compare
String.Compare(String, String, false) is a case-sensitive compare

String.Contains(string) - it has no case-insensitive version;
use Indexof instead -

```
IndexOf(string, StringComparison.OrdinalIgnoreCase) >= 0
```

Other useful functions:

Substring, Trim, Replace, Remove, Split, StartsWith, EndsWith

[https://msdn.microsoft.com/en-us/library/system.string_methods\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.string_methods(v=vs.110).aspx)

<https://code.sololearn.com/c7AaAzamDyZD/#cs>

Passing Parameters to Function/Method

- A function (or method) can have no parameter, 1 or more parameters, or a variable number of parameters
- Functions can take different types of parameters – called function overload

```
static void Main(string[] args)
{
    methodA();
    methodB(5);
    methodB("nice");
}
static void methodA()
{
    Console.WriteLine("I have nothing");
}
static void methodB(int x)
{
    int y = x*x;
    Console.WriteLine(x + "'s square is " + y);
}
static void methodB(string x)
{
    Console.WriteLine(x + "'s square is -- I don't know, because it's a string");
}
```

<https://code.sololearn.com/c7dW8f71TW77/#cs>

Passing Parameters to Function/Method

- Functions can also return nothing – (void), or return different data types
- Function can call other functions; see examples in the link
- Use “Function Prototype” to figure out how to use a function

```
//functions can call other functions
static void functionDemo(int x)
{
    int sum = 0;
    for (int i=0; i<x; i++)
    {
        sum = sum + squared(i);
    }
    Console.WriteLine(sum);
}

static int squared(int x)
{
    int y = x*x;
    return y;
}
```

<https://code.sololearn.com/c7dW8f71TW77/#cs>



Homework

- ▶ Check codrlab.com for homework #5