**KIT506 Programming Practical 5: Repeat actions foreach item in a list**

[ [↶ Close all sections](https://mylo.utas.edu.au/content/enforced/214790-AW_EAI_17S2_15468_0_0_0_1_1/kit506_pracs/KIT506_Prac_05.html?d2lSessionVal=dx0bynATNq1YR9mA2AHARswOX&ou=214790&d2l_body_type=3) ] Only visible sections will be included when printing this document.

**Aims:**

* to practise reading **while** loops;
* to play with Lists of strings; and
* to practise using C#’s **foreach**...in loop construct.

**∨ 1 Reading exercises (< 10 minutes)**

**1.1 A little counting**

What is the value of a after the following loop has executed?

int a = 1;

**while** (a < 10)

{

a = a + 1;

} // a = 10

[**Solution**](javascript:void(0);)

10, as when a becomes 10 the loop condition becomes false and the loop ends.

**1.2 Disappearing cat**

What is the output from executing the following C# statements? (Predict first, then run them in a program to find out.) As a hint, the Substring(i) method of strings creates a substring starting at position i that continues to the end of the string.

string s = "cat";

**while** (s.Length > 0)

{

Console.WriteLine(s);

s = s.Substring(1);

}

// cat

at

t

[**Solution**](javascript:void(0);)

First identify the stopping condition: when the string s is empty. Next, identify the statement inside the loop that *updates*the value of s: the Substring() method call.

Given how Substring() works we can then work out how s is changed each time through the loop. Then you’ll see why the output is:

[**Solution**](javascript:void(0);)

cat

at

t

**∨ 2 For each item in a List**

Before continuing, create a new Console Application to hold the code you will write.

In addition to (fixed-sized) arrays, C# includes a wide variety of collection types which will grow as necessary when new items are added. We won’t be exploring that feature today (you’ll get to do that soon enough in the main tutorials), but instead declaring and initialising a List with some values and then iterating over them.

Each collection type needs to be told what data type its elements will have (a programming language feature called *generics*in C# and Java, and *templates* in C++). So, to *declare* a List of strings, you can write:

List<string> myStringList;

and to *instantiate* it (create a new empty List object):

myStringList = **new** List<string>();

which is how you’ll create many of them in the assignment, since you’ll then fill them from a database in a separate step.

C# also supports List (and other collection) *initialisers*, which reduces the code required for you to initialise a non-empty List. Add the following code to your program, replacing the animal names with any of your choice:

List<string> animals = **new** List<string> { "elephant", "cat", "mouse", "snake", "giraffe" };

Now let’s display its contents. Lists support the same index notation as arrays, so we *could* write a **for**-loop to iterate over the indices in this List (0 through 4) and display each value in turn:

*//Note that collections like Lists have Count instead of an array's Length property*

**for** (int i = 0; i < animals.Count; i++)

{

Console.WriteLine(animals[i]);

}

But, if you know you’re going to do something with every element and you don’t need to know its position, you can use a foreach loop instead. It has the general form:

**foreach** (*element\_type* *loop\_variable* **in** *collection*)  
{  
   *body*  
}

where *element\_type* matches the type name inside <> when the List was declared, *loop\_variable* is a the name of the variable that will hold the current element’s value each time through the loop, and *collection* is the List, array or other iterable data structure.

**Task:** Try to write a **foreach** to print the values in the animals List.

[**Solution**](javascript:void(0);)

List<string> animals = **new** List<string> { "replace", "these", "values", "with", "your", "own" };

**foreach** (string animal **in** animals)

{

Console.WriteLine(animal);

}

A lot simpler, isn’t it?

**∨ 3 Next week: final test + loop debugging**

At the start of next week’s practical class you’ll do the final 20-minute online quiz on programming skills. After that you’ll debug loop code involving **for**, **while** and **foreach**.