HNDIT1012 Visual Application Programming

Week 6

Array

- Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.
- To declare an array, define the variable type with **square brackets**:

Eg: string[] cars;

int[] marks;

Array …

To insert values to it, we can use an array literal - place the values in a comma-separated list, inside curly braces:

```
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

To create an array of integers, you could write:

```
int[] myNum = {10, 20, 30, 40};
```

Access the Elements of an Array

You access an array element by referring to the index number.

This statement assign the value of the first element in cars to the string variable mycar:

```
string mycar=cars[0];
```

Array Length

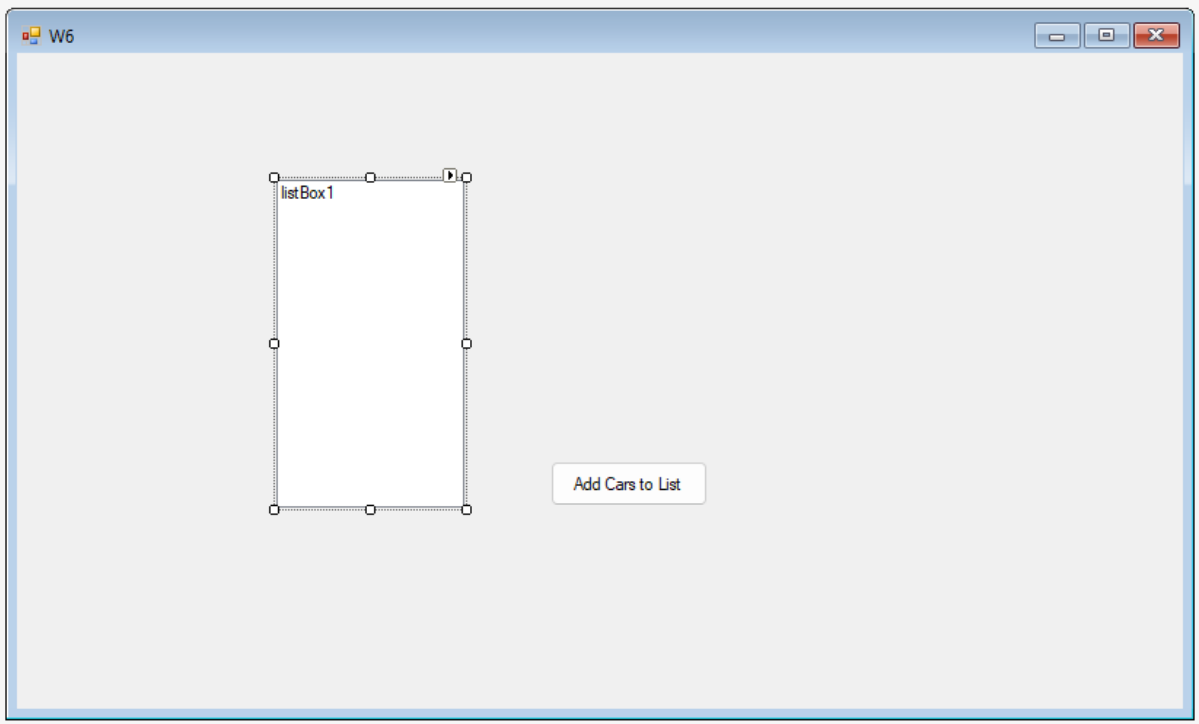
To find out how many elements an array has, use the Length property:

Eg:

```
int x=cars.Length;
```

The following example add all elements in the cars array to the listbox:

- Create a C# solution and design the form as shown below:



Add following code to Button Click event

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string[] cars= { "Volvo", "BMW", "Ford", "Mazda" };
    for (int i = 0; i < cars.Length; i++)
        listBox1.Items.Add(cars[i]);

}
```

Above code will add the elements of car array to the listbox items.
The method Add(item) will add an item to the Items collection of the listbox1.

Same example using foreach loop

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string[] cars= { "Volvo", "BMW", "Ford", "Mazda" };
    foreach(string car in cars)
        listBox1.Items.Add(car);


}
```

For single-dimensional arrays, the foreach statement processes elements in increasing index order, starting with index 0 and ending with index Length - 1:

Sorting an Array

There are many array methods available, for example Sort(), which sorts an array alphabetically or in an ascending order:

Eg:

```
// Sort a string
string[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
Array.Sort(cars);

// Sort an int
int[] myNumbers = {5, 1, 8, 9};
Array.Sort(myNumbers);
```

Reverse(Array)

Reverses the sequence of the elements in the entire one-dimensional Array.

Reverse(Array, Int32, Int32)
Reverses the sequence of a subset of the elements in the one-dimensional Array.

Parameters

Array
The one-dimensional Array to reverse.

index
Int32
The starting index of the section to reverse.

length
Int32
The number of elements in the section to reverse.

Example

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    string[] cars = { "Volvo", "BMW", "Toyota", "Mazda", "Suzuki" };
    Array.Sort(cars);
    Array.Reverse(cars,0,2);
    comboBox1.Items.AddRange(cars);

}
```

AddRange() method add an array to Items collection of a ListBox or ComboBox.

In the above example, the Reverse() method reverse the first 2 elements of the array. To reverse all elements, change the statement as given below:

Array.Reverse(cars,0,cars.Length);

Here the second parameter 0 indicates beginning of the array and second parameter is the total number of elements to be reversed.

Thank You