VARIABLE ARRAYS

- Say everyone in grade 12 (300 students) wrote skills test and your principal wanted you to write a program that would store the names and test scores for every student that took the test. The program should rank (put in order) the scores and computer the average score. The code to do this is not hard. The problem here is how do we declare all the variables we need? To write this test score program, you would need 300 **string** variables to store student names and 300 **int** variables to store student scores. We are required to declare every variable we use. Do you really want to write 600 lines of code.....

- An array is a way to store large number of variables under the same name. Each variable in an array, called an element, must have the same data type, and they are distinguished from each other by an array **index**.

- A variable array is declared in a way similar to other variables. To indicate the variable is an array, you use two square brackets (**[ ]**) after the type. Square brackets are used a lot with arrays. At the same time you declare an array,you create it using the **new** keyword.

- The number in brackets is called the array **dimension**.

```
string[] stuNames = new string[5];
```

- Each variable in an array is referred to by its declared name and index. The first student name in the array would be **student[0]** and the last name would be **student[4]**, not student[5]. If you try to refer to student[5], you will get a run-time error saying an array value is out of bounds. This is a common mistake! Always be aware arrays are zero-based.

- For example, an array containing **five** String elements could be:

0	1	2	3	4
Jamie	Tim	Ben	Matt	Justin
stuNames				

- The elements of an array have an index value, with the first element being index 0, the second being index 1, and so on. In the example above, Ben is the 3rd element of the array and has an index of 2.

- The example below has nine elements, ranging from item[0] to item[8].

```
int[] item = new int[9];
```

- It is also possible to have arrays of controls. For example, to have 20 button types available to use. The utility of such a declaration will become apparent at a later date.

```
Button[] myButtons = new Button[20];
```

- You can even initialize arrays with this technique. You can let Visual C# Express figure out the dimension by counting the number of elements used to initialize the array. An example is:

```
int[] item = new int[] {0,1,2,3,4,5,6,7,8,9};
```

- C# will know that this array item has 10 elements.

USING ARRAYS

- As an example of using an array, to assign information to the student with index of 3 (actually, the 4th student in the array because of the zero base), we could write two lines of code like this:

```
stuNames[3] = "Billy Bob";
stuScore[3] = 87;
```

- Array variables can be used anywhere regular variables are used. They can be used on the left side of assignment statements or in expressions. To add up the first three test scores, you would write:

```
sum = score[0] + score[1] + score[2];
```

- Arrays are objects of the Array class that includes many properties and methods. One property is the Length, which returns the number of elements which can come in handy at times.

```
lblStudentNames.Test = stuNames.Length.ToString();
```

USING A FOR LOOP

- The **For Loop** is very useful when working with large numbers of similar variables.

- In the school score example at the top, we need to go through 300 scores to compute an average. C# offers a convenient way to do counting: the for loop.

- Notice I use the variable name studentNumber instead of the generic **i** in this instance. (giving it a name in this instance just makes it easier to read as i would have worked just as well).

```
scoreSum = 0;

for (int studentNumber = 0; studentNumber < 300; studentNumber++)

{
    scoreSum = scoreSum + score[StudentNumber];
}
average = scoreSum / 300;
```