Arrays

There are times when it is convenient to store a number of data items in memory to be used (and reused) in the computations of a form. The data items may be input from a file or they may be created by a program.

One-dimensional Arrays

One-dimensional arrays give a variable name to a number of data elements. We number the data elements from 0 to some maximum value; each of these numbers is an index, sometimes called a subscript, of the array. Each data element must have the same type.

Array variables are used to store and access data elements one at a time. Individual elements are referred to by their subscript, or index, number.

Declaration of a one-dimensional array:

```
Dim <arrayname> ( 0 to <maxnumber> ) as <type>
```

The maximum number is the last valid subscript number of the array. Since the subscript numbers start at 0, the number of elements is <maxnumber> + 1.
Examples:

```
Dim students ( 0 to 24 ) as String    (25 elements in the array students)
Dim prices ( 0 to 50 ) as Single
Dim scores ( 0 to 99 ) as Integer
```

Setting and using elements of an array:

Individual data elements are referred to by putting a valid subscript number in parentheses:

The third element of the prices array:

```
prices ( 3 )
```

This can be used in expressions or in assignment like a variable.

Assigning to the tenth element of the scores array:

```
scores ( 10 ) = 59
```

You may need to get the values of an array from a file, using the input command. Suppose that you have a file with prices that has been opened with number 1:

```
Dim n as Integer = 0
While Not EOF(1)
        Input ( 1, prices (n))
        n = n + 1
End While
```

This will read each number from the file and store it into the prices array.

A standard For loop to use each element of an array would have the form:

```
Dim students (0 to 24) as String
For index = 0 to 24
    ‘ do whatever processing needs to be done to each element here
    labelstring = labelstring & students ( index ) & “ “
Next
```

You should not try to subscript an array element that has not been defined (this is also true for regular variables) and you should not try to use subscript numbers that are beyond the declaration of the maximum number.

Two-dimensional Arrays (and beyond)

For each additional dimension, add a subscript range to the declaration and use an additional subscript number to access elements.

Declaration of a two-dimensional array:

The dimensions of a two-dimensional array can be thought of as the row and column numbers of a matrix.

```
Dim <arrayname> ( 0 to <maxrownumber>, 0 to <maxcolnumber> ) as <type>
```

As with one-dimensional arrays, the maximum number is the last valid subscript number of the array.
Example:

```
Dim prices ( 0 to 24, 0 to 9 ) as Single
```

This is an array with 25 rows and 10 columns.

Setting and using elements of 2-D arrays:

Individual data elements are referred to by putting a two valid subscript numbers in parentheses:

The element in the row numbered 9 and the column numbered 3 would be:

```
prices ( 9, 3 )
```

Additional dimensions would add additional subscript numbers within the parentheses and separated by commas.

Accessing each element of a two-dimensional array can be achieved by a double loop, for example:

```
For row = 0 to 24
    For col = 0 to 9
        ‘ do processing of array element here
        prices ( row, col ) =  prices ( row, col ) + tax
    Next
Next
```