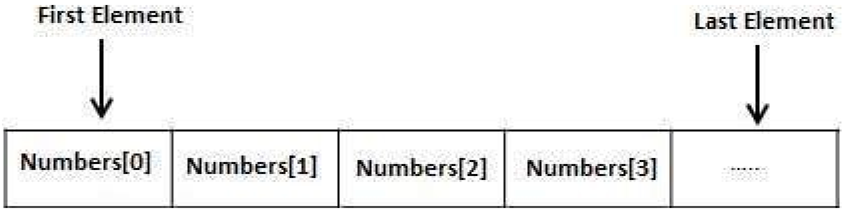
Self-Learning Module 13(Week 14)

I. Topic: Arrays and Loops.

V. Key Point

An array stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.



VI. Core Content

By definition, an array is a variable with a single name that represents many different items. When we work with a single item, we only need to use one variable. However, if we have a list of items which are of similar type to deal with, we need to declare an array of variables instead of using a variable for each item. For example, if we need to enter one hundred names, it is difficult to declare 100 different names. Besides, if we want to process those data that involves decision making, we might have to use hundreds of if...then statements, this is a waste of time and efforts. So, instead of declaring one hundred different variables, we need to declare only one array. So far you've been using variables quite a lot. You've put numbers into variables, and you've put text into variables. But you've only done this one at a time: you've put one number into a variable, or one string of text. You've been doing this:

```
Dim MyNumber As Integer
MyNumber = 5
```

Or this

```
Dim MyText As String
MyText = "A String is really just text"
```

Or even this:

```
Dim MyNumber As Integer = 5
```

So one variable was holding one piece of information. An array is a variable that can hold more than one piece of information at a time. The MyNumber variable above held one number 5. If you had an array variable called MyNumbers - plural - you could hold more than one number at a time. You set them up like this:

```
Dim MyNumbers(4) As Integer
MyNumbers(0) = 1
MyNumbers(1) = 2
MyNumbers(2) = 3
MyNumbers(3) = 4
MyNumbers(4) = 5
```

When you set up an array with the Dim word, you put the name of your array variable, and tell Visual Basic how many items you want to store in the array. But you need to use parentheses around your figure. You then assign your data to a position in the array. In the example above we've set up an Integer array with 5 items in it. We've then said put number 1 into array position 0, put number 2 into array position 1, put number 3 into array position 2, and so on.

You might be thinking that the array was set to the number 4 - MyNumbers(4) - but always remember that an array starts counting at zero, and the first position in your array will be zero.

So that's what an array is - a variable that can hold more than one piece of data at a time -but how do they work? An example below will display the output to make the lesson clear;

Example:

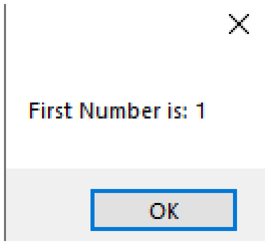
Step 1: Start a new VB project.

Step 2: Add a Button to your Form.

Step 3: Set the Text property of the Button to "Integer Array"

Step 4: Put the following code behind your button:

```
Dim MyNumbers(4) As Integer
MyNumbers(0) = 1
MyNumbers(1) = 2
MyNumbers(2) = 3
MyNumbers(3) = 4
MyNumbers(4) = 5
MessageBox.Show("First Number is: " & MyNumbers(0))
MessageBox.Show("Second Number is: " & MyNumbers(1))
MessageBox.Show("Third Number is: " & MyNumbers(2))
MessageBox.Show("Fourth Number is: " & MyNumbers(3))
MessageBox.Show("Fifth Number is: " & MyNumbers(4))
```



First Number is: 1

The image above displays the output based on the code above,

In the code, we first set up an Integer array with 5 items in it.

Dim MyNumbers(4) As Integer

We then assigned values to each position in the array.

MyNumbers(0) = 1

To get at the values in the array, and display them in messages boxes, we just used the array name, followed by the position in the array.

MessageBox.Show("First Number is: " & MyNumbers(0))

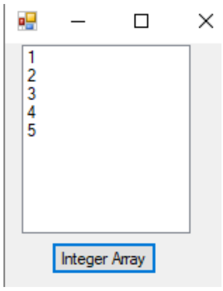
So we've said, "Display whatever number is in array position 0, then display whatever number is in array position 1 ... " and so on.

Now here comes the part where array plays an important role in a loop depending on what the program is all about. So basically, from the above example, the numbers from the array will be displayed using message boxes, which can be annoying. On the next example, there will be modifications on the codes and a loop will be inserted so that those numbers will appear on a single list box.

```
Public Class Form1
   Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
      Dim MyNumbers(4) As Integer
      MyNumbers(0) = 1
      MyNumbers(1) = 2
      MyNumbers(2) = 3
      MyNumbers(3) = 4
      MyNumbers(4) = 5

      For i = 0 To 4
         ListBox1.Items.Add(MyNumbers(i))
      Next i
   End Sub
End Class
```

The output will be like this;



```
1
2
3
4
5
```

From our previous, example, we can also write this codes as follows;

```
Public Class Form1
   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
      Dim MyNumbers() As Integer = {1, 2, 3, 4, 5}
      For i = 0 To 4
         ListBox1.Items.Add(MyNumbers(i))
      Next i
   End Sub
End Class
```

Activity 1

Put two textboxes on your form. The first box asks users to enter a start position for a For Loop; the second textbox asks user to enter an end position for the *For loop*. When a button is clicked, the program will add up the numbers between the start position and the end position. Display the answer in a message box. You can use this For Loop code