Arrays

An array is a sequence of variables that have the same name but use index numbers to differentiate them from one another.

Arrays are useful when there are many similar bits of information that need to be stored.

Imagine a class of 10 students that have a test mark that needs to be recorded.
One way to do this is to declare 10 variables.

```
Dim mark1 as Integer
Dim mark2 as Integer
Dim mark3 as Integer
Dim mark4 as Integer
Dim mark5 as Integer
Dim mark6 as Integer
Dim mark7 as Integer
Dim mark8 as Integer
Dim mark9 as Integer
Dim mark10 as Integer
```

Now each variable will need to be assigned a value, in this case a mark.

```
mark1 = 85
mark2 = 65
mark3 = 56
mark4 = 78
mark5 = 45
mark6 = 72
mark7 = 91
mark8 = 63
mark9 =  77
mark10 = 65
```

This can get very tedious even with just 10 variables.
Imagine a hundred or even thousands.

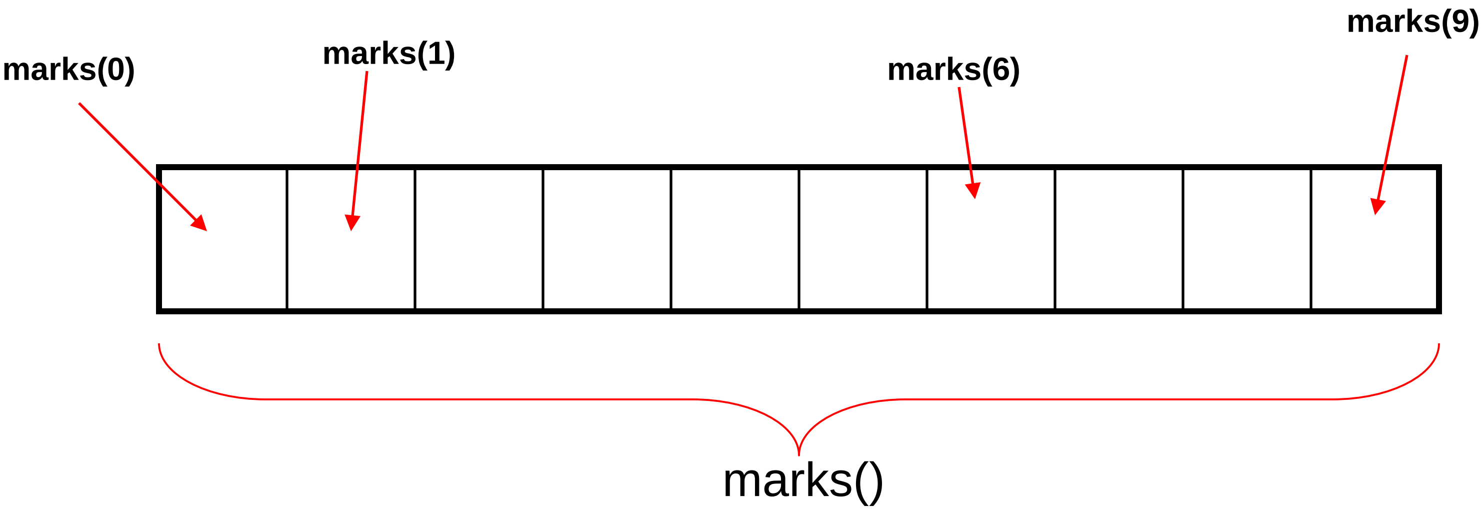
Arrays to the Rescue

Lets take our 10 variables from the previous slide and replace them with an array.
An array that could hold the marks for 10 students might be declared as follows.

Dim marks(10) As Integer

We describe this as an integer array containing 10 elements. Each element in the array will hold an integer value. The elements are numbered with index numbers beginning with 0. That is to say the first element of the array has an index of 0.

Think of an array as a series of mailbox slots. Each mailbox has its own index number and each box can hold a value.
A 10 element array called marks might be visualized as follows.



Since the elements are numbered beginning with a zero, the last element of a 10 element array is marks(9)

Assigning Values To Each Element In An Array

Because the array was declared as an integer array, the value assigned to each element in the array must be of an integer type. Values can be assigned as follows.

```
marks(0) = 85
marks(1) = 65
marks(2) = 56
marks(3) = 78
marks(4) = 45
marks(5) = 72
marks(6) = 91
marks(7) = 63
marks(8) = 77
marks(9) = 65
```

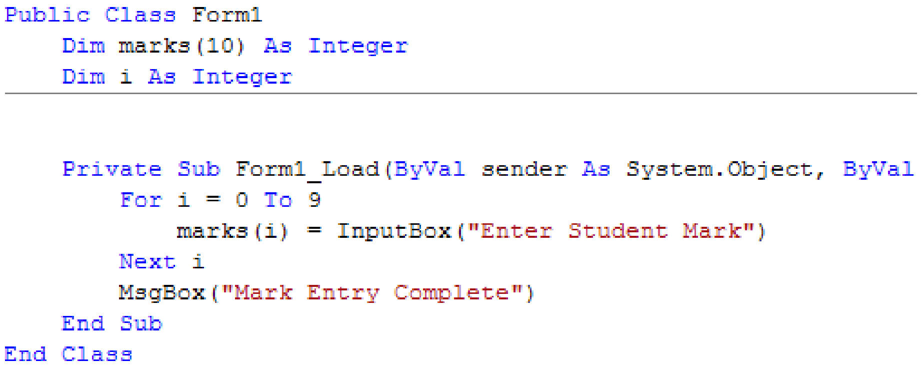
```
marks(0) =  76.4
```

error!!

Using Variables to Represent Index Numbers

10 lines of code to assign 10 values is not particularly convenient. There is an easier way.
The real power of arrays occurs when we substitute variables instead of literals to represent the index numbers. Doing this, we can substitute a *For Loop for the 10 lines we used earlier.*

Example



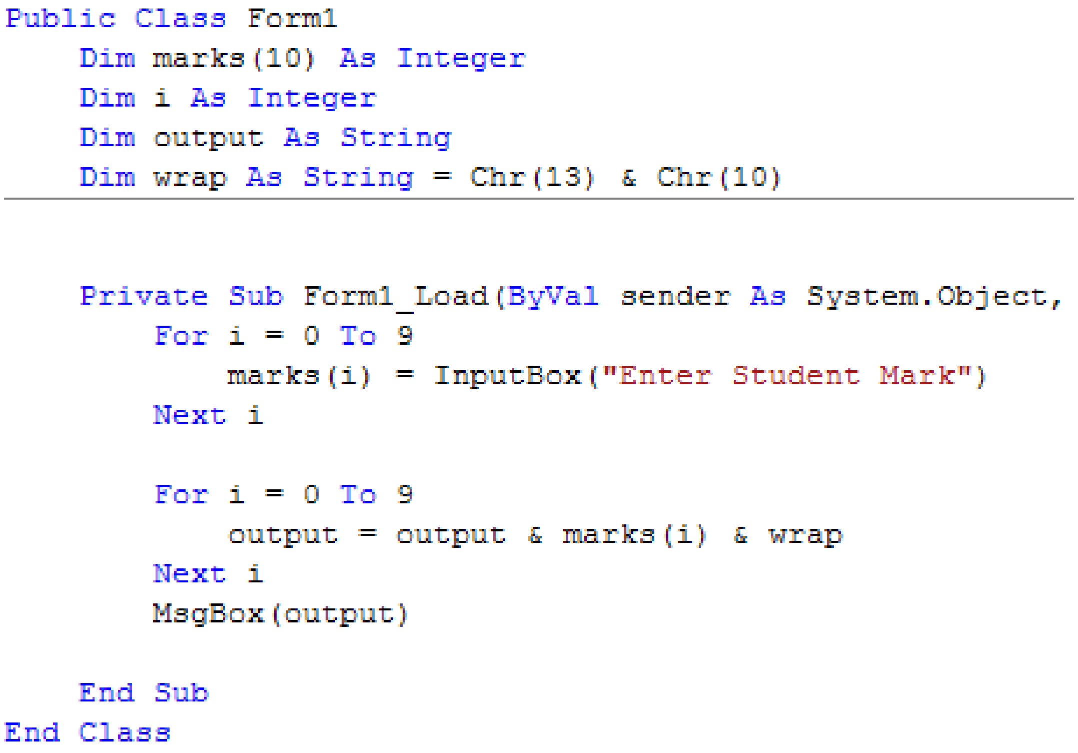
```
Public Class Form1
    Dim marks(10) As Integer
    Dim i As Integer

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal
        For i = 0 To 9
            marks(i) = InputBox("Enter Student Mark")
        Next i
        MsgBox("Mark Entry Complete")
    End Sub
End Class
```

With one simple *for loop* we have coded the same results as 10 lines in the previous example!

Displaying Information

Once we have the values in the array, it is just as simple to display the information using variables as indexes combined with a *for loop*.

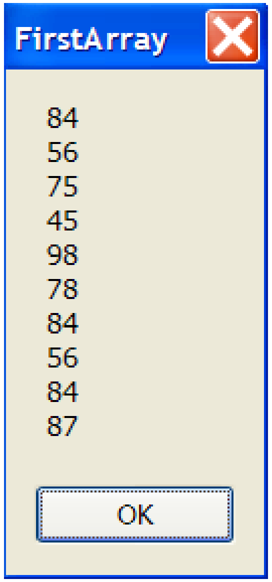


```
Public Class Form1
    Dim marks(10) As Integer
    Dim i As Integer
    Dim output As String
    Dim wrap As String = Chr(13) & Chr(10)


    Private Sub Form1_Load(ByVal sender As System.Object,
        For i = 0 To 9
            marks(i) = InputBox("Enter Student Mark")
        Next i

        For i = 0 To 9
            output = output & marks(i) & wrap
        Next i
        MsgBox(output)

    End Sub
End Class
```



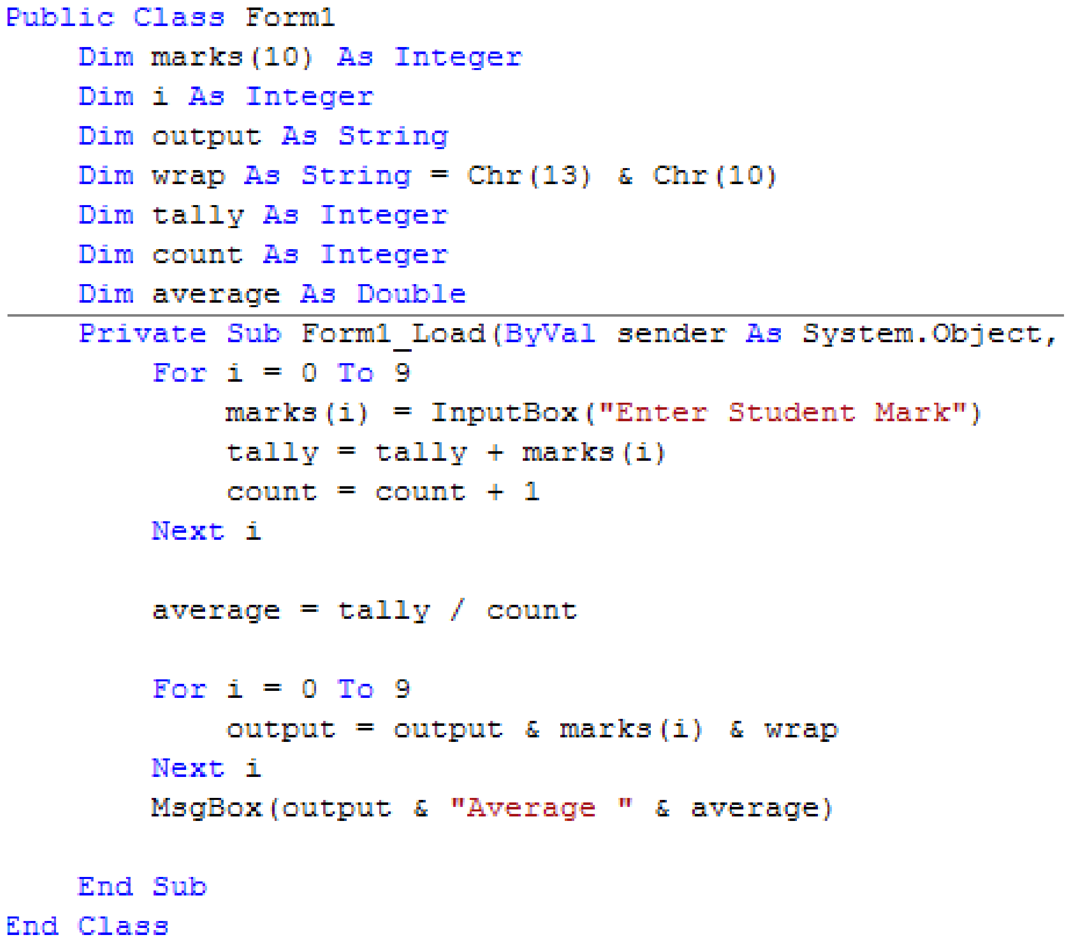
```
84
56
75
45
98
78
84
56
84
87
```

Determining Averages

With the addition of a few lines of code the for loop in the previous slide can also tally the marks as they are entered. Then, by dividing the tally by the number of students a class average can be determined.

After the loop is completed the number of students is equal to i + 1.

Alternatively, you could use a separate variable which we will call count which will track the number of marks entered.

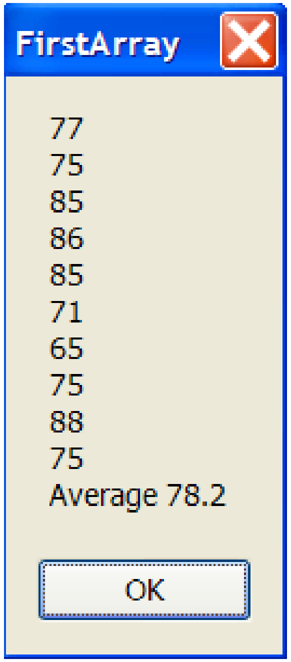


```
Public Class Form1
    Dim marks(10) As Integer
    Dim i As Integer
    Dim output As String
    Dim wrap As String = Chr(13) & Chr(10)
    Dim tally As Integer
    Dim count As Integer
    Dim average As Double
    Private Sub Form1_Load(ByVal sender As System.Object,
        For i = 0 To 9
            marks(i) = InputBox("Enter Student Mark")
            tally = tally + marks(i)
            count = count + 1
        Next i

        average = tally / count

        For i = 0 To 9
            output = output & marks(i) & wrap
        Next i
        MsgBox(output & "Average " & average)

    End Sub
End Class
```



```
77
75
85
86
85
71
65
75
88
75
Average 78.2
```