

8.3

Sub Procedures and Functions That Work With Arrays

You May Pass Arrays As Arguments to Sub Procedures and Functions

You May Also Return an Array From a Function

This Allows You to Write Sub Procedures and Functions That Perform General Operations With Arrays





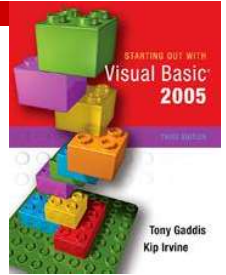
Passing Arrays as Arguments

- Array *numbers* passed to *DisplaySum* sub

```
Dim numbers() As Integer = { 2, 4, 7, 9, 8, 12, 10 }  
DisplaySum(numbers)
```

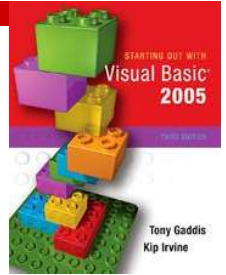
- Sub computes/shows sum of array elements
- Can pass any integer array to *DisplaySum*

```
Sub DisplaySum(ByVal intArray() As Integer)  
    Dim total As Integer = 0           ' Accumulator  
    Dim count As Integer               ' Loop counter  
    For count = 0 To (intArray.Length - 1)  
        total += intArray(count)  
    Next  
    MessageBox.Show("Total is " & total.ToString)  
End Sub
```



Passing Arrays: ByVal and ByRef

- When passing an array ByVal
 - Calling procedure “sees” sub procedure changes to element values
 - Simple variables don’t work this way
 - If sub assigns array argument to another array, no effect on array in calling procedure
- When passing an array ByRef
 - Calling procedure “sees” sub procedure changes to element values
 - If sub assigns array argument to another array, calling procedure array values affected

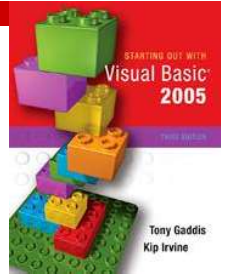


Passing Arrays Example

- After *ResetValues* procedure executes, *numbers* array still contains { 1, 2, 3, 4, 5 }
- If array passed ByRef, *numbers* array will contain { 0, 0, 0, 0, 0 } after procedure runs

```
Dim numbers() As Integer = { 1, 2, 3, 4, 5 }  
ResetValues(numbers)
```

```
Sub ResetValues(ByVal intArray() As Integer)  
Dim newArray() As Integer = {0, 0, 0, 0, 0}  
intArray = newArray  
End Sub
```



An Array Returned From a Function

- Return type `String()` indicates an array of strings
- Thus the function result must be assigned to an array of strings

```
Function GetNames() As String() ←
    ' Get four names from user
    ' Return them as an array of strings.
    Dim strNames(3) As String
    Dim strInput As String
    Dim count As Integer
    For count = 0 To 3
        strInput = InputBox("Enter name " & _
            (count + 1).ToString)
        strNames(count) = strInput
    Next
    Return strNames
End Function
```