

Arrays

*CE 311 K - Introduction to Computer
Methods*

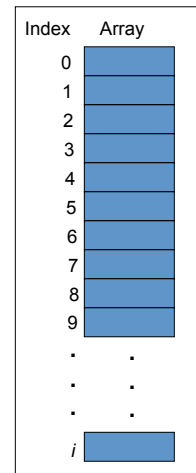
Daene C. McKinney

Introduction

- Arrays
- Two-Dimensional Arrays
- String Arrays & Splitting

Arrays

- A group of related things
- Sequentially indexed data structure (matrix)
- All elements in an array have same data type
- An element of an array is accessed using the array name and an index, e.g., $A(i)$

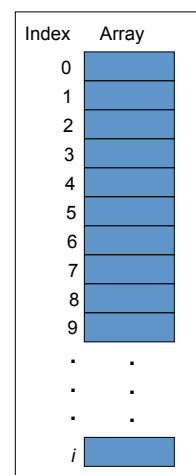


Array Definition and Initialization

- An array is defined using a declaration statement.

`Dim arrayName(0 To end) As dataType`

- allocates memory for “end” elements
- Index of first element is 0
 - `arrayName(0)` is the first element
 - `arrayName(end)` is the last element



Example

```

Private Sub Button1_Click(ByVal sender As System.Object,
    Dim A(0 To 4) As Double

    For i As Long = 0 To 4
        A(i) = i * 2
        ListBox1.Items.Add(" i = " & i & vbTab & _
            " A(" & i & ") = " & A(i))
    Next
End Sub

```

Array w/ 5 elements

Assign array values

Flow Example - Revisited

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim flow(0 To 100) As Double
    Dim sr As IO.StreamReader = IO.File.OpenText("C:\temp\flow.txt")

    sum = 0
    count = 0
    Do While sr.Peek <> -1
        flow(count) = sr.ReadLine
        sum = sum + flow(count)
        count = count + 1
    Loop
    sr.Close()
    average = sum / count
    sum = 0
    For i As Long = 0 To count - 1
        sum = sum + (flow(i) - average) ^ 2
    Next
    stdev = Math.Sqrt(sum / (count - 1))
    ListBox1.Items.Add("Average = " & Math.Round(average, 0))
    ListBox1.Items.Add("Std. Dev. = " & Math.Round(stdev, 0))
End Sub

```

Array w/ 101 elements

Assign array values

Perform calculations using array

Need the array values again for calculations

Flow Example – Revised Again

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim flow(0 To 100), count, average, stdev As Double
    Dim sr As IO.StreamReader = IO.File.OpenText("C:\temp\flow.txt")

    count = 0
    Do While sr.Peek <> -1
        flow(count) = sr.ReadLine
        count = count + 1
    Loop
    sr.Close()

    average = Compute_Ave(flow, count)
    stdev = Compute_Stdev(flow, count, average)

    ListBox1.Items.Add("Average = " & Math.Round(average, 0))
    ListBox1.Items.Add("Std. Dev. = " & Math.Round(stdev, 0))
End Sub

```

Pass Array to function
"average" (see next slide)

Pass Array to function
"stdev" (see next slide)

Flow Example - Revised

Function to
compute
average

```

Function Compute_Ave(ByVal flow() As Double, _
                    ByVal count As Integer) As Double
    Dim sum As Long = 0
    For i As Long = 0 To count
        sum = sum + flow(i)
    Next
    Return sum / count
End Function

```

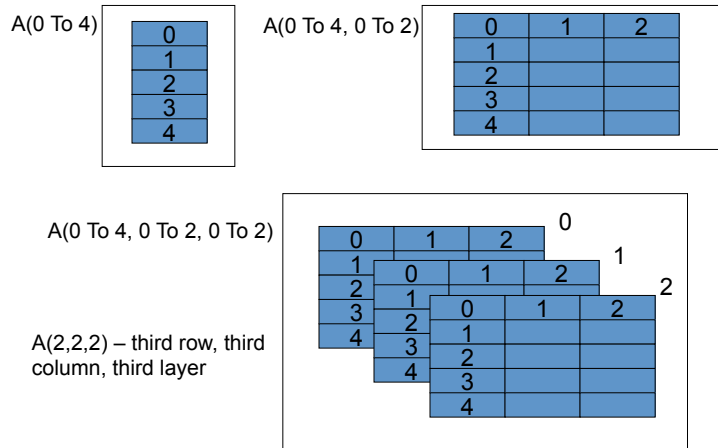
Function to
compute
standard
deviation

```

Function Compute_Stdev(ByVal flow() As Double, _
                      ByVal count As Long, _
                      ByVal average As Double) As Double
    Dim sum As Long = 0
    For i As Long = 0 To count - 1
        sum = sum + (flow(i) - average) ^ 2
    Next
    Return Math.Sqrt(sum / (count - 1))
End Function

```

Multi-Dimensional Arrays



Example – 2D Arrays

- Distances between Texas cities in km

	Austin	Houston	El Paso	Dallas
Austin	0	237	844	283
Houston	237	0	1081	362
El Paso	844	1081	0	916
Dallas	283	362	916	0

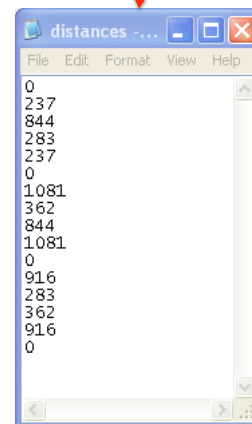
2-D Array to Store Contents of Table

Array
 Dim mile(0 To 3, 0 To 3) As Double

	Austin	Houston	El Paso	Dallas
Austin	0	237	844	283
Houston	237	0	1081	362
El Paso	844	1081	0	916
Dallas	283	362	916	0

Matrix containing data

Text file containing data



Example

Declare array

Load data into array before hitting the button

Use data in array after entering origin and destination and hitting the button

```
Public Class Form1
    Dim mile(0 To 3, 0 To 3) As Double

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Dim sr As IO.StreamReader = IO.File.OpenText("C:\temp\distances.txt")
        For row As Integer = 0 To 3
            For col As Integer = 0 To 3
                mile(row, col) = CDb1(sr.ReadLine)
            Next
        Next
        sr.Close()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Dim row, col As Integer
        row = CInt(TextBox1.Text)
        col = CInt(TextBox2.Text)
        If (row >= 1 And row <= 4) And (col >= 1 And col <= 4) Then
            ListBox1.Items.Add("The distance is " & mile(row - 1, col - 1) & " km ")
        Else
            MsgBox("Origin and Destination must be numbers from 1 to 4", , "Error")
        End If
    End Sub
End Class
```

Flow Duration Curve

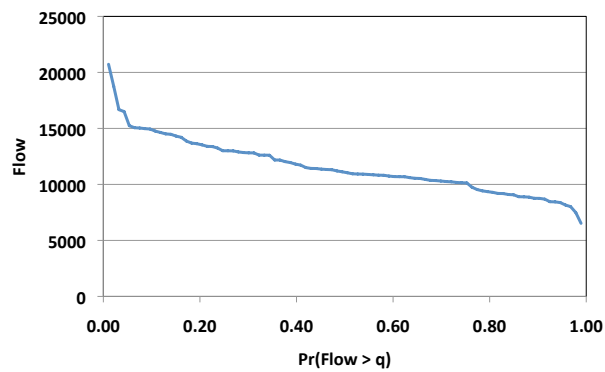
Exceedence
Probability

$$\Pr\{Flow \leq q\} = \frac{i}{n+1}$$

$$\Pr\{Flow > q\} = 1 - \frac{i}{n+1}$$

Unsorted Flows	Sorted Flows	Rank	P{Flow < q}	P{Flow > q}
Million m3	Million m3			
10817	6525	1	0.01	0.99
11126	7478	2	0.02	0.98
11503	8014	3	0.03	0.97
...
12615	18754	91	0.98	0.02
16675	20725	92	0.99	0.01

Flow Duration Curve



String Arrays & Splitting

```

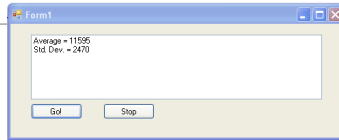
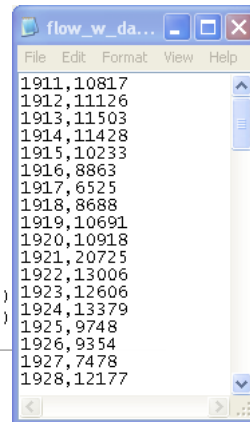
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.E
    Dim flow(0 To 100), average, stdev As Double
    Dim count As Long
    Dim sr As IO.StreamReader = IO.File.OpenText("C:\temp\flow_w_date.csv")
    Dim line, field(0 To 1) As String

    count = 0
    Do While sr.Peek <> -1
        line = sr.ReadLine
        field = line.Split(",")
        flow(count) = field(1)
        count = count + 1
    Loop
    sr.Close()

    average = Compute_Ave(flow, count)
    stdev = Compute_Stdev(flow, count, average)

    ListBox1.Items.Add("Average = " & Math.Round(average, 0))
    ListBox1.Items.Add("Std. Dev. = " & Math.Round(stdev, 0))
End Sub

```



Summary

- Arrays
- Two-Dimensional Arrays
- String Arrays & Splitting