

# Visual Web Development

Terry Marris October 2008

## 8 Iterations - For ... Each

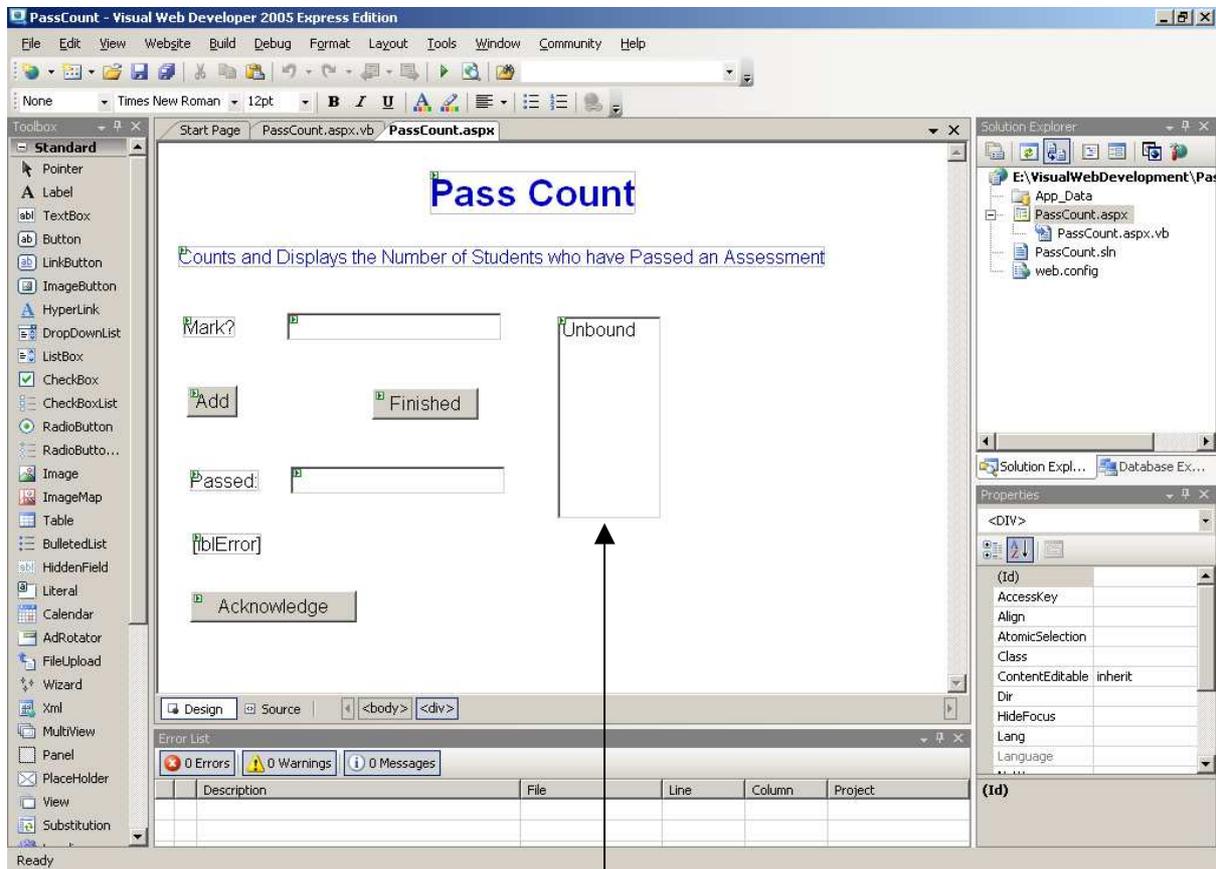
Iterations, or loops as they are commonly known, allow a single code fragment to be repeatedly executed. We look at the For Each ... Next loop construct.

### 8.1 Counting

We count the number of students who have passed an assessment.

<b>Use Case</b>	PassCount
<b>Goal</b>	to count and display the number of students who have passed an assessment
<b>Pre-condition</b>	integers representing marks between 0 and 100 inclusive are entered
<b>Post-condition</b>	the number of students who have passed is displayed
<b>Initiating Actor</b>	the user
<b>Main Success Scenario</b>	
1	user enters marks until finished
2	system calculates and displays the number who have passed
3	exit success
<b>Exceptions</b>	
1a	non-numeric input
1a1	system displays error message
1a2	resume 1
1b	mark < 0
1b1	system displays error message
1b2	resume 1
1c	mark > 100
1c1	system displays error message
1c2	resume 1

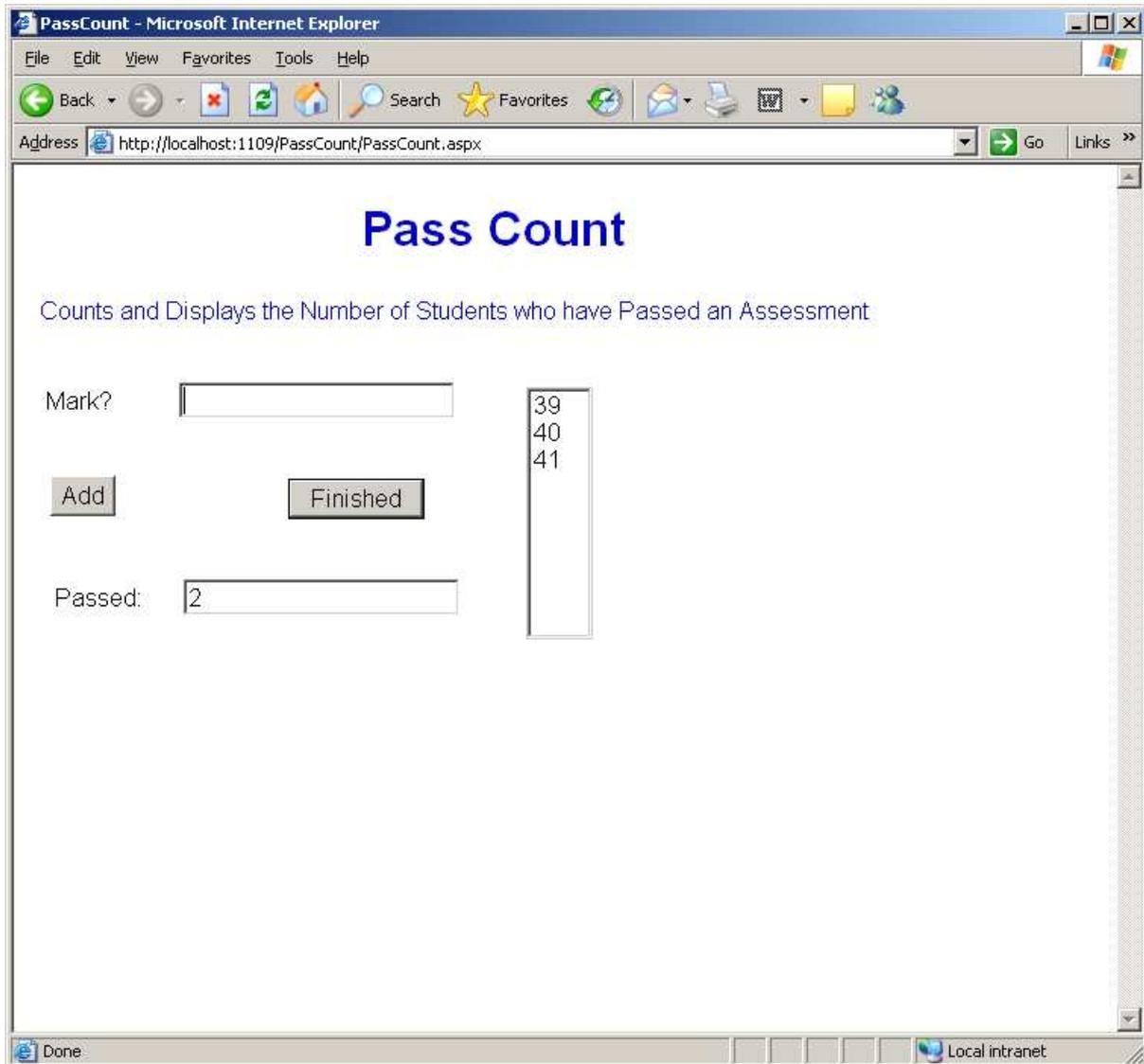
## 8.2 The Interface



ListBox

When you drag a ListBox onto your form, a ListBox Task Window appears: ignore it. Just drag the box to where you want it.

## 8.3 Program Run



On clicking Add the mark entered by the user is placed in the list box.

On clicking Finished the number of students who have passed is shown. Here, the programmer has set the pass mark to 40.

## 8.4 For Each ... Next

We have a list of marks: 39, 40, 41. We examine the marks one-by-one: if a mark is more than (or equal to) the pass-mark, we add 1 to a counter. If the pass-mark is 40 then two students have passed.

```
Dim intMark As Integer = 0
Dim intCount As Integer = 0
For Each item As ListItem In lstMarks.Items
    intMark = Convert.ToInt32(item.Value)
    If intMark >= intPassMark Then
        intCount = intCount + 1
    End If
Next
txtPassed.Text = intCount.ToString()
```

<p>Initialise count</p> <p>For each item in the list:</p> <ul style="list-style-type: none"><li>• convert it to an integer</li><li>• if it <math>\geq</math> pass mark then<ul style="list-style-type: none"><li>○ add 1 to count</li></ul></li></ul> <p>EndFor</p> <p>Display count</p>
--

For Each ... looks at each item in a collection in turn.

A ListBox is an example of a collection. A collection contains zero or more ListItems. We are obliged to convert ListItems into the appropriate type if we want to process them.

Next marks the end of For Each ....

Statements between For Each and Next are executed zero, once or many times depending on the number of items in the collection.

## 8.5 The VB Code

```
Partial Class _Default
    Inherits System.Web.UI.Page

    ' GLOBAL CONSTANTS
    Const intPassMark As Integer = 40

    Sub ProcessError(ByVal errorMessage As String)
        lstMarks.Items.Clear()
        lblError.Visible = True
        lblError.Text = "Error: " + errorMessage
        btnAcknowledge.Visible = True
    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        lblError.Visible = False
        btnAcknowledge.Visible = False
        txtMark.Focus()
    End Sub

    Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
        Try
            Dim intMark As Integer = Convert.ToInt32(txtMark.Text)
            If intMark < 0 Then
                ProcessError("Mark is less than zero")
            ElseIf intMark > 100 Then
                ProcessError("Mark is greater than 100")
            Else
                lstMarks.Items.Add(New ListItem(intMark))
                txtMark.Text = ""
                txtMark.Focus()
                Add the mark to the list as a new list
                item
            End If
        Catch ex As FormatException
            ProcessError("Mark is not a whole number")
        Catch ex As OverflowException
            ProcessError("Mark is too large")
        End Try
    End Sub

    Protected Sub btnFinished_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnFinished.Click
        Dim intMark As Integer = 0
        Dim intCount As Integer = 0
        For Each item As ListItem In lstMarks.Items
            intMark = Convert.ToInt32(item.Value)
            If intMark >= intPassMark Then
                intCount = intCount + 1
            End If
        Next
        txtPassed.Text = intCount.ToString()
    End Sub
```

Pass mark is declared here. Its scope is the entire program.

```
Protected Sub btnAcknowledge_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnAcknowledge.Click
    lblError.Text = ""
    lblError.Visible = False
    btnAcknowledge.Visible = False
    txtMark.Text = ""
    txtMark.Focus()
End Sub
End Class
```

## 8.6 Exercises

1. Try out the program, to count the number of students who have passed an assessment, shown above.
2. Design, write and test a program that will sum a collection of numbers.
3. Design, write and test a program that will output the sum and average of a collection of numbers. Ensure that your program remains in control if no numbers are entered.
4. Design, write and test a program that will output the range of a collection of numbers (where range is defined to be the largest number minus the smallest number plus 1).
5. Design, write and test a program that will output the sum of the squares of a collection of numbers.

## 8.7 Conclusion

We saw how to use For Each ... loops to process each item in a collection. Next we look at For ... Next loops.