

# REPETITION

Robert Burtch  
Surveying Engineering Department  
Ferris State University

## DO LOOPS

- Used to repeat sequence of statements a number of times
- Each pass, statements act on variables whose values are changing
- Do Loop
  - Repeats statements as long or until a certain condition is true
  - Keywords: Do, Loop, While, Until

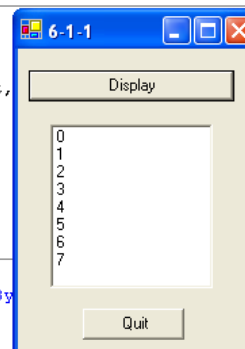
# DO LOOP

- Form
  - Do While *condition*  
statement(s)  
Loop
- VB.NET first checks condition
  - If false, statements inside loop not executed and program continues with line after Loop statement
  - If true, statements inside executed
    - When Loop statement encountered, entire process repeated including testing the condition

# EXAMPLE DO LOOP

Windows Form Designer generated code

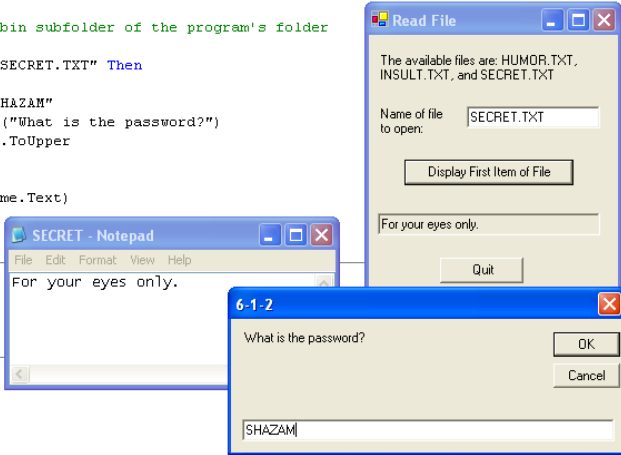
```
Private Sub btnDisplay_Click(ByVal sender As System.Object,  
    'Display the numbers from 1 to 7  
    Dim num As Integer  
    Do While num <= 7  
        lstNumbers.Items.Add(num)  
        num += 1      'Add 1 to the value of num  
    Loop  
End Sub  
  
Private Sub btnQuit_Click(ByVal sender As System.Object, By  
    End  
End Sub
```



# EXAMPLE DO LOOP

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDi
    Dim password As String
    'Assume the file is in the bin subfolder of the program's folder
    Dim sr As IO.StreamReader
    If txtName.Text.ToUpper = "SECRET.TXT" Then
        password = ""
        Do While password <> "SHAZAM"
            password = InputBox("What is the password?")
            password = password.ToUpper
        Loop
    End If
    sr = IO.File.OpenText(txtName.Text)
    txtItem.Text = sr.ReadLine
    sr.Close()
End Sub

Private Sub btnQuit_Click(ByVal
End
End Sub
Class
```



The screenshot shows three overlapping windows. The 'Read File' dialog box is in the background, listing available files: HUMOR.TXT, INSULT.TXT, and SECRET.TXT. The 'SECRET - Notepad' window is in the middle ground, displaying the text 'For your eyes only.'. The '6-1-2' dialog box is in the foreground, asking 'What is the password?' and showing the input 'SHAZAM'.

# DO LOOP

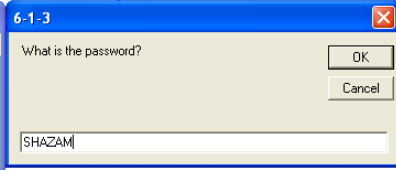
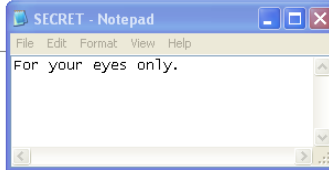
- Alternative form is to check condition at bottom of the loop
- Form:

```
Do
    statement(s)
Loop Until condition
```
- Statements executed inside loop and then checks truth value of condition
  - If true, program continues program
  - If false, entire process within loop executed again.

# EXAMPLE DO LOOP

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnD
    Dim passWord As String
    'Assume the file is in the bin subfolder of the program's folder
    Dim sr As IO.StreamReader
    If txtName.Text.ToUpper = "SECRET.TXT" Then
        Do
            passWord = InputBox("What is the password?")
            passWord = passWord.ToUpper
            Loop Until passWord = "SHAZAM"
        End If
        sr = IO.File.OpenText(txtName.Text)
        txtItem.Text = sr.ReadLine
        sr.Close()
    End Sub
```

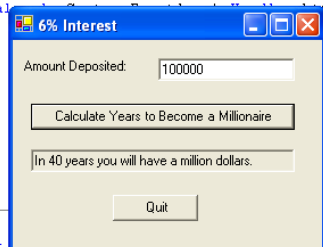
```
Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnQuit
    End
End Sub
End Class
```



# EXAMPLE DO LOOP

```
Private Sub btnYears_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnYears
    'Compute years required to become a millionaire
    Dim balance As Double, numYears As Integer
    balance = Cdbl(txtAmount.Text)
    Do While balance < 1000000
        balance += 0.06 * balance
        numYears += 1
    Loop
    txtWhen.Text = "In " & numYears & _
        " years you will have a million dollars."
End Sub
```

```
Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnQuit
    End
End Sub
```



## DO LOOPS

- Avoid infinite loops - loops never exited
- Word “While” – testing at top of loop
- Word “Until” – testing at bottom of loop

## PROCESSING LISTS OF DATA

- Devices useful when working with lists
  - Counters – calculate number of elements in lists
  - Accumulators – sum numerical values in lists
  - Flags – record whether certain events have occurred
  - Peek method – used to determine when end of text file has been reached

## PEEK METHOD

- Data often retrieved from file using Do loop
- VB.NET has way to tell us if we reached end of file
- Assume file opened as StreamReader object names sr
- The value of

sr.Peek

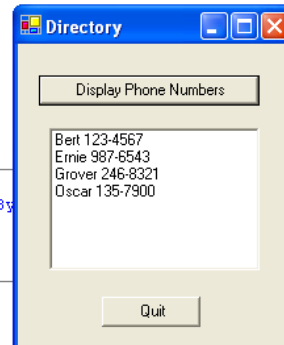
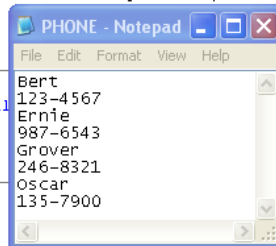
is ANSI value of first character of line about to be read with ReadLine

– If end of file reached, value of sr.Peek is -1

## PEEK METHOD EXAMPLE

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim name, phoneNum As String
    Dim sr As IO.StreamReader = IO.File.OpenText("PHONE.TXT")
    lstNumbers.Items.Clear()
    Do While sr.Peek <> -1
        name = sr.ReadLine
        phoneNum = sr.ReadLine
        lstNumbers.Items.Add(name & " " & phoneNum)
    Loop
    sr.Close()
End Sub

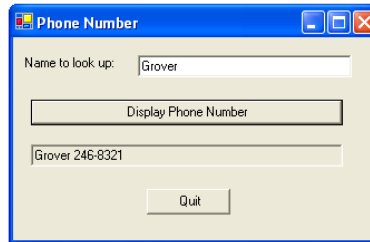
Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    End
End Sub
End Class
```



# PEEK METHOD EXAMPLE

- Example of search for name specified by user
- Do While statement compound logical expression with operator And
  - Looking for name and not end of file

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDisp
    Dim name, phoneNum As String
    Dim sr As IO.StreamReader = IO.File.OpenText("PHONE.TXT")
    Do While (name <> txtName.Text) And (sr.Peek <> -1)
        name = sr.ReadLine
        phoneNum = sr.ReadLine
    Loop
    If (name = txtName.Text) Then
        txtNumber.Text = name & " " & phoneNum
    Else
        txtNumber.Text = "name not found."
    End If
    sr.Close()
End Sub
```



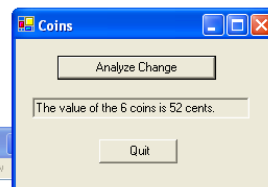
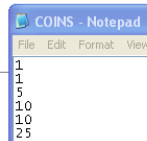
# COUNTERS & ACCUMULATORS

- Counter – numeric variable keeping track of number of items processed
- Accumulator – numeric variable that totals numbers

**Counter**  
**Accumulator**

```
Private Sub btnAnalyze_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAnalyze.
    Dim numCoins As Integer, coin As Single
    Dim sum As Double
    Dim sr As IO.StreamReader = IO.File.OpenText("COINS.TXT")
    numCoins = 0
    sum = 0
    Do While sr.Peek <> -1
        coin = sr.ReadLine
        numCoins += 1 'Add 1 to the value of numCoins
        sum += Cdbl(coin) 'Add the value of the current coin to the sum
    Loop
    sr.Close()
    txtValue.Text = "The value of the " & numCoins & " coins is " & sum & " cents."
End Sub

Private Sub btnQuit_Click(ByVal sender As System.Object,
    End
End Sub
```



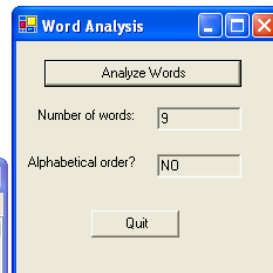
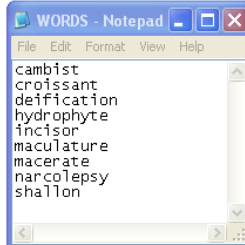
# FLAGS

- Flag – variable that tracks whether certain situation has occurred
- Data type most suited to flags is Boolean data type
  - Can be assigned only 2 values: True and False
  - Default is False
- Provide information that will be utilized after loop terminates
- Provide alternative method of terminating loop

# FLAGS EXAMPLE

- Program counts number of words in file and if words are in alphabetical order

```
Private Sub btnAnalyze_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
'Count words. Are they in alphabetical order?
Dim orderFlag As Boolean, wordCounter As Integer
Dim word1, word2 As String
Dim sr As IO.StreamReader = IO.File.OpenText("WORDS.TXT")
orderFlag = True
Do While (sr.Peek <> -1)
    word2 = sr.ReadLine
    wordCounter += 1 'Increment the word count by 1
    If word1 > word2 Then
        orderFlag = False
    End If
    word1 = word2
Loop
sr.Close()
txtNumber.Text = CStr(wordCounter)
If orderFlag = True Then
    txtAlpha.Text = "YES"
Else
    txtAlpha.Text = "NO"
End If
End Sub
End Sub
```



# FOR...NEXT LOOPS

- Used when we know how many times the loop should be executed

- General form:

For i = m To n  
statement(s)

Next

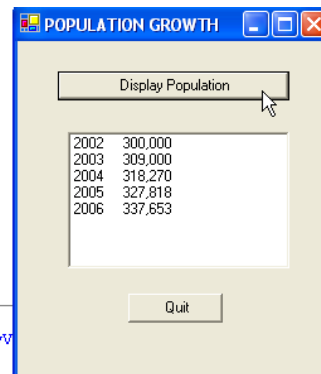
- i = control variable
- m = initial value
- n = terminating value
- statements → body of loop

# FOR...NEXT LOOPS

- Example

Windows Form Designer generated code

```
Private Sub btnDisplay_Click(ByVal sender As System.Object,  
    'Display population from 2002 to 2006  
    Dim pop As Double = 300000  
    Dim yr As Integer  
    Dim fmtStr As String = "{0,4}{1,12:N0}"  
    lstTable.Items.Clear()  
    For yr = 2002 To 2006  
        lstTable.Items.Add(String.Format(fmtStr, yr, pop))  
        pop += 0.03 * pop  
    Next  
End Sub  
  
Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal  
    End  
End Sub
```

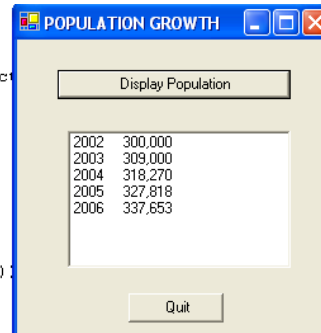


## FOR...NEXT LOOPS

- Initial and terminating values can be literals, variables, expressions

Shows Form Designer generated code

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDisplay.Click
    'Display population from 2002 to 2006
    Dim pop As Double = 300000
    Dim yr As Integer
    Dim fmtStr As String = "{0,4}{1,12:N0}"
    lstTable.Items.Clear()
    Dim firstYr As Integer = 2002
    Dim lastYr As Integer = 2006
    For yr = firstYr To lastYr
        lstTable.Items.Add(String.Format(fmtStr, yr, pop))
        pop += 0.03 * pop
    Next
End Sub
```



## FOR...NEXT LOOPS

- Can use any number to increment (normally 1)
- General form:

For i = m To n Step s

- s = step value
- m, n, s do not have to be whole numbers

# FOR...NEXT LOOPS

Windows Form Designer generated code

```
Private Sub btnDisplay_Click(ByVal sender As System.Object,  
    'Display values of index ranging from 0 to n Step s  
    Dim n, s, index As Double  
    n = CDb1(txtEnd.Text)  
    s = CDb1(txtStep.Text)  
    lstValues.Items.Clear()  
    For index = 0 To n Step s  
        lstValues.Items.Add(index)  
    Next  
End Sub  
  
Private Sub btnEnd_Click(ByVal sender As System.Object, ByV  
    End  
End Sub  
1 Class
```

For index = 0 To n Step s

n: 3.2 s: .5

Display Values of index

- 0
- 0.5
- 1
- 1.5
- 2
- 2.5
- 3

End

# FOR...NEXT LOOPS

- Can also use negative step values

```
Private Sub btnReverse_Click(ByVal sender As System  
    txtBackwards.Text = Reverse(txtWord.Text)  
End Sub  
  
Function Reverse(ByVal info As String) As String  
    Dim m, j As Integer, temp As String = ""  
    m = info.Length  
    For j = m - 1 To 0 Step -1  
        temp &= info.Substring(j, 1)  
    Next  
    Return temp  
End Function
```

Write Backwards

Enter Word: Suez

Reverse Letters

zeuS

End

# NESTED FOR...NEXT LOOPS

- Second loop must be completely contained in first loop and must have different control variable

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDisplay.Click
    Dim j, k As Integer
    Dim row, entry As String
    lstTable.Items.Clear()
    For j = 1 To 3
        row = ""
        For k = 1 To 3
            entry = j & " x " & k & " = " & (j * k)
            row &= entry & " "
        Next
        lstTable.Items.Add(row)
    Next
End Sub

Private Sub btnEnd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEnd.Click
    End
End Sub
```

