COMPUTER PROGRAMMING

FILE INPUT / OUTPUT

by
LIM KAR SING

FACULTY OF CIVIL ENGINEERING & EARTH RESOURCES
UNIVERSITI MALAYSIA PAHANG

limks@ump.edu.my

File Input / Output in VB

What is File Input / Output?

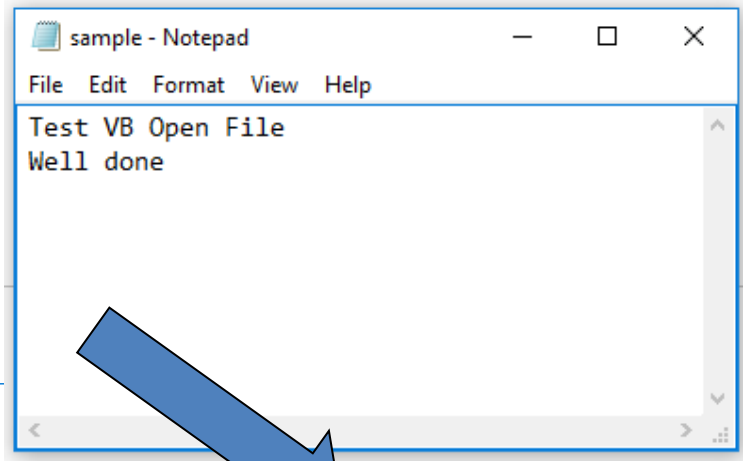
- File Input/Output a.k.a file I/O
- Storage of data in variables and arrays is temporary
- For permanent retention of data, file are used
- All of us are familiar with files. We save our work (e.g.: documents, graphics, VB project) in a file
- If we want to read/edit our work, we need to retrieve the file

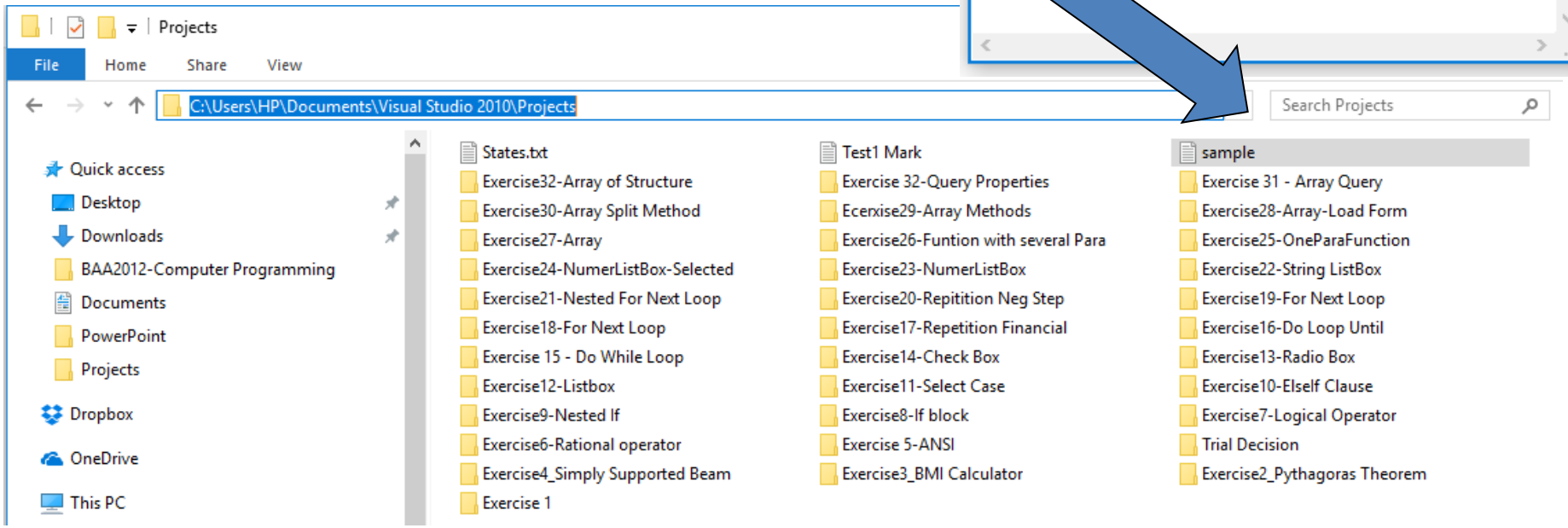
What is File Input / Output?

- Files are normally stored in hard disk / other disks
- A file is just a collection of related data stored in one unit (under one name) on a disk
- Programs use files for several reasons:
  1. Data generated by one program can be read by other program
  2. Data used in one run of a program will be available the next time the program is executed
  3. Reports can be generated and saved in the form of a file. The report can be viewed at anytime by displaying or printing the file

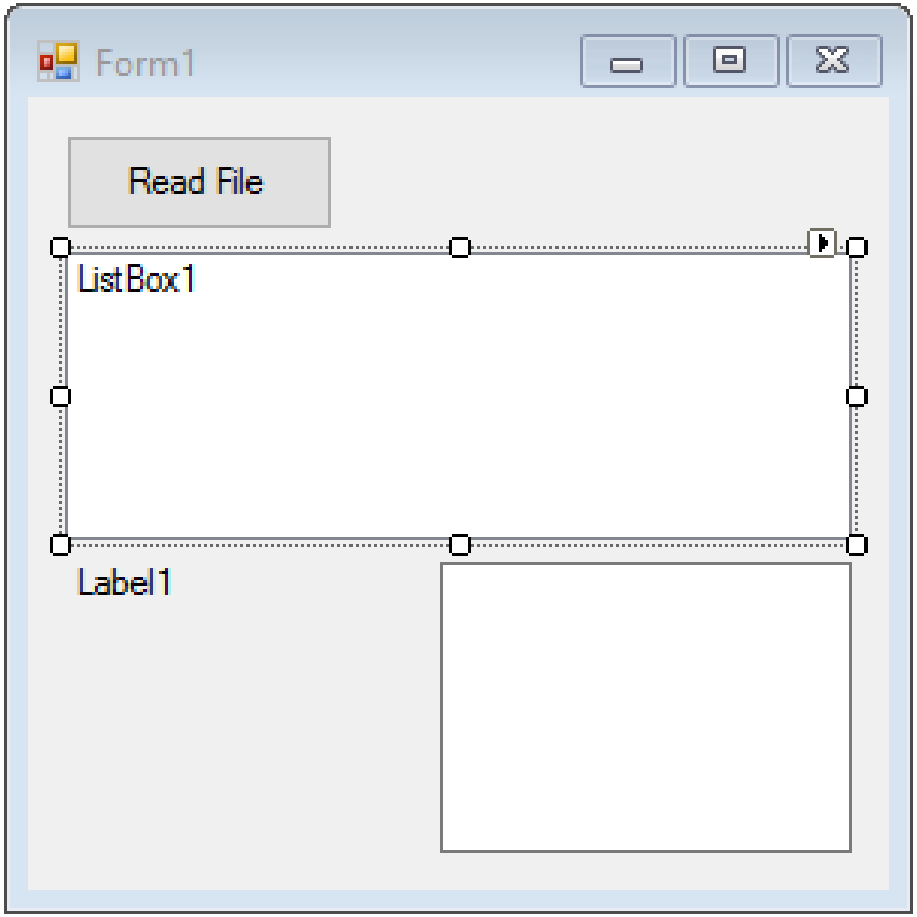
Exercise 34 – Open File: Input (Creating File)

- Create a file in notepad and save the file in any location of your computer

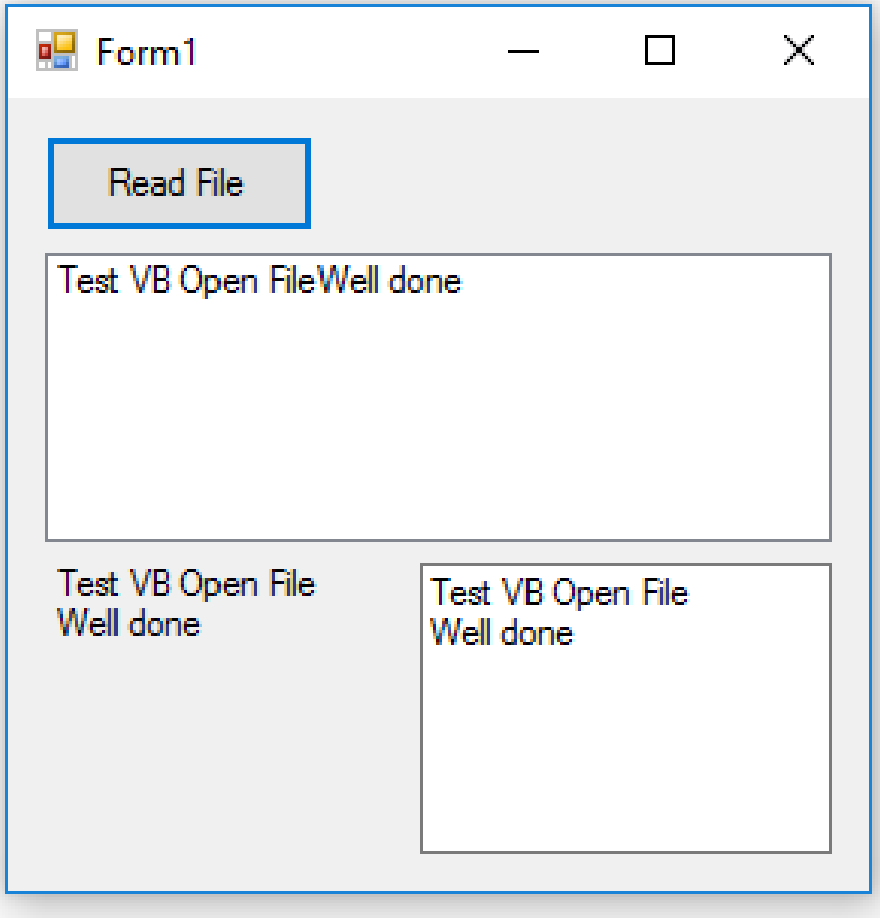




Exercise 34 – Open File: Input (Form and Output)



Form



Output

Exercise 34 – Open File: Input (Code)

```
Public Class FrmOpenFile

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim fileNum As Integer = 1
        Dim TemS As String = ""
        Dim TemL As String
        FileOpen(1, "C:\Users\HP\Documents\Visual Studio 2010\Projects\sample.txt",
OpenMode.Input)
        Do Until EOF(1)
            TemL = LineInput(1)
            TemS += TemL + vbCrLf
        Loop
        FileClose(1)
        ListBox1.Items.Add(TemS)
        Label1.Text = TemS
        TextBox1.Text = TemS
    End Sub
End Class
```

Note:

- Every file open will have a file number, in this case, we assigned #1 as file number
- General syntax for FileOpen is:

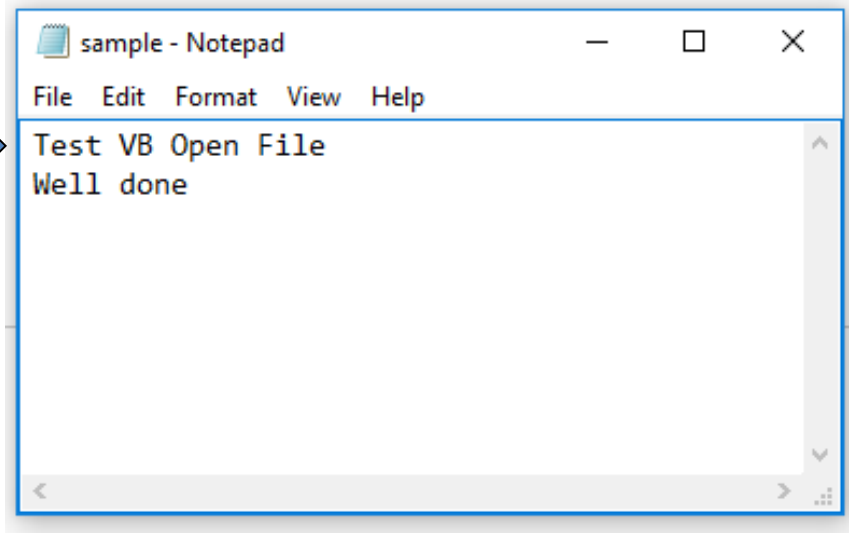
  FileOpen(file_number, “filepath”, Mode)

- FileOpen(1, "C:\Users\HP\Documents\Visual Studio 2010\Projects\sample.txt", OpenMode.Input)

- EOF stands for **End of File**
- vbCrLf represent a **line-break**
  - This line-break command is not function when using listbox
  - **Try to delete** vbCrLf in your program and see the difference

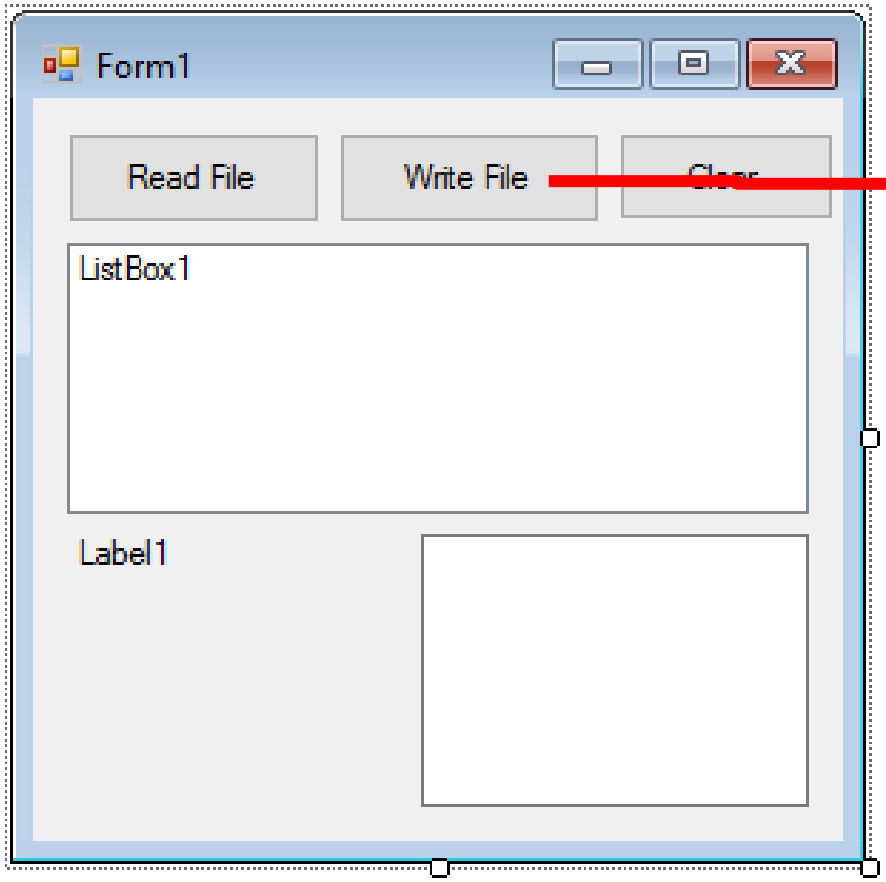
Exercise 35 – Write to File: Output (Modify Exercise 34)

- The existing data in file "sample.txt"



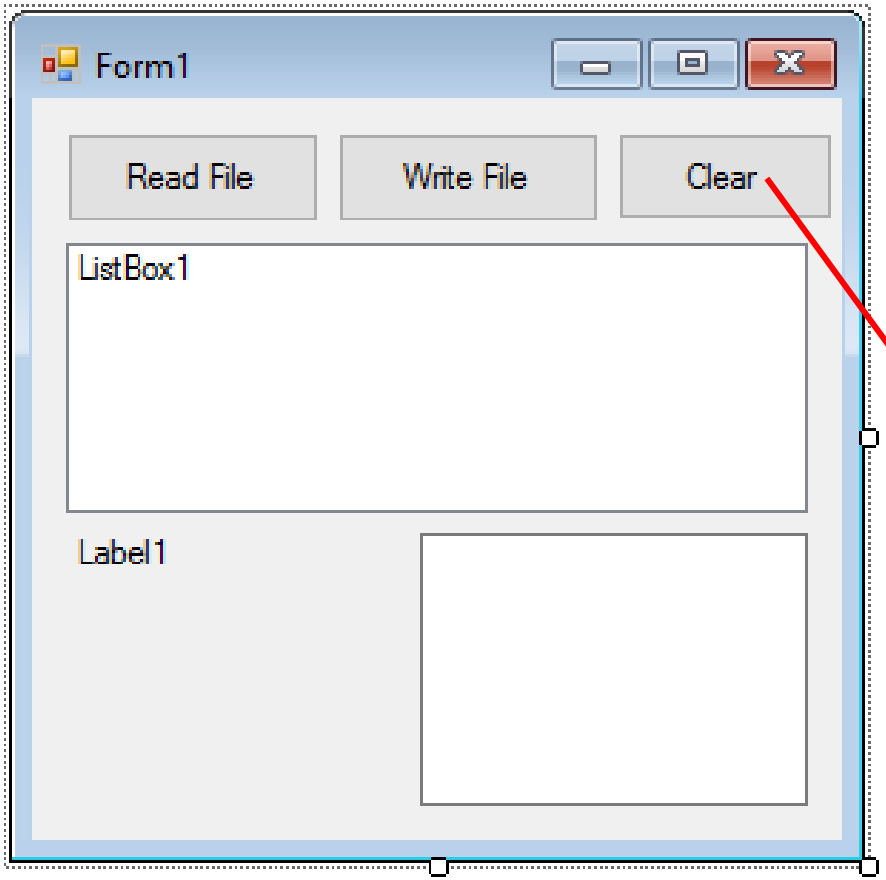
- Now we want to write new information (add data) into this file using VB
- First open Exercise 34 that you have saved and rename (give any name you like)
- Then add 2 new buttons (refer next slide)

Exercise 35 – Write to File: Output (Form and Codes)




```
Private Sub BtnWrite_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
BtnWrite.Click
        Dim fileNum As Integer = 1
        FileOpen(1,
"C:\Users\HP\Documents\Visual
Studio 2010\Projects\sample.txt",
OpenMode.Output)
        PrintLine(1, TextBox1.Text)
        FileClose(1)
    End Sub
```

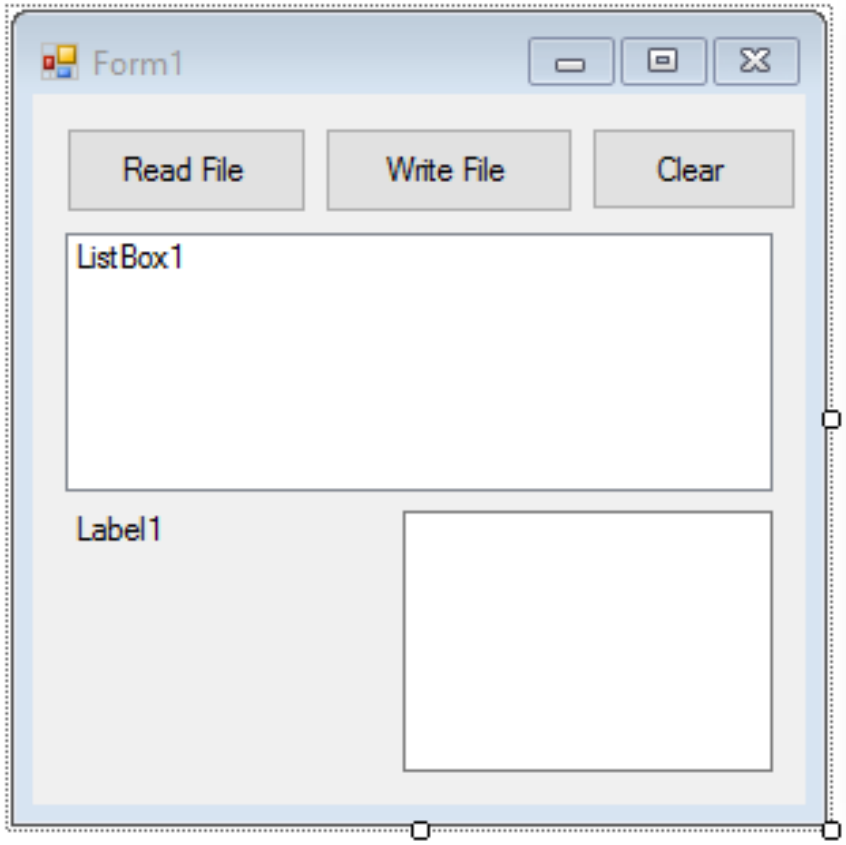
Exercise 35 – Write to File: Output (Form and Codes)



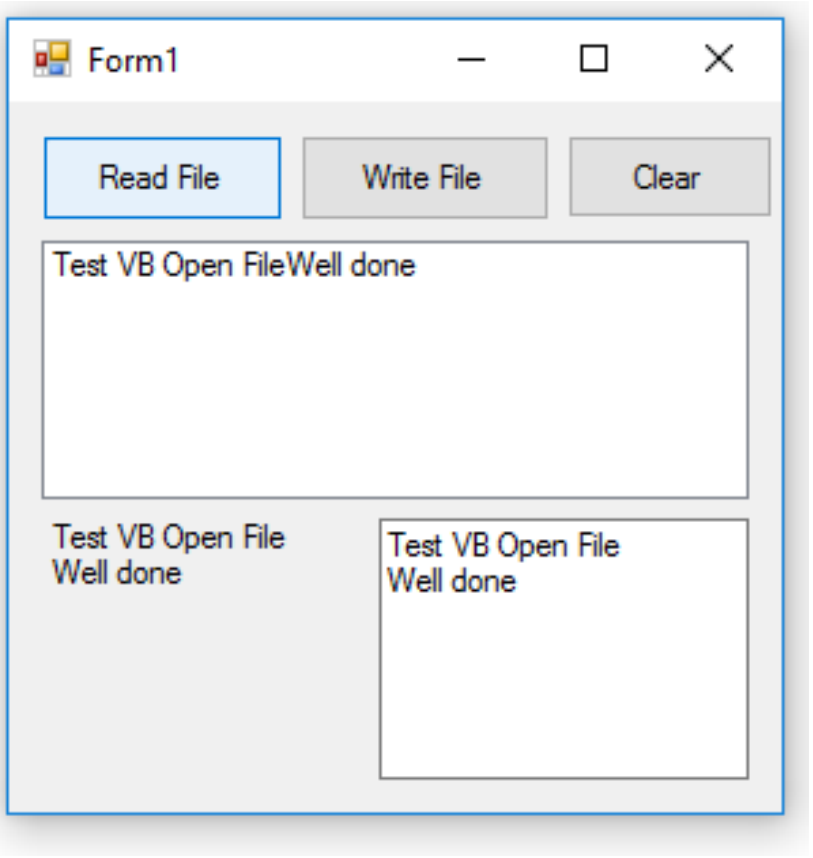

```
Private Sub BtnClear_Click(ByVal
sender As System.Object, ByVal e
As System.EventArgs) Handles
BtnClear.Click
        ListBox1.Items.Clear()
        Label1.Text = ""
        TextBox1.Clear()
    End Sub
```

Exercise 35 – Write to File: Output (Form and Outputs)

- Click at “Read File” Button – the output is similar to Exercise 34



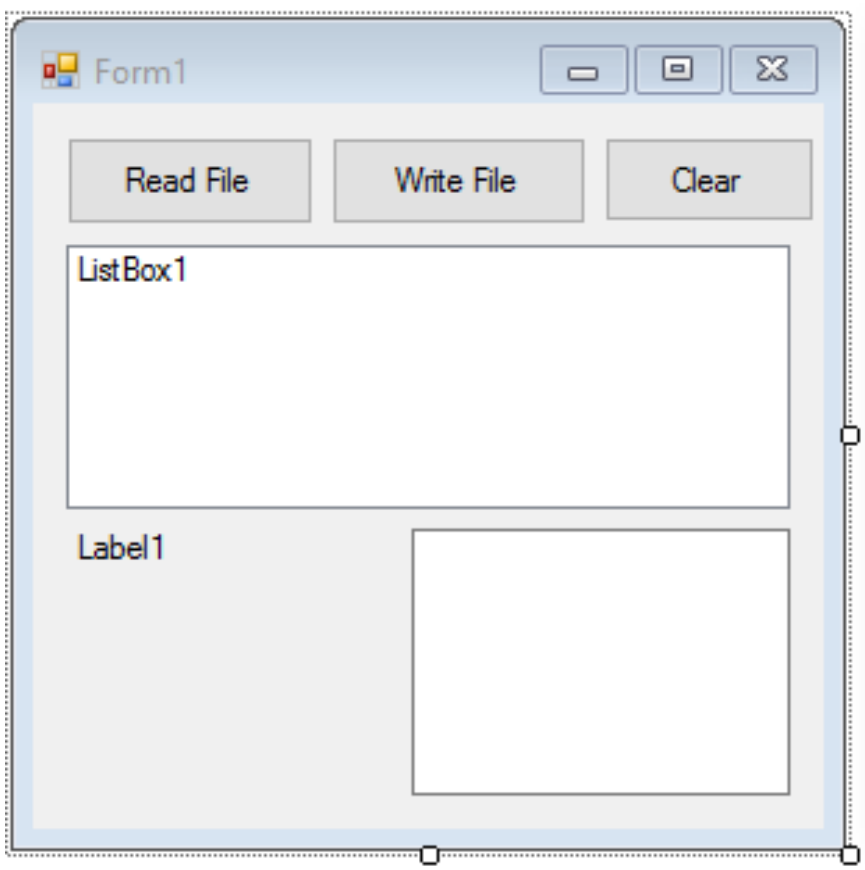
Form



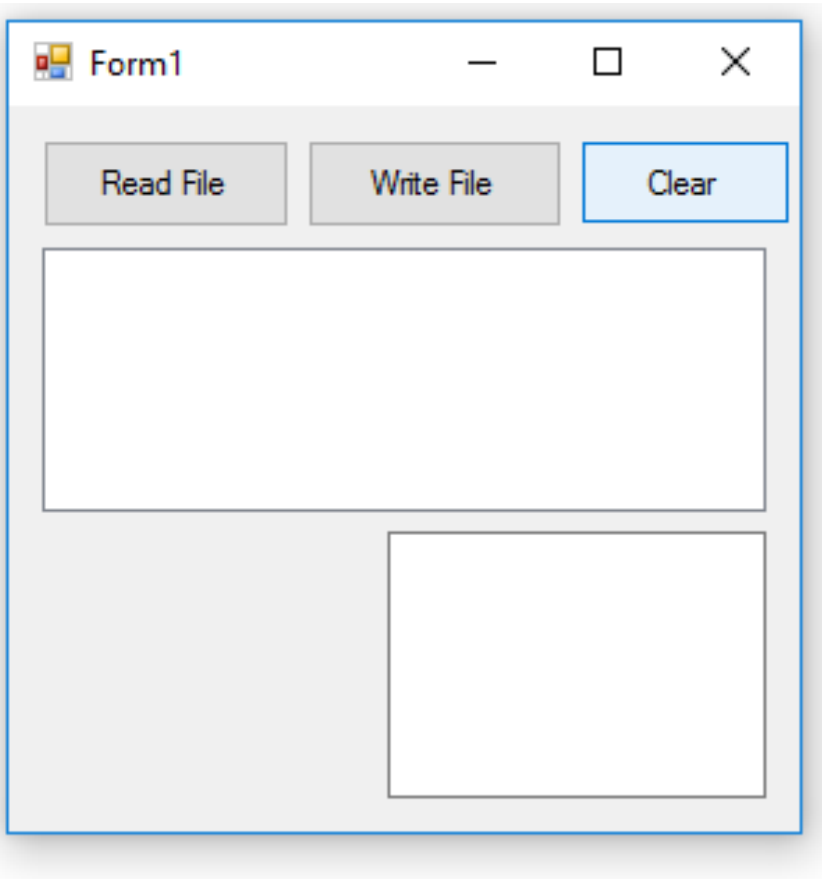
Output

Exercise 35 – Write to File: Output (Form and Outputs)

Click at “Clear” Button – the output for listbox, lable and textbox are all cleared



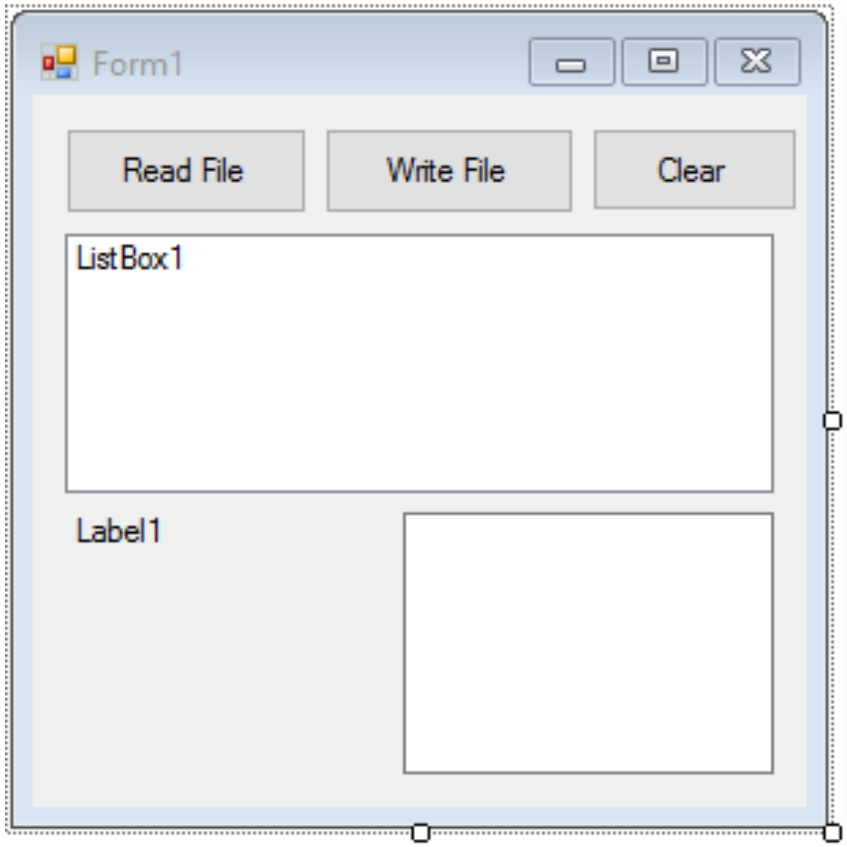
Form



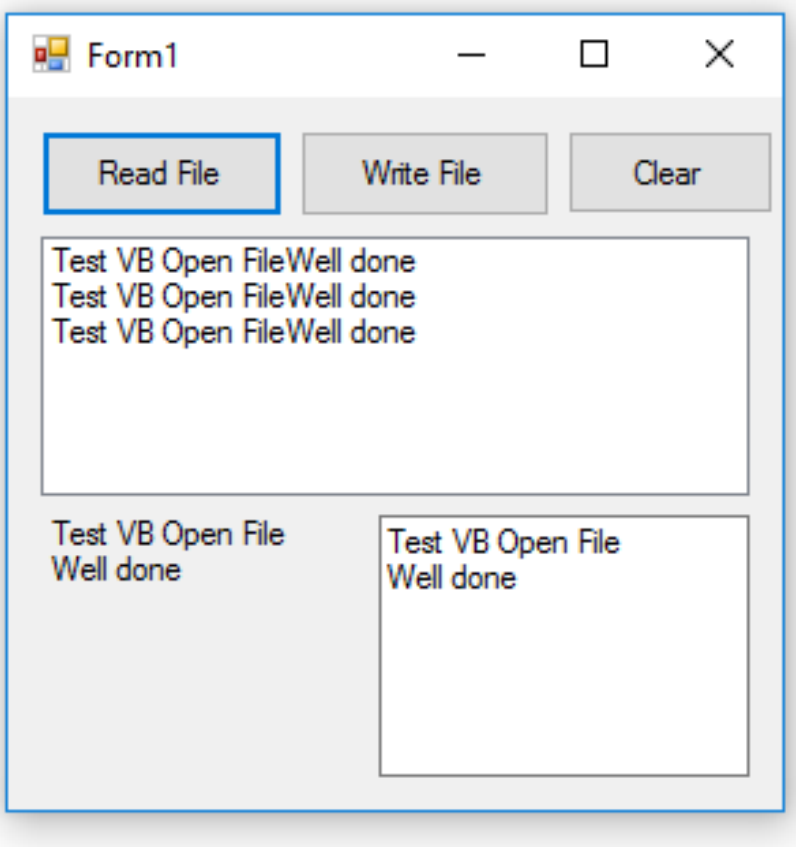
Output

Exercise 35 – Write to File: Output (Form and Outputs)

Click at “Read File” Button **AGAIN for 2-3 times** and observe the difference



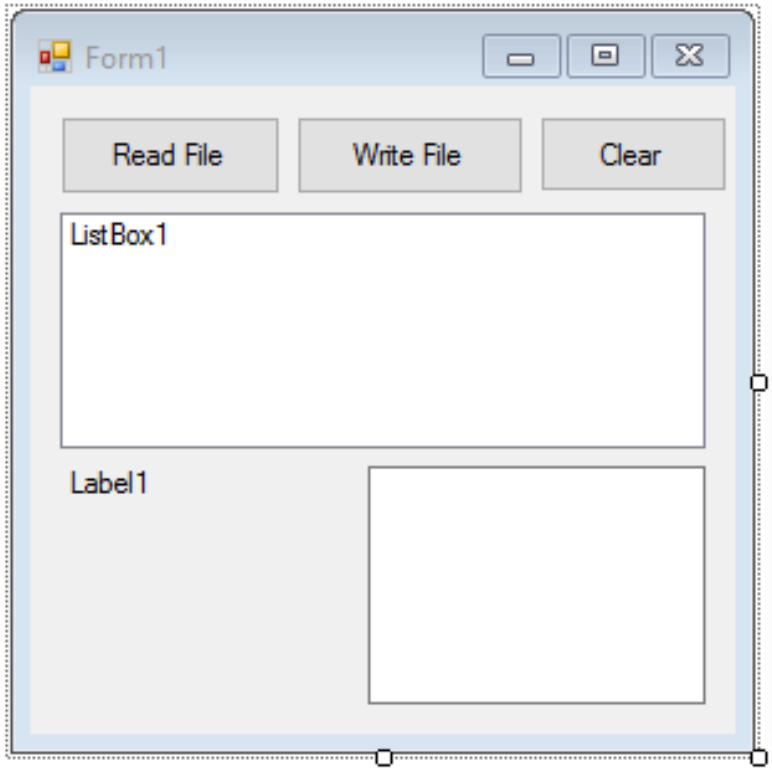
Form



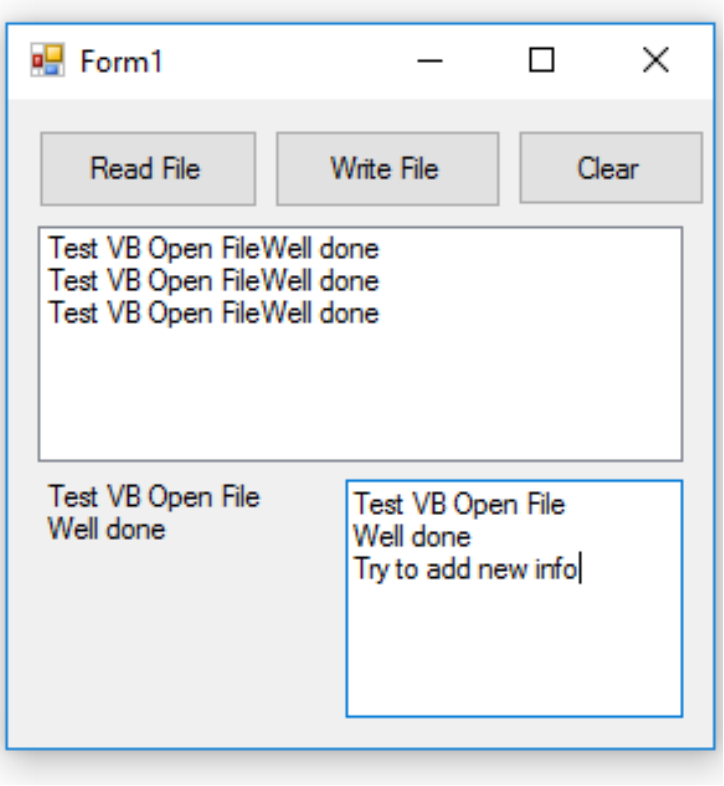
Output

Exercise 35 – Write to File: Output (Form and Outputs)

- Now try to write some new information in listbox, lable, textbox
  - You will notice only textbox allows you to write/add info



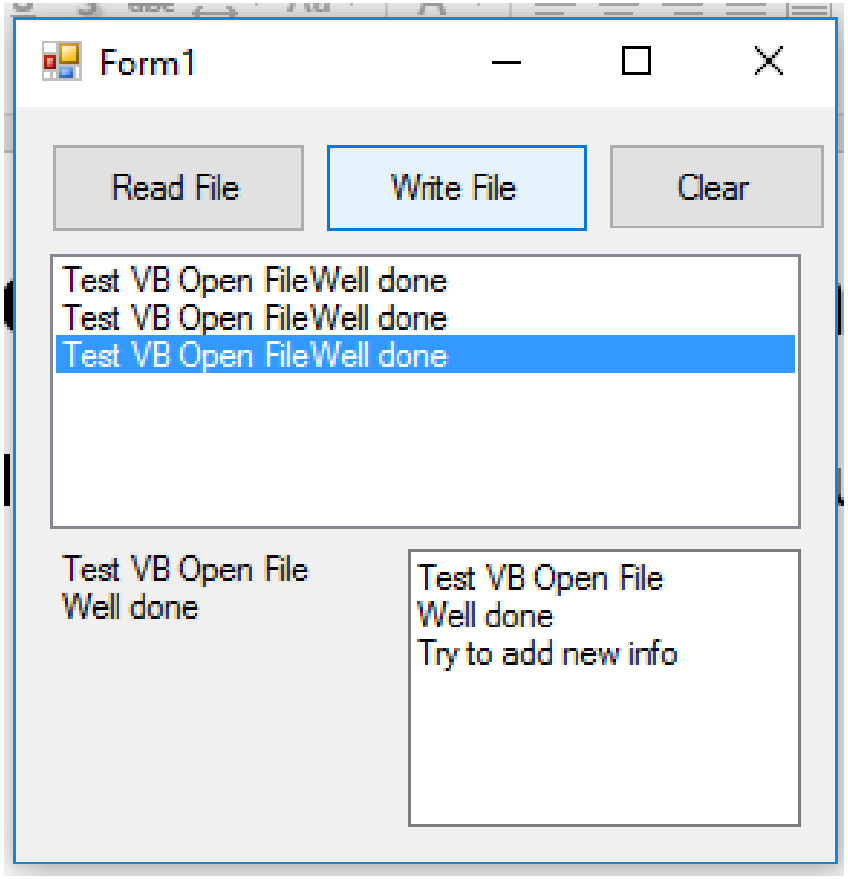
Form



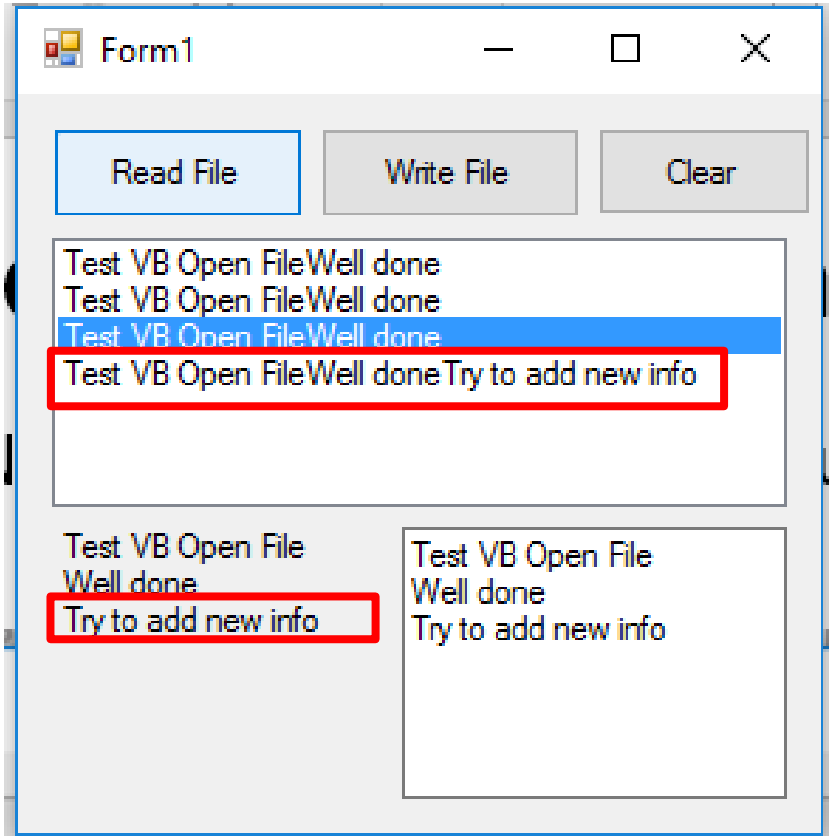
Output

Exercise 35 – Write to File: Output (Form and Outputs)

- Now click the “Write File” Button

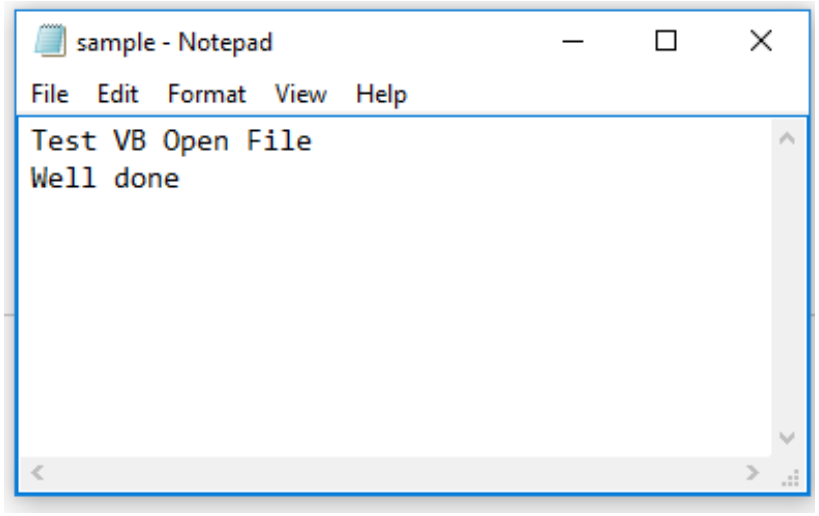


- Then click the “Read File” Button AGAIN
- New data is added

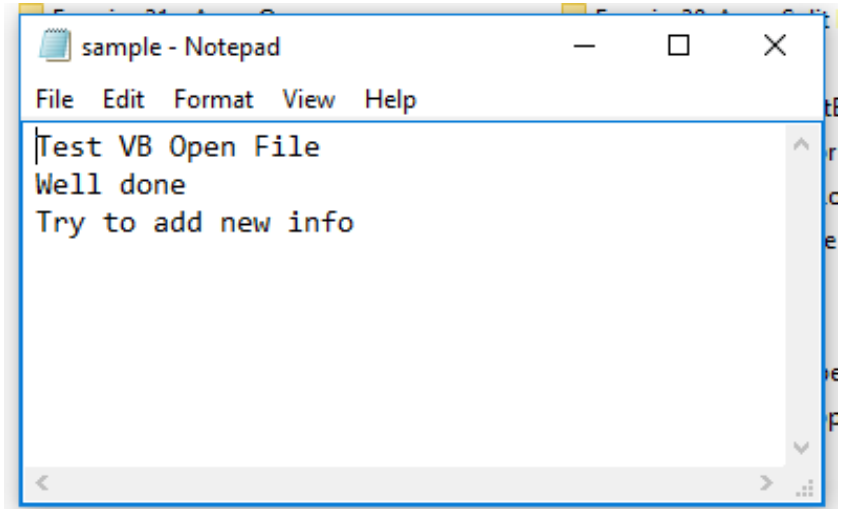


Exercise 35 – Write to File: Output (Check the modified file)

- Now open the file in your computer and see the difference



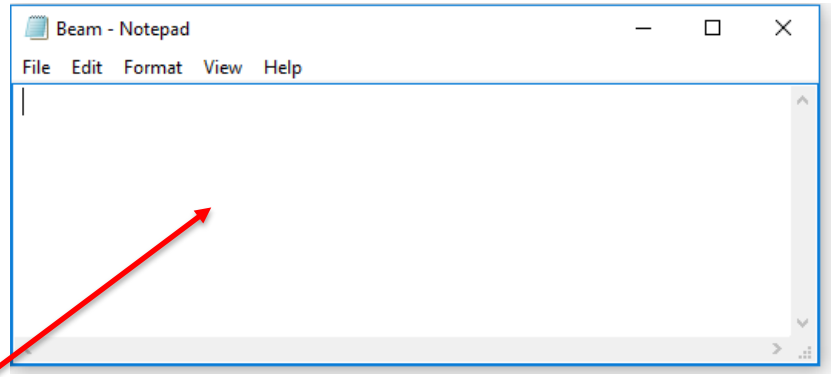
Original file



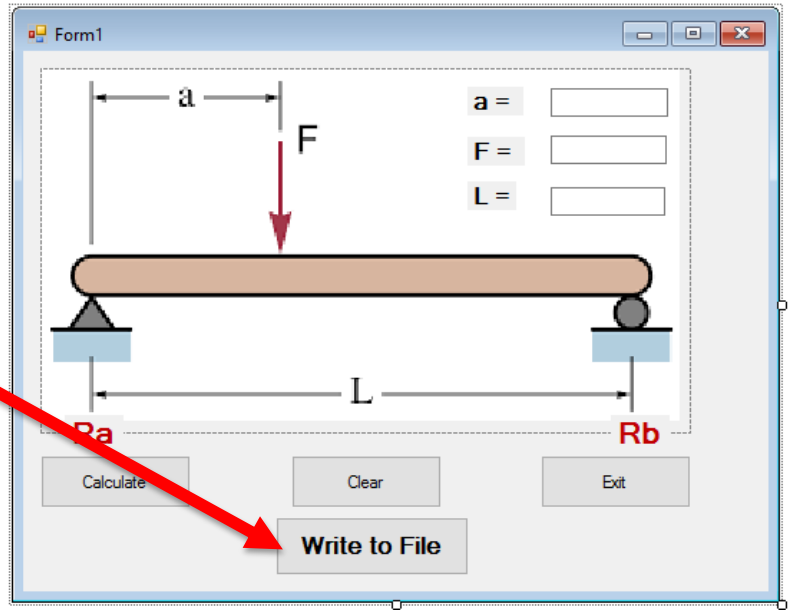
Modified file

Exercise 36 – Write to File: (Modify from Existing Exercises)

- Assuming we have developed a VB program for solving a problem and we want the results to be written in a file (e.g: notepad)



- We can create a file (notepad) and add button to write the output file for the program



- Let's modify for one of the existing program, says Exercise 4 (Simply support beam)

Exercise 36 – Write to File: (Codes for Exercise 4)

- For Calculate button (btnCal)

```
Private Sub btnCal_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCal.Click
        Dim Load, Span, A, Ra, Rb As Single
        Load = Val(txtF.Text)
        A = Val(txta.Text)
        Span = Val(txtL.Text)
        Ra = Load * (Span - A) / Span
        Rb = Load - Ra
        lblRa.Text = Ra
        lblRb.Text = Rb
    End Sub
```

Exercise 36 – Write to File: (Codes for Exercise 4)

- For Clear button (btnClear)

```
Private Sub btnClear_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnClear.Click
        lblRa.Text = "Ra"
        lblRb.Text = "Rb"
        txtF.Text = ""
        txta.Text = ""
        txtL.Text = ""
    End Sub
```

Exercise 36 – Write to File: (Codes for Exercise 4)

- For Exit button (btnExit)

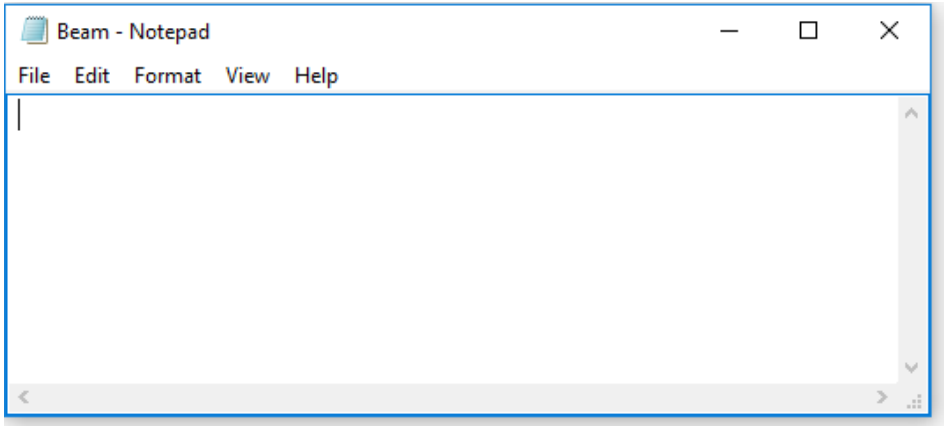
```
Private Sub btnExit_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles btnExit.Click
        End
    End Sub
```

Exercise 36 – Write to File: (Code for "Write to File" button)

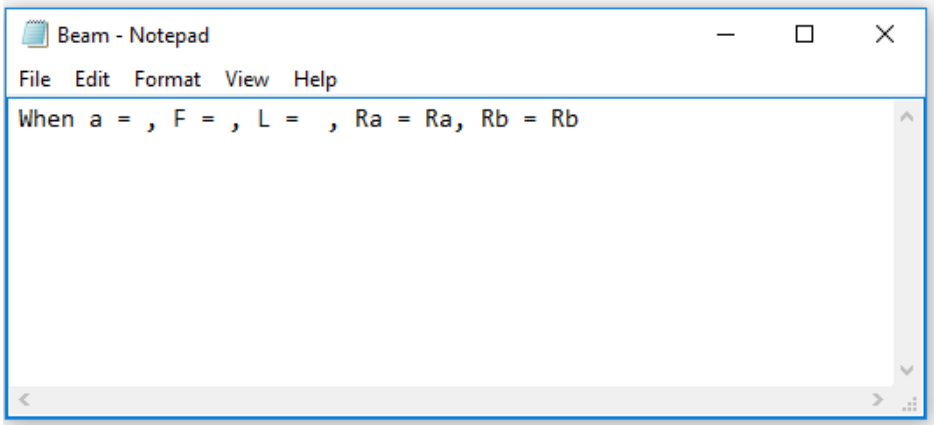
```
Private Sub BtnWrite_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles BtnWrite.Click
        Dim fileNum As Integer = 1
        FileOpen(1,
"C:\Users\HP\Documents\Visual Studio
2010\Projects\Beam.txt", OpenMode.Output)
        PrintLine(1, "When a = " & txta.Text &
", F = " & txtF.Text & ", L = " & txtL.Text &
" , " & "Ra = " & lblRa.Text & ", Rb = " &
lblRb.Text)
        FileClose(1)
    End Sub
```

Exercise 36 – Write to File: (Outputs in file - notepad)

- Once you have done the coding, start the debugging process
- First click at “Write to File” button and the data/info have been written into your file



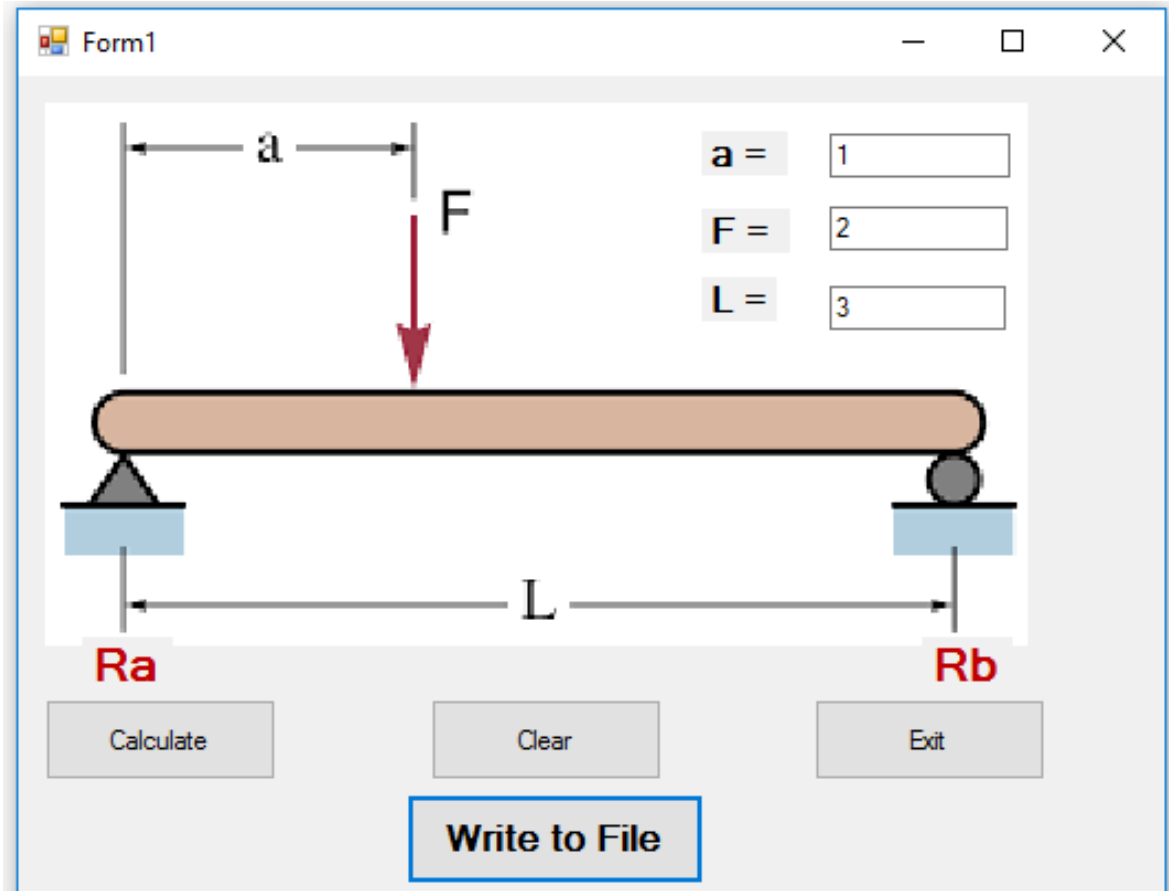
Original file (empty)



Modified file

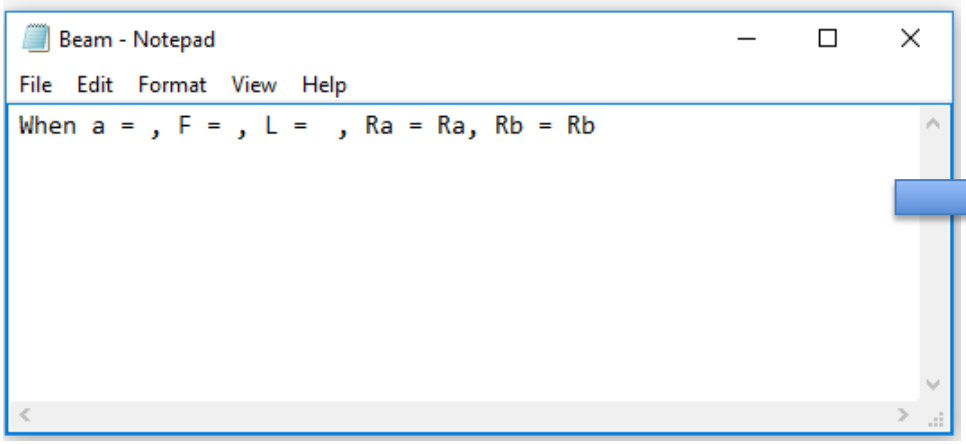
Exercise 36 – Write to File: (Outputs in file - notepad)

- Now input values for a, F and L then click at “Write to File” button



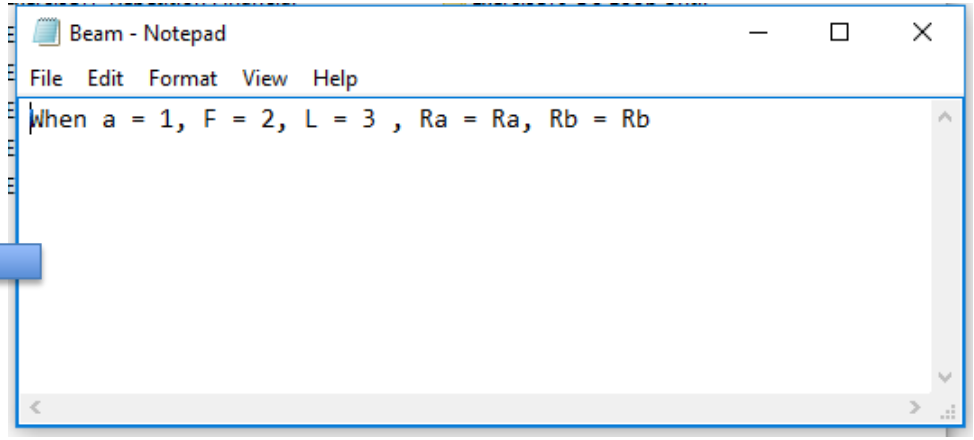
Exercise 36 – Write to File: (Outputs in file - notepad)

- Now input values for a, F and L then click at “Write to File” button



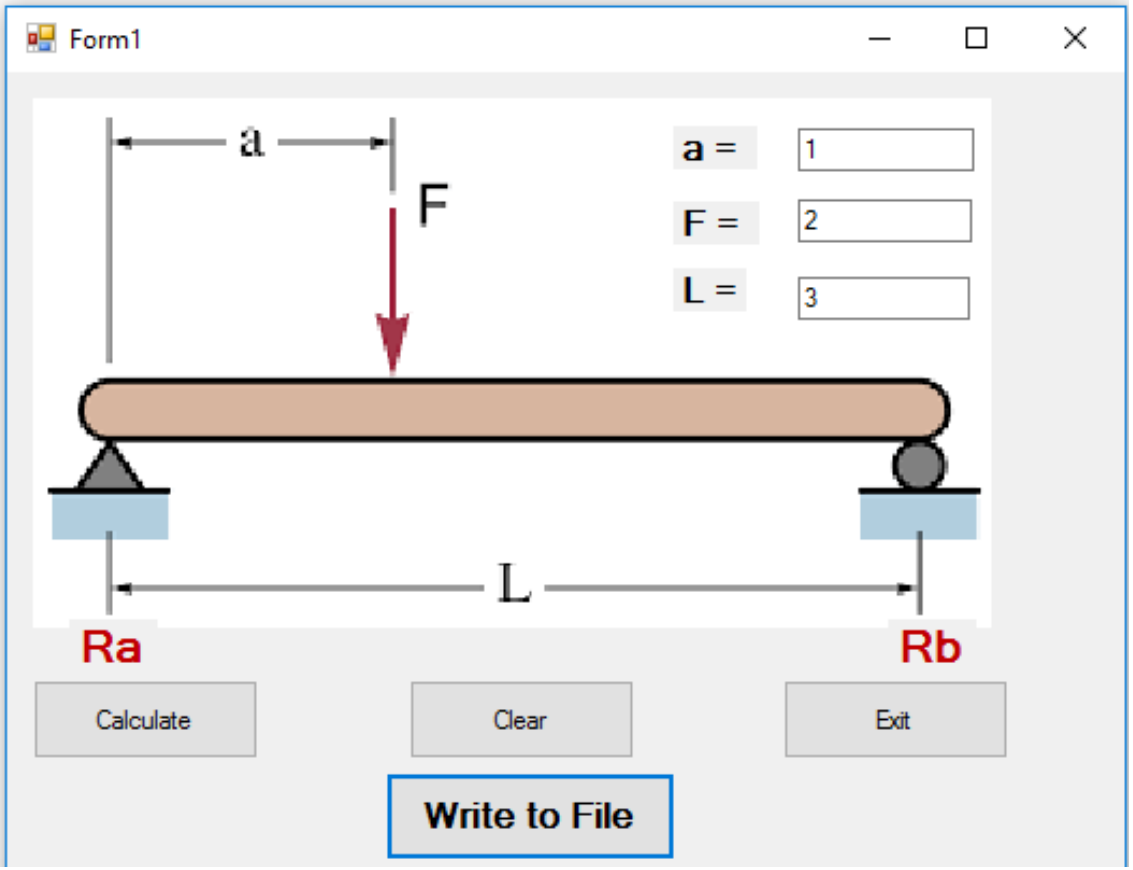
Original file

Modified file



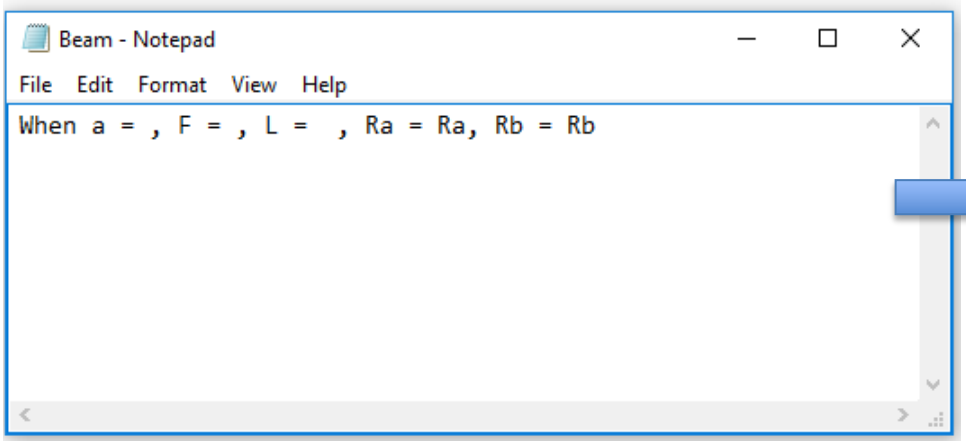
Exercise 36 – Write to File: (Outputs in file - notepad)

- Now input values for a, F and L , click at “Calculate” button **before** click at “Write to File” button



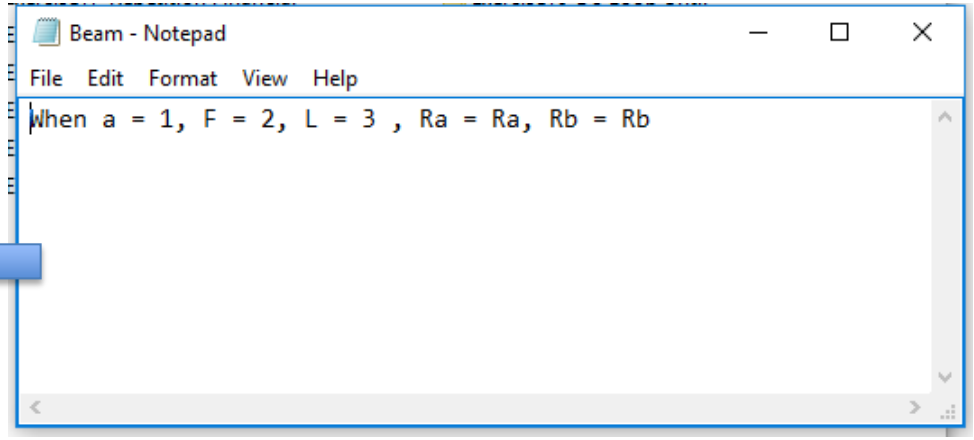
Exercise 36 – Write to File: (Outputs in file - notepad)

Now input values for a, F and L , click at “Calculate” button **before** click at “Write to File” button

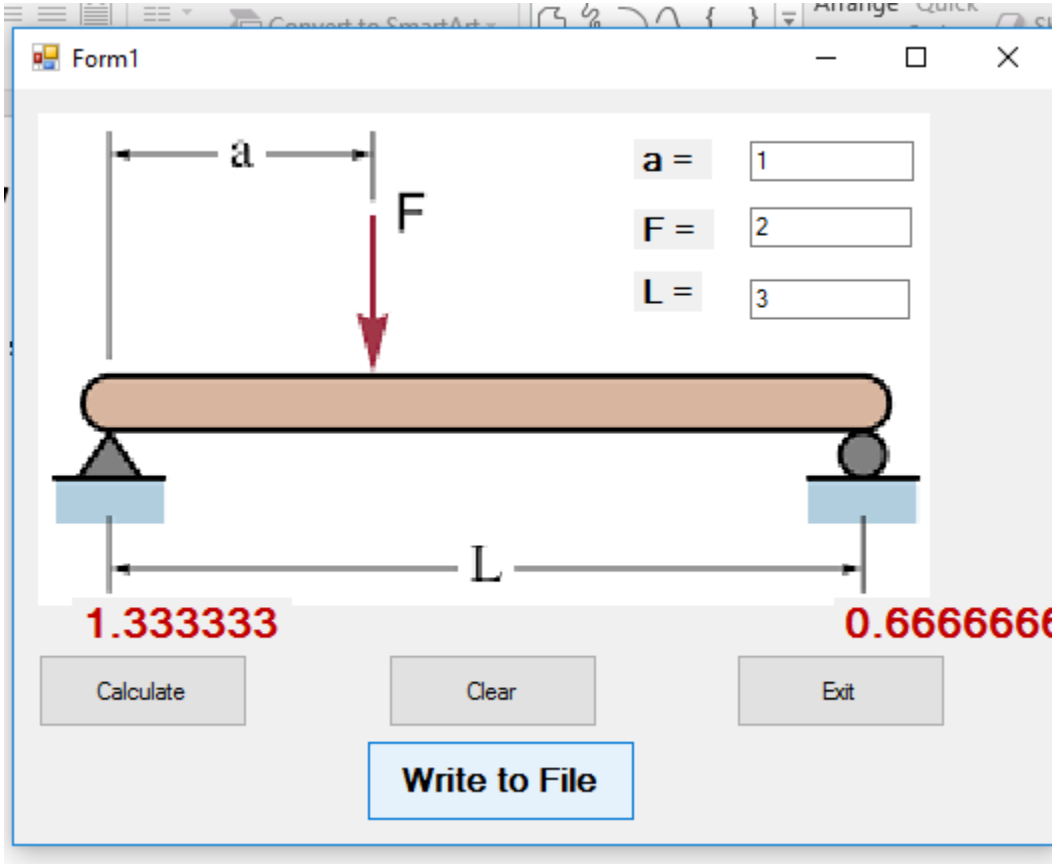


Original file

Modified file

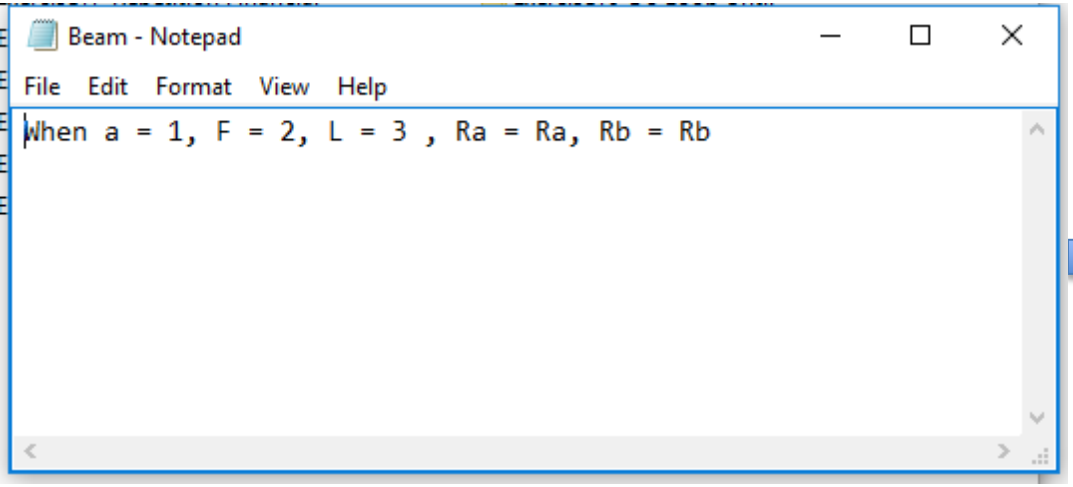


- Now input values for a, F and L , click at “Calculate” button **Followed by** click at “Write to File” button

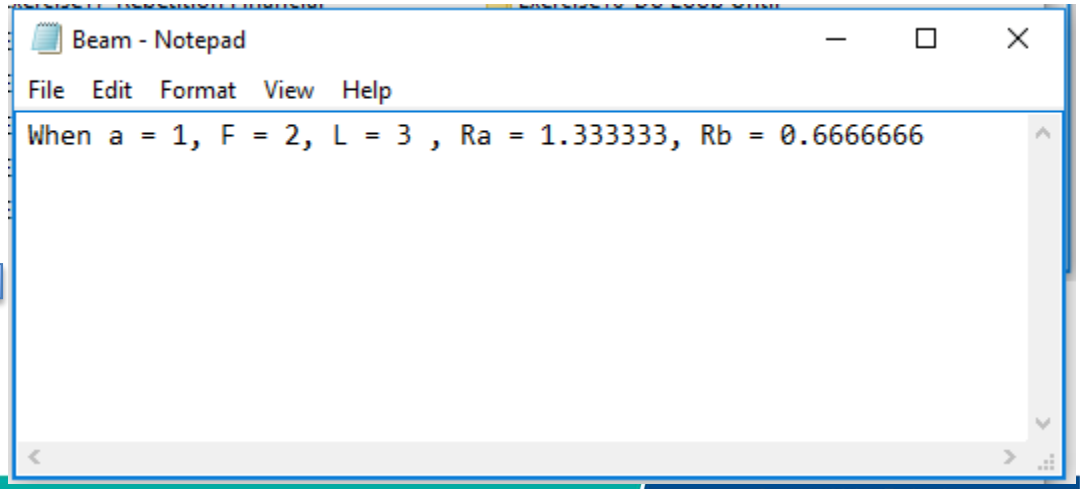


Exercise 36 – Write to File: (Outputs in file - notepad)

Now input values for a, F and L , click at “Calculate” button **before** click at “Write to File” button



Original file



Modified file

Have Fun ☺

Lets try to write the output file with other existing program