Software Design and Development

9 Debugging

Terry Marris July 2009

Debugging involves looking for errors in your program, and fixing them. We have already used some of the debugging facilities provided by Visual Basic 2008. The compiler tells us about our compile-time errors (e.g. spelling mistakes). We have used Try ... Catch to deal with run-time errors (e.g. number format errors). Now we look at a debugging facility to help find logic errors.

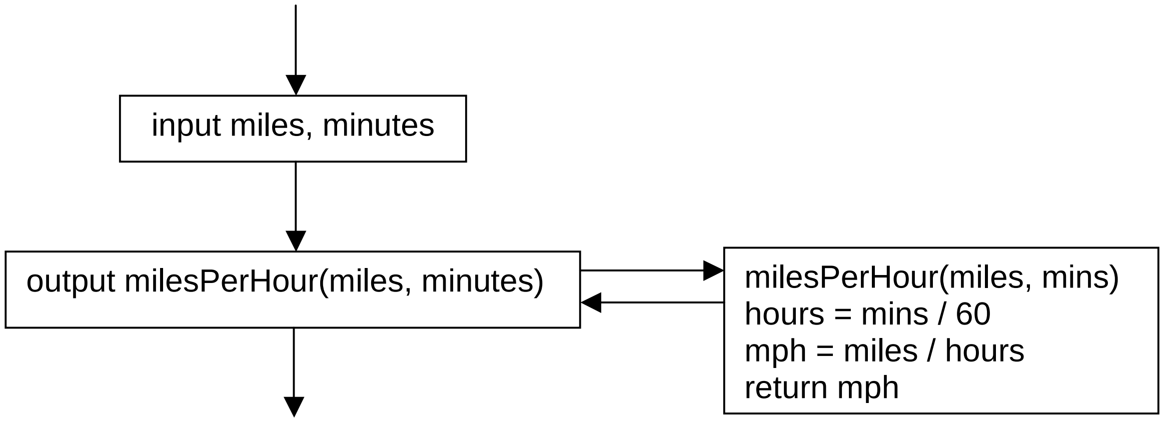
9.1 Logic Error

Specification. We are training for the Leicester Marathon. We note that we can run 10 miles in 60 minutes. What is that in miles per hour? (The answer is 10 - pretty slow when Haile Gebrselassie runs a marathon at about 15 miles per hour!). Anyway, a function is required that will convert miles and minutes into miles per hour.

Design - Structured English

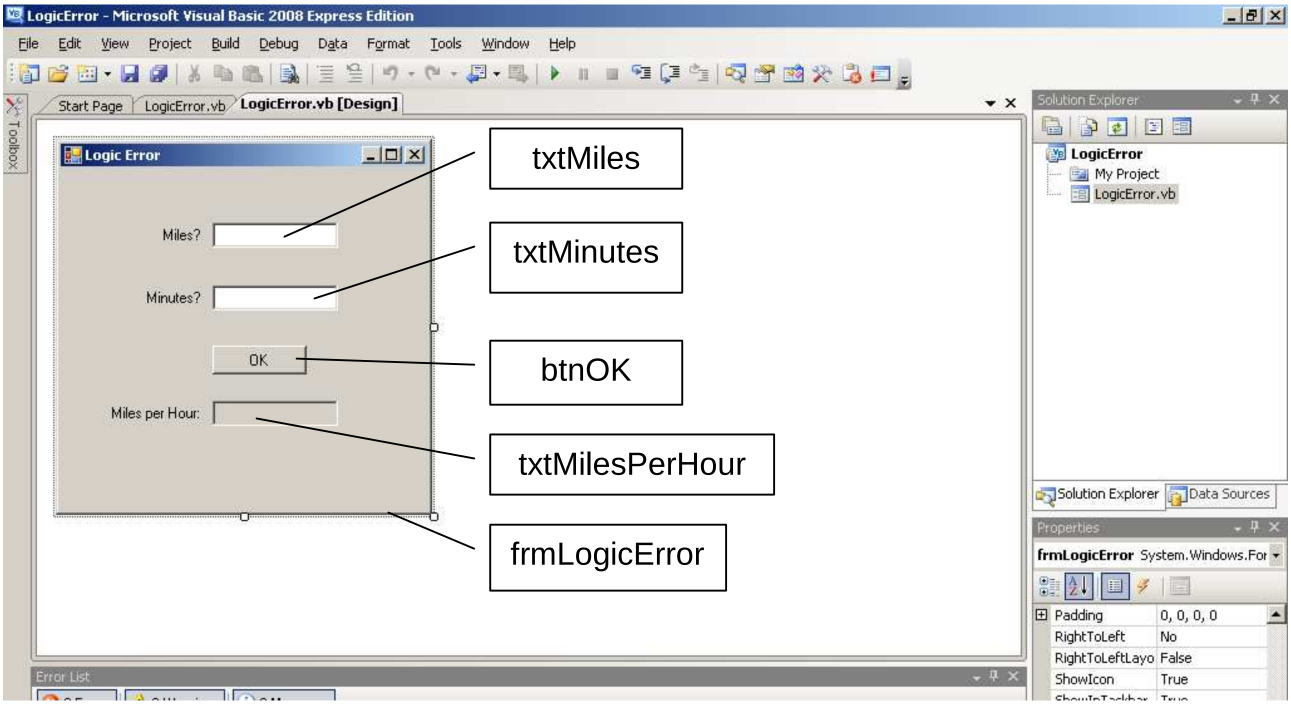
```
function milesPerHour(miles, minutes)
    hours = minutes / 60
    mph = miles / hours
    return mph
endfunction


input miles, minutes
output milesPerHour(miles, minutes)
```

Design - Program Flowchart



User Interface

1. Start a new project and name it *LogicError*
2. Set File Name = *LogicError.vb*
3. Set
   a. Form
      i. Name = *frmLogicError*
      ii. Text = *Logic Error*
   b. Label
      i. Name = *lblMiles*
      ii. Text = *Miles?*
   c. Text Box
      i. Name = *txtMiles*
   d. Label
      i. Name = *lblMinutes*
      ii. Text = *Minutes?*
   e. Text Box
      i. Name = *txtMinutes*
   f. Button
      i. Name = *btnOK*
      ii. Text = *OK*
   g. Label
      i. Name = *lblMilesPerHour*
      ii. Text = *Miles per Hour:*
   h. Text Box
      i. Name = *txtMilesPerHour*
      ii. ReadOnly = *True*



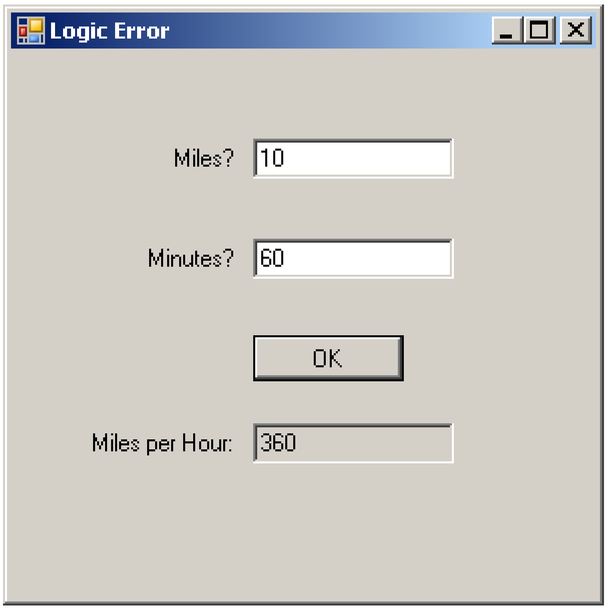
Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftLayo	False
ShowIcon	True

Programming Code

4. The entire VB File:

```
Public Class frmLogicError
' Terry Marris  29 July 2009
' Demonstrates logic errors

Function getMilesPerHour(ByVal dblMiles As Double,
                         ByVal dblMinutes As Double) As Double
    ' converts miles and minutes into miles per hour
    Dim dblHours As Double = dblMinutes / 60
    Dim dblMilesPerHour = dblMiles / dblHours
    Return dblMilesPerHour
End Function

Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e
                         As System.EventArgs) Handles btnOK.Click
    ' inputs miles and minutes, outputs miles per hour
    Dim dblMinutes As Double = Convert.ToDouble(txtMinutes.Text)
    Dim dblMiles As Double = Convert.ToDouble(txtMiles.Text)
    txtMilesPerHour.Text = getMilesPerHour(dblMinutes,dblMiles)
End Sub
End Class
```

Program Run



Miles? 10

Minutes? 60

Miles per Hour: 360

We have a problem. 10 miles in 60 minutes is 10 miles per hour, not 360!

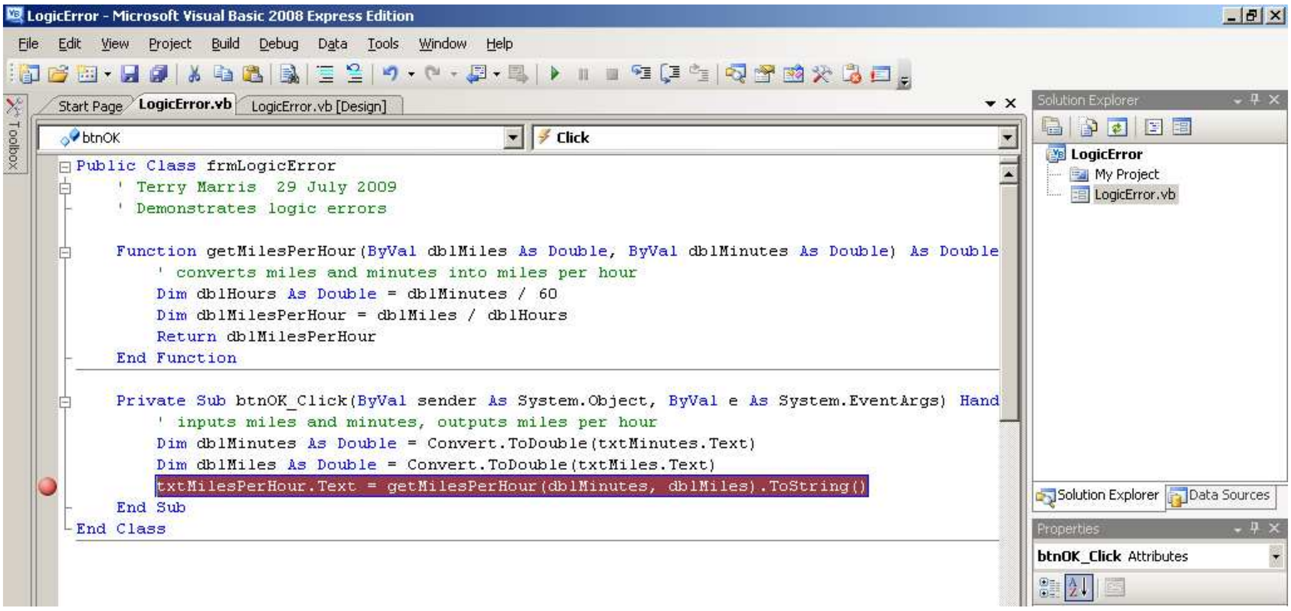
9.2 Using the Debugger

We see how to set a breakpoint , step through our VB code and inspect the values of variables.

1. Save your work: choose **File, Save All**

2. In the code window click the left hand margin next on the line

```
txtMilesPerHour.Text = getMilesPerHour(dblMinutes,dblMiles)
```



```
Public Class frmLogicError
    ' Terry Marris  29 July 2009
    ' Demonstrates logic errors

    Function getMilesPerHour(ByVal dblMiles As Double, ByVal dblMinutes As Double) As Double
        ' converts miles and minutes into miles per hour
        Dim dblHours As Double = dblMinutes / 60
        Dim dblMilesPerHour = dblMiles / dblHours
        Return dblMilesPerHour
    End Function

    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Hand
        ' inputs miles and minutes, outputs miles per hour
        Dim dblMinutes As Double = Convert.ToDouble(txtMinutes.Text)
        Dim dblMiles As Double = Convert.ToDouble(txtMiles.Text)
        txtMilesPerHour.Text = getMilesPerHour(dblMinutes, dblMiles).ToString()
    End Sub
End Class
```

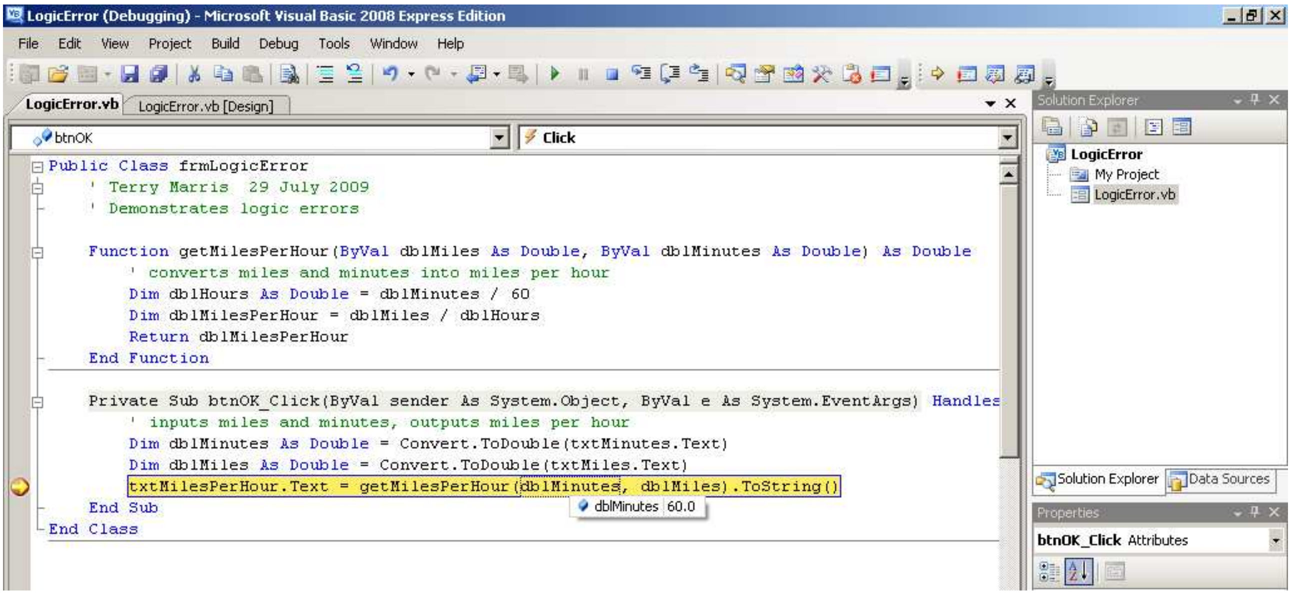
A red dot is shown in the margin and the line is highlighted in red. This sets a breakpoint.

3. Press F5 to run the program.

4. Drag your run-time window clear of your programming code.

5. Enter 10 miles and 60 minutes, and click OK.

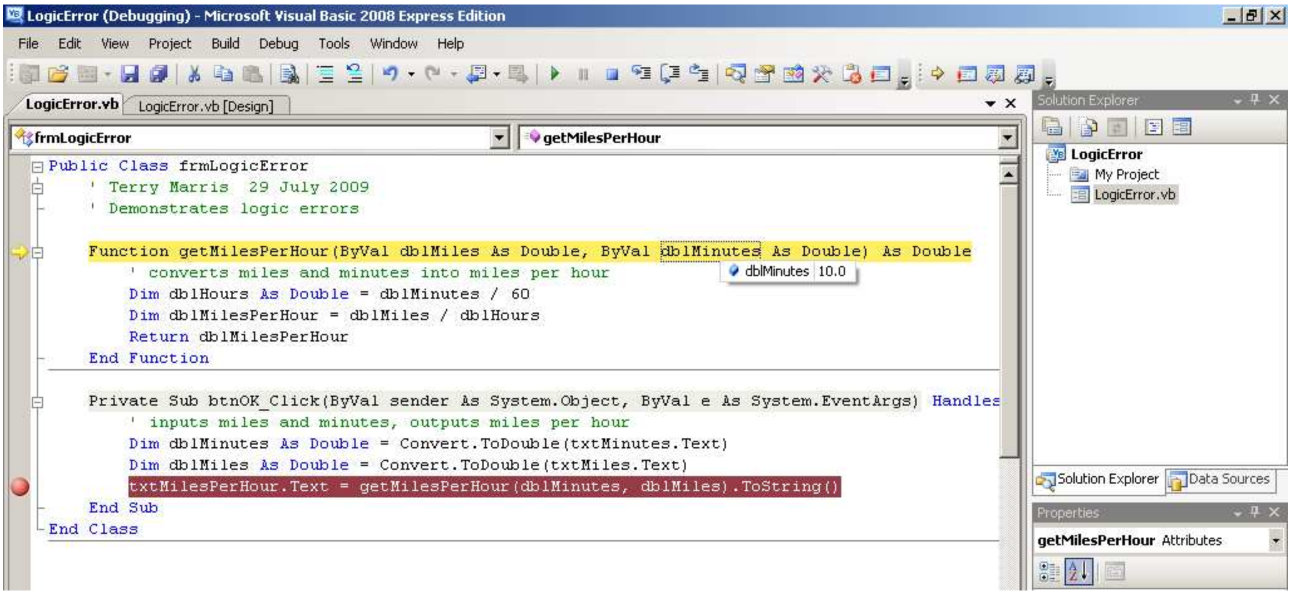
6. Position the cursor over *dblMinutes* - a little window shows its value, 60



```
Public Class frmLogicError
    ' Terry Marris  29 July 2009
    ' Demonstrates logic errors

    Function getMilesPerHour(ByVal dblMiles As Double, ByVal dblMinutes As Double) As Double
        ' converts miles and minutes into miles per hour
        Dim dblHours As Double = dblMinutes / 60
        Dim dblMilesPerHour = dblMiles / dblHours
        Return dblMilesPerHour
    End Function

    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        ' inputs miles and minutes, outputs miles per hour
        Dim dblMinutes As Double = Convert.ToDouble(txtMinutes.Text)
        Dim dblMiles As Double = Convert.ToDouble(txtMiles.Text)
        txtMilesPerHour.Text = getMilesPerHour(dblMinutes, dblMiles).ToString()
    End Sub
End Class
```

7. Press F8 to go to the next line.

8. Position the cursor over the *dblMinutes* in the highlighted function heading. It shows 10. *dblMinutes* on the previous line was 60. We have found the error. The *dblMiles* and the *dblMinutes* in the function heading are in the wrong order, they need changing round.



```
Public Class frmLogicError
    ' Terry Marris  29 July 2009
    ' Demonstrates logic errors

    Function getMilesPerHour(ByVal dblMiles As Double, ByVal dblMinutes As Double) As Double
        ' converts miles and minutes into miles per hour
        Dim dblHours As Double = dblMinutes / 60
        Dim dblMilesPerHour = dblMiles / dblHours
        Return dblMilesPerHour
    End Function

    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
        ' inputs miles and minutes, outputs miles per hour
        Dim dblMinutes As Double = Convert.ToDouble(txtMinutes.Text)
        Dim dblMiles As Double = Convert.ToDouble(txtMiles.Text)
        txtMilesPerHour.Text = getMilesPerHour(dblMinutes, dblMiles).ToString()
    End Sub
End Class
```

9. Choose **Debug, Stop Debugging** to halt the program.

10. Click on the red dot to remove the breakpoint.

11. Fix the error and re-test.

Exercise 9.1

1. Try out the debugger with the Miles and Minutes to Miles Per Hour program as shown above.

Bibliography VB2008/help

Next we look at programming languages.