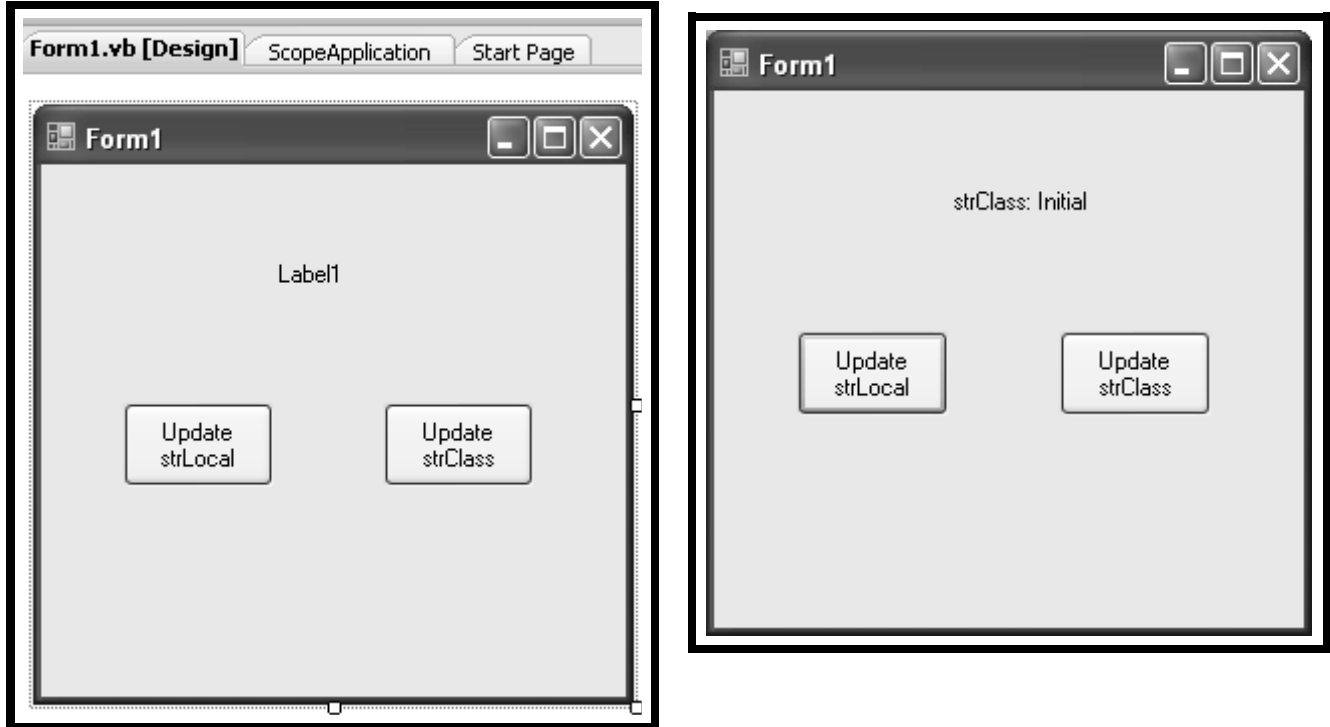


1. ScopeApplication form layout in Visual Studio is shown on the left and the application when it initially starts running is shown on the right. The code for the application is given below.



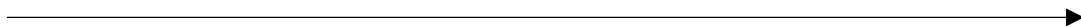
```
Public Class Form1
    Dim strClass As String = "strClass: Initial  "
    Private Sub Form1_Load(ByVal sender As System.Object, _
                           ByVal e As System.EventArgs) Handles MyBase.Load
        Label1.Text = strClass
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, _
                              ByVal e As System.EventArgs) Handles Button1.Click
        Dim strLocal As String = "strLocal: Initial"
        strLocal = "strLocal: Updated"
        Label1.Text = strClass + strLocal
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, _
                              ByVal e As System.EventArgs) Handles Button2.Click
        strClass = "strClass: Updated  "

        REM  Label1.Text = strClass + strLocal          ←←←← Causes Error
        Label1.Text = strClass
        REM  Dim strLocal As String = "strLocal: Updated"
        REM      Label1.Text = strClass + strLocal
    End Sub
End Class
```

- What causes Label1 to change from its initial Text of "Label1" when the application starts running?
- What do you predict will happen if each buttons is clicked?
- If we uncomment only the first REM in Button2_Click, then.what compiler error do you expect?
- If we uncomment only the second and third REM's, then how would the program behave?

2. *Implicit type conversion* occurs when you assign a value of one type to a variable of a different data type. In chapter 1 (Figure 1-15) when we were setting up the Visual Basic programming environment, we set Option Strict to “On”. This has the effect of only allowing *widening* conversions so that numeric accuracy is less likely to be lost. For integer numeric types, the following implicit numeric conversions are allowed with Option Strict “On”:

Byte 8-bit unsigned	Integer 32-bit signed	Long 64-bit signed -9,223,372,036,854,775,808 to +9,223,372,036,854,775,807	Decimal (29 significant digits) Used in financial calculations	Single (7 sign. digits) $\sim 1.0 \times 10^{38}$	Double (15 sign. digits) $\sim 1.0 \times 10^{308}$
---------------------------	-----------------------------	---	--	--	--


 Allowed widening

Even with Option Strict “On”, give some examples of when numeric accuracy is lost.

3. When we convert a numeric value to a string, you can use a formatting string in the ToString function. Complete the following table:

Number Value	Format String	ToString() Value
12.3	n4 (number 4 decimal places and commas)	
12.348	n1 (number 1 decimal place and commas)	
1234567.1	n (number default of 2 decimal places and commas)	
1234.560	f1 (fixed-point 1 decimal place)	
123456.0	e3 (exponential: normalized so 1 digit is to left of decimal point and 3 to right)	
.234678	p2 percent with 2 decimal places and ‘%’	
-1234567.8	c3 (currency with \$ and 3 decimal places)	