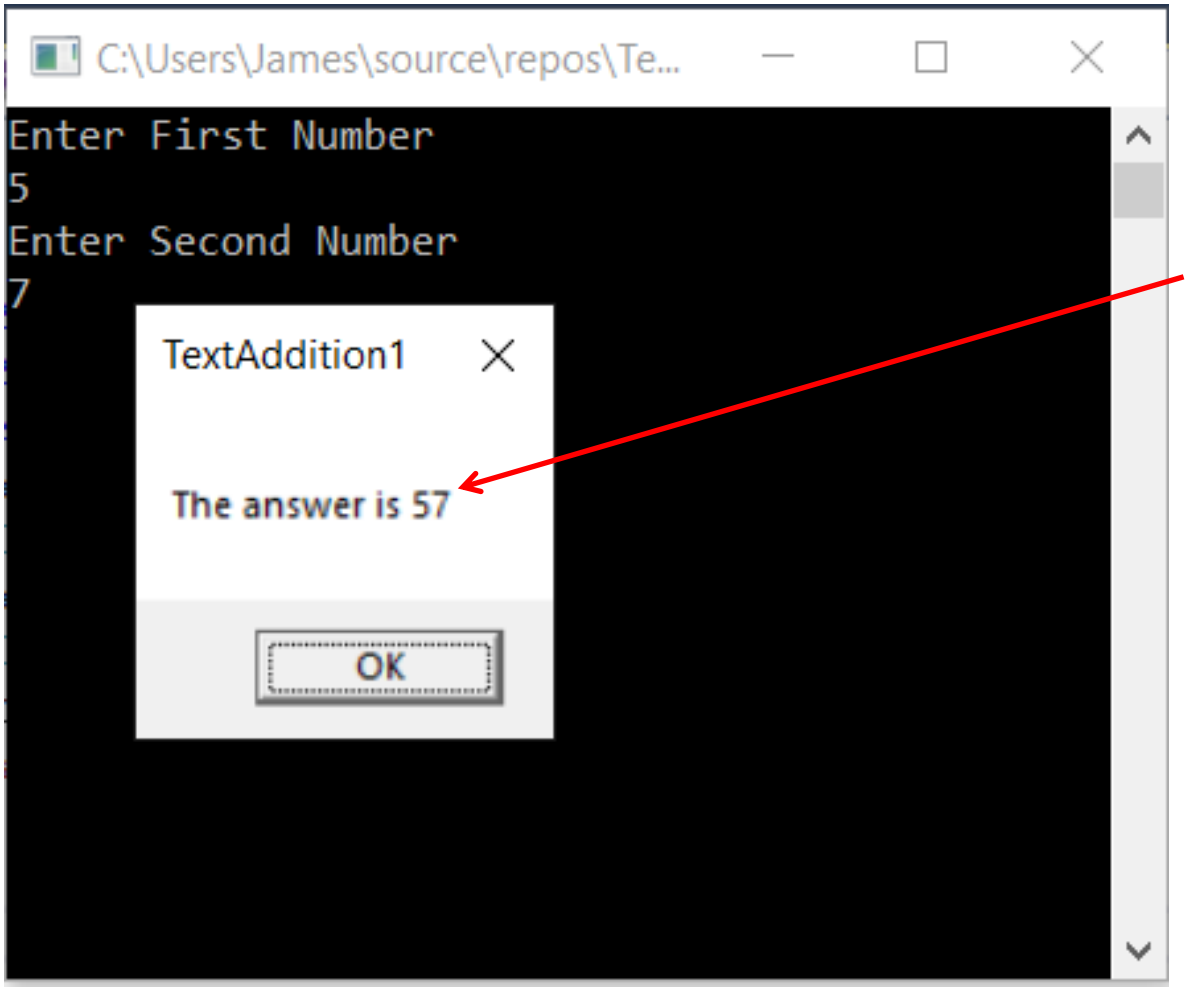
A Simple Calculator…

```
1   Module Module1
2
3       Sub Main()
4           Dim num1 As String
5           Dim num2 As String
6           Dim answer As String
7           Console.WriteLine("Enter First Number")
8           num1 = Console.ReadLine()
9           Console.WriteLine("Enter Second Number")
10          num2 = Console.ReadLine()
11          answer = num1 + num2
12          MsgBox("The answer is " & answer)
13
14      End Sub
15
16  End Module
```

…that doesn't work!



'Numbers' become concatenated instead of added

Strings just don’t ‘add up’

A String variable’s purpose is to hold strings of text.
Letters or text are not something we associate mathematical functions with.

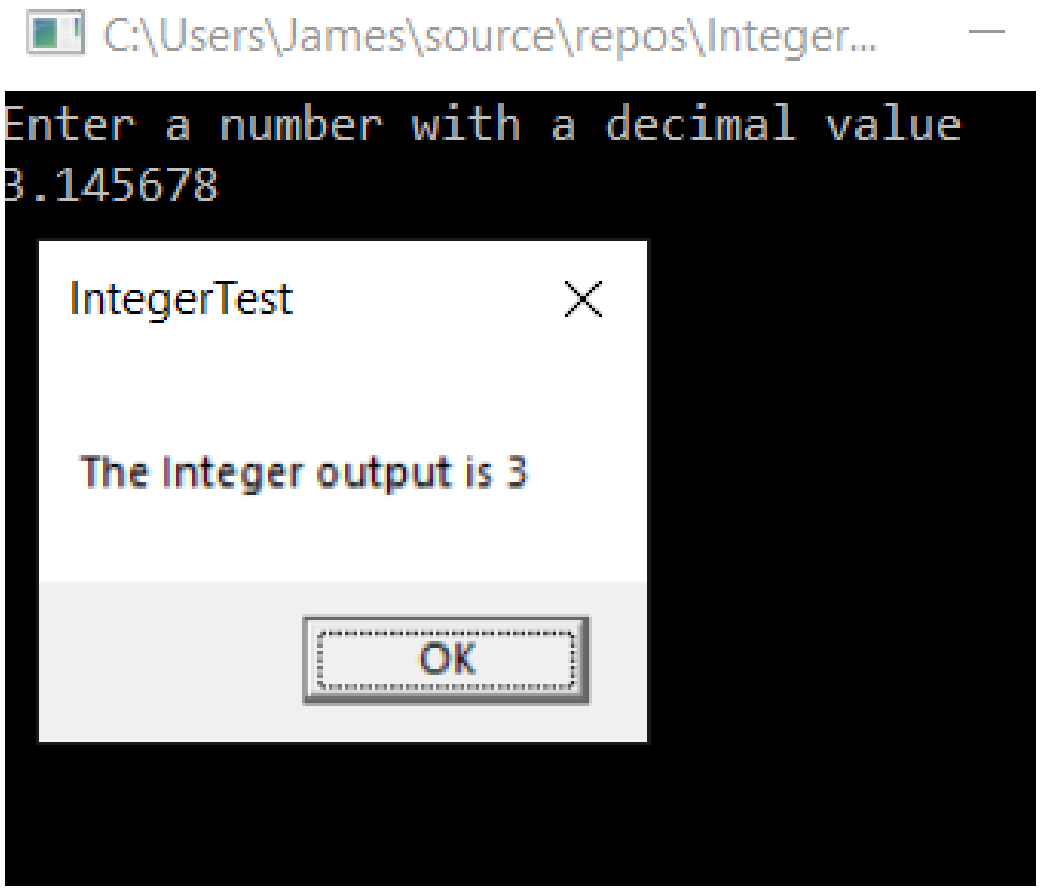
If we want to perform mathematical functions on this input, we need to store it in a variable that is better suited to hold numbers.

Integers to the Rescue

Integers are variables that are able to store whole numbers.
These variables can than be modified and manipulated exactly the same way
a number could be.

Note: As the name implies, Integer variables can only store whole numbers and can not contain any decimal values.

Truncation



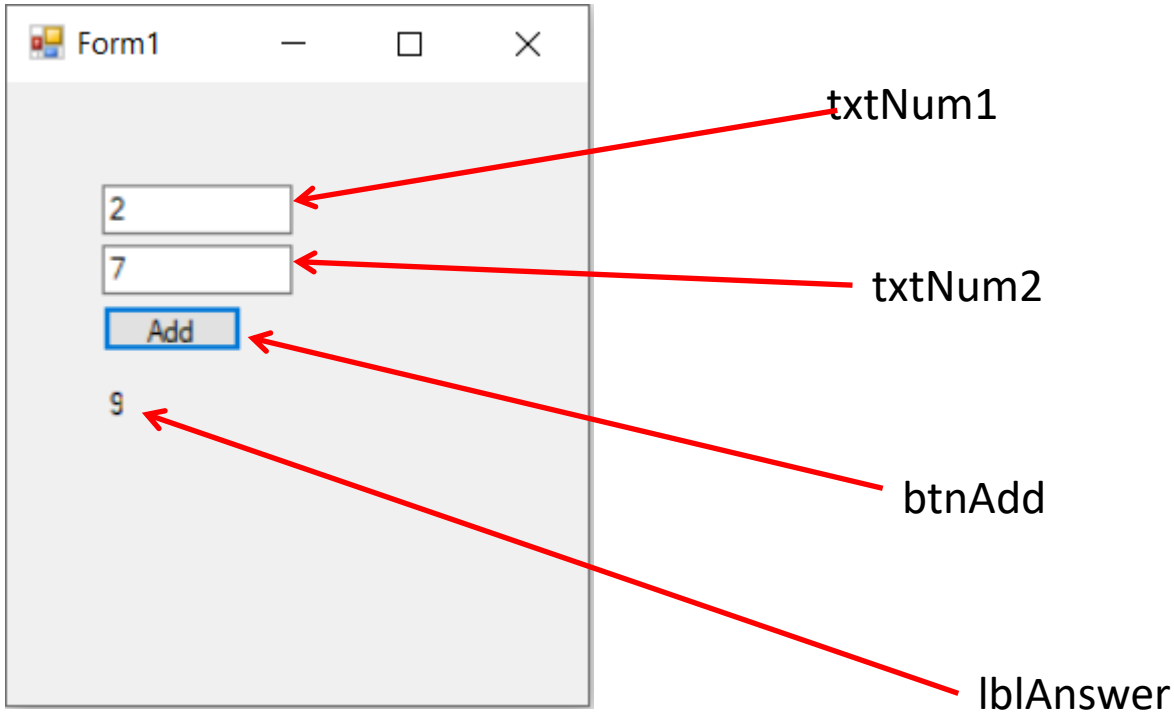
Every value after the decimal place is cut off or truncated

Addition With Integers

TextAddition1 — Module1

```
1   Module Module1
2
3       Sub Main()
4           Dim num1 As Integer
5           Dim num2 As Integer
6           Dim answer As Integer
7           Console.WriteLine("Enter First Number")
8           num1 = Console.ReadLine()
9           Console.WriteLine("Enter Second Number")
10          num2 = Console.ReadLine()
11          answer = num1 + num2
12          MsgBox("The answer is " & answer)
13      End Sub
14
15  End Module
16
```

FirstGUICalculator



FirstGUICalculator Code

```
1   Public Class Form1
2       Private Sub btnAdd_Click(sender As Object, e As EventArgs) Handles btnAdd.Click
3           Dim num1 As Integer
4           Dim num2 As Integer
5           Dim answer As Integer
6           num1 = txtNum1.Text
7           num2 = txtNum2.Text
8           answer = num1 + num2
9           lblAnswer.Text = answer
10      End Sub
11  End Class
12
```

A Better Way

The Visual Basic language can be very forgiving, at times, too much so.
One example is where we are putting numbers into the text boxes in our program and VB is ‘smart’ enough to realize that these are actually numbers.

Even though we can get away with it, it is not a good practice to allow the running of our programs on this kind of ‘luck’.
From now on, whenever we are putting numbers into a text box, we shall add the following bit of code to format our code properly.

The Val Function

FirstGUICalculator | Form1 | InitializeComponent

```
1   Public Class Form1
2       Private Sub btnAdd_Click(sender As Object, e As EventArgs) Handles btnAdd.Click
3           Dim num1 As Integer
4           Dim num2 As Integer
5           Dim answer As Integer
6           num1 = Val(txtNum1.Text)
7           num2 = Val(txtNum2.Text)
8           answer = num1 + num2
9           lblAnswer.Text = answer
10      End Sub
11  End Class
```

This code translates as follows…whatever value that is typed into the TextBox will be converted into a numeric value.
If it can not be identified as a numeric value, it will be interpreted as a zero.