We are starting with funcinVB.
I added in the MsgBox to track our code.
```vbnet
Public Class frmFuncConvert
    Inherits System.Windows.Forms.Form

    #Windows Form Designer generated code

    Private Sub btnAdr.Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAdr.Click
        Dim wkCSI As String, wkComma As Integer, wkCityLen As Integer
        Dim wkState As Integer, wkState1 As Integer
        wkCSI = InputBox("Enter City followed by a comma and space then State space Zip", "Address")
        wkComma = InStr(wkCSI, ",")
        MsgBox(wkComma)
        wkLen = Left(wkCSI, wkComma - 1)
        MsgBox(wkLen)
        txtCity.Text = Mid(wkCSI, wkLen + 1, 1)
        txtCityLen = Len(wkCSI) - (txtCity.Text.Length + 1)
        txtCity.Text = Microsoft.VisualBasic.Left(wkCSI, wkCityLen)
        txtState.Text = Mid(wkCSI, wkLen + 1, 1)
        wkState = InStr(wkCSI, " ", 1, vbTextCompare)
        MsgBox(wkState)
        txtState.Text = Mid(wkCSI, wkLen + 1, 1)
        wkState1 = InStr(wkCSI, ",", 1, vbTextCompare)
        MsgBox(wkState1)
        txtZip.Text = Mid(wkCSI, wkLen + 1, 1)
    End Sub
End Class
```
Public Class frmFuncConvert
    Inherits System.Windows.Forms.Form

    Private Sub btnAddr_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddr.Click
        Dim wkCSZ As String, wkComma As Integer, wkCityLen As Integer
        Dim wki As Integer, wkCSZ As String
        wkCSZ = InputBox("Enter City followed by a comma and space then State space Zip", "Address")
        wkComma = InStr(wkCSZ, ",")
        MsgBox(wkComma)
        wkCityLen = wkCSZ.Length
        MsgBox(wkCityLen)
        wkCityLen = wkComma - 1
        MsgBox(wkCityLen)
        txtCity1.Text = wkCSZ.Substring(0, wkCityLen)
        txtState.Text = wkCSZ.Substring(wkComma + 1, 2)
        txtState = wkCSZ.Substring(wkComma + 2)
        MsgBox(wkState)
        txtZip.Text = wkCSZ.Substring(wkComma + 3, 5)
        txtZip.Text = Microsoft.VisualBasic.Right(wkCSZ, 5)
    End Sub

    End Class
Starting functions you can write.
The function is named and the data to pass is in parenthesis on the right of the assignment sign.
The place to receive the results from the function are on the left.
Procedures

```
Public Class FirstProc
    Private Sub btnGreet_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGreet.Click
        Dim myDate As Date
        myDate = Now()
        lstGreet.Items.Add("Hello, it is " & myDate)
    End Sub

    Public Sub toString()
        Dim myDate As Date
        myDate = Now()
        lstGreet.Items.Add("Hello, it is " & myDate)
    End Sub

    Public Sub ToTotal()
        Dim wkTotal As Double
        Dim wkRate As Double
        wkRate = InputBox("Enter the amount of the check", "Total of Checks")
        wkTotal = wkRate * wkTotal
        lstTotal.Items.Add("The total is " & wkTotal)
    End Sub

    Private Sub btnTotal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnTotal.Click
        lstTotal.Items.Add("The total is " & wkTotal)
    End Sub

    Private Sub btnPassByRef_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPassByRef.Click
        Dim wkAmount As Single
        Dim wkRate As Single
        Dim wkNum As String
        wkAmount = InputBox("Enter the amount of the transaction", "Transaction Amount")
        wkRate = InputBox("Enter the account you are using" "Percent")
    End Sub
```

Hello, it is 2/28/2013 12:22 PM

The total is 4
The total is 6
The total is 12
Public Class firstProc
  Private Sub btnGreet_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGreet.Click
    End Sub
  
  Public Sub Todreet()
    Dim wDate As Date
    wDate = Now()
    lstDays.Items.Add(“Hello, it is " & wDate)
    End Sub
  
  Public Sub Tototal()
    Static wkTotal As Double
    ‘Now comment out static and use this line
    ‘Dim wkTotal As Double
    Dim wkAs Double
    wkAs = InputBox(“Enter the amount of the check”, “Total of Checks”)  
    wkTotal = wkTotal + wkAs
    lstDays.Items.Add(“The total is " & wkTotal)
    End Sub
  
  Private Sub btnTotal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnTotal.Click
    End Sub
  
  Private Sub btnPassByRef_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPassByRef.Click
    Dim wkAs Single
    Dim wkStr As String
    wkStr = InputBox(“Enter the amount of the transaction”, “Transaction Amount”)
    wkAs = CSingle(wkStr)
    lstDays.Items.Add(“Amount: " & wkAs)  
  End Sub
End Class
Be sure to try this program and see the difference with pass by val
ByVal
There is another program that does pass by reference - check it out!
I missed some slides so I am rerunning at home to show ByVal and byRef.
Used 1000 and .2 again.

ByRef references the original while ByVal works on a copy.
Argument Passing ByVal and ByRef

Variable and Nonvariable Arguments

The programming elements underlying an argument can be either a variable element, capable of having its value changed, or a nonvariable element. The following lists key variables:

Variable elements:

- Declared variables, including object variables
- Methods and subroutines
- Arrays
- Structures

Nonvariable elements:

- Constants
- Literals
- Enumerations
- Expressions

Nonvariable arguments are never modified in the calling code, even if they are passed by value. The called procedure manipulates a copy of such an argument, but the modification does not affect the underlying element in the calling code.