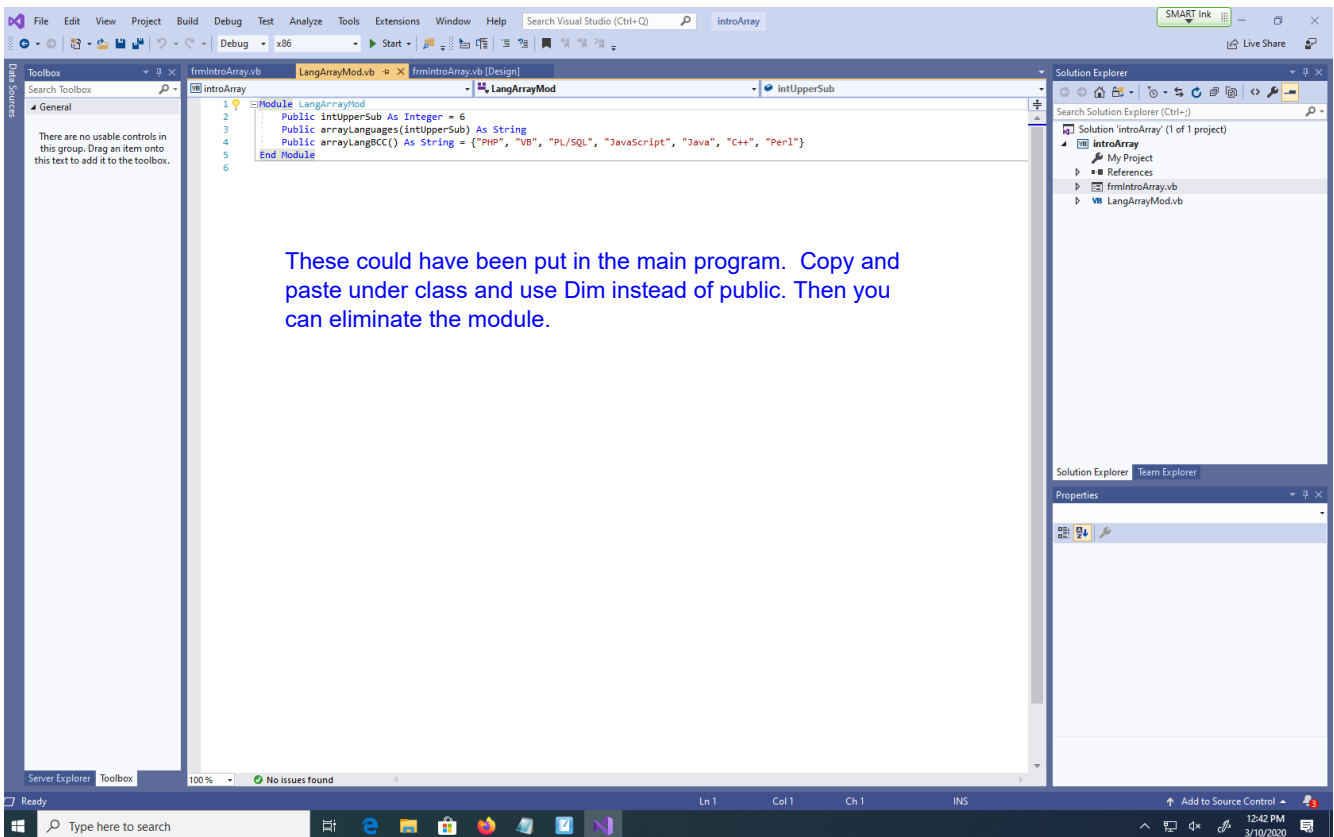


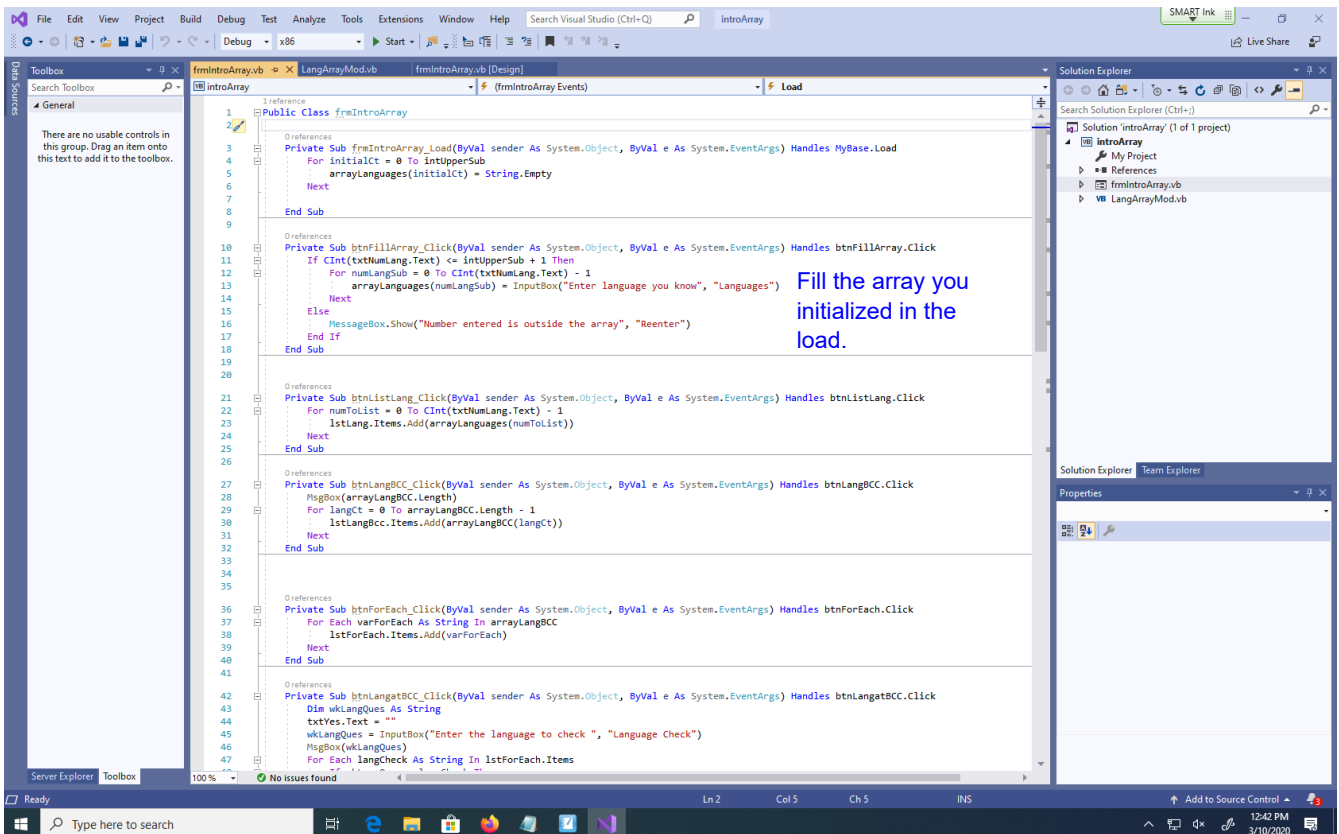
Examples using VB 2010

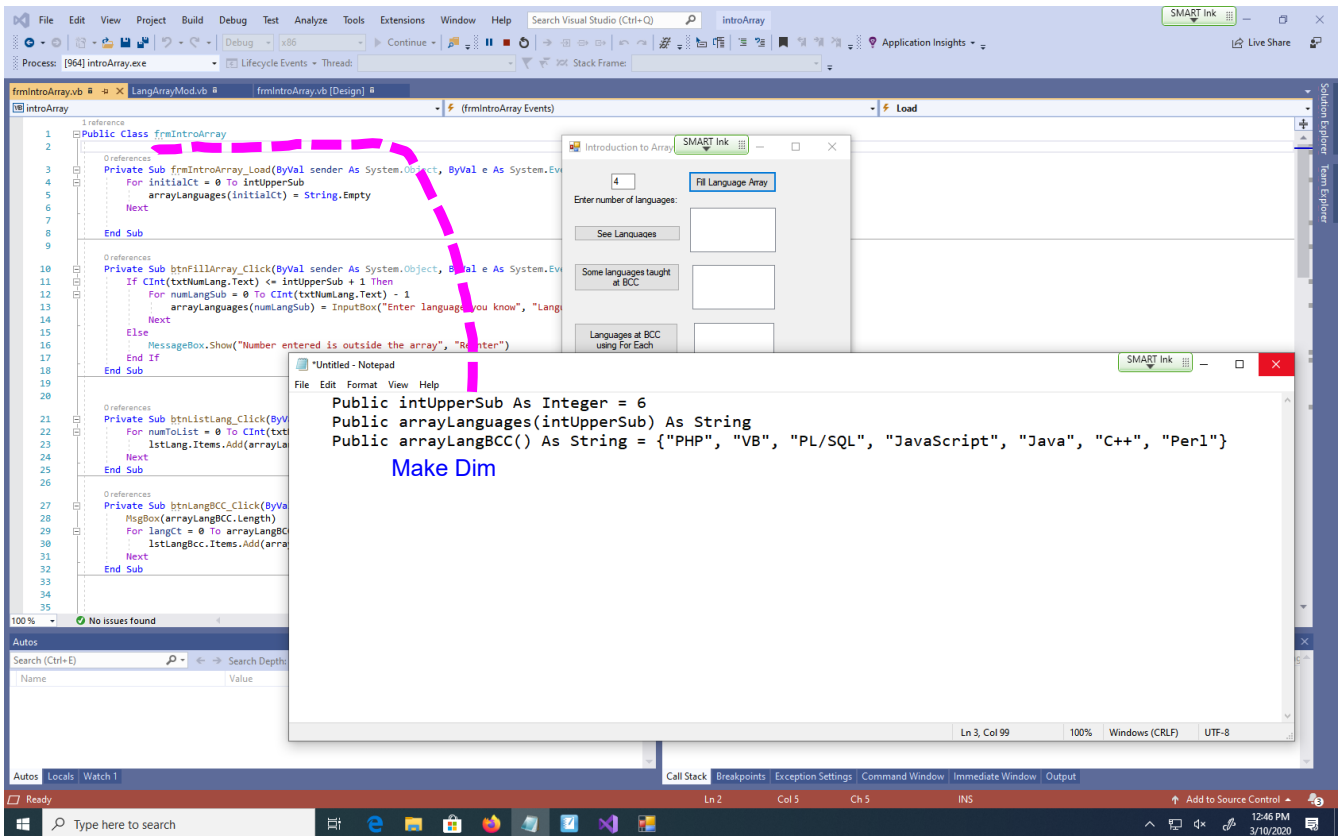
www.pgrocer.net/Cis56/VB2010examples.html

Examples using VB 2012/2010

| | |
|---|--|
| Examples | Click on links to retrieve |
| Payroll Calculation (intro) | PayCalc.zip |
| Basic Project (chap #2) | cisBasicProj.zip |
| Variables and input (chap #3) | introCalcProj.zip varCalcProj.zip introExceptionProj.zip useGroupProg.zip projMath.zip |
| Decisions (chap #4) | projIntroIF.zip otherDecisions.zip projRadioCheck.zip inputVal.zip |
| Lists and Loops (chap #5) | useBoxes.zip basicLoops.zip diffBoxes.zip GuessGames2010.zip |
| Procedures and functions (chap #6) | firstProcedures.zip firstFunctions.zip |
| VB functions | VBfunctions.zip |
| Multiple forms, modules and menus (chap #7) | basicMultiForms.zip menuGrades.zip |
| Arrays (chap #8) | introArray.zip calcArray.zip moreArrays.zip |
| Files (chap #9) | ReadVB2010.zip writeVB2010.zip |
| Database examples using SQLServer | Basic VBGrid bindonclick DonorFormSQL DonorTwoTables TwoGrids combinewithSQL ConvertedtoSQLServer contains all of the examples above |
| ADO using Access 2010 - writing code | DonorFormSQL12.zip OR DonorFormSQL12.zip bookADO.zip OR bookADO.zip ADOCurr12.zip OR ADOcurr12.zip VBGrid12.zip OR VBGrid12.zip bindonclick12.zip combinewithSQL.zip SQLdiffcode12.zip DonorTwoTables12.zip |







The screenshot displays the Visual Studio IDE with a VB.NET project named 'frmIntroArray'. The code in the main window includes three subroutines: `btnForEach_Click`, `btnLangatBCC_Click`, and `btnSearchLang_Click`. The `btnLangatBCC_Click` subroutine is currently active, showing a message box for language selection and a loop for checking each language. A pink arrow points from the `MsgBox(wkLangQues)` line to a 'Language Check' dialog box. This dialog box has a title bar 'Language Check', a label 'Enter the language to check', and a text box containing 'C++'. To the right, an 'Introductions SMART link' dialog box is open, showing a 'Fill Language Array' section with a list of languages (VB, PHP, Perl, PL/SQL) and a 'Languages at BCC using For Each' section with a list (PHP, VB, PL/SQL, JavaScript, Java). The status bar at the bottom indicates '100%' zoom and 'No issues found'.

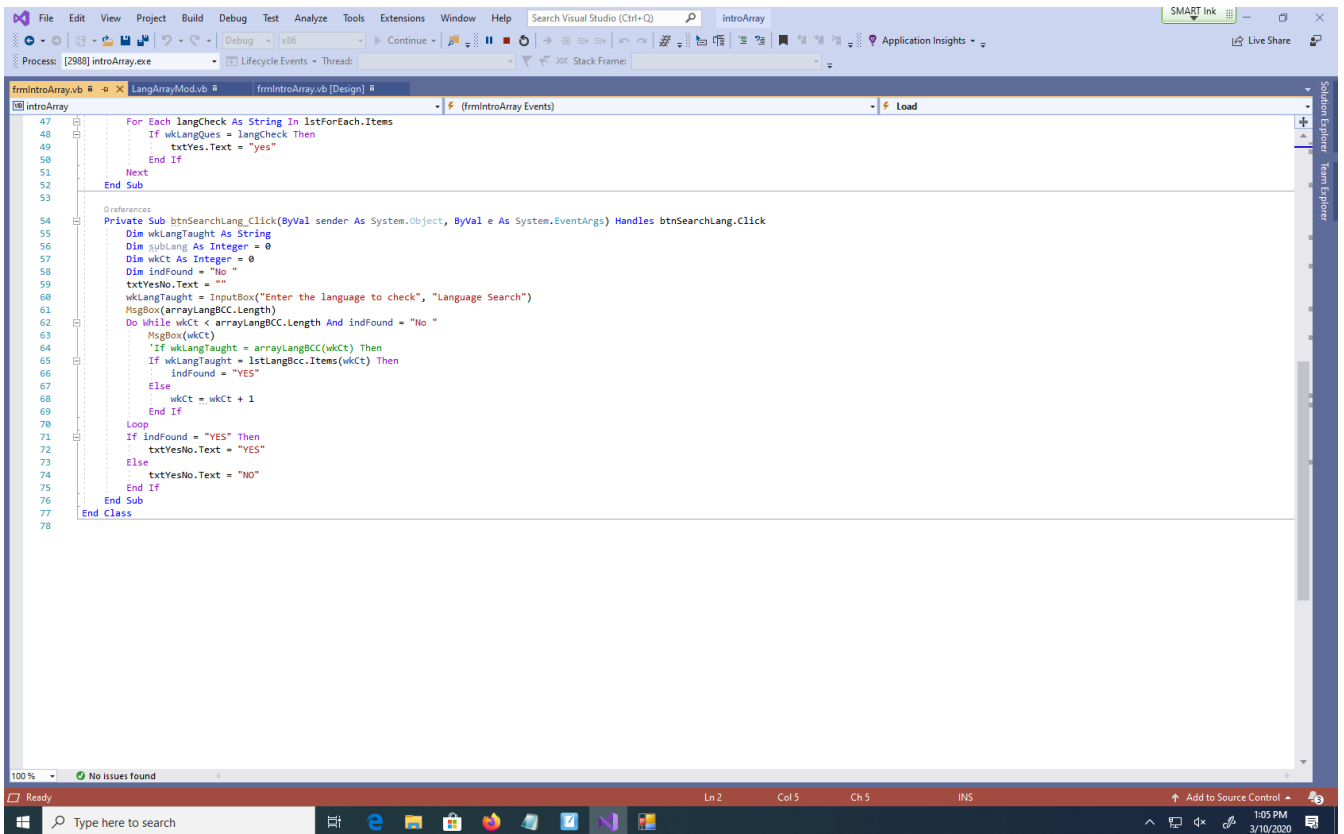
The screenshot displays the Visual Studio IDE with a VB.NET project named 'IntroArray'. The code in the 'LangArrayMod.vb' file includes three event handlers: 'btnForEach_Click', 'btnLangatBCC_Click', and 'btnSearchLang_Click'. A dialog box titled 'Introduce SMART Ink' is overlaid on the code. The dialog has a text input field containing '4' and a 'Fill Language Array' button. Below this, there are two lists of languages: 'Some languages taught at BCC' (VB, Php, Perl, PL/SQL) and 'Languages at BCC using For Each' (PHP, VB, PL/SQL, JavaScript, Java). A checkbox labeled 'Is language taught at BCC for each?' is checked, and a 'Search Language Taught at BCC?' button is at the bottom. A dashed pink line connects the dialog's 'Is language taught at BCC for each?' checkbox to the 'If wkLangQuces = langCheck Then' line in the 'btnLangatBCC_Click' method.

```
33
34
35
36 Private Sub btnForEach_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnForEach.Click
37     For Each varForEach As String In arrayLangBCC
38         lstForEach.Items.Add(varForEach)
39     Next
40 End Sub
41
42 Private Sub btnLangatBCC_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLangatBCC.Click
43     Dim wkLangQuces As String
44     txtYes.Text = ""
45     wkLangQuces = InputBox("Enter the language to check ", "Language Check")
46     MsgBox(wkLangQuces)
47     For Each langCheck As String In lstForEach.Items
48         If wkLangQuces = langCheck Then
49             txtYes.Text = "yes"
50         End If
51     Next
52 End Sub
53
54 Private Sub btnSearchLang_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSearchLang.Click
55     Dim wkLangTaught As String
56     Dim subLang As Integer = 0
57     Dim wkCt As Integer = 0
58     Dim indFound = "No"
59     txtYesNo.Text = ""
60     wkLangTaught = InputBox("Enter the language to check", "Language Search")
61     MsgBox(arrayLangBCC.Length)
62     Do While wkCt < arrayLangBCC.Length And indFound = "No"
63         MsgBox(wkCt)
64         If wkLangTaught = arrayLangBCC(wkCt) Then
65             If wkLangTaught = lstLangBcc.Items(wkCt) Then
66                 indFound = "YES"
67             Else
68                 wkCt = wkCt + 1
69             End If
70         Loop
71         If indFound = "YES" Then
72             txtYesNo.Text = "yes"
73         Else
74             txtYesNo.Text = "No"
75         End If
76     End Sub
77 End Class
78
```

The screenshot displays the Visual Studio IDE with a VB.NET project named 'introArray'. The code in the main window includes three subroutines: `btnForEach_Click`, `btnLangatBCC_Click`, and `btnSearchLang_Click`. The `btnSearchLang_Click` subroutine contains a loop that iterates through an array of languages, checking if each language is present in a list of languages taught at BCC. A pink dashed arrow points from the `MsgBox` call in the code to a dialog box titled 'Introducti SMART Ink'. The dialog box has a title bar with 'SMART Ink' and a close button. It contains a text box with the number '4', a 'Fill Language Array' button, and a list box showing 'VB', 'PHP', 'Perl', and 'PL/SQL'. Below this is a 'See Languages' button. There are two scrollable lists: 'Some languages taught at BCC' (containing 'PHP', 'VB', 'PL/SQL', 'JavaScript') and 'Languages at BCC using For Each' (containing 'PHP', 'VB', 'PL/SQL', 'JavaScript', 'Java'). At the bottom, there is a text box with 'Search: Language Taught at BCC?' and a 'NO' button.

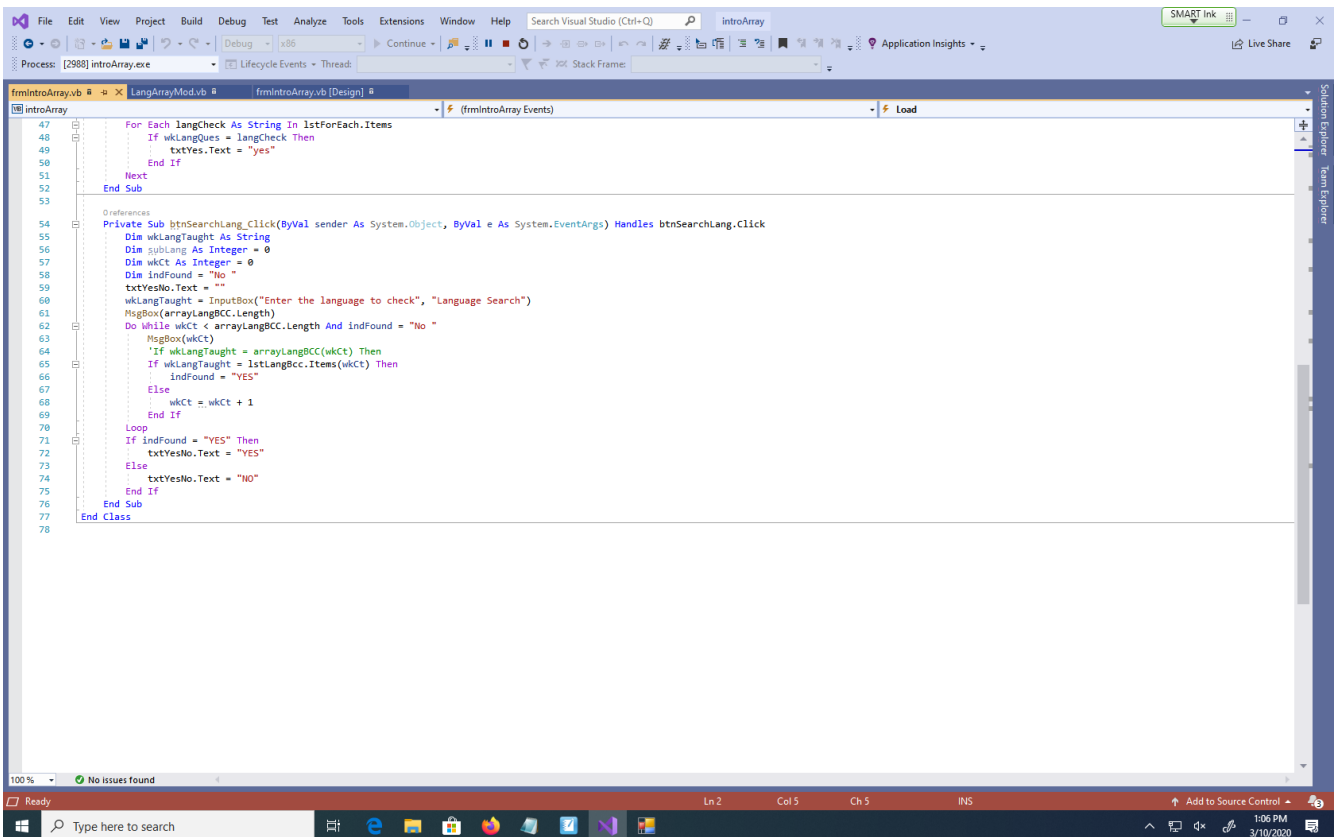
The screenshot shows the Visual Studio IDE with a VB.NET project named 'frmIntroArray'. The code in 'LangArrayMod.vb' defines a 'Public Class frmIntroArray' with several event handlers. The first handler, 'frmIntroArray_Load', uses a 'For' loop to initialize 'arrayLanguages' with 'String.Empty' for 'initialCt' from 0 to 'intUpperSub'. The second handler, 'btnFillArray_Click', uses an 'If' statement and a 'For' loop to populate 'arrayLanguages' based on 'txtNumLang.Text'. The third handler, 'btnListLang_Click', uses a 'For' loop to add items to 'lstLang.Items'. The fourth handler, 'btnLangBCC_Click', uses a 'For' loop to add items to 'lstLangBcc.Items'. The fifth handler, 'btnForEach_Click', uses a 'For Each' loop to add items to 'lstForEach.Items'. The sixth handler, 'btnLangatBCC_Click', uses a 'Dim' statement and an 'InputBox' to get a language to check. A blue text overlay 'Examples of For and For Each' is positioned in the center of the code editor.

```
1 Public Class frmIntroArray
2
3     Private Sub frmIntroArray_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
4         For initialCt = 0 To intUpperSub
5             arrayLanguages(initialCt) = String.Empty
6         Next
7     End Sub
8
9
10    Private Sub btnFillArray_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnFillArray.Click
11        If cint(txtNumLang.Text) <= intUpperSub + 1 Then
12            For numLangSub = 0 To Cint(txtNumLang.Text) - 1
13                arrayLanguages(numLangSub) = InputBox("Enter language you know", "Languages")
14            Next
15        Else
16            MessageBox.Show("Number entered is outside the array", "Reenter")
17        End If
18    End Sub
19
20
21    Private Sub btnListLang_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnListLang.Click
22        For numToIist = 0 To Cint(txtNumLang.Text) - 1
23            lstLang.Items.Add(arrayLanguages(numToIist))
24        Next
25    End Sub
26
27    Private Sub btnLangBCC_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLangBCC.Click
28        MsgBox(arrayLangBCC.Length)
29        For langCt = 0 To arrayLangBCC.Length - 1
30            lstLangBcc.Items.Add(arrayLangBCC(langCt))
31        Next
32    End Sub
33
34
35
36    Private Sub btnForEach_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnForEach.Click
37        For Each varForEach As String In arrayLangBCC
38            lstForEach.Items.Add(varForEach)
39        Next
40    End Sub
41
42    Private Sub btnLangatBCC_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLangatBCC.Click
43        Dim wkLangQues As String
44        txtYes.Text = ""
45        wkLangQues = InputBox("Enter the language to check ", "Language Check")
46        MsgBox(wkLangQues)
```

The screenshot shows the Visual Studio IDE with a VB.NET project named 'introArray'. The code is displayed in the main editor window, showing a class with a 'Load' event handler and a 'btnSearchLang_Click' method. The 'Load' event handler iterates through a list of language checks. The 'btnSearchLang_Click' method prompts the user to enter a language to check, then searches for it in a list of languages. It uses a 'Do While' loop to iterate through the list until the language is found or the end is reached. The results are displayed in text boxes.

```
47 For Each langCheck As String In lstForEach.Items
48     If wkLangQueues = langCheck Then
49         txtYes.Text = "yes"
50     End If
51 Next
52 End Sub
53
54 @references:
55 Private Sub btnSearchLang_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSearchLang.Click
56     Dim wkLangTaught As String
57     Dim subLang As Integer = 0
58     Dim wkCt As Integer = 0
59     Dim indFound = "No "
60     txtYesNo.Text = ""
61     wkLangTaught = InputBox("Enter the language to check", "Language Search")
62     MsgBox(arrayLangBCC.Length)
63     Do While wkCt < arrayLangBCC.Length And indFound = "No "
64         MsgBox(wkCt)
65         If wkLangTaught = arrayLangBCC(wkCt) Then
66             If wkLangTaught = lstLangBCC.Items(wkCt) Then
67                 indFound = "YES"
68             Else
69                 wkCt = wkCt + 1
70             End If
71         Loop
72     If indFound = "YES" Then
73         txtYesNo.Text = "YES"
74     Else
75         txtYesNo.Text = "NO"
76     End If
77 End Sub
78 End Class
```



The screenshot shows the Visual Studio IDE with a VB.NET project named 'IntroArray'. The code is in the 'frmIntroArray.vb' file, which is in Design view. The code defines a class 'frmIntroArray' with a 'Load' event handler and a 'btnSearchLang_Click' event handler. The 'Load' event handler iterates through a list of language checks, displaying 'yes' for each. The 'btnSearchLang_Click' event handler prompts the user to enter a language to check, then searches through an array of languages ('arrayLangBCC') to find a match. If found, it displays 'YES'; otherwise, it displays 'NO'.

```
47 For Each langCheck As String In lstForEach.Items
48     If wkLangQues = langCheck Then
49         txtYes.Text = "yes"
50     End If
51 Next
52 End Sub
53
54 References
55 Private Sub btnSearchLang_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSearchLang.Click
56     Dim wkLangTaught As String
57     Dim subLang As Integer = 0
58     Dim wkCt As Integer = 0
59     Dim indFound = "No "
60     txtYesNo.Text = ""
61     wkLangTaught = InputBox("Enter the language to check", "Language Search")
62     MsgBox(arrayLangBCC.Length)
63     Do While wkCt < arrayLangBCC.Length And indFound = "No "
64         MsgBox(wkCt)
65         If wkLangTaught = arrayLangBCC(wkCt) Then
66             If wkLangTaught = IstLangBcc.Items(wkCt) Then
67                 indFound = "YES"
68             Else
69                 wkCt = wkCt + 1
70             End If
71         Loop
72         If indFound = "YES" Then
73             txtYesNo.Text = "YES"
74         Else
75             txtYesNo.Text = "NO"
76         End If
77     End Sub
78 End Class
```

The screenshot displays the Visual Studio IDE with a VB.NET project named 'frmCalcArray.vb'. The code defines a class 'frmCalcArray' with several methods: 'frmCalcArray_Load' for initializing an array, 'btnShowArray_Click' for displaying array elements, 'btnTotal_Click' for calculating the sum, 'btnAvg_Click' for calculating the average, 'btnHigh_Click' for finding the maximum value, and 'btnLow_Click' for finding the minimum value. A 'Calculate' dialog box is open, showing the results of these operations for the array [12.34, 23.45, 34.56, 45.67, 56.78, 67.89].

```
Public Class frmCalcArray
    Dim numArray(5) As Single
    Dim wkTotal As Double = 0

    Private Sub frmCalcArray_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        numArray(0) = 12.34
        numArray(1) = 23.45
        numArray(2) = 34.56
        numArray(3) = 45.67
        numArray(4) = 56.78
        numArray(5) = 67.89
    End Sub

    Private Sub btnShowArray_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnShowArray.Click
        For Each showNum As Single In numArray
            lstShowArray.Items.Add(showNum)
        Next
    End Sub

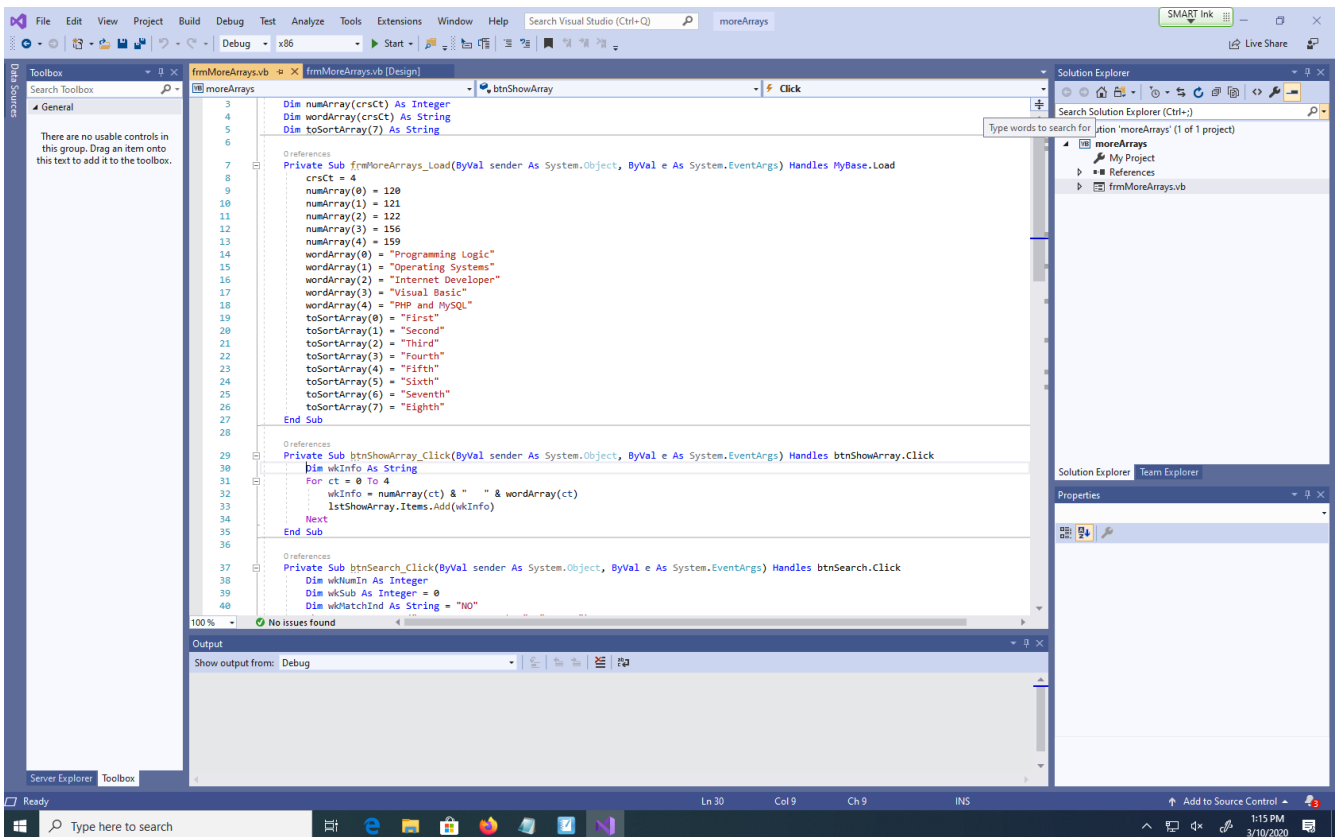
    Private Sub btnTotal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnTotal.Click
        For Each addNum As Single In numArray
            wkTotal = wkTotal + addNum
        Next
        txtTotal.Text = Format(wkTotal, "n2")
    End Sub

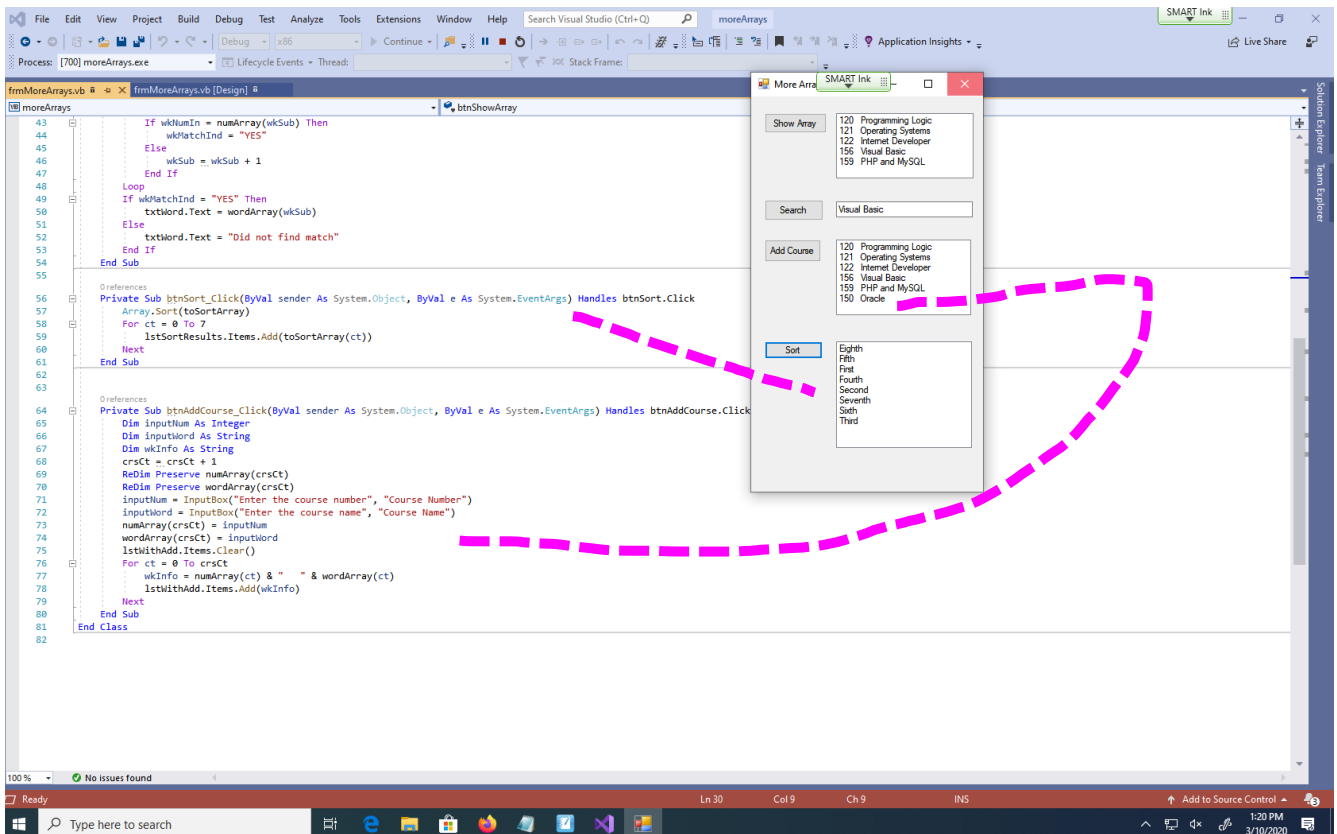
    Private Sub btnAvg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAvg.Click
        For Each aNum As Single In numArray
            wkTotal += aNum
        Next
        MsgBox(numArray.Length)
        txtAvg.Text = Format(wkTotal / numArray.Length, "n2")
    End Sub

    Private Sub btnHigh_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnHigh.Click
        Dim wkHigh As Single = 0
        For forCt = 0 To numArray.Length - 1
            If numArray(forCt) > wkHigh Then
                wkHigh = numArray(forCt)
            End If
        Next
        txtHigh.Text = wkHigh
    End Sub

    Private Sub btnLow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLow.Click
        Dim wkLow As Single
        wkLow = numArray(0)
    End Sub
End Class
```

| | |
|--------------|--|
| Show Array | 12.34 23.45 34.56 45.67 56.78 67.89 |
| Show Total | 240.69 |
| Show Average | 80.23 |
| Highest | |
| Lowest | |



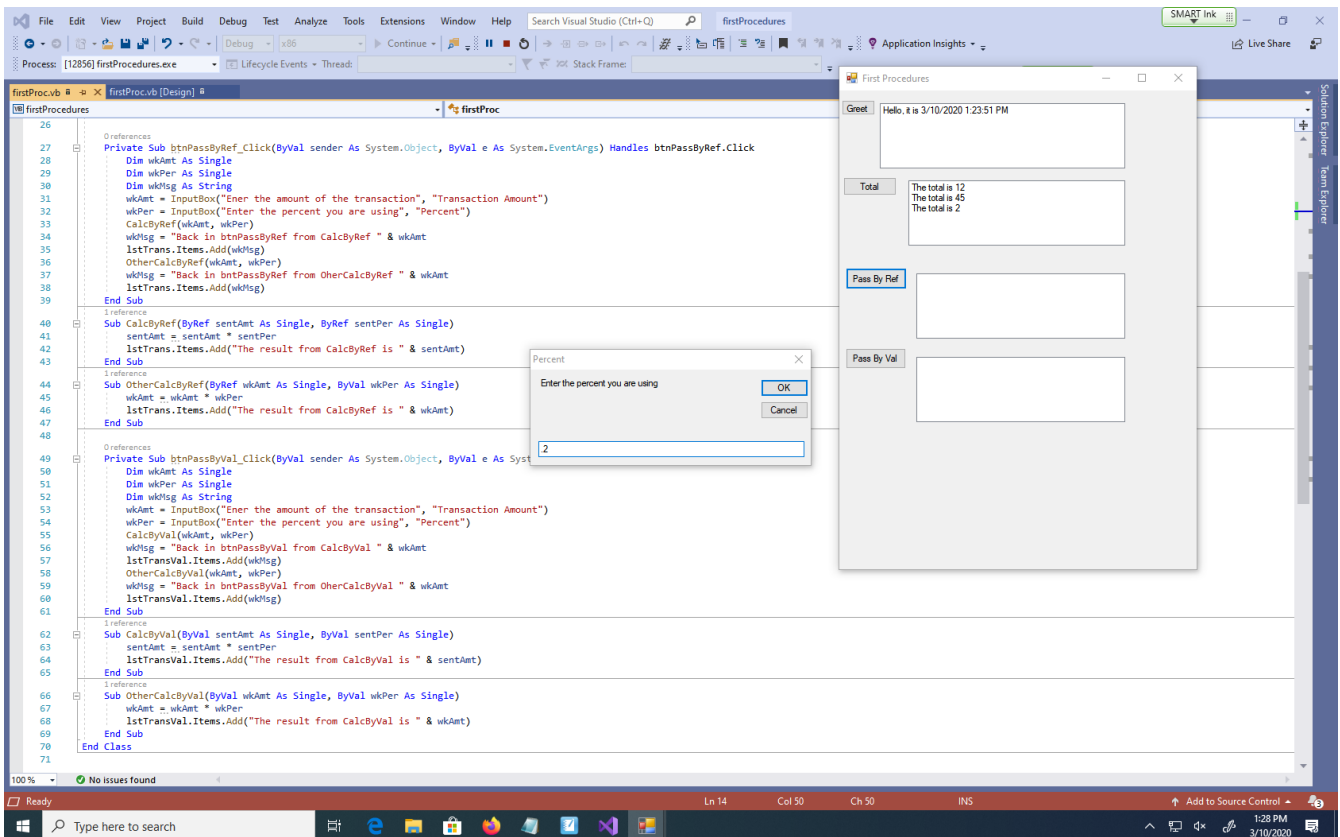


The screenshot displays the Visual Studio IDE with a C# project named 'firstProc'. The code in the 'firstProc.cs' file includes several methods demonstrating different parameter passing techniques: 'CalcByRef', 'OtherCalcByRef', 'CalcByVal', and 'OtherCalcByVal'. A 'Transaction Amount' dialog box is open, showing the input '1000'. The 'First Procedures' application window is also visible, displaying a 'Greet' message and a 'Total' calculation.

```
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71
```

Transaction Amount dialog box:
Enter the amount of the transaction
1000
OK Cancel

First Procedures application window:
Greet: Hello, it is 3/10/2020 1:23:51 PM
Total: The total is 12, The total is 45, The total is 2
Pass By Ref
Pass By Val



The screenshot displays a Visual Studio IDE with a C# project named 'firstProcedures'. The code in the 'firstProc' class is as follows:

```
26  
27 References:  
28 Private Sub btnPassByRef_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPassByRef.Click  
29     Dim wkAmt As Single  
30     Dim wkPer As Single  
31     Dim wkMsg As String  
32     wkAmt = InputBox("Enter the amount of the transaction", "Transaction Amount")  
33     wkPer = InputBox("Enter the percent you are using", "Percent")  
34     CalcByRef(wkAmt, wkPer)  
35     wkMsg = "Back in btnPassByRef from CalcByRef " & wkAmt  
36     lstTrans.Items.Add(wkMsg)  
37     OtherCalcByRef(wkAmt, wkPer)  
38     wkMsg = "Back in btnPassByRef from OtherCalcByRef " & wkAmt  
39     lstTrans.Items.Add(wkMsg)  
40 End Sub  
41  
42 Reference:  
43 Sub CalcByRef(ByRef sentAmt As Single, ByRef sentPer As Single)  
44     sentAmt = sentAmt * sentPer  
45     lstTrans.Items.Add("The result from CalcByRef is " & sentAmt)  
46 End Sub  
47  
48 Reference:  
49 Sub OtherCalcByRef(ByRef wkAmt As Single, ByVal wkPer As Single)  
50     wkAmt = wkAmt * wkPer  
51     lstTrans.Items.Add("The result from CalcByRef is " & wkAmt)  
52 End Sub  
53  
54 References:  
55 Private Sub btnPassByVal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPassByVal.Click  
56     Dim wkAmt As Single  
57     Dim wkPer As Single  
58     Dim wkMsg As String  
59     wkAmt = InputBox("Enter the amount of the transaction", "Transaction Amount")  
60     wkPer = InputBox("Enter the percent you are using", "Percent")  
61     CalcByVal(wkAmt, wkPer)  
62     wkMsg = "Back in btnPassByVal from CalcByVal " & wkAmt  
63     lstTransVal.Items.Add(wkMsg)  
64     OtherCalcByVal(wkAmt, wkPer)  
65     wkMsg = "Back in btnPassByVal from OtherCalcByVal " & wkAmt  
66     lstTransVal.Items.Add(wkMsg)  
67 End Sub  
68  
69 Reference:  
70 Sub CalcByVal(ByVal sentAmt As Single, ByVal sentPer As Single)  
71     sentAmt = sentAmt * sentPer  
72     lstTransVal.Items.Add("The result from CalcByVal is " & sentAmt)  
73 End Sub  
74  
75 Reference:  
76 Sub OtherCalcByVal(ByVal wkAmt As Single, ByVal wkPer As Single)  
77     wkAmt = wkAmt * wkPer  
78     lstTransVal.Items.Add("The result from CalcByVal is " & wkAmt)  
79 End Sub  
80  
81 End Class  
82
```

Handwritten annotations in red and blue highlight values: '1000' and '2' are written near line 32; '200' and '.2' are written near line 44; '40' is written near line 51. A blue arrow points from the 'Pass By Ref' button in the dialog box to the 'CalcByRef' method signature.

The 'First Procedures' dialog box shows the following output:

- Greet: Hello, it is 3/10/2020 1:23:51 PM
- Total: The total is 12, The total is 45, The total is 2
- Pass By Ref: The result from CalcByRef is 200, Back in btnPassByRef from CalcByRef 200, The result from CalcByRef is 40, Back in btnPassByRef from OtherCalcByRef 40
- Pass By Val: (empty)

The screenshot displays the Visual Studio IDE with a C# project named 'firstProcedures'. The code is divided into two main sections: 'btnPassByRef_Click' and 'btnPassByVal_Click'. Handwritten annotations in green and red ink are present throughout the code, including the values '1000', '200', and '0.2'. A blue arrow points from the 'CalcByRef' sub-routine to the 'Pass By Ref' section of the dialog box. The dialog box, titled 'First Procedures', contains a 'Greet' message, a 'Total' calculation, and two sections: 'Pass By Ref' and 'Pass By Val', each showing the results of calculations performed with different parameter passing methods.

```
26  
27 0 references  
28 Private Sub btnPassByRef_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPassByRef.Click  
29     Dim wkAmt As Single  
30     Dim wkPer As Single  
31     Dim wkMsg As String  
32     wkAmt = InputBox("Enter the amount of the transaction", "Transaction Amount")  
33     wkPer = InputBox("Enter the percent you are using", "Percent")  
34     CalcByRef(wkAmt, wkPer)  
35     wkMsg = "Back in btnPassByRef from CalcByRef " & wkAmt  
36     lstTrans.Items.Add(wkMsg)  
37     OtherCalcByRef(wkAmt, wkPer)  
38     wkMsg = "Back in btnPassByRef from OtherCalcByRef " & wkAmt  
39     lstTrans.Items.Add(wkMsg)  
40 End Sub  
41 1 reference  
42 Sub CalcByRef(ByRef sentAmt As Single, ByRef sentPer As Single)  
43     sentAmt = sentAmt * sentPer  
44     lstTrans.Items.Add("The result from CalcByRef is " & sentAmt)  
45 End Sub  
46 1 reference  
47 Sub OtherCalcByRef(ByRef wkAmt As Single, ByVal wkPer As Single)  
48     wkAmt = wkAmt * wkPer  
49     lstTrans.Items.Add("The result from CalcByRef is " & wkAmt)  
50 End Sub  
51 0 references  
52 Private Sub btnPassByVal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPassByVal.Click  
53     Dim wkAmt As Single  
54     Dim wkPer As Single  
55     Dim wkMsg As String  
56     wkAmt = InputBox("Enter the amount of the transaction", "Transaction Amount")  
57     wkPer = InputBox("Enter the percent you are using", "Percent")  
58     CalcByVal(wkAmt, wkPer)  
59     wkMsg = "Back in btnPassByVal from CalcByVal " & wkAmt  
60     lstTransVal.Items.Add(wkMsg)  
61     OtherCalcByVal(wkAmt, wkPer)  
62     wkMsg = "Back in btnPassByVal from OtherCalcByVal " & wkAmt  
63     lstTransVal.Items.Add(wkMsg)  
64 End Sub  
65 1 reference  
66 Sub CalcByVal(ByVal sentAmt As Single, ByVal sentPer As Single)  
67     sentAmt = sentAmt * sentPer  
68     lstTransVal.Items.Add("The result from CalcByVal is " & sentAmt)  
69 End Sub  
70 1 reference  
71 Sub OtherCalcByVal(ByVal wkAmt As Single, ByVal wkPer As Single)  
72     wkAmt = wkAmt * wkPer  
73     lstTransVal.Items.Add("The result from CalcByVal is " & wkAmt)  
74 End Sub  
75 End Class
```

Pass By Ref
The result from CalcByRef is 200
Back in btnPassByRef from CalcByRef 200
The result from CalcByRef is 40
Back in btnPassByRef from OtherCalcByRef 40

Pass By Val
The result from CalcByVal is 200
Back in btnPassByVal from CalcByVal 1000
The result from CalcByVal is 200
Back in btnPassByVal from OtherCalcByVal 1000