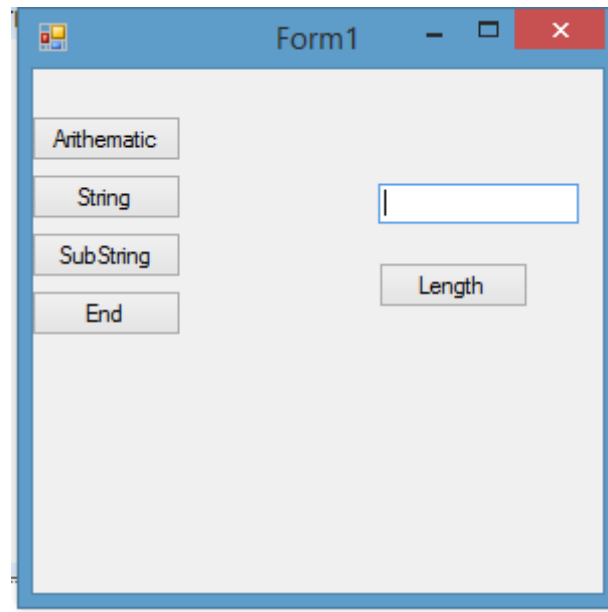


Arithmetic



```
Public Class Form1

    Private Sub btnArithmetic_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnArithmetic.Click
        'Declare variable
        Dim intNumber As Integer
        'Set number, add numbers, and display results
        intNumber = 16
        intNumber = intNumber + 8
        MessageBox.Show("Addition test. " & intNumber.ToString, "Integer Math")
        'Set number, subtract numbers, and display results
        intNumber = 24
        intNumber = intNumber - 2
        MessageBox.Show("Subtraction test. " & intNumber.ToString, "Integer Math")
        'Set number, multiply numbers, and display results
        intNumber = 6
        intNumber = intNumber * 10
        MessageBox.Show("Multiplication test. " & intNumber.ToString, "Integer Math")
        'Set number, divide numbers, and display results
        intNumber = 12
        intNumber = CType(intNumber / 6, Integer)
        MessageBox.Show("Division test. " & intNumber.ToString, "Integer Math")
    End Sub

    Private Sub btnString_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnString.Click
        'Declare variable
        Dim strResults As String
        'Set the string value
```

```

        strResults = "Hello World!"
        'Display the results
        MessageBox.Show(strResults, "Strings")
    End Sub

    Private Sub btnSubString_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnSubString.Click
        'Declare variable
        Dim strResults As String
        'Get the text from the TextBox
        strResults = txtStrings.Text
        'Display the first three characters
        MessageBox.Show(strResults.Substring(0, 3), "Strings")
        'Display the middle three characters
        MessageBox.Show(strResults.Substring(3, 3), "Strings")
        'Display the last three characters
        MessageBox.Show(strResults.Substring(strResults.Length - 3),
"Strings")
    End Sub

    Private Sub btnEnd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEnd.Click
        End
    End Sub

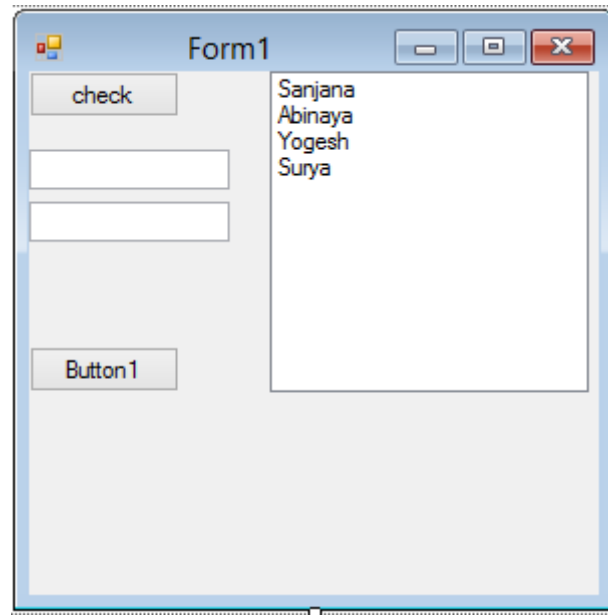
    Private Sub btnLength_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLength.Click
        'Declare variable
        Dim strResults As String
        'Get the text from the TextBox
        strResults = txtStrings.Text
        'Display the length of the string
        MessageBox.Show(strResults.Length.ToString & " characters(s)",
"Strings")
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        End Sub
End Class

```

Decision



```
Public Class Form1
```

```
    Private Sub btnCheck_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles btnCheck.Click
```

```
        'Declare variables
```

```
        Dim strName1 As String, strName2 As String
```

```
        'Get the names
```

```
        strName1 = txtIF1.Text
```

```
        strName2 = txtIF2.Text
```

```
        'Is one of the names Wendy?
```

```
        If strName1 = "VIT" Or strName2 = "VIT" Then
```

```
            MessageBox.Show("One of the names is VIT.", "And Or Demo")
```

```
        Else
```

```
            MessageBox.Show("Neither of the names is VIT.", "And Or Demo")
```

```
        End If
```

```
    End Sub
```

```
    Private Sub lstData_SelectedIndexChanged(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles lstData.SelectedIndexChanged
```

```
        'Declare variables
```

```
        Dim strName As String
```

```
        Dim strFavoriteColor As String
```

```
        'Get the selected name
```

```
        strName = lstData.Items(lstData.SelectedIndex).ToString
```

```
        'Use a Select Case statement to get the favorite color
```

```
        'of the selected name
```

```
        Select Case strName
```

```
            Case "Sanjana"
```

```
                strFavoriteColor = "Pink"
```

```
            Case "Abinaya"
```

```
                strFavoriteColor = "Sea Blue"
```

```
            Case "Yogesh"
```

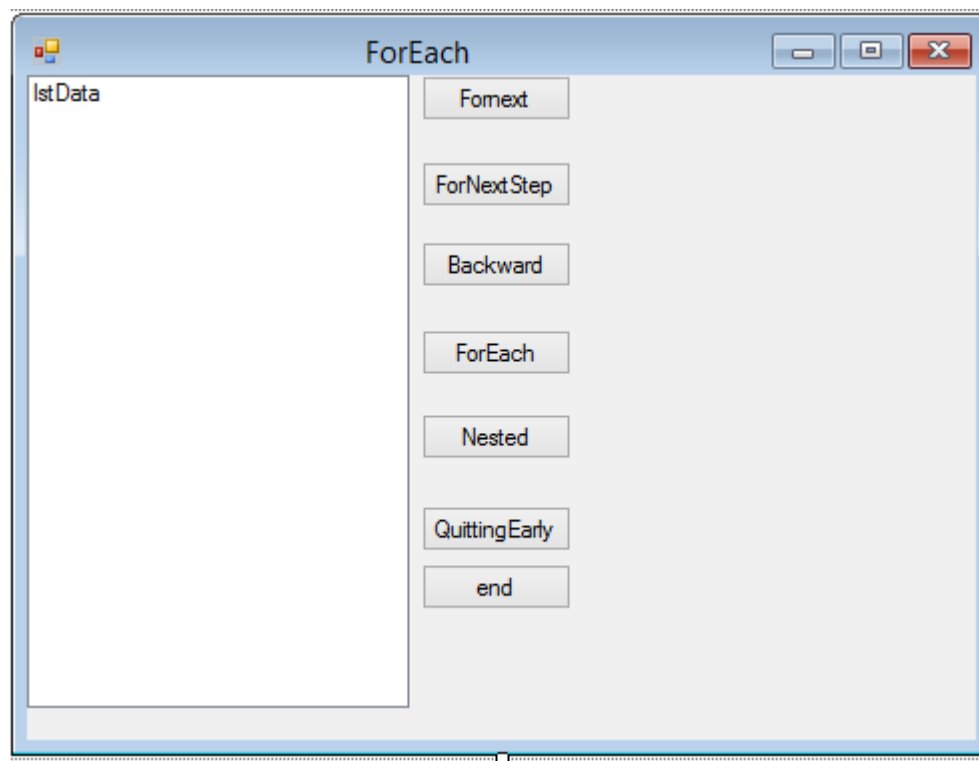
```
        strFavoriteColor = "Morning Mist"
    Case "Surya"
        strFavoriteColor = "Passionate Purple"

End Select
'Display the favorite color of the selected name
MessageBox.Show(strName & "'s favorite color is " &
strFavoriteColor, "Select Demo")
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub
End Class
```

Loops



```
Public Class form1

    Private Sub btnFornext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFornext.Click
        'Declare variable
        Dim intCount As Integer
        'Clear the list
        ClearList()
        'Perform a loop
        For intCount = 1 To 5
            'Add the item to the list
            lstData.Items.Add("I'm item " & intCount.ToString & " in the
list!")
        Next
    End Sub
    Private Sub ClearList()
        'Clear the list
        lstData.Items.Clear()
    End Sub

    Private Sub btnForNextStep_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnForNextStep.Click
        'Clear the list
        ClearList()
        'Perform a loop
        For intCount As Integer = 4 To 62 Step 7
            'Add the item to the list
            lstData.Items.Add(intCount.ToString)
        Next
    End Sub
```

```

    Private Sub btnBackward_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBackward.Click
        'Clear the list
        ClearList()
        'Perform a loop
        For intCount As Integer = 10 To 1 Step -3
            'Add the item to the list
            lstData.Items.Add(intCount.ToString)
        Next
    End Sub

    Private Sub btnForEach_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnForEach.Click
        'Clear the list
        ClearList()
        'List each folder at the root of your C drive
        For Each strFolder As String In
My.Computer.FileSystem.GetDirectories("C:\")
            'Add the item to the list
            lstData.Items.Add(strFolder)
        Next
    End Sub

    Private Sub btnNested_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNested.Click
        'Clear the list
        ClearList()
        'Process an outer loop
        For intOuterLoop As Integer = 1 To 2
            'Process a nested (inner) loop
            For intInnerLoop As Integer = 1 To 3
                lstData.Items.Add(intOuterLoop.ToString & ", " &
intInnerLoop.ToString)
            Next
        Next
    End Sub

    Private Sub btnQuittingEarly_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnQuittingEarly.Click
        'Clear the list
        ClearList()
        'List each folder at the root of your C drive
        For Each strFolder As String In
My.Computer.FileSystem.GetDirectories("C:\")
            'Add the item to the list
            lstData.Items.Add(strFolder)
            'Do you have the folder C:\Program Files?
            If String.Compare(strFolder, "c:\program files", True) = 0 Then
                'Tell the user
                MessageBox.Show("Found it, exiting the loop now.", "Loops")
                'Quit the loop early
                Exit For
            End If
        Next
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        End
    End Sub

```

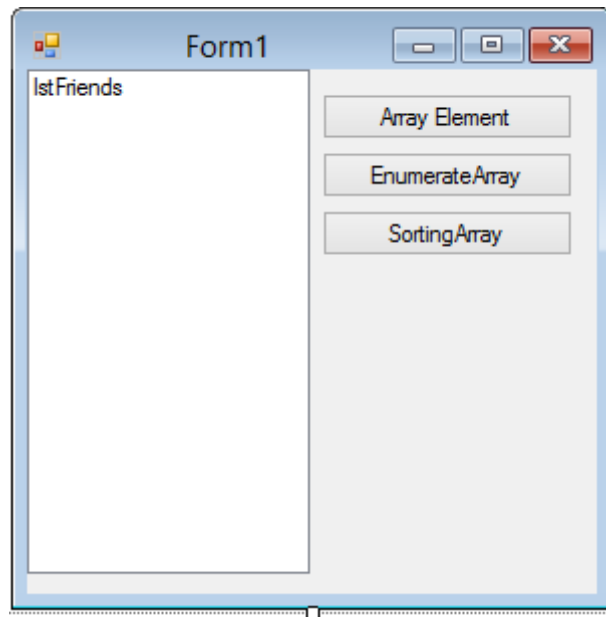
```

    Private Sub form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        End Sub
End Class

```

Array



```

Public Class Form1
    'Declare a form level array
    Private strFriends(4) As String

    Private Sub btnArrayElement_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnArrayElement.Click
        'Clear the list
        ClearList()
        'Declare an array
        Dim strFriends(4) As String
        'Populate the array
        strFriends(0) = "Wendy"
        strFriends(1) = "Harriet"
        strFriends(2) = "Jay"
        strFriends(3) = "Michelle"
        strFriends(4) = "Richard"
        'Add the first array item to the list
        lstFriends.Items.Add(strFriends(0))
    End Sub
    Private Sub ClearList()
        'Clear the list
        lstFriends.Items.Clear()
    End Sub

```

```

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        'Populate the array
        strFriends(0) = "Wendy"
        strFriends(1) = "Harriet"
        strFriends(2) = "Jay"
        strFriends(3) = "Michelle"
        strFriends(4) = "Richard"
    End Sub

    Private Sub btnEnumerateArray_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnEnumerateArray.Click
        'Clear the list
        ClearList()
        'Enumerate the array
        For Each strName As String In strFriends
            'Add the array item to the list
            lstFriends.Items.Add(strName)
        Next
    End Sub

    Private Sub btnSortingArray_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnSortingArray.Click
        'Clear the list
        ClearList()
        'Sort the array
        Array.Sort(strFriends)
        'List your friends
        AddItemsToList(strFriends)
    End Sub
    Private Sub AddItemsToList(ByVal arrayList As String())
        'Enumerate the array
        For Each strName As String In arrayList
            'Add the array item to the list
            lstFriends.Items.Add(strName)
        Next
    End Sub

    Public Sub New()

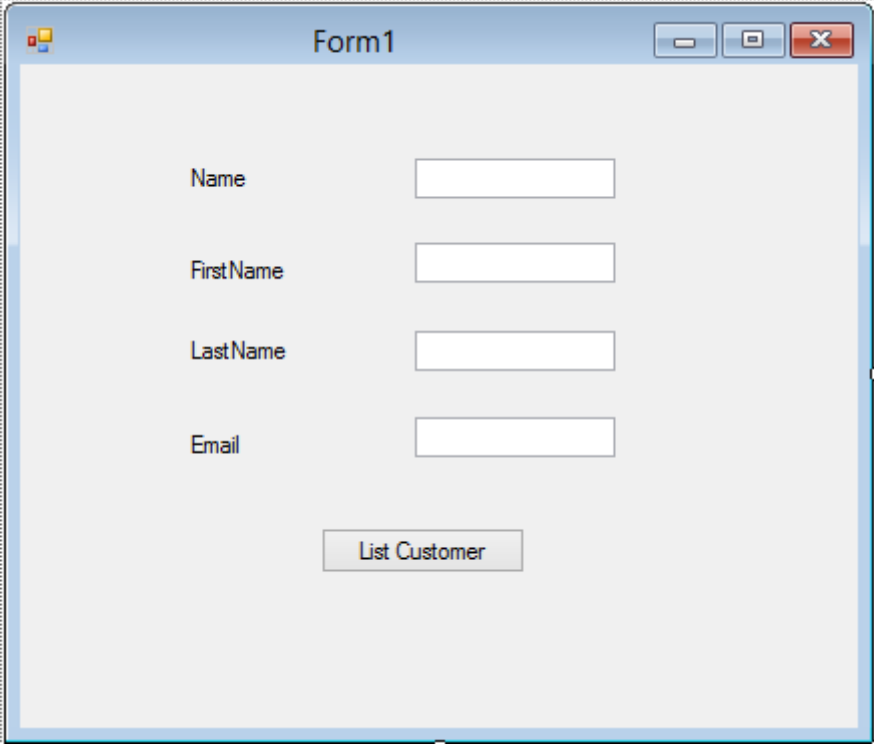
        ' This call is required by the Windows Form Designer.
        InitializeComponent()

        ' Add any initialization after the InitializeComponent() call.

    End Sub
End Class

```

Structures



The screenshot shows a standard Windows application window titled "Form1". Inside the window, there are four text input fields arranged vertically, each with a label to its left: "Name", "FirstName", "LastName", and "Email". Below these input fields is a single button labeled "List Customer". The window has a standard title bar with minimize, maximize, and close buttons.

```
Public Class Form1
    Public Structure customer
        'Public members
        Public FirstName As String
        Public LastName As String
        Public Email As String
        'Name property
        Public ReadOnly Property Name() As String
            Get
                Return FirstName & " " & LastName
            End Get
        End Property
    End Structure
    Public Sub DisplayCustomer(ByVal customer As customer)
        'Display the customer details on the form
        txtName.Text = customer.Name
        txtFirstName.Text = customer.FirstName
        txtLastName.Text = customer.LastName
        txtEmail.Text = customer.Email
    End Sub

    Private Sub btnList_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnList.Click
        'Create a new customer
        Dim objCustomer As customer
        objCustomer.FirstName = "Michael"
        objCustomer.LastName = "Dell"
```

```

        objCustomer.Email = "mdell@somecompany.com"
        'Display the customer
        DisplayCustomer(objCustomer)
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        End Sub
End Class

```

ArrayList Structures

The screenshot shows a Windows application window titled "Form1". Inside the window, there is a large, empty list box at the top with the label "lstCustomers". Below the list box, there are four text input fields arranged horizontally, labeled "Name", "FirstName", "LastName", and "Email". At the bottom of the form, there are two buttons: "List Customer" and "Delete Customer".

```

Public Class Form1
    Private objCustomers As New ArrayList
    Public Sub CreateCustomer(ByVal firstName As String, ByVal lastName As
String, ByVal email As String)
        'Declare a Customer object
        Dim objNewCustomer As customer
        'Create the new customer
        objNewCustomer.FirstName = firstName
        objNewCustomer.LastName = lastName
        objNewCustomer.Email = email
        'Add the new customer to the list
        objCustomers.Add(objNewCustomer)
        'Add the new customer to the ListBox control
        lstCustomers.Items.Add(objNewCustomer)
    End Sub
    Public Structure customer
        'Public members
        Public FirstName As String
        Public LastName As String
        Public Email As String
    End Structure
End Class

```

```

        'Name property
        Public ReadOnly Property Name() As String
            Get
                Return FirstName & " " & LastName
            End Get
        End Property
        Public Overrides Function ToString() As String
            Return Name & " (" & Email & ")"
        End Function
    End Structure
    Public ReadOnly Property SelectedCustomer() As customer
        Get
            If lstCustomers.SelectedIndex <> -1 Then
                'Return the selected customer
                Return CType(objCustomers(lstCustomers.SelectedIndex),
customer)
            End If
        End Get
    End Property
    Public Sub DisplayCustomer(ByVal customer As customer)
        'Display the customer details on the form
        txtName.Text = customer.Name
        txtFirstName.Text = customer.FirstName
        txtLastName.Text = customer.LastName
        txtEmail.Text = customer.Email
    End Sub

    Private Sub btnList_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnList.Click
        'Create some customers
        CreateCustomer("Darrel", "Hilton", "dhilton@somecompany.com")
        CreateCustomer("Frank", "Peoples", "fpeoples@somecompany.com")
        CreateCustomer("Bill", "Scott", "bscott@somecompany.com")
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub btnDeleteCustomer_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnDeleteCustomer.Click
        'If no customer is selected in the ListBox then.
        If lstCustomers.SelectedIndex = -1 Then
            'Display a message
            MessageBox.Show("You must select a customer to delete.",
"Structure Demo")
            'Exit this method
            Exit Sub
        End If
        'Prompt the user to delete the selected customer
        If MessageBox.Show("Are you sure you want to delete " &
SelectedCustomer.Name & "?", "Structure Demo", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) = DialogResult.Yes Then
            'Get the customer to be deleted
            Dim objCustomerToDelete As customer = SelectedCustomer
            'Remove the customer from the ArrayList
            objCustomers.Remove(objCustomerToDelete)
            'Remove the customer from the ListBox
            lstCustomers.Items.Remove(objCustomerToDelete)
        End If
    End Sub

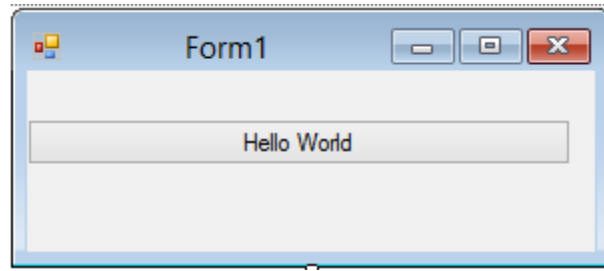
```

```

End Sub
Private Sub lstCustomers_SelectedIndexChanged(ByVal sender As Object,
ByVal e As EventArgs) Handles lstCustomers.SelectedIndexChanged
    'Display the customer details
    DisplayCustomer(SelectedCustomer)
End Sub
End Class

```

Button Events



```

Public Class ButtonEvents

    Private Sub btnSayHello_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSayHello.Click
        MessageBox.Show("Hello World!", Me.Text)
    End Sub

    Private Sub btnSayHello_MouseEnter(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnSayHello.MouseEnter
        'Change the Button text
        btnSayHello.Text = "The mouse is here!"
    End Sub

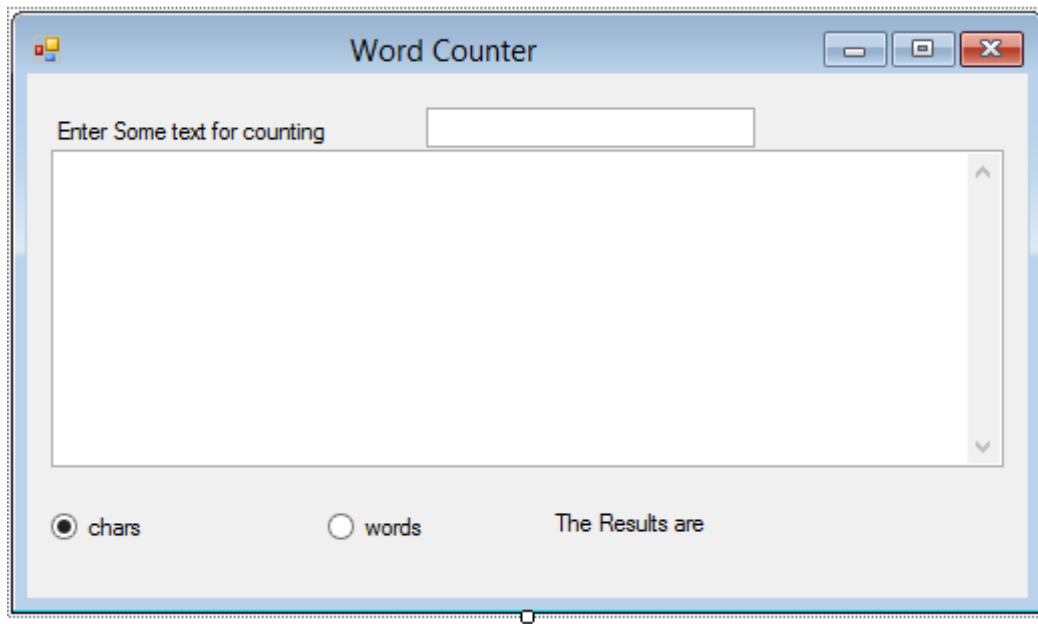
    Private Sub btnSayHello_MouseLeave(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnSayHello.MouseLeave
        'Change the Button text
        btnSayHello.Text = "The mouse has gone!"
    End Sub

    Private Sub ButtonEvents_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub
End Class

```

Word Counter



```
Public Class Form1
    Private Function CountCharacters(ByVal text As String) As Integer
        Return text.Length
    End Function
    Private Function CountWords(ByVal text As String) As Integer
        'Is the text empty?
        If text.Trim.Length = 0 Then Return 0
        'Split the words
        Dim strWords() As String = text.Split(" "c)
        'Return the number of words
        Return strWords.Length
    End Function
    Private Sub UpdateDisplay()
        'Do we want to count words?
        If radCountWords.Checked Then
            'Update the results with words
            lblResults.Text = CountWords(txtWords.Text) & " words"
        Else
            'Update the results with characters
            lblResults.Text = CountCharacters(txtWords.Text) & "
characters"
        End If
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub txtWords_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtWords.TextChanged
        'Count the number of characters
        'Dim intChars As Integer = CountCharacters(txtWords.Text)
        'Display the results
```

```

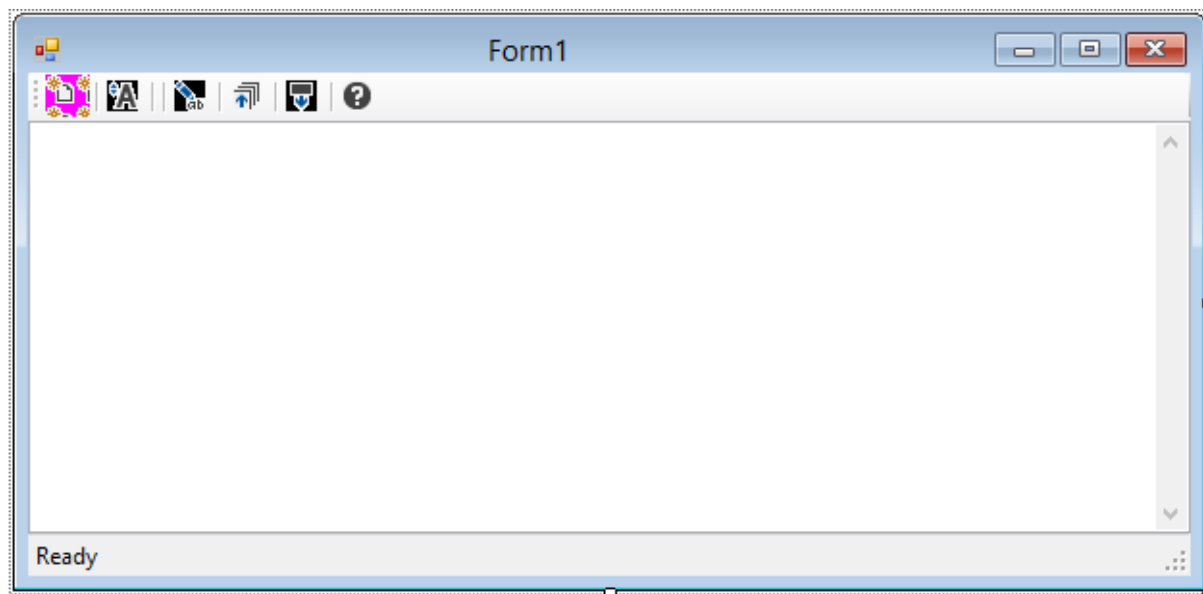
        'lblResults.Text = intChars & " characters"
        UpdateDisplay()
    End Sub

    Private Sub radCountWords_CheckedChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles radCountWords.CheckedChanged
        'Something changed so display the results
        UpdateDisplay()
    End Sub

    Private Sub radCountChars_CheckedChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles radCountChars.CheckedChanged
        'Something changed so display the results
        UpdateDisplay()
    End Sub
End Class

```

Toolbar Statusbar Multiple Forms



```

Public Class Form1

    'Get or set the text on the status bar
    Public Property StatusText() As String
        Get
            Return sslStatus.Text
        End Get
        Set(ByVal value As String)
            sslStatus.Text = value
        End Set
    End Property

    'Gets or sets the text that you're editing
    Public Property EditText() As String

```

```

        Get
            Return txtEdit.Text
        End Get
        Set(ByVal value As String)
            txtEdit.Text = value
        End Set
    End Property
    'Clears the txtEdit control
    Public Sub ClearEditBox()
        'Set the EditText property
        EditText = String.Empty
        'Reset the font color
        txtEdit.ForeColor = Color.Black
        'Set the status bar text
        StatusText = "Text box cleared"
    End Sub

    Private Sub txtEdit_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles txtEdit.TextChanged
        'Reset the status bar text
        StatusText = "Ready"
    End Sub

    Private Sub tbrClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tbrClear.Click
        'Clear the edit box
        ClearEditBox()
    End Sub
    Public Sub RedText()
        'Make the text red
        txtEdit.ForeColor = Color.Red
        'Update the status bar text
        StatusText = "The text is red"
    End Sub

    Private Sub tbrRed_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tbrRed.Click
        'Make the text red
        RedText()
    End Sub
    Public Sub BlueText()
        'Make the text blue
        txtEdit.ForeColor = Color.Blue
        'Update the status bar text
        StatusText = "The text is blue"
    End Sub

    Private Sub tbrBlue_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tbrBlue.Click
        'Make the text blue
        BlueText()
    End Sub
    Public Sub UpperCaseText()
        'Make the text uppercase
        EditText = EditText.ToUpper
        'Update the status bar text

```

```

        StatusText = "The text is all uppercase"
    End Sub

    Private Sub tbrUppercase_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles tbrUppercase.Click
        'Make the text uppercase
        UpperCaseText()

    End Sub
    Public Sub LowerCaseText()
        'Make the text lowercase
        EditText = EditText.ToLower
        'Update the status bar text
        StatusText = "The text is all lowercase"
    End Sub

    Private Sub tbrLowercase_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles tbrLowercase.Click
        'Make the text lowercase
        LowerCaseText()
    End Sub
    Public Sub ShowAboutBox()
        'Display the About dialog box
        Using objAbout As New About
            objAbout.ShowDialog(Me)
        End Using
    End Sub

    Private Sub tbeHelp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tbeHelp.Click
        'Display the About dialog box
        ShowAboutBox()
    End Sub

    Private Sub ToolStrip1_ItemClicked(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.ToolStripItemClickedEventArgs) Handles
ToolStrip1.ItemClicked

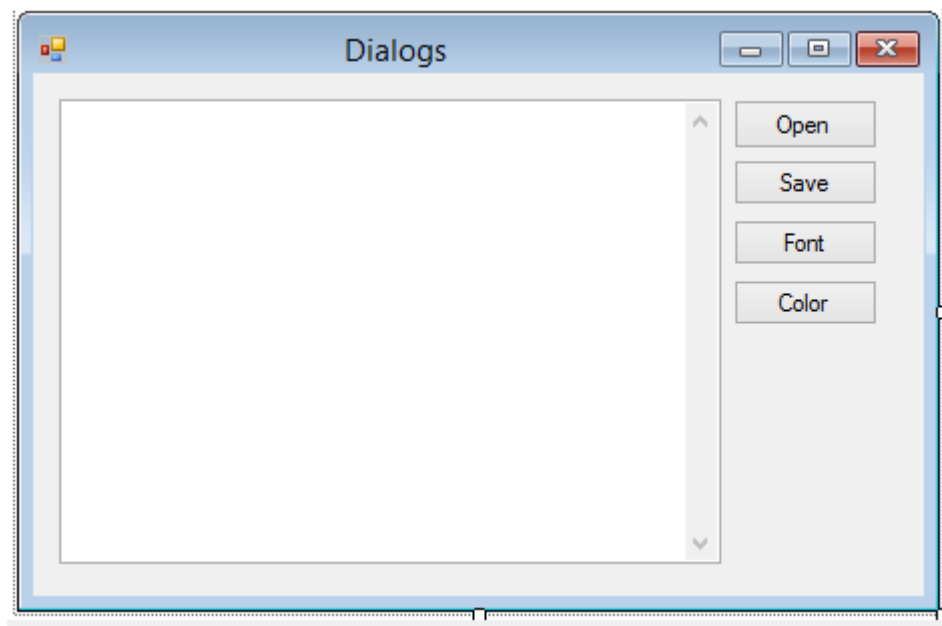
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub
End Class

```

Common Dialog Controls



```
Public Class Dialogs

    'Declare variable
    Private strFileName As String

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        'Set the Open dialog properties
        OpenFileDialog1.Filter = "Text Documents (*.txt)|*.txt|All Files
(*.*)|*.*"
        OpenFileDialog1.FilterIndex = 1
        OpenFileDialog1.Title = "Demo Open File Dialog"
        'Show the Open dialog if the user clicks the Open button,
        'load the file
        If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
            Try
                'Save the file path and name
                strFileName = OpenFileDialog1.FileName
                Dim fileContents As String
                fileContents =
My.Computer.FileSystem.ReadAllText(strFileName)
                Dim fileContents1 As String

                txtFile.Text = fileContents
            Catch ex As Exception
                MessageBox.Show(ex.Message, My.Application.Info.Title,
MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End If
    End Sub

    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
```

```

        'Set the Save dialog properties
With SaveFileDialog1
    .DefaultExt = "txt"
    .FileName = strFileName
    .Filter = "Text Documents (*.txt)|*.txt|All Files (*.*)|*.*"
    .FilterIndex = 1
    .OverwritePrompt = True
    .Title = "Demo Save File Dialog"
End With
'Show the Save dialog and if the user clicks the Save button,
'save the file
If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
    Try
        'Save the file path and name
        strFileName = SaveFileDialog1.FileName
        My.Computer.FileSystem.WriteAllText(strFileName,
txtFile.Text, False)
    Catch ex As Exception
        MessageBox.Show(ex.Message, My.Application.Info.Title,
MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End If
End Sub

    Private Sub btnFont_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFont.Click
        'Show the Font dialog and if the user clicks the OK button,
        'update the font and color in the text box
        If FontDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
            txtFile.Font = FontDialog1.Font
            txtFile.ForeColor = FontDialog1.Color
        End If
    End Sub

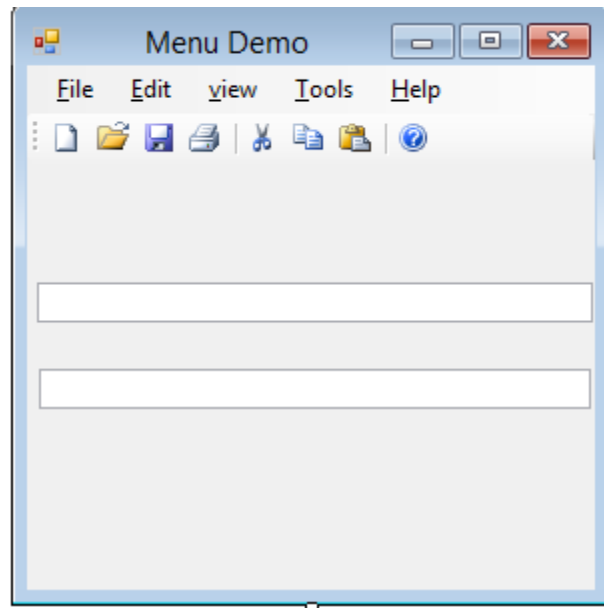
    Private Sub btnColour_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnColour.Click
        If ColorDialog1.ShowDialog = Windows.Forms.DialogResult.OK Then
            Me.BackColor = ColorDialog1.Color
            txtFile.ForeColor = ColorDialog1.Color
        End If
    End Sub

    Private Sub Dialogs_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub
End Class

```

Menu and Pop Up (Context) Menu



```
Public Class frmMenu
```

```
    Private Sub NewToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles NewToolStripMenuItem.Click
        'Clear the text boxes
        TextBox1.Text = String.Empty
        TextBox2.Text = String.Empty
        'Set focus to the first text box
        TextBox1.Focus()
    End Sub
```

```
    Private Sub NewToolStripButton_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click
        'Call the NewToolStripMenuItem_Click procedure
        NewToolStripMenuItem_Click(sender, e)
    End Sub
```

```
    Private Sub ExitToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click
        'Close the form and end
        Me.Close()
    End Sub
```

```
    Private Sub UndoToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles UndoToolStripMenuItem.Click
        'Undo the last operation
        If TypeOf Me.ActiveControl Is TextBox Then
            CType(Me.ActiveControl, TextBox).Undo()
        End If
    End Sub
```

```

    Private Sub CutToolStripMenuItem_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles CutToolStripMenuItem.Click
    'Copy the text to the clipboard and clear the field
    If TypeOf Me.ActiveControl Is TextBox Then
        CType(Me.ActiveControl, TextBox).Cut()
    End If
End Sub

    Private Sub CutToolStripButton_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles CutToolStripButton.Click
    'Copy the text to the clipboard and clear the field
    If TypeOf Me.ActiveControl Is TextBox Then
        CType(Me.ActiveControl, TextBox).Cut()
    End If
End Sub

    Private Sub CopyToolStripMenuItem_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles CopyToolStripMenuItem.Click
    'Copy the text to the clipboard
    If TypeOf Me.ActiveControl Is TextBox Then
        CType(Me.ActiveControl, TextBox).Copy()
    End If
End Sub

    Private Sub CopyToolStripButton_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles CopyToolStripButton.Click
    'Call the CopyToolStripMenuItem_Click procedure
    CopyToolStripMenuItem_Click(sender, e)
End Sub

    Private Sub PasteToolStripMenuItem_Click(ByVal sender As Object, ByVal
e As System.EventArgs) Handles PasteToolStripMenuItem.Click
    'Copy the text from the clipboard to the text box
    If TypeOf Me.ActiveControl Is TextBox Then
        CType(Me.ActiveControl, TextBox).Paste()
    End If
End Sub

    Private Sub PasteToolStripButton_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles PasteToolStripButton.Click
    'Call the PasteToolStripMenuItem_Click procedure
    PasteToolStripMenuItem_Click(sender, e)
End Sub

    Private Sub SelectAllToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles SelectAllToolStripMenuItem.Click
    'Select all the text in the text box
    If TypeOf Me.ActiveControl Is TextBox Then
        CType(Me.ActiveControl, TextBox).SelectAll()
    End If
End Sub

    Private Sub MainToolStripMenuItem2_Click(ByVal sender As Object, ByVal
e As System.EventArgs) Handles MainToolStripMenuItem2.Click
    'Toggle the visibility of the Main toolbar
    'based on the menu item's Checked property

    If MainToolStripMenuItem2.Checked Then
        tspMain.Visible = True
    Else

```

```

        tspMain.Visible = False
    End If
End Sub

Private Sub FromatingToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles FromatingToolStripMenuItem.Click

    'Toggle the visibility of the Formatting toolbar
    'based on the menu item's Checked property
    If FromatingToolStripMenuItem.Checked Then
        tspFormatting.Visible = True
    Else
        tspFormatting.Visible = False
    End If

End Sub

Private Sub ToolStripMenuItem4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ContextCutToolStripMenuItem.Click
    'Call the CutToolStripMenuItem_Click procedure
    CutToolStripMenuItem_Click(sender, e)
End Sub

Private Sub ContextUndoToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ContextUndoToolStripMenuItem.Click
    'Call the UndoToolStripMenuItem_Click procedure
    UndoToolStripMenuItem_Click(sender, e)
End Sub

Private Sub ContextCopyToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ContextCopyToolStripMenuItem.Click
    'Call the CopyToolStripMenuItem_Click procedure
    CopyToolStripMenuItem_Click(sender, e)
End Sub

Private Sub ContextPasteToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ContextPasteToolStripMenuItem.Click
    'Call the PasteToolStripMenuItem_Click procedure
    PasteToolStripMenuItem_Click(sender, e)
End Sub

Private Sub ContextSelectAllToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ContextSelectAllToolStripMenuItem.Click
    'Call the SelectAllToolStripMenuItem_Click procedure
    SelectAllToolStripMenuItem_Click(sender, e)
End Sub

Private Sub frmMenu_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

End Sub
End Class

```

