Separate if statement logic associated with #4.
Private Sub btnFifth_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnFifth.Click
    'Write the code to test for code A and either field1 > 10 or field2 > 20 and field3 > 40
    'Write different message for each of the options meeting the criteria and different messages for each of the options not meeting the criteria
End Sub

Private Sub btnSixth_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSixth.Click
    'Write the code for code A and field1 > 100 or field2 > 200 or code B and either field1 or field2 greater than 100 or code C and field 3 < 50
    IF txtCode = "A" And either Then
End Sub

Private Sub btnSeventh_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSeventh.Click
    'Use the case structure to test the code for A, B, C or D.
End Sub
End Class
Informatica Informatica

End If
' write the code for code A and field1 > 100 or field2 > 200 or
' code B and either field1 or field2 greater than 100 or
' code C and field 3 < 50
If txtCode.Text = "A" Then
  If txtField1.Text > 100 Then
    txtResult.Text = "OK"
  Else
    If txtField2.Text > 200 Then
      txtResult.Text = "OK"
    End If
  End If
Else
  If txtField1.Text > 100 Then
    txtResult.Text = "OK"
  End If
End If
Else
  If txtCode.Text = "B" Then
    If txtField1.Text > 100 Then
      txtResult.Text = "OK"
    Else
      If txtField2.Text > 200 Then
        txtResult.Text = "OK"
      End If
    End If
  End If
Else
  If txtCode.Text = "C" Then
    If txtField1.Text < 50 Then
      txtResult.Text = "OK"
    End If
  End If
End If
End If
End Sub
Example using case that will help with doing the seventh problem.
Partial test
Else If lstJob.SelectedIndex = 1 Or lstJob.SelectedIndex = 3 Then
    wktWithRaise = wktWithRaise + 1.06
End If
End If
End Sub
End Class

Private Sub btnOtherthings_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOtherthings.Click
    Dim wktNewJob As String
    wktNewJob = InputBox("Enter new job: ", "Job", ", 48, 48)
    Dim i As Integer = lstJob.Items.Add(wktNewJob)
    lstJob.Items(i) = lstJob.Items(i) + 1
    lstJob.Items(i) = lstJob.Items(i) + 1
End Sub

Private Sub btnAddJob_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddJob.Click
    Dim wktNewJob As String
    wktNewJob = InputBox("Enter new job: ", "Job", ", 48, 48)
    Dim i As Integer = lstJob.Items.Add(wktNewJob)
    lstJob.Items(i) = lstJob.Items(i) + 1
    lstJob.Items(i) = lstJob.Items(i) + 1
End Sub
Else
    If lstJob.SelectedIndex = 1 Or lstJob.SelectedIndex = 3 Then
        WidthRaise = wkPay * 1.86
    End If
    txtWidthRaise.Text = wkWidthRaise
    txtJob.Text = lstJob.Items(lstJob.SelectedIndex)
End Sub

Private Sub btnotherthings_Click(MyVal sender As System.Object, ByVal e As System.EventArgs) Handles btnotherthings.Click
    Dim wkNewJob As String
    wkNewJob = InputBox("Key in new job", "Job", 48, 48)
    lstJob.Items.Add(wkNewJob)
    txtAddedJob.Text = lstJob.Items(lstJob.Items.Count - 1)
    txtNewCount.Text = lstJob.Items.Count
End Sub

End Class
Potential of not doing code in loop

Always do code in loop once.
Condition to get out
```vbnet
Private Sub btnDoWhileLoop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim wKt As Integer = 1
    Dim wAns As Double = CDbl(txtStartAmount.Text)
    Dim wMsg As String
    Do
        wAns = wAns - CDBL(txt2Amount.Text)
        wMsg = vbCrLf & "Amount Now " & CStr(wAns)
        lstTrace.Items.Add(wMsg)
        wKt = wKt + 1
    Loop Until wKt > txtNumTimes.Text
    txtAmountLeft.Text = wAns
End Sub
```

```vbnet
Private Sub btnForLoop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim wAns As Double = CDbl(txtStartAmount.Text)
    Dim wMsg As String
    For wKt = 1 To 4
        wAns = wAns - CDBL(txt2Amount.Text)
        wMsg = vbCrLf & "Amount Now " & CStr(wAns)
        lstTrace.Items.Add(wMsg)
    Next
End Sub
```

![Visual Studio code editor with VBA code snippet](image-url)
Loop not entered so no processing in loop.
```vbnet
Private Sub btnDoWhileLoop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDoWhileLoop.Click
    Dim wct As Integer = 6
    Dim wkmgs As String = "Hello World!
    Do While wct <> txtNumTimes.Text
        wkmgs = wkmgs + CStr(wkans) + "\n"
        lstTrace.Items.Add(wkmgs)
        wct = wct + 1
    Loop
    txtAtLeft.Text = wkmgs
End Sub
```

```vbnet
Private Sub btnDoWhilePost_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDoWhilePost.Click
    Dim wkmgs As String = "Hello World!
    Do
        wkmgs = wkmgs + CStr(txtAtStart.Text)
        wkmgs = "Amount Now " + CStr(wkans)
        lstTrace.Items.Add(wkmgs)
        wct = wct + 1
    Loop While wct <> txtNumTimes.Text
    txtAtLeft.Text = wkmgs
End Sub
```

```vbnet
Private Sub btnDoUntilLoop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDoUntilLoop.Click
    Dim wkans As Double = CDb(txtAtStart.Text)
    Dim wkmgs As String = "Hello World!
    Do Until wkmgs <> txtNumTimes.Text
        wkmgs = wkmgs + CStr(wkans) + "\n"
        lstTrace.Items.Add(wkmgs)
        wct = wct + 1
    Loop
    txtAtLeft.Text = wkmgs
End Sub
```
Enter loop since no condition. Did it once and then since wkCt started at 6, we exit.