

Forum:[Hack Forums](#)

Thread: [95 percent done with this program](#)

SIM-1895121 - October 27, 2017, 9:08 am

I am currently working on a project called TG automotive. It is a simple form with pretty basic calculations. I have the program running flawlessly except for the validation. I am supposed to have a function that tests to make sure inputs are numeric and not negative. I have written the function, i just cannot seem to figure out how to actually implement it in the program. Any tips or suggestions greatly appreciated. The ValidateInputs() is the function i created that i need to implement. Here is the full code.

Code Public Class Form1

Private Sub GroupBox5_Enter(sender As Object, e As EventArgs) Handles GroupBox5.Enter

End Sub

Private Sub Label11_Click(sender As Object, e As EventArgs) Handles Label11.Click

End Sub

Function CalcOilAndLube() As Decimal

Dim decOil As Decimal

If chkOilChange.Checked = True Then

decOil += 36

End If

If chkLube.Checked = True Then

decOil += 28

End If

Return decOil

End Function

Function Misc() As Decimal

Dim decMisc As Decimal

If chkInspection.Checked = True Then

decMisc += 15

End If

If chkMuffler.Checked = True Then

decMisc += 200

End If

If chkTireRotation.Checked = True Then

decMisc += 20

End If

Return decMisc

End Function

Function RadandTran() As Decimal

Dim decRadCost As Decimal

If chkRadFlush.Checked = True Then

decRadCost += 50

End If

If chkTranFlush.Checked = True Then

```
        decRadcCost += 120
    End If
    Return decRadcCost
End Function
Function Parts() As Decimal
```

```
    Dim decParts As Decimal
    decParts = FormatCurrency(txtParts.Text)
    Return decParts
End Function
```

```
Function Labor() As Decimal
    Dim decLabor As Decimal
    decLabor = FormatCurrency(txtLabor.Text)
    Return decLabor
End Function
```

```
Function Tax() As Decimal
    Dim dectax As Decimal
    dectax = FormatCurrency(txtParts.Text) * 0.06
    Return dectax
End Function
```

```
End Function
```

```
Private Sub btnClear_Click(sender As Object, e As EventArgs) Handles btnClear.Click
    ClearMisc()
    ClearOil()
    ClearOther()
    ClearPartsAndLabor()
    ClearRad()
End Sub
```

```
Private Sub btnCalculate_Click(sender As Object, e As EventArgs) Handles btnCalculate.Click
```

```
    Dim decParts As Decimal
    Dim decService As Decimal
    Dim decTax As Decimal
    Dim decTotal As Decimal
    decService = CalcOilAndLube() + Misc() + RadandTran() + Labor()
    decParts = Parts()
    decTax = Tax()
    decTotal = FormatCurrency(decService + decParts + decTax)
    lblParts.Text = FormatCurrency(decParts)
    lblServicesAndLabor.Text = FormatCurrency(decService)
    lblTax.Text = FormatCurrency(decTax)
    lblTotal.Text = FormatCurrency(decTotal)
```

```
End Sub
```

```
Sub ClearOil()
    chkOilChange.Checked = False
    chkLube.Checked = False
```

```
End Sub
```

```

Sub ClearRad()
    chkRadFlush.Checked = False
    chkTranFlush.Checked = False

End Sub
Sub ClearMisc()
    chkInspection.Checked = False
    chkMuffler.Checked = False
    chkTireRotation.Checked = False
End Sub
Sub ClearPartsAndLabor()
    txtLabor.Text = String.Empty
    txtParts.Text = String.Empty
End Sub
Sub ClearOther()
    lblParts.Text = String.Empty
    lblServicesAndLabor.Text = String.Empty
    lblTax.Text = String.Empty
    lblTotal.Text = String.Empty
End Sub
Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles btnExit.Click
    Me.Close()
End Sub
Function ValidInputs() As Boolean
    Dim decVal1 As Decimal
    Dim decVal2 As Decimal
    If Not Decimal.TryParse(txtParts.Text, decVal1) Then
        MessageBox.Show("Enter a Numeric value please")
        Return False
    End If
    If decVal1 < 0 Then
        MessageBox.Show("Number must be positive")
    End If
    If Not Decimal.TryParse(txtLabor.Text, decVal2) Then
        MessageBox.Show("Enter a Numeric value please")
        Return False
    End If
    If decVal2 < 0 Then
        MessageBox.Show("Number Must be positive")
    End If
    Return True
End Function
End Class

```

CoderEX - October 27, 2017, 9:19 am

Have you tried implementing it into the textbox text change event?

Note: I did not look through the entire code.

SIM-1895121 - October 27, 2017, 9:25 am

(Oct 27, 2017 09:19 AM)CoderEX Wrote: [\[CITE\]](#) Have you tried implementing it into the textbox text change event?

Note: I did not look through the entire code.

So i would be adding it to Parts function and the Labor function? I cant figure out correct syntax for referencing it while still keeping the currency format.

This is where the text boxes get converted.

Edit: By changing it to a procedure i can simply call it using ValidInputs() wich works, however the assignment is asking me to use it as a function.

Code Function Labor() As Decimal

Dim decLabor As Decimal

decLabor = FormatCurrency(txtLabor.Text)

Return decLabor

End Function

Function Tax() As Decimal

Dim dectax As Decimal

dectax = FormatCurrency(txtParts.Text) * 0.06

Return dectax

End Function

He B TeMy - October 27, 2017, 9:52 am

You're not calling your ValidInputs() method anywhere, either you call them on the Button that you are proceeding with the validation or you handle both textbox's textchanged events and handle it that way.

CoderEX - October 27, 2017, 9:56 am

Refer to this link to see how to use control events (specifically textbox controls):

<https://youtu.be/Z4SQwEVwjJg>

You will be implementing the ValidInputs() function into the event.

SIM-1895121 - October 27, 2017, 10:09 am

Very useful video, thank you for that. I seemed to have solved my issue, well atleast as good as its going to get. Would anyone be interested in possibly tutoring me in vb.net for this semester? I work full time and dont have time to make it to my professors office. I have like 4 large assignments left. At no point would i be asking anyone to do my homework, just provide support for when i screw up. I can compensate depending on what you want.

PM me if so

gigajew - October 30, 2017, 9:21 pm

Non-negative integers would imply unsigned integers

"The value of an item with an integral type is the mathematical integer that it corresponds to. Integral types may be unsigned (capable of representing only non-negative integers) or signed (capable of representing negative integers as well)." - Wikipedia

Here's an example solution (from Stackoverflow user Isak Savo) modified for decimals:

0xDEAD10CC - October 31, 2017, 7:35 am

Instead of throwing exceptions you should implement a TryParse().

Dofeus - October 31, 2017, 10:00 am

make a parameter for your ValidateInput method.

Code Function ValidateInput(Byval input as Integer) As Boolean

```
" Your code here
```

```
End Function
```

```
then if you are going to use it:
```

```
Code If ValidateInput(txtParts.Text) Then
```

```
    " true
```

```
else
```

```
    " false
```

```
End If
```

0xDEAD10CC - *October 31, 2017, 10:19 am*

(Oct 31, 2017 10:00 AM)Dofeus Wrote: [CITE](#) make a parameter for your ValidateInput method.

```
Code Function ValidateInput(Byval input as Integer) As Boolean
```

```
" Your code here
```

```
End Function
```

```
then if you are going to use it:
```

```
Code If ValidateInput(txtParts.Text) Then
```

```
    " true
```

```
else
```

```
    " false
```

```
End If
```

You define the function to take an integer yet you pass in a string... LOL.

Why not ByVal the string and ByRef an int to return a boolean and return the result in the reference param passed in the way TryParse() methods work?