```vb
'Nathan Carlson
'RCET0265
'asg3-3
'Inventory Calculator
'https://github.com/carlnathF19/NDC-VS-F19/tree/master/assignments/ASG3-3/Inventory_Calculator

Option Explicit On
Option Strict On
Public Class Inventory_Calculator
    Dim startInv As Integer
    Dim endInv As Integer
    'costOfGoods, avgInv, and turnover must be declared as double.  Currency is a double unit; avgInv, and turnover
    'are results of division.  CostofGoods being static allows another Sub to change textbox format.
    Dim costOfGoods As Double
    Dim avgInv As Double
    Dim turnover As Double
    Dim errorMessage As String

    'When calculate is clicked a series of codes execute to calculate avgInv and turnover while catching
    'invalid user input.
    Private Sub CalculateButton_Click(sender As Object, e As EventArgs) Handles calculateButton.Click
        'sets errorMessage to empty
        errorMessage = ""

        'attempts to parse cost of goods text into a double, if it fails it puts a message with desired text
        'in the box, displays an errormessage and adds the bottommost line of the variable passed to the
        'messagebox.  Also selects the bottommost box if input is invalid
        Try
            costOfGoods = Double.Parse(costOfGoodsTextBox.Text)
            costOfGoodsTextBox.Text = costOfGoods.ToString("C")
        Catch ex As Exception


            costOfGoodsTextBox.Text = "Please enter currency with no $"
                costOfGoodsTextBox.Select()
                errorMessage = "Please enter currency with no $" & vbNewLine & errorMessage

        End Try
        'each of these has the same function for their individual textbox as described in the previous comment
        'but selects the current box, by running the bottommost box first, and
```

```
            the code execution from top
41          'to bottom this results in the focus on the highest box with an error.
42          Try
43              endInv = Integer.Parse(endingInventoryTextBox.Text)
44          Catch ex As Exception
45              endingInventoryTextBox.Text = "Please enter only Whole Numbers in
                  Ending Inventory"
46              endingInventoryTextBox.Select()
47              errorMessage = "Please enter only Whole Numbers" & vbNewLine &
                  errorMessage
48          End Try
49          'each of these has the same function for their individual textbox as
              described in the previous comment
50          Try
51              startInv = Integer.Parse(beginningInventoryTextBox.Text)
52          Catch ex As Exception
53              beginningInventoryTextBox.Text = "Please enter only Whole Numbers
                  in Beginning Inventory"
54              beginningInventoryTextBox.Select()
55              errorMessage = "Please enter only Whole Numbers" & vbNewLine &
                  errorMessage
56          End Try
57
58          'if errorMessage is not empty, a messagebox is shown from the
              concatenated Catch statements.
59          If errorMessage <> "" Then
60              MessageBox.Show(errorMessage)
61
62          End If
63
64          'If errorMessage is empty then no Try's failed.  The calculations are
              then performed, and output written.
65          'This ensures that if invalid data is entered calculations will not
              output invalid data based on the data
66          'it does receive.
67          If errorMessage = "" Then
68              avgInv = Convert.ToDouble(startInv + endInv)
69              avgInv /= 2
70              turnover = costOfGoods / avgInv
71              avgInventoryTextBox.Text = avgInv.ToString("C")
72              turnoverTextBox.Text = turnover.ToString("F1")
73          End If
74      End Sub
75
76      'This sub clears all fields and values for a new user calculation
77      Private Sub ClearButton_Click(sender As Object, e As EventArgs) Handles
          clearButton.Click
78          beginningInventoryTextBox.Text = ""
79          endingInventoryTextBox.Text = ""
```

```
80          costOfGoodsTextBox.Text = ""
81          avgInventoryTextBox.Text = ""
82          turnoverTextBox.Text = ""
83          startInv = 0
84          endInv = 0
85          costOfGoods = 0
86          avgInv = 0
87          turnover = 0
88      End Sub
89
90      'Closes the form
91      Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles
          exitButton.Click
92          Me.Close()
93      End Sub
94
95      'The next 3 subs check to see if the textbox text has been set in a
          previous Catch, if it has when the user
96      'clicks on the textbox, the textbox will clear for new data.  If the data
          was valid though, the content of the
97      'textbox is unchanged.
98
99      Private Sub beginningInventoryTextBox_MouseDown(sender As Object, e As
          MouseEventArgs) Handles beginningInventoryTextBox.MouseDown
100         If (beginningInventoryTextBox.Text = "Please enter only Whole Numbers
              in Beginning Inventory") Then
101             beginningInventoryTextBox.Text = ""
102         End If
103     End Sub
104
105     Private Sub endingInventoryTextBox_MouseDown(sender As Object, e As
          MouseEventArgs) Handles endingInventoryTextBox.MouseDown
106         If (endingInventoryTextBox.Text = "Please enter only Whole Numbers in
              Ending Inventory") Then
107             endingInventoryTextBox.Text = ""
108         End If
109     End Sub
110
111     Private Sub CostOfGoodsTextBox_MouseDown(sender As Object, e As EventArgs)
          Handles costOfGoodsTextBox.MouseDown
112         If (costOfGoodsTextBox.Text = "Please enter currency with no $") Then
113             costOfGoodsTextBox.Text = ""
114         End If
115     End Sub
116
117
118 End Class
119
```