```
'-----------------------------------------------------------------------
' Title:    GISC9304-Assignment 2 – VB .NET Application Development
' Author:   Travis Vanos
' Date:     11/15/2015
' Purpose:  VB .NET Mapping Scale calculator with data validation
' ----------------------------------------------------------------------

'Option Strict On to prevent data type conversion related errors
Option Strict On

' Module declaration
Module Module1

    Sub Main()

        ' Declaration of all variables to be used in calculations and string concatenations
        Dim strFocalLength As String = ""
        Dim dblFocalLength As Double
        Dim strHeight As String = ""
        Dim dblHeight As Double
        Dim strElevation As String = ""
        Dim dblElevation As Double
        Dim intScale As Integer
        Dim LoopValidation As Boolean = False
        Dim ExitProgram As Boolean = False

        Console.Write("-------------------------------------------------------------------------" & vbCrLf)
        Console.Write(vbTab & vbTab & "Travis Vanos' Map Scale Program V 1.0" & vbCrLf)
        Console.Write("-------------------------------------------------------------------------" & vbCrLf & vbCrLf &
vbCrLf)

        ' Do ... Until loop for user to re-calculate scale until keystroke is pressed
        Do
            ' Set the Decimal values as 0 to clear input for when loop is excecuted
            dblFocalLength = 0
            dblHeight = 0
            dblElevation = 0
            Console.Write("Please enter the focal length (f) of the aerial camera (in centimetre) " & vbCrLf)
            ' Reading of user's input for Focal Length of aerial camera
            strFocalLength = Console.ReadLine()

            ' Loop for data validation of specified range, will break when boolean "LoopValidation" = True
            Do
                ' Tryparse method is used to validate input as numerical and cast into Decimal type if, when returned,
TryParse() = True
                ' If, else logic for converted decimal to adhere to specified range
                If Double.TryParse(strFocalLength, dblFocalLength) And dblFocalLength >= 0.1 And dblFocalLength <= 99.9
Then
                    ' Convert sanitized string
                    dblFocalLength = System.Convert.ToDecimal(strFocalLength)
                    ' cm to m conversion for aerial camera focal length
                    dblFocalLength = dblFocalLength * 1 / 100
                    ' LoopValidation is True to break loop
                    LoopValidation = True
                    Console.ForegroundColor = ConsoleColor.Green
                    Console.Write(vbCrLf & "OK!" & vbCrLf)
                    Console.ResetColor()
                Else
                    ' Visual error message for invalid responses
                    Console.ForegroundColor = ConsoleColor.Red
                    Console.Write("-------------------------------------------------------------" & vbCrLf)
                    Console.Write("| Error! Please Valid Numeric Value from 0.1 to 99.9         |" & vbCrLf)
                    Console.Write("-------------------------------------------------------------" & vbCrLf & vbCrLf)
                    Console.ResetColor()
                    ' Waits for user input of any keystroke
                    Console.Write("Press any key to continue . . . ")
                    Console.ReadKey(True)
                    strFocalLength = ""
                    Console.Clear()
                    Console.Write("Please re-enter the focal length (cm) " & vbCrLf)
                    strFocalLength = Console.ReadLine()
                    Continue Do
                End If
```

```
            ' Loop until Validator breaks loop
        Loop Until LoopValidation = True

        ' Reset to false for next loop iteration
        LoopValidation = False
        Console.Write(vbCrLf & "Please enter the flying height (H) of the aerial camera (in metre) " & vbCrLf)
        strHeight = Console.ReadLine()

        Do
            ' Tryparse method is used to validate input as numerical and cast into Decimal type if, when returned,
TryParse() = True
            ' If, else logic for converted decimal to adhere to specified range
            If Double.TryParse(strHeight, dblHeight) And dblHeight >= 1.0 And dblHeight <= 1000000.0 Then
                ' Convert sanitized string
                dblHeight = System.Convert.ToDecimal(strHeight)
                ' LoopValidation is True to break loop
                LoopValidation = True
                Console.ForegroundColor = ConsoleColor.Green
                Console.Write(vbCrLf & "OK!" & vbCrLf)
                Console.ResetColor()
            Else
                ' Visual error message for invalid responses
                Console.ForegroundColor = ConsoleColor.Red
                Console.Write("-------------------------------------------------------------" & vbCrLf)
                Console.Write("| Error! Please Valid Numeric Value from 1.0 to 1000000.0   |" & vbCrLf)
                Console.Write("-------------------------------------------------------------" & vbCrLf & vbCrLf)
                Console.ResetColor()
                ' Waits for user input of any keystroke
                Console.Write("Press any key to continue . . . ")
                Console.ReadKey(True)
                strHeight = ""
                Console.Clear()
                Console.Write("Please re-enter the height (m) " & vbCrLf)
                strHeight = Console.ReadLine()
                Continue Do
            End If
            ' Loop until Validator breaks loop
        Loop Until LoopValidation = True

        ' Reset to false for next loop iteration
        LoopValidation = False
        Console.Write(vbCrLf & "Please enter the elevation (e) of the selected point on the photo (in metre) " &
vbCrLf)
        strElevation = Console.ReadLine()


        Do
            ' Tryparse method is used to validate input as numerical and cast into Decimal type if, when returned,
TryParse() = True
            ' If, else logic for converted decimal to adhere to specified range
            If Double.TryParse(strElevation, dblElevation) And dblElevation >= -413.0 And dblElevation <= 9000.0 Then
                ' Convert sanitized string
                dblElevation = System.Convert.ToDecimal(strElevation)
                ' LoopValidation is True to break loop
                LoopValidation = True
                Console.ForegroundColor = ConsoleColor.Green
                Console.Write(vbCrLf & "OK!" & vbCrLf)
                Console.ResetColor()
            Else
                ' Visual error message for invalid responses
                Console.ForegroundColor = ConsoleColor.Red
                Console.Write("-------------------------------------------------------------" & vbCrLf)
                Console.Write("| Error! Please Valid Numeric Value from -413.00 to 9,000.09 |" & vbCrLf)
                Console.Write("-------------------------------------------------------------" & vbCrLf & vbCrLf)
                Console.ResetColor()
                ' Waits for user input of any keystroke
                Console.Write("Press any key to continue . . . ")
                Console.ReadKey(True)
                strElevation = ""
                Console.Clear()
                Console.Write("Please re-enter the elevation (m) " & vbCrLf)
                strElevation = Console.ReadLine()
                Continue Do
```

```
            End If
            ' Loop until Validator breaks loop
        Loop Until LoopValidation = True

        ' Height - Elevation cannot be lessthan or equal to 0 -- also ensures not dividing by 0
        If (dblHeight - dblElevation) <= 0 Then
            Console.ForegroundColor = ConsoleColor.Red
            Console.Write("--------------------------------------------------------------------------------" & vbCrLf)
            Console.Write("| Error! Height - Elevation cannot be less than 0, please re-enter values!|" & vbCrLf)
            Console.Write("--------------------------------------------------------------------------------" & vbCrLf &
vbCrLf)
            Console.ResetColor()
        Else
            ' Casting validated decimal inputs to integer
            intScale = CInt((dblHeight - dblElevation) / dblFocalLength)
            ' Display of the final map scale in formatted form
            Console.WriteLine(vbTab & vbTab & vbTab & "    The Map Scale is: ")
            Console.WriteLine(vbTab & vbTab & vbTab & "**********************")
            ' Displaying the scale with commas when larger numbers are present
            Console.WriteLine(vbTab & vbTab & vbTab & vbTab & "1:" & (intScale.ToString("N0")))
            Console.WriteLine(vbTab & vbTab & vbTab & "**********************")
        End If

        ' After the main code segment executes the user can quit or recalculate if the result was correct or if
invalid data was entered for elevation and height
        Console.Write(vbCrLf & vbCrLf & "------------------------------------------------------" & vbCrLf)
        Console.Write(vbTab & "Press")
        Console.ForegroundColor = ConsoleColor.DarkMagenta
        Console.Write(" Q ")
        Console.ResetColor()
        Console.Write("to quit or any key to recalculate")
        Console.Write(vbCrLf & vbCrLf & "------------------------------------------------------" & vbCrLf)

        ' Do loop will break and thus terminate the program
        If Console.ReadKey(True).KeyChar = "q"c Then
            Console.Write("END TEST" & vbCrLf)
            ' Loop validator to True to exit the program
            ExitProgram = True
        Else
            ' catch all to re-iterate through loop
            ExitProgram = False
        End If
    Loop Until ExitProgram = True

End Sub
End Module
```