Comparing Numbers with Logical and Relational Operators in VB.Net

Working with operators and conditional statements in a Windows Forms Application

In this tutorial we build a program that compares two user-entered numbers. We'll delve deeper into logical operators (AndAlso, OrElse) and relational operators (>, <, =) to construct conditional statements that evaluate the relationship between the numbers.

Operators

1. **Logical Operators** These operators combine conditions to create more complex expressions.
    a. **AndAlso**: Both conditions must be true for the entire expression to be true. (Think "AND")
    b. **OrElse**: At least one condition must be true for the expression to be true. (Think "OR")
    c. **Not**: Reverses the logical state of a condition. (Think "NOT")
2. **Relational Operators** These operators compare values and return true or false based on the relationship.
    a. > Greater than
    b. < Less than
    c. = Equal to
    d. >= Greater than or equal to

e. <= Less than or equal to

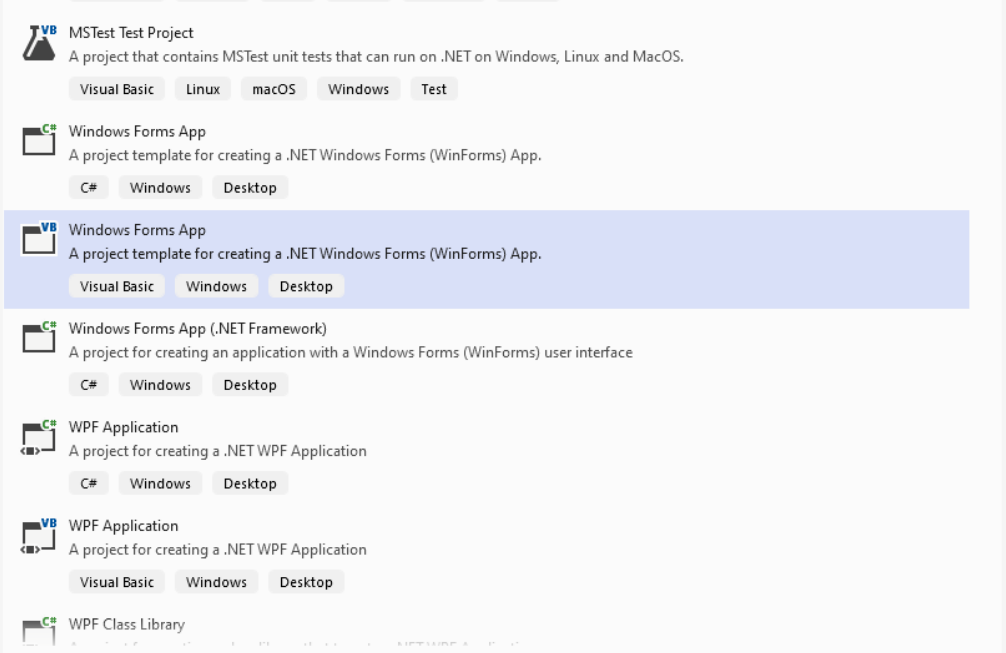
Setting up the project

Now that we have defined the building blocks, let us create our project.

Try It: Create VB Windows Forms Application.

1. We'll use Visual Studio to create and code our application. If you don't have it, download the free Community Edition from Microsoft Visual Studio Community.

2. Launch Visual Studio and create a new **Windows Forms App (.NET Framework)** project with VB.Net as the language.



Building the Number Comparison App

Our app will allow a user to enter their name, and the get a personalized greeting.

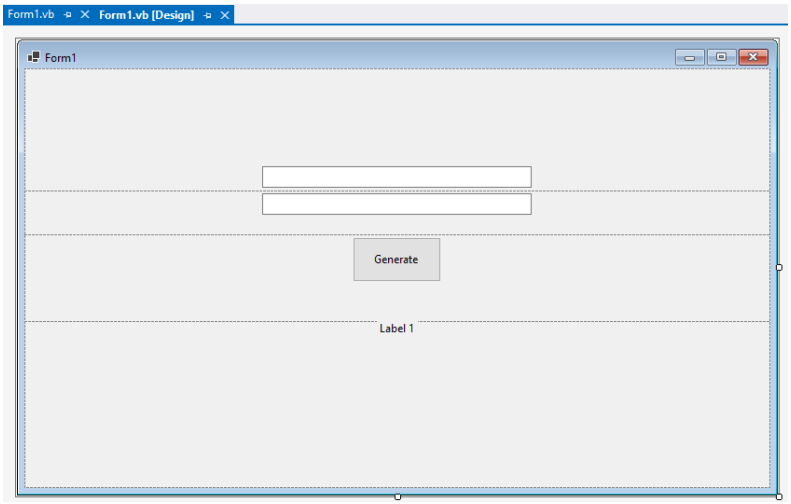
Try It: Create the Hello World Application.

1. The designer window shows the visual layout of your application. Here's what we'll add:

    a. Two TextBoxes Name them "txtNumber1" and "txtNumber2" to hold the numbers.

    b. A Button Name it "btnGenerate" to trigger the comparison.

    c. A Label Name it "lblEvaluation" to display the results.



2. Double-click the "btnGenerate" button to open the code window. This is where the magic happens! Write the following code:

```
Public Class Form1
    Private Sub btnGenerate_Click(sender As Object, e As EventArgs) Handles btnGenerate.Click
        Dim numberOne As Integer
        Dim numberTwo As Integer

        If Integer.TryParse(txtNumber1.Text, numberOne) AndAlso Integer.TryParse(txtNumber2.Text,
numberTwo) Then
            If numberOne <> numberTwo Then
                lblEvaluation.Text = "The two numbers are not equal" & Environment.NewLine
            End If
            If numberOne > numberTwo Then
                lblEvaluation.Text += numberOne & " is greater than " & numberTwo
            End If
            If numberOne < numberTwo Then
                lblEvaluation.Text += numberOne & " is less than " & numberTwo
            End If
        Else
            lblEvaluation.Text = "Invalid input. Enter integers"
        End If

    End Sub
End Class
```

Explanation

- We declare two integer variables, `numberOne` and `numberTwo`, to store the user-entered values.
- `Integer.TryParse(...)`: This method attempts to convert the text from the textboxes to integers. If successful, the converted values are stored in the variables, and True is returned. We use the `AndAlso` operator to ensure both conversions are successful before proceeding.
- The `If` statements create conditional blocks that execute based on the comparison results:
- We've added checks for >= (greater than or equal to) and <= (less than or equal to) using the respective relational operators.

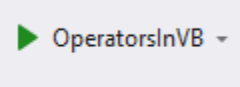
Try It: Add a conditional block that checks whether the two numbers are equal to each other

Why We Use the += Operator In our code, we use the += operator to concatenate (join) strings displayed in the `lblEvaluation` label. The += operator is a shorthand for assignment followed by addition. We leverage the += operator to concatenate (join) strings within the

lblEvaluation.Text property. Assuming lblEvaluation.Text initially contains an empty string (""), the first If statement (not equal) might assign "The two numbers are not equal" to lblEvaluation.Text. Subsequent If statements use +=. For instance, if the first number is greater, it adds " 10 is greater than 5" to the existing text inside the label. This allows us to display multiple statements after the comparison instead of only displaying one statement at a time.

Running the Application

Press **F5** or click the **Run** button in Visual Studio. This will compile and execute your code.



Enter your name in the textbox and click the "Generate" button.

The label should now display relationships that have been evaluated from comparison of the two numbers.