## MicroTest 2

Name:\_\_\_\_\_

## Part A [10 points]

- Operators such as >, >=, <, <=, Not, And, Or are all Boolean T / F operators.
- 2) The operands in a Boolean expression (such as x > y) should in principle both T / F be the same data type.
- 3) A Boolean expression such as  $\mathbf{x} = \mathbf{y}$ , where  $\mathbf{x}$  and  $\mathbf{y}$  are floating point values T / F resulting from division, might not be true when you expect it should be.
- When writing an If statement you must always use a comparison operator, as in T / F
   If a <= b Then ....</li>
- 5) An **If** statement always needs an **Else** block to deal with processing when the T / F Boolean expression evaluates to False.
- 6) In an **If** statement with many **Else If** blocks, where more than one Boolean T / F expression evaluates to True, only the last block where the condition is True is executed.
- 7) An **if** statement that has an **Else** block will always execute one of the blocks, **T** / F no matter what the values of data used in the program.
- 8) An expression can contain arithmetic, logical, and comparison operators. Indicate the order of precedence for these operators. Place the type that will be done first on the top line, the second group on the next line, ... logical

## Part B [10 points] -- based on Exercise 4-4: Improving the Guessing Game.

The chapter concludes with the following paragraph:

Notice that we have not tried to validate user input, and that consequently the program crashes if the user enters data that is not a number. Additionally the program allows the user to enter values less than 1 and greater than 100.

A third issue is that the program allows fractions in the guesses.

There is a categorical difference between *validating* user input and *processing* user input.

- Results of *processing* input are displayed in labels within the interface.
- Warnings generated by *validating* input should use dialog boxes.

To the sample solution on the next page, add code to most efficiently handle ONE of these:

- A. insuring that the guess has no illegal charaters, OR
- B. insuring that the guess is an integer, OR
- C. insuring that the guess is in the correct range

Make sure to place the code in the appropriate location(s).

## Marking Notes:

- 1 point for an If...Then...Else...End If structure
- 1 point if it's in the right location
- 3 points for appropriate condition (as indicated by <u>underscores</u> on the next page)
- 3 points for <a>MessageBox.Show( "an appropriate error message")</a>
- 1 point for setting the focus
- 1 point for selecting the text

```
Private Sub btnCheck Click(...) Handles btnCheck.Click
A 'Test the input to insure legal characters
  If <u>IsNumeric</u> (txtGuess .Text) Then
    'Declare local variable - for easy testing use the same integer type as theNumber
    Dim theGuess As Short
    'Collect guess from text box
    theGuess = CShort(txtGuess.Text)
в
    'Test that the guess is an integer
    If <u>theGuess</u> = <u>txtGuess.Text</u> Then 'MANY other tests for fractions are possible
      'Test that the guess is in the right range (including not negative)
С
      If theGuess >= 0 And theGuess <= 100 Then
        If theGuess = theNumber Then
          lblPrompt.Text = "You got it!"
          btnPlayAgain.Visible = True
          btnCheck.Enabled = False
          txtGuess.Text = vbNullString
          btnPlayAgain.Focus()
        Else
          'Decrement the guess counter
          guessCounter -= 1
          lblGuessesLeft.Text = CStr(guessCounter) & " guesses left"
          'Is the game over?
             If guessCounter = 0 Then
               lblPrompt.Text = "You blew it!"
               btnCheck.Enabled = False
               btnPlayAgain.Visible = True
               txtGuess.Text = vbNullString
               btnPlayAgain.Focus()
             Else
                'Is the guess too low?
                If theGuess < theNumber Then
                 lblPrompt.Text = "Too Low"
               Else
                  'The guess must be too high
                 lblPrompt.Text = "Too High"
               End If
              'In either case, prepare for a new guess
             txtGuess.Focus()
             txtGuess.SelectAll()
           End If
         End If
С
       Else
         MessageBox.Show("Must be between 0 and 100!!", "Bad Data")
         txtGuess.Focus()
         txtGuess.SelectAll()
       End If
в
     Else
       MessageBox.Show("INTEGER please!!", "Bad Data")
       txtGuess.Focus()
       txtGuess.SelectAll()
     End If
A Else
     MessageBox.Show("Enter a NUMBER", "Bad Data")
     txtGuess.Focus()
     txtGuess.SelectAll()
   End If
 End Sub
```