Test 1 Review

Short Answer Coding

Chapters 1, 2, 3
Assignment Statements for a Label or Text Box

- Write assignment statements for a label or text box.

MessageLabel.Text = “Hello, how are you?”

PromotionTextBox.Text = _
    “Buy one, get one for half price”

GreetingsTextBox.Text = NameTextBox.Text _
    & “: “ & “Welcome to CPT 186”
Selecting/Unselecting a Radio Button

- Set the Checked property for a radio button to be selected or unselected.

- Make a radio button selected
  RedRadioButton.Checked = True

- Make a radio button unselected
  RedRadioButton.Checked = False
Selecting/Unselecting a Check Box

- Set the Checked property for a check box to be checked or unchecked.

- Make a check box checked

  `VisibleCheckBox.Checked = True`

- Make a check box unchecked

  `VisibleCheckBox.Checked = False`
Set the Visibility of a Control

- Set the Visibility of a control so that it becomes visible or invisible.

- Make a control visible (appear)
  SloganLabel.Visible = True

- Make a control invisible (disappear)
  SloganLabel.Visible = False
Set the Visibility of a Control Based on a Check Box

- Set the Visibility of a control based on the value in a check box.

- If the check box is checked then the value of the Checked property is True and the control will be Visible. If the check box is unchecked, the value of the Checked property is False and the control will not be Visible.

  CompanyNameLabel.Visible = _
  CompanyCheckBox.Checked
Change the Text Color of a Text Box or Label

- Change the color of text in a text box or label.

- Make the text color of a text Box Blue
  AmountTextBox.ForeColor = Color.Blue

- Make the text color of a label Red
  MessageLabel.ForeColor = Color.Red

**Note that if the text box is ReadOnly, the color will not be able to be changed. If you want it to change color, change the ReadOnly text box to a label. The ForeColor property is the text color. The BackColor property is the background color.**
Display a Message in a MessageBox

- Display a message box given the message, title, buttons and icons to use.

- Display the message “Amount must be numeric” in a MessageBox. Use “Error in Data” for the title bar message, an OK button, and an Exclamation icon.

  MessageBox.Show(“Amount must be numeric”, _
  “Error in Data”, MessageBoxButtons.OK, _
  MessageBoxIcon.Exclamation)
Convert a TextBox Value to Integer or Decimal

- Convert a value from a text box to numeric (integer or decimal) using Integer.Parse or Decimal.Parse.
- Dim AmountDecimal as Decimal
- Dim QuantityInteger as Integer
- Convert to Decimal
  AmountDecimal = Decimal.Parse (AmountTextBox.Text)
- Convert to Integer
  QuantityInteger = Integer.Parse (QuantityTextBox.Text)
Use Try/Catch with `Integer.Parse`:

- Use the Try/Catch block with numeric conversion to display an error message if the conversion causes an error.
- Convert the value in `PointsTextBox` to `Integer`. If the value is not valid, display the message “Value must be numeric”.

```
Try
    PointsInteger = _
    Integer.Parse(PointsTextBox.Text)
    'Value is valid – processing statements go here

Catch PointsException as FormatException
    MessageBox.Show("Value must be numeric")
    With PointsTextBox
        .Focus()
        .SelectAll()
    End With
End Try
```
Use Try/Catch with Decimal.Parse

- Use the Try/Catch block with numeric conversion to display an error message if the conversion causes an error.
- Convert the value in PriceTextBox to Decimal. If the value is not valid display the message “Price must be numeric”

```csharp
Try
  PriceDecimal = _
  Decimal.Parse(PriceTextBox.Text)
  'Price is valid – processing statements go here

Catch PriceException as FormatException
  MessageBox.Show("Price must be numeric")
  With PriceTextBox
    .Focus()
    .SelectAll()
  End With
End Try
```
Coding Portion of the Test (20 pts)
Be able to use these controls on a form

- Labels
- TextBoxes
- Buttons
- RadioButtons
- CheckBoxes
- PictureBoxes
- Group Boxes

- Remember when you refer to a control within the code you MUST change the name from the default. Add the name of the control type to the end of the name.
- EX: OutputLabel, NameTextBox, RedRadioButton, VisibleCheckBox, ExitButton, LogoPictureBox
Setting the Property Values

- You will be given a chart, similar to the ones for your Hands On, that contains the control names, and the properties values that need to be set.
- **NOTE: The output will be displayed in a Label instead of a ReadOnly TextBox because you will be changing the color of the Label during run time and the ReadOnly TextBox does not allow you to change the color.
- Be sure to set the properties of each control accurately by following the table.
- For the Form, be sure you can set the Name, AcceptButton, CancelButton, Text and StartPosition properties.
Setting the Property Values

- For Labels, be sure that you can set the Name, Text, ForeColor, and TextAlign properties.
- For TextBoxes that will be used for input, be sure that you can set the Name, Text, and TextAlign properties.
- For GroupBoxes, be sure you can set the Name and Text Properties.
- For CheckBoxes and RadioButtons, be sure you can set the Name, Text, and Checked properties.
- For Buttons, be sure you can set the Name and Text properties and set up an Access key by using the &
Coding for Your Controls

- Be able to code the CheckedChanged event for the RadioButtons so that when selected, the text color of the SquareLabel used for output will be changed to the selected color. Remember to use the ForeColor property.
  - SquareLabel.ForeColor = Color.Red
- Be able to code the CheckedChanged event for the CheckBoxes so that when checked, the corresponding TextBox or Label will be Visible (Visible property set to True) and when Unchecked, the corresponding TextBox or Label will be Invisible (Visible property set to False). See p. 88 in your book.
  - InputTextBox.Visible = InputCheckCheckBox.Checked
Coding for Your Controls

- Be able to code the Button_Click event for the EnterButton that will:
  1) Use a Try/Catch and a Decimal.Parse to convert the value from a InputTextBox to Decimal storing it in a decimal variable. Remember to add the word Decimal to the end of the variable name.
  2) Calculate the square of the number by multiplying the number by itself and storing the result in another Decimal variable (adding Decimal to the end of the name).
  3) Display the square value you calculated above in the SquareLabel used for output using the Number format with 2 decimal positions. (Use .ToString(“N”))
4) If the Decimal.Parse causes an error, add a Catch at the end, just above the End Sub to display an error message. (Catch InputException as FormatException). Make this a FormatException and display the error message “Enter numeric data only”, with “Data EntryError” for the title bar, an OK button and an Exclamation icon using MessageBox.Show. Set the .Focus() back to the InputTextBox and .SelectAll() to select any bad data.

5) Before the End Try, add a 2nd Catch to catch all of the other errors using Catch AnyException as Exception. Use “Error: “ & AnyException.Message as the error message in MessageBox.Show.

6) Add the End Try right above the End Sub
Coding for Your Controls

- Be able to code the Button_Click event for the ExitButton. It should close the project with Me.Close()
- Go back to the top of your program and add 4 comments to document the Project name, Programmer, Date, and Description.
- Make sure that Option Strict On is above the Public Class statement
- Make sure you add at least 1 comment to each event procedure to describe what that procedure does
Testing Your Code for Correctness

- After completing your project, test to see if it produces the correct output.
  1) Run the project and click the Enter button without entering any values at all. You should get the error message “Enter numeric data only “ and the focus should go back to the input text box.
  2) Enter the value 2 and click the Enter button. The value 4.00 should display in the output label as Red.
  3) Click each of the Radio Buttons to change the text color to each of the colors then back to Red.
  4) Uncheck each of the check boxes and see if the corresponding text box or label disappears when unchecked. Click the check boxes again to check them and see if the corresponding text box or label reappears .
  5) Click the Exit button and see if the project ends.