CPT 186 TEST 2 REVIEW
Chapters 4 and 5
IF-THEN-ELSE STATEMENTS

- Write the code for the different versions of the if-then-else statement. P. 156-157, 166-167
- If-Then statement
  If AmountDecimal > 0 Then
    TotalDecimal += AmountDecimal
  End If

- If-Then-Else statement
  If Over21RadioButton.Checked Then
    AdultCountInteger += 1
  Else
    MinorCountInteger += 1
  End If
**IF-THEN-ELSE (CONTINUED)**

- If-ElseIf statement
  - If RedRadioButton.Checked Then
    - MessageLabel.Forecolor = Color.Red
  - ElseIf BlueRadioButton.Checked Then
    - MessageLabel.Forecolor = Color.Blue
  - ElseIf GreenRadioButton.Checked Then
    - MessageLabel.Forecolor = Color.Green
  - Else
    - MessageLabel.Forecolor = Color.Black
  - End If

- NOTE: If-ElseIf uses only 1 End If
IF-THEN-ELSE (CONTINUED)

- Nested If-Then-Else statements
- If TemperatureDecimal > 80 Then
  - If TemperatureDecimal > 90 Then
    - MessageLabel.Text = “It is very hot today”
  - Else
    - MessageLabel.Text = “It is a nice warm day”
  - End If
- Else
  - MessageLabel.Text = “It is getting colder “
- End If

- NOTE: Uses an End If for each If statement
IF-THEN-ELSE (CONTINUED)

- Nested If-Then-Else statements
- If HoursDecimal <= 40 Then
  NetDecimal = HoursDecimal * RateDecimal
  Else
  If HoursDecimal > 50 Then
    NetDecimal = HoursDecimal * (2 * RateDecimal)
  Else
    NetDecimal = HoursDecimal * (1.5 * RateDecimal)
  End If
End If

- NOTE: Uses an End If for each If statement
Using If-ElseIf with Radio Buttons

- Write the code to use nested if-else-if statements to test the values in a set of radio buttons and based upon the results, perform an action. P. 168-169
- If FreshmanRadioButton.Checked Then
  FreshmanCountInteger += 1
ElseIf SophomoreRadioButton.Checked Then
  SoftmoreCountInteger += 1
ElseIf JuniorRadioButton.Checked Then
  JuniorCountInteger += 1
Else
  SeniorCountInteger += 1
End If
Using If-Then-Else with Check Boxes

- Write the code to use a separate if statement to test the value in multiple check boxes and based upon the results, perform an action. P. 169-170

- If CompanyCheckBox.Checked Then
  CompanyLabel.Visible = _
  CompanyCheckBox.Checked
End If

- If LogoCheckBox.Checked Then
  LogoPictureBox.Visible = _
  LogoCheckBox.Checked
End If

- NOTE: Each CheckBox has its own If and End If
CONVERT TO AN INTEGER VALUE

- Write the code to validate a value from a text box to see if it is numeric using Parse and Try/Catch and display an error message if it is not. P. 176

- Try

  ```csharp
  QuantityInteger = Integer.Parse( _
      QuantityTextBox.Text)
  AmountDueDecimal = QuantityInteger *
      11.95D
  AmountTextBox.Text = _
      AmountDecimal.ToString("C")
  ```

  Catch AmountException as FormatException
  ```csharp
  MessageBox.Show("Quantity must be numeric", _
      "Error", MessageBoxButtons.OK, _
      MessageBoxIcon.Exclamation)
  ```

End Try
CONVERT TO UPPER OR LOWER CASE

- Write the code to convert a string to upper or lower case. p. 162-163, 178
- If ResponseTextBox.Text.ToLower = “yes” Then
  TuitionLabel.Text = SemesterCostDecimal. _
  ToString(“C”)
Else
  TuitionLabel.Text = ZeroDecimal. _
  ToString(“C”)
End If

- If LanguageTextBox.Text.ToUpper = “BASIC” _
  Then
  MessageLabel.Text = _
  “You are programming in Visual Basic!”
End If
ENABLE OR DISABLE A MENU SELECTION

- Write the code to enable or disable a menu selection. P. 217
- Enable
  - SummaryToolStripMenuItem.Enabled = True
- Disable (grayed out)
  - SummaryToolStripMenuItem.Enabled = False
CHECKING AND UNCHECKING A MENU SELECTION

- Write the code for a menu selection that uses the Checked property. When the menu is checked and the selection is clicked, uncheck the menu and make a control invisible. Otherwise, check the menu and make a control visible. P. 217-218

- If VisibleToolStripMenuItem.Checked then
  VisibleToolStripMenuItem.Checked = False
  MessageLabel.Visible = False

  Else
  VisibleToolStripMenuItem.Checked = True
  MessageLabel.Visible = True

End If
CHANGE THE COLOR USING COLORDIALOG1

- Write the code for a menu selection that changes the color of the text in a label or text box using the ColorDialog1 box. Remember that you must include 3 steps in your code: p. 242

- Private Sub ColorToolStripMenuItem_Click(ByVal sender as System.Object, ByVal e as System.EventArgs) _ Handles ColorToolStripMenuItem.Click
  'Change the color of SloganLabel
  With ColorDialog1
    .Color = SloganLabel.ForeColor 'Initialize Color
    .ShowDialog()    'Display the color dialog
    SloganLabel.ForeColor = .Color    'Change the Color
  End With
End Sub
**CHANGE THE FONT USING FONTDIALOG1**

- Write the code for a menu selection that changes the font properties of the text in a label or text box using the FontDialog1 box. Remember that you must include 3 steps in your code: p. 241

- Private Sub FontToolstripMenuItem_Click(ByVal sender as System.Object, ByVal e as System.EventArgs) Handles FontToolstripMenuItem.Click
  'Change the font of SloganLabel
  With FontDialog1
    .Font = SloganLabel.Font 'Initialize Font
    .ShowDialog() 'Display the font dialog
    SloganLabel.Font = .Font 'Change the Font
  End With
End Sub
WRITE A GENERAL SUBPROCEDURE WITH ARGUMENTS

- Write a sub procedure that calculates a value using arguments passed to the procedure and displays the calculated value in a label. P. 226-227

- Private Sub CalculatePay(ByVal HoursDecimal As Decimal, ByVal PayDecimal As Decimal)
  
  ‘Calculate Gross Pay
  Dim GrossDecimal As Decimal
  GrossDecimal = HoursDecimal * PayDecimal
  GrossTextBox.Text = GrossDecimal.ToString("C")

End Sub
WRITE A FUNCTION PROCEDURE WITH ARGUMENTS

- Write a function procedure that calculates a value using arguments passed to the function and returns the calculated value back to the calling procedure. P. 227-229

- Using the Return statement

- Private Function CalculatePay(ByVal HoursDecimal As Decimal, ByVal PayDecimal As Decimal) As Decimal
  - 'Calculate Gross Pay
  - Dim GrossDecimal As Decimal
  - GrossDecimal = HoursDecimal * PayDecimal
  - Return GrossDecimal
  End Function
Alternate code for Function procedure that uses the Return statement

Private Function CalculatePay(ByVal HoursDecimal As Decimal, ByVal PayDecimal As Decimal) As Decimal
    'Calculate Gross Pay
    Return HoursDecimal * PayDecimal
End Function
Alternate code for a Function Procedure with value returned in a variable with the same name as the function name

Private Function **CalculatePay**(ByVal _ HoursDecimal As Decimal, ByVal PayDecimal _ As Decimal) As Decimal
    ‘Calculate Gross Pay
    **CalculatePay** = HoursDecimal * PayDecimal
End Function
Coding Section for Test 2

Given a program that has already been started for you, you should be able to:

- Download and unzip the folder and open the project in Visual Studio
- Add a MenuStrip component to the component tray and set up the menu given the names for the menu items and their access keys
- Set the forms Startup Position to CenterScreen
- Change to the coding window and add comments above the Public Class statement to document the project name, the programmer, the date, and a brief description
CODING FOR TEST 2 (CONTINUED)

- Add the statement above the Public Class statement to turn Option Strict On
- Add 2 module level integer variables, ReaderCountInteger and TotalBooksInteger, used to accumulate and count summary values given the names of the variables. Do not use module level variables for any other variables. Make them local within the procedure where they are used.
- Copy and Paste the code for the Exit, Clear, About and Points Button_Click events to the corresponding MenuItem_Click events. Then delete the Button_Click events and the Buttons for those that you copied and pasted.
CODING FOR TEST 2 (CONTINUED)

- In Help/About event procedure replace my name with your name.
- Go to the PointsToolstripMenuItem_Click event and add the code to update the module level variables you created where indicated in the code. Add 1 to ReaderCountInteger using += 1 and add BooksInteger to TotalBooksInteger using += BooksInteger.
  - Go to the end of the PointsToolstripMenuItem_Click event and press Enter twice to start the code for your function. Write the function header:
    Private Function CalculatePoints(ByVal BooksInteger as Integer) As Integer
  - Press Enter and you will get the End Function statement for the function
  - Add a comment telling what the procedure does
CODING FOR TEST 2 (CONTINUED)

- Declare the Integer variable PointsInteger with a Dim statement.
- In the Points menu item click event, find the nested If/ElseIf statements that calculate PointsInteger for the reader based on the number of books. Cut and paste it to the CalculatePoints function to the line after your variable declaration.
- Add a return statement after the If statements to return the points using: Return PointsInteger.

- Go back to the PointsToolStripMenuItem_Click event to the place where you cut the calculations. Place a call to the function here, returning the value to the PointsInteger variable.

PointsInteger = CalculatePoints(BooksInteger)
Write the File/Summary event procedure.

- Add a comment describing what the procedure does
- Declare the Decimal variable AverageDecimal using Dim
- Create an If statement to display the summary only if the ReaderCountInteger>0.
- On the Then branch, calculate the average using the statement: `AverageDecimal = _ Convert.ToDecimal(TotalBooksInteger/ReaderCountInteger)`
- Display the message: “Average Number of Books Read: “ concatenated with AverageDecimal.ToString(“N”)
- On the Else branch for ReaderCountInteger>0, display an error message (“No Data to Summarize”) with a MessageBox.Show.
- End with an End If
Coding for Test 2 (continued)

- Write the code for the Edit/Color event procedure using code similar to Slide 12. You will change the Forecolor of BonusLabel instead of SloganLabel used in that procedure.
- Write the code for the Edit/Font event procedure using code similar to Slide 13. You will change the Font of BonusLabel instead of SloganLabel used in that procedure.
- Follow the remainder of the instructions to finish your program and test it.