**Unit 4 Case Study**

**The Calculator**

Case studies are used to learn problem solving techniques. This case study illustrates the sequence of steps and thinking that goes into the construction of a substantial application. Learning to program is much more than simply learning a particular language. It is learning a process for creating an application.

The four steps in creating an application are 1. specifications, 2. design, 3. coding and 4. testing and debugging. In this first case study, each of these steps is explained.

**Step Number 1: Specification**

The first step in creating an application is clearly defining what the application is to accomplish. This definition is called the specification, or ***spec***, because it specifies what the application should do. In real world situations, the specification is developed by talking with the application user and other computer professionals. In this case study the specifications will be provided to you.

The specifications for this case study are:

The calculator application prompts the user for two numbers (operands) and then displays the result of an expression formed with the operands and operator (^, \*, /,\, Mod, +,-) selected by the user. Selecting an operator automatically displays the result of the expression.

**Sept Number 2: Design**

Application design includes how the interface looks and how the code is written to accomplish the specification. The best way to design the interface is to draw interface designs on paper. The code design is a description of each object’s event procedures.

The interface design for this Case Study is:

First Operand

Second Operand

Select and Operator:

^ \* / \ Mod + -

Result:

The program code design describes how to accomplish the spec. The code design for this case study is:

Global variables of type Single can be used to store the values entered by the user and the value of the expression. This is good programming style because all the radio button Click event procedures evaluates an expression and then assigns the value to a label. TextChanged event procedures will be coded to clear the radio buttons and label.

**Step Number 3: Coding**

Coding is creating the interface and writing the program code. The interface and code for this case study are:

|  |  |  |
| --- | --- | --- |
| **Object** | **Name** | **Text** |
| Form 1 |  | Calculator |
| Label1 | lblOp1Prompt | First Operand: |
| Label2 | lblOp2Prompt | Second Operand: |
| TextBox1 | txtOperand1 | *Empty* |
| TextBox2 | txtOperand2 | *Empty* |
| GroupBox1 | grpOperators | Select and Operand |
| RadioButton1 | radExponentiation | ^ |
| RadioButton2 | radMultiplication | \* |
| RadioButton3 | radDivision | \ |
| RadioButton4 | radIntDivision | / |
| RadioButton5 | radModDivision | Mod |
| RadioButton6 | radAddition | + |
| RadioaButton7 | radSubtraction | - |
| Label3 | lblResult | Result: |
| Label4 | lblExpressionValue | *Empty* |

**Code**

Public Class Form1

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

Private Sub txtOperand1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtOperand1.TextChanged

Me.radAddition.Checked = False

Me.radDivision.Checked = False

Me.radExponentiation.Checked = False

Me.radMultiplication.Checked = False

Me.radIntDivision.Checked = False

Me.radModDivision.Checked = False

Me.radSubtraction.Checked = False

Me.lblExpressionValue.Text = Nothing

End Sub

Private Sub txtOperand2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtOperand2.TextChanged

Me.radAddition.Checked = False

Me.radDivision.Checked = False

Me.radExponentiation.Checked = False

Me.radIntDivision.Checked = False

Me.radModDivision.Checked = False

Me.radMultiplication.Checked = False

Me.radSubtraction.Checked = False

Me.lblExpressionValue.Text = Nothing

End Sub

Private Sub radExponentiation\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radExponentiation.CheckedChanged

Dim sngAnswer As Single

sngAnswer = Val(Me.txtOperand1.Text) ^ Val(Me.txtOperand2.Text)

Me.lblExpressionValue.Text = sngAnswer

End Sub

Private Sub radMultiplication\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radMultiplication.CheckedChanged

Dim sngAnswer As Single

sngAnswer = Val(Me.txtOperand1.Text) \* Val(Me.txtOperand2.Text)

Me.lblExpressionValue.Text = sngAnswer

End Sub

Private Sub radDivision\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radDivision.CheckedChanged

Dim sngAnswer As Single

sngAnswer = Val(Me.txtOperand1.Text) \ Val(Me.txtOperand2.Text)

Me.lblExpressionValue.Text = sngAnswer

End Sub

Private Sub radIntDivision\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radIntDivision.CheckedChanged

Dim sngAnswer As Single

sngAnswer = Val(Me.txtOperand1.Text) / Val(Me.txtOperand2.Text)

Me.lblExpressionValue.Text = sngAnswer

End Sub

Private Sub radModDivision\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radModDivision.CheckedChanged

Dim sngAnswer As Single

sngAnswer = Val(Me.txtOperand1.Text) Mod Val(Me.txtOperand2.Text)

Me.lblExpressionValue.Text = sngAnswer

End Sub

Private Sub radAddition\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radAddition.CheckedChanged

Dim sngAnswer As Single

sngAnswer = Val(Me.txtOperand1.Text) + Val(Me.txtOperand2.Text)

Me.lblExpressionValue.Text = sngAnswer

End Sub

Private Sub radSubtraction\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles radSubtraction.CheckedChanged

Dim sngAnswer As Single

sngAnswer = Val(Me.txtOperand1.Text) - Val(Me.txtOperand2.Text)

Me.lblExpressionValue.Text = sngAnswer

End Sub

Private Sub ExitToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click

Application.Exit()

End Sub

End Class

**Step Number 4: Testing and Debugging**

Testing is the process of running the application and entering data to test different possibilities to reveal any bugs. Debugging is the process of getting an application to work correctly.

This case study should be tested by entering values that are positive, negative, and zero. What will happen when 0 is entered in the second text box?

**Getting credit for this assignment requires you demonstrating the code functionality to Mr. Pluchino and getting “ticked off”.**