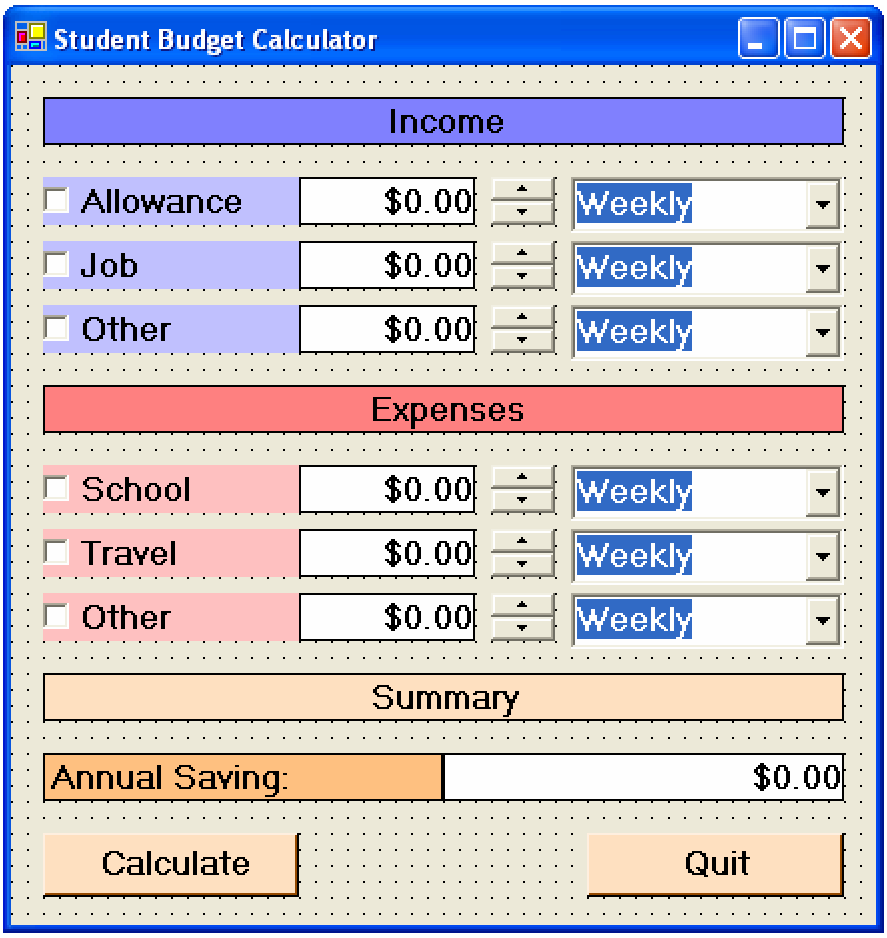
Student Budget Calculator

Problem Description: Design a program that can be used by a student to keep track of their expenses in terms of school based and outside activities. The program should be intuitive, easy to use and should provide for quick data entry and validation of data. The program should already contain common options for the student to include in their budget, but could also offer the facility for the student to add expenses of their own. The program should also allow the student to view summaries of their expenses by specific time periods, such as weekly, monthly and yearly.

Programming Level: Beginner

Skills Covered: Check boxes, Combo boxes, Vertical scroll bars.

This problem can be approached in a variety of different ways. What follows is a suggestion for one program that focuses on up to three sources of income and three types of expenses. You could create this program and then add other elements using a similar coding style if you wish. This program also presents a summary of the user's expenses calculated annually. Summaries for other time periods could easily be added using similar techniques to those shown.



□ Allowance	$0.00	Weekly
□ Job	$0.00	Weekly
□ Other	$0.00	Weekly

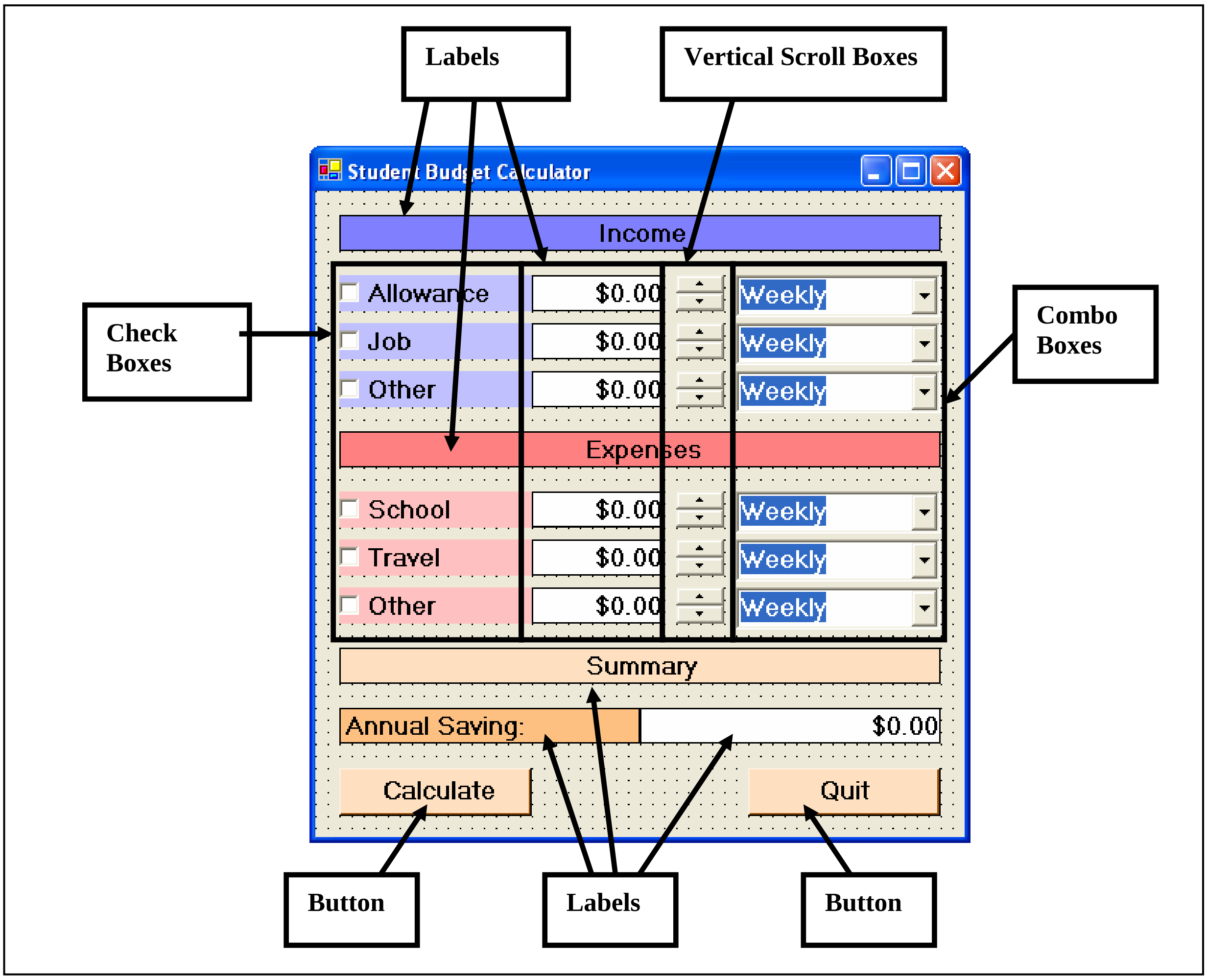
□ School	$0.00	Weekly
□ Travel	$0.00	Weekly
□ Other	$0.00	Weekly

Annual Saving: $0.00

Setting up the objects for our program

Follow these steps to create the program.

1. Start a new Visual Basic.NET program called ‘Student Budget’. As the layout of this program is quite complicated, you may like to refer to the diagram below to see where specific objects need to be placed.



2. Select the Label tool from the toolbox. Use it to draw a rectangular box in the center and at the top of the form. Stretch this label out so that it is almost the width of the entire form. This will be our first heading. Name it ‘lblHeading1’. Place two more labels near the middle and the bottom of the form, the same size as the first. These will also be headings for different sections. Name them ‘lblHeading2’ and ‘lblHeading3’ respectively.

3. Underneath the first label, place three smaller labels in a column. These labels will hold currency amounts for the student’s income. Rename them ‘lblIncome1’, ‘lblIncome2’ and ‘lblIncome3’. Three more labels need to be placed underneath the second heading label ‘lblHeading2’. Make sure they are the same size and column position as the labels you just placed. Rename these labels ‘lblExpenses1’, ‘lblExpenses2’, and ‘lblExpenses3’.

4. Underneath the third heading label ‘lblHeading3’, place two more labels side by side. Name these ‘lblHeading4’ and ‘lblAnnual’.

5. Use the Button tool to draw a button in the bottom left hand corner of the form. Name the button ‘btnCalculate’. This button will be used to perform the calculations required.

6. Use the Button tool to draw a button in the bottom right hand corner of the form. Name the button ‘btnQuit’. This button will be used to end the program.

7. Use the CheckBox tool to add two groups of three check boxes (three underneath ‘lblHeading1’ and three underneath ‘lblHeading2’ as shown in the diagram). These check boxes will allow the user to select or deselect various items. Name the first three ‘chbIncome1’, ‘chbIncome2’ and ‘chbIncome3’ respectively. Name the second group of three ‘chbExpense1’, ‘chbExpense2’ and ‘chbExpense3’.

8. Just as with the check boxes, use the VerticalScrollBar tool to add two groups of three vertical scroll bars (three underneath ‘lblHeading1’ and three underneath ‘lblHeading2’ as shown in the diagram). These vertical scroll bars will allow the user to modify the values contained in the labels beside each control. Name the first three ‘vsbIncome1’, ‘vsbIncome2’ and ‘vsbIncome3’ respectively. Name the second group of three ‘vsbExpense1’, ‘vsbExpense2’ and ‘vsbExpense3’.

9. Once again, but this time using the ComboBox tool, add two groups of three combo boxes (three underneath ‘lblHeading1’ and three underneath ‘lblHeading2’ as shown in the diagram on the right hand side of the form). These combo boxes will allow the user to select the time period to use for each item in the budget. Name the first three ‘cmbIncome1’, ‘cmbIncome2’ and ‘cmbIncome3’ respectively. Name the second group of three ‘cmbExpense1’, ‘cmbExpense2’ and ‘cmbExpense3’.

Changing the properties of the objects

Click on each ‘object’ in turn and set the following properties. Note that properties such as colours and fonts have not been prescribed, but should be set according to your own preferences.

Form1

StartPosition: CenterScreen
Text: Student Budget Calculator
FormBorderStyle: FixedSingle

By setting the form’s properties in this way, the form will appear in the centre of the screen and will not be able to be resized by the user. The form will also display the title ‘Student Budget Calculator’ at the top.

lblHeading1

Text: Income
TextAlign: MiddleCenter
BorderStyle: FixedSingle
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

lblHeading2

Text: Expenses
TextAlign: MiddleCenter
BorderStyle: FixedSingle
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

lblHeading3

Text: Summary
TextAlign: MiddleCenter
BorderStyle: FixedSingle
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

lblHeading4

Text: Annual Saving:
TextAlign: MiddleLeft
BorderStyle: FixedSingle
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.
the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

lblIcome1 – lblIncome3, lblExpense1 – lblExpense3, lblAnnual

Text: $0.00
TextAlign: MiddleRight
BorderStyle: FixedSingle
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

All of the labels (besides the ones used for the headings), will be used to hold dollar amounts. For that reason, they have been given a default value of '$0.00' and have been right aligned.

chbIncome1

Text: Allowance
TextAlign: MiddleLeft
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

chbIncome2

Text: Job
TextAlign: MiddleLeft
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

chbIncome3

Text: Other
TextAlign: MiddleLeft
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

chbExpense1

Text: School
TextAlign: MiddleLeft
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

chbExpense2

Text: Travel
TextAlign: MiddleLeft
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

chbExpense3

Text: Other
TextAlign: MiddleLeft
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

These check boxes will allow the user to select or deselect each of these options respectively. When the calculate button is pressed, we will write code that will only include those items which are checked, in the total amount. Ensure that each check box is wide enough so that the text is displayed correctly. Also ensure that all of the check boxes are lined up vertically.

vsbIncome1 – vsbIncome3, vsbExpense1 – vsbExpense3

Value: 0
Maximum: 5000
SmallChange: 10

Vertical scroll bars can be used to move up and down through a range of values specified by the minimum and maximum properties. In this case, the range will be from 0-5000. The starting value of the vertical scroll bar is given by the Value property, which in this case is set to 0. The SmallChange property can be used to change the value by which numbers are modified when the vertical scroll bar is clicked on. In this case, we want the values to change in increments of 10 cents for ease of use. Remember, you can always modify these setting later to allow for a greater range or more accuracy if required.

cmbIncome1 – cmbIncome3, cmbExpense1 – cmbExpense3

Items: Add ‘Daily’, ‘Weekly’, ‘Monthly’, ‘Annually’
Text: Weekly
Set the Font and BackColor properties yourself, so that the heading is colourful and easy to read.

A combo box provides a method by which a user can select from a collection of items. The collection is entered into the combo box using the items property, placing one item on each line. The default property of the combo box is then set by placing one of the item names in the Text property. In this case, our default property will be ‘weekly’.

btnCalculate

Text: Calculate
TextAlign: MiddleCenter
Set the Font and BackColor properties yourself, so that the button is colourful and easy to read.

btnQuit

Text: Quit
TextAlign: MiddleCenter
Set the Font and BackColor properties yourself, so that the button is colourful and easy to read.

These buttons will be used to perform the summary calculation and to quit the program respectively.

Adding code to our program

1. The first step in creating the code for this task will be programming the vertical scroll boxes to change their respective labels.

   Double click on the first vertical scroll bar ‘vsbIncome1’ and type in the following code. Note that the ‘Private Sub’ and ‘End Sub’ lines appear automatically and do not need to be typed in.

   ```
   Private Sub vsbIncome1_Scroll(ByVal sender As System.Object, _
         ByVal e As System.Windows.Forms.ScrollEventArgs) Handles _
         vsbIncome1.Scroll

         lblIncome1.Text = Format(vsbIncome1.Value / 100, "currency")

   End Sub
   ```

   When either button of the vertical scroll bar is clicked, the value will be placed into the corresponding label. To display the value in currency layout, it is divided by 100 and then formatted using the ‘Format’ command.

   Type code into each of the vertical scroll bars, so that they change their respective labels in the same way.

2. Switch back to designer, and double click on the calculate button. Enter the following code:

   ```
   Private Sub btnCalculate_Click(ByVal sender As System.Object, ByVal _
         e As System.EventArgs) Handles btnCalculate.Click

         Dim Total As Decimal = 0.0
         If chbIncome1.Checked = True Then
               If cmbIncome1.Text = "Daily" Then Total = Total + _
                     (vsbIncome1.Value / 100 * 365)
               If cmbIncome1.Text = "Weekly" Then Total = Total + _
                     (vsbIncome1.Value / 100 * 52)
               If cmbIncome1.Text = "Monthly" Then Total = Total + _
                     (vsbIncome1.Value / 100 * 12)
               If cmbIncome1.Text = "Annually" Then Total = Total + _
                     (vsbIncome1.Value / 100)
         End If
         lblAnnual.Text = Format(Total, "Currency")

   End Sub
   ```

   The variable ‘Total’ is declared at the start of this code, and is a local variable that will hold the total of the various items in the budget as they are added up. It is declared as a decimal type and given the starting value of ‘0.0’.

   The next line checks to see if the first check box is in fact selected. If it is, the amount that is in the corresponding label is added to it, after being multiplied by the appropriate time period. Note that when comparing the selected time periods, the strings are case sensitive.

   The last line of code places the value of the variable ‘Total’ into the label which is being used to display the total. Once again, it is converted into currency format, to ensure that the value is displayed with 2 decimal places and a dollar sign.

3. Copy and paste the above code so that all of the income and expenses options have been covered. Note that when copying this code for the expenses, that the values need to be subtracted from the total rather than added.

4. Type the following code into the click event of the button ‘btnQuit’, which will end the program:

   ```
   Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e _
         As System.EventArgs) Handles btnQuit.Click

       End

   End Sub
   ```

Testing the program

Run and test the program and ensure that it behaves as expected. Test that each of the options can be selected and deselected, and note the effect that this has on the final calculation. Ensure that each of the vertical scroll bars changes the correct value and that changing the time periods using the combo boxes also works correctly.

Further ideas to develop

- Add more income and expenses items.
- Add the facility for the user to rename the items that are under each category.
- Change the incremental values of the vertical scroll bars or allow the user to set the maximum and minimum values.