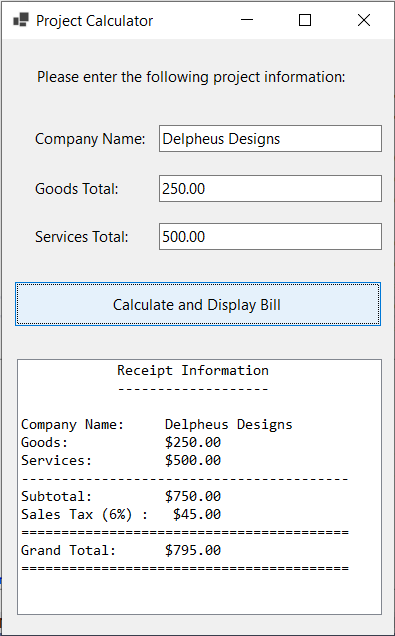
***Sample Program Documentation and Layout***

Some people are interested in seeing what a well-commented program looks like and the order in which to put in the various pieces. Here is a simple project calculator which will add up the goods and services expended on a project, calculate Michigan sales tax on the subtotal and then calculate a grand total (goods + services + sales tax). Here’s the form with the application in use:



One thing that I do, which I don’t require from you, but you will see it in the code is that all of the functions and subs are arranged in order such that can item which is called, is located above where the call takes place. In other words if A calls B, in scroll from the top of the program to the bottom, you would encounter the code for B before you would run into A.

You might not like the level of commenting that I expect, but you have to admit that when you are done reading it, there’s not one thing left that hasn’t been discussed. This allows much easier maintenance and modification of the system by people other than the original author(s) – and that’s the whole point! I shouldn’t have to pay twice for duplicate effort! If things are documented properly in the first place, anyone who come along after the fact should have a whole bunch easier time figuring out **what** and **how** the system does what it does.

I also understand that this is a very simple program and the documentation seems like overkill. It’s arguable that perspective is true. However, I can also tell you first hand how many simple programs that would only be used temporarily became permanent solutions and amalgamated into unmaintainable behemoths that no one wanted to touch. Imagine if all programs, simple and complex, were always fully documented. We would save a lot of money over time; since software is expensive to build (and maintain), anything that we can do to reduce effort and save a few dollars is well spent.

Here’s the fully commented program – also notice the order of items such as constants, structures, variables and subprograms. I put in a few page breaks to allow you to neatly view this in Word:

Public Class frmMain

'------------------------------------------------------------

'- File Name : frmMain.frm -

'- Part of Project: ProjectCalculator -

'------------------------------------------------------------

'- Written By: Scott D. James -

'- Written On: January 12, 20XX -

'------------------------------------------------------------

'- File Purpose: -

'- -

'- This file contains the main form for the entire appli- -

'- cation. All user input is gathered on this form. The -

'- calculations which are performed by the application -

'- reside in this file as well. Finally all generated -

'- output is contained here too. -

'------------------------------------------------------------

'- Program Purpose: -

'- -

'- The purpose of this program is to handle the calculations-

'- and bill generation for a project carried out by our -

'- company, which we give to our client. Project costs are -

'- made up of goods and services. These two amounts, along -

'- with the name of the company for whom the project was -

'- completed, will be entered by a user. Once all data has -

'- been input, the user will be able to instruct the program-

'- to calculate (1) a subtotal for the amounts, (2) Michigan-

'- state sales tax applied to the subtotal and (3) a grand -

'- total which consists of the subtotal and sales tax added -

'- together. The program will then display a neatly -

'- formatted receipt. -

'------------------------------------------------------------

'- Global Variable Dictionary (alphabetically): -

'- aCompanyBill - A user defined structure which will hold –

'- the company name along with all numeric -

'- values for the project currently being -

'- calculated. -

'------------------------------------------------------------

'---------------------------------------------------------------------------------------

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'--- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS --- GLOBAL CONSTANTS ---

'---------------------------------------------------------------------------------------

'FYI - Constants are not placed in the Global Variable Dictionary above because

'they are not variables. If you want to provide a comment in for a constant, make

'it inline as shown on the next line with the constant declaration...

Const MI\_SALES\_TAX = 0.06 'State of Michigan sales tax rate

'-------------------------------------------------------------------------------------------

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'--- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES --- GLOBAL STRUCTURES ---

'-------------------------------------------------------------------------------------------

'FYI - Structures are not placed in the Global Variable Dictionary (nor are the pieces

'that make the structure up). Treat these like constants and just place inline comments

'to provide details about the structure...

'This structure will hold all information related to a given project that is being calculated

'by the program

Structure udtBillInfo

Dim strCompanyName As String 'The name of the company for which the project was completed

Dim sngGoodsValue As Single 'The total amount of goods consumed on the project

Dim sngServicesValue As Single 'The total amount of services consumed on the project

Dim sngSubTotal As Single 'The subtotal is the amount of goods added to services

Dim sngSalesTax As Single 'The Michigan state sales tax calculated from the subtotal

Dim sngGrandTotal As Single 'The total cost of the project (subtotal added to sales tax)

End Structure

'---------------------------------------------------------------------------------------

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'--- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES --- GLOBAL VARIABLES ---

'---------------------------------------------------------------------------------------

'FYI - These are the variables that are placed in the Global Variable Dictionary in the

'first comment block at the top of the file.

Dim aCompanyBillInfo As udtBillInfo 'This variable instance of udtBillInfo will

' hold all of the project information for the

' current project being examined in this program.

' This is the main data structure that supports

' the entire application.

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'-----------------------------------------------------------------------------------

Private Sub SetInitialCompanyInformation(ByRef aCompanyBillInfo As udtBillInfo)

'------------------------------------------------------------

'- Subprogram Name: SetInitialCompanyInfomation -

'------------------------------------------------------------

'- Written By: Scott D. James -

'- Written On: January 12, 20XX -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine extracts the values entered from the user-

'- into the form and places them in the global structure –

'- variable. Any data conversions from string to numeric -

'- which are necessary are performed here as well. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- aCompanyBillInfo - The structure variable which contains -

'- all information for the project from -

'- which the program is creating a bill. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'Pull all three textbox values and place them in the structure variable for

'convenient use. We will convert the strings that look like numeric amounts

'to their respective numeric amounts.

aCompanyBillInfo.strCompanyName = txtCompanyName.Text

aCompanyBillInfo.sngGoodsValue = CSng(txtTotalGoods.Text)

aCompanyBillInfo.sngServicesValue = CSng(txtTotalServices.Text)

End Sub

Function CalculateSubTotal(ByVal sngGoodsValue As Single, ByVal sngServicesValue As Single) As Single

'------------------------------------------------------------

'- Subprogram Name: CalculateSubTotal -

'------------------------------------------------------------

'- Written By: Scott D. James -

'- Written On: January 12, 20XX -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This function calculates the subtotal of the project's -

'- value, which is the total value of the goods costs added -

'- to the total value of the services costs incurred on the -

'- project. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sngGoodsValue - The single value of the goods cost for -

'- the project. -

'- sngServicesValue - The single value of the services cost -

'- for the project. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- sngSubTotal - A single variable which will hold the sum -

'- of the goods cost added to the services -

'- cost in the project. This value will be -

'- returned from the function to whatever -

'- calls the function. -

'------------------------------------------------------------

'- Returns: -

'- Single – The subtotal amount of the project goods and -

'- services costs added together. -

'------------------------------------------------------------

'I normally wouldn't use a temporary variable for something as

'simple as a function adding two numbers, but I wanted to show

'how to document local variables in the comment block above...

Dim sngSubtotal As Single

'Add the two components that make up the project subtotal together and

'then return that value.

sngSubtotal = sngGoodsValue + sngServicesValue

Return sngSubtotal

End Function

Function CalculateSalesTax(ByVal sngSubtotal As Single) As Single

'------------------------------------------------------------

'- Subprogram Name: CalculateSalesTax -

'------------------------------------------------------------

'- Written By: Scott D. James -

'- Written On: January 12, 20XX -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This function computes the sales tax amount from the -

'– project's cost subtotal. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sngSubtotal - The current project's subtotal for all -

'- costs added together -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'- Returns: -

'- Single - The amount of sales tax calculated for this -

'- project. -

'------------------------------------------------------------

'The sales tax is simply computed by multiplying the project cost subtotal

'by the sales tax rate.

Return sngSubtotal \* MI\_SALES\_TAX

End Function

Function CalculateGrandTotal(ByVal sngSubtotal As Single, ByVal sngSalesTax As Single) As Single

'------------------------------------------------------------

'- Subprogram Name: CalculateGrandTotal -

'------------------------------------------------------------

'- Written By: Scott D. James -

'- Written On: January 12, 20XX -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This function computes the project's grand total, which -

'- is the project subtotal and sales tax amounts added -

'- together. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sngSubtotal - The project's subtotal cost value. -

'- sngSalesTax - The project's sales tax value. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'- Returns: -

'- Single - The grand total cost for this project. -

'------------------------------------------------------------

'The grand total is calculated by adding the project subtotal and sales

'tax amounts together.

Return sngSubtotal + sngSalesTax

End Function

Private Sub GenerateReceipt(ByVal aCompanyBillInfo As udtBillInfo)

'------------------------------------------------------------

'- Subprogram Name: GenerateReceipt -

'------------------------------------------------------------

'- Written By: Scott D. James -

'- Written On: January 12, 20XX -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine will generate a neatly formatted receipt -

'- in the form's lstReceipt listbox that appears at the -

'- bottom of the form. The listbox will be cleared prior to-

'- receipt generation. The receipt will detail the name of -

'- the company for whom the project was carried out along -

'- with all project cost values. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- aCompanyBillInfo - This is the structure containing all -

'- of the current project's information. -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

Const RECEIPT\_HEADING = "Receipt Information" 'Heading for the receipt

Const RECEIPT\_ITEM\_FORMAT = "{0,-17}{1,8:C}" 'A reusable format string that specifies

' how any cost lines are to appear

'Clear the listbox to ensure only the current receipt's information appears

lstReceipt.Items.Clear()

'Place the receipt heading in the listbox

lstReceipt.Items.Add(StrDup(12, " ") & RECEIPT\_HEADING)

lstReceipt.Items.Add(StrDup(12, " ") & StrDup(RECEIPT\_HEADING.Length, "-"))

lstReceipt.Items.Add(" ")

'Place the project's company name in the listbox

lstReceipt.Items.Add(String.Format("{0, -18}{1,-30}", "Company Name:",

aCompanyBillInfo.strCompanyName))

'Place all project cost information details in the listbox

lstReceipt.Items.Add(String.Format(RECEIPT\_ITEM\_FORMAT, "Goods:",

aCompanyBillInfo.sngGoodsValue))

lstReceipt.Items.Add(String.Format(RECEIPT\_ITEM\_FORMAT, "Services:",

aCompanyBillInfo.sngServicesValue))

lstReceipt.Items.Add("-----------------------------------------")

lstReceipt.Items.Add(String.Format(RECEIPT\_ITEM\_FORMAT, "Subtotal:",

aCompanyBillInfo.sngSubTotal))

lstReceipt.Items.Add(String.Format(RECEIPT\_ITEM\_FORMAT, "Sales Tax (" &

FormatPercent(MI\_SALES\_TAX, 0) & ") :", aCompanyBillInfo.sngSalesTax))

lstReceipt.Items.Add("=========================================")

lstReceipt.Items.Add(String.Format(RECEIPT\_ITEM\_FORMAT, "Grand Total:",

aCompanyBillInfo.sngGrandTotal))

lstReceipt.Items.Add("=========================================")

End Sub

Private Sub cmdCalculateAndDisplayBill\_Click(sender As Object, e As EventArgs) Handles

cmdCalculateAndDisplayBill.Click

'------------------------------------------------------------

'- Subprogram Name: cmdCalculateAndDisplayBill -

'------------------------------------------------------------

'- Written By: Scott D. James -

'- Written On: January 12, 20XX -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- form's command button. This routine will invoke all -

'- work performed by the program. There is an assumption -

'- that the user has entered all necessary information into -

'- the form prior to invoking this routine. No data valid- -

'- ation is performed. Therefore if incorrect data types or-

'- blank textboxes are encountered, the program will not -

'- deal with those issues. -

'------------------------------------------------------------

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control that raised -

'- the click event -

'- e – Holds the EventArgs object sent to the routine -

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- (None) -

'------------------------------------------------------------

'Pull user information into the global data structure

SetInitialCompanyInformation(aCompanyBillInfo)

'Peform all calculations to compute project cost details

aCompanyBillInfo.sngSubTotal = CalculateSubTotal(aCompanyBillInfo.sngGoodsValue,

aCompanyBillInfo.sngServicesValue)

aCompanyBillInfo.sngSalesTax = CalculateSalesTax(aCompanyBillInfo.sngSubTotal)

aCompanyBillInfo.sngGrandTotal = CalculateGrandTotal(aCompanyBillInfo.sngSubTotal,

aCompanyBillInfo.sngSalesTax)

'Create and display the receipt

GenerateReceipt(aCompanyBillInfo)

End Sub

End Class