Brilliant Fitness Inventory Management



By John Meyer

May 2, 2011

Table of Contents

Section #1 (User Guide)

Introduction 3

Mission Statement 3

Splash Screen 4

Login Form 5

Main Entry Form 6

  Data Entry 7

  Buttons 8

  Tab Controls 10

Form Menu 12

The Text Editor 13

  Password 13

  Editor Menu 14

The Inventory Report viewer 16

  Report Viewer Menu 17

Section #2 (Programmer's Reference)

Divisions of Code Regions 18

Public Variable Declarations 19

Load 19

Buttons 20

Calcs 21

Labels & Inputs 22

Timer & Loan 24

Loop & Accumulator 25

Check Box & Menu 26

Link, Array & Database 28

File 30

Edit & Format 31

Tools & Toolstrip 32

The Splash Screen Code 33

Login Form Code 34

About Box Code 35

Database SQL Code 36

Introduction

Thank you for choosing Brilliant Fitness Inventory Management Systems. This guide will exhibit how to become proficient with the Brilliant Fitness Inventory Management Program. This program is designed to provide users with quality inventory management and invoicing. Follow the step by step guides in this documentation package and follow the visual direction to become familiar with this software.

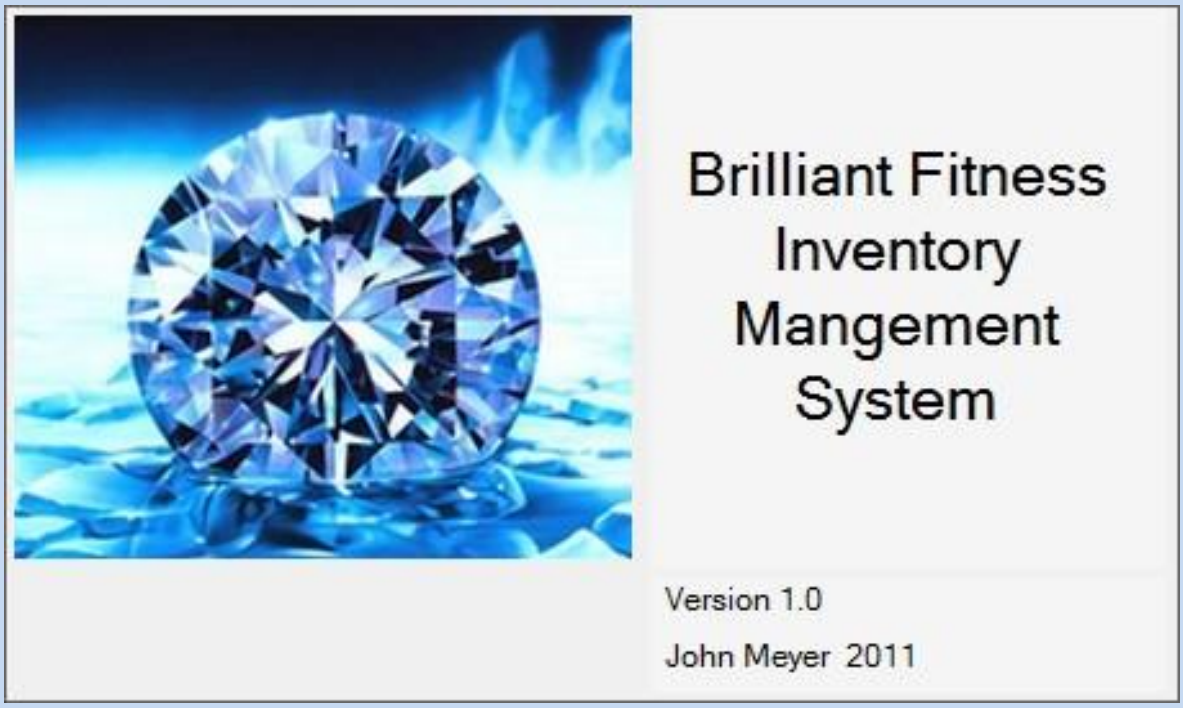
The first section of this document will give users directions on how to use the inventory program while the second part will be useful for a programmer's reference of code. It will also show how the code was implemented into this program to provide accurate and efficient inventory management and invoicing.

Mission Statement

Brilliant Fitness Inventory Management Systems' mission is to produce quality software for our clients through attention to detail, commitment and working and listening closely to the specific needs of our clients' business. It is our goal to succeed together with our clients in bringing quality programs to the business and economic table.

The Splash Screen

The Splash Screen shows up when the application starts to run. It shows a brief synopsis of the program with minimum information including the name of the program, the programmer's name, the company logo and the version.



The Login Form

The Login Form appears once the program has loaded and the Splash screen has gone away. Here you enter your Username and Password to gain entry to the Main Form.



If the password entered is not correct than a pop up window will appear, prompting you to enter the correct password. The user has 3 attempts to enter the correct password before the program shuts down.

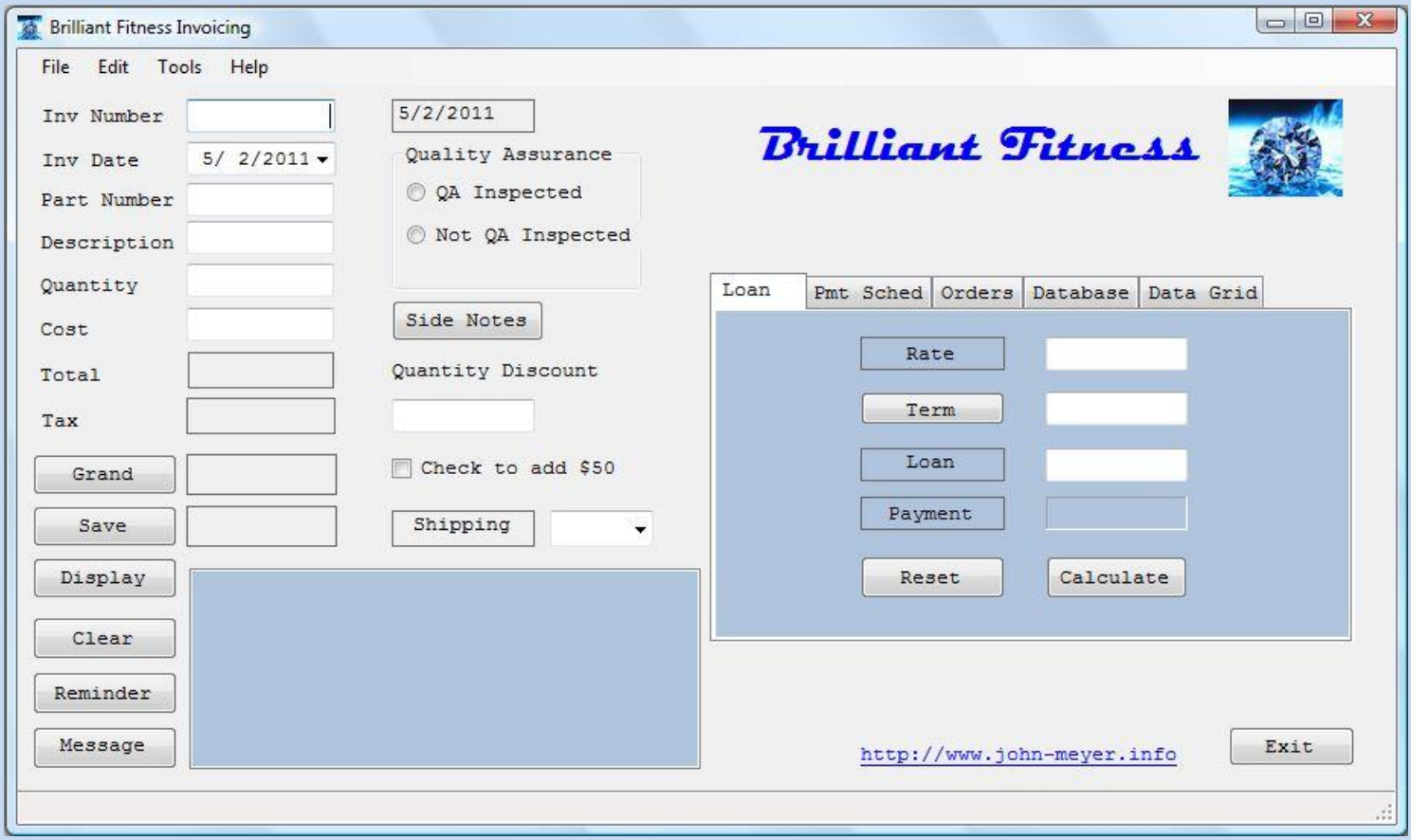


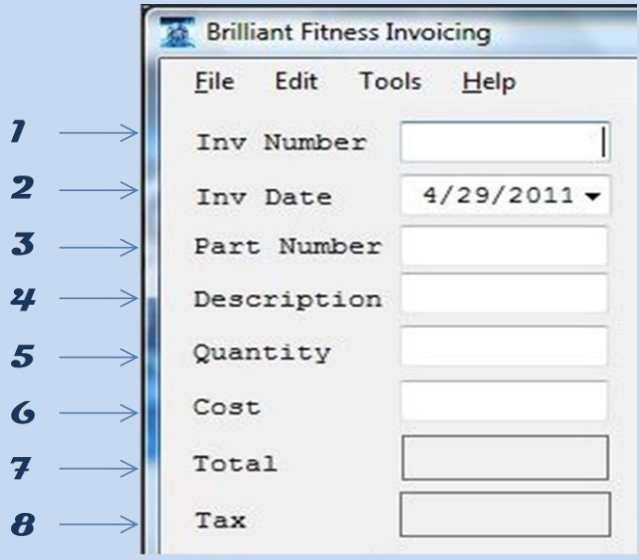


The shaded area represents a "back door." This is an entry point that the system administrator can enter the program if the password has been compromised.

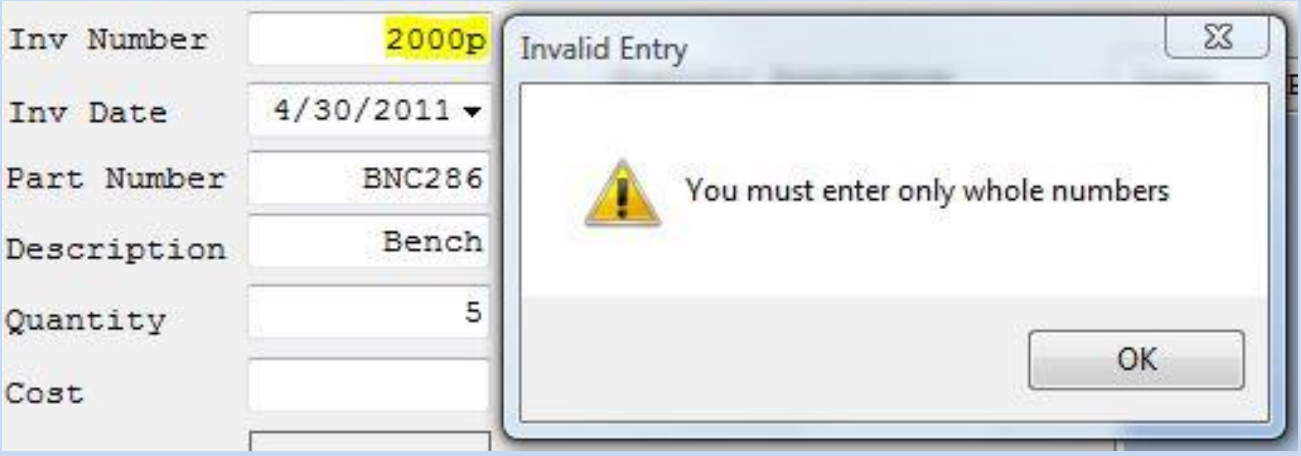
The Main Entry Form

The Main Entry form appears once the user has successfully logged in. Here the user will enter Items purchased into their corresponding fields. The events that occur on this form include calculating the quantity times the cost, tax, discounts based on quantity, loan amount with interest and a payment schedule. The following will explain in detail.





1. **Invoice Number:** This field is formatted to only accept whole numbers.



2. **Inventory Date:** Is formatted with a date time picker so the user can choose any date to enter the invoice.
3. **Part Number:** This field is formatted to hold only 6 characters and upon leaving the text box it will format the text to all caps.
4. **Description:** This field has no restrictions so the user may type any description.

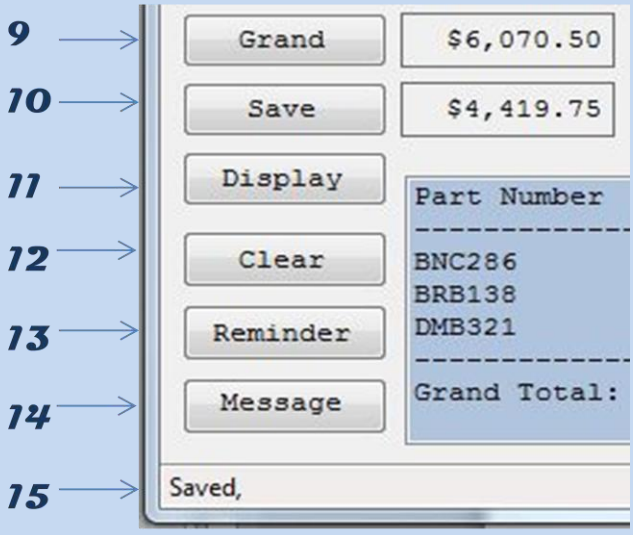
5. **Quantity:** Is formatted to only accept whole numbers, show as currency and apply discounts based on the quantity.



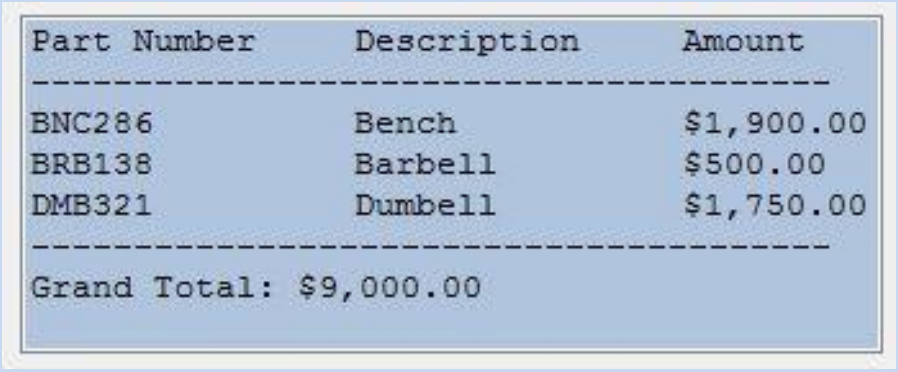
6. **Cost:** Is formatted to currency. Upon tabbing out of the text box a calculation will occur displaying the total in the next label.

7. **Total:** Is a label that displays the total after leaving the cost text box.

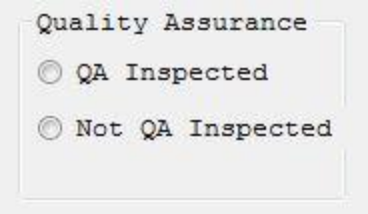
8. **Tax:** Will display the tax amount when the Grand button is clicked.



9. **Grand:** When clicked, this button is formatted to display the tax, in the Tax label, and the grand total with tax in the Grand label.
10. **Save:** This button is formatted to save the records and clear the input boxes to prepare for the next invoice entry (note: item 15 the word "Saved" in the tray indicates the record has been saved).
11. **Display:** Is formatted to display all the saved records in the list box to the right of the button.

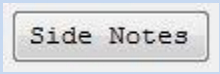


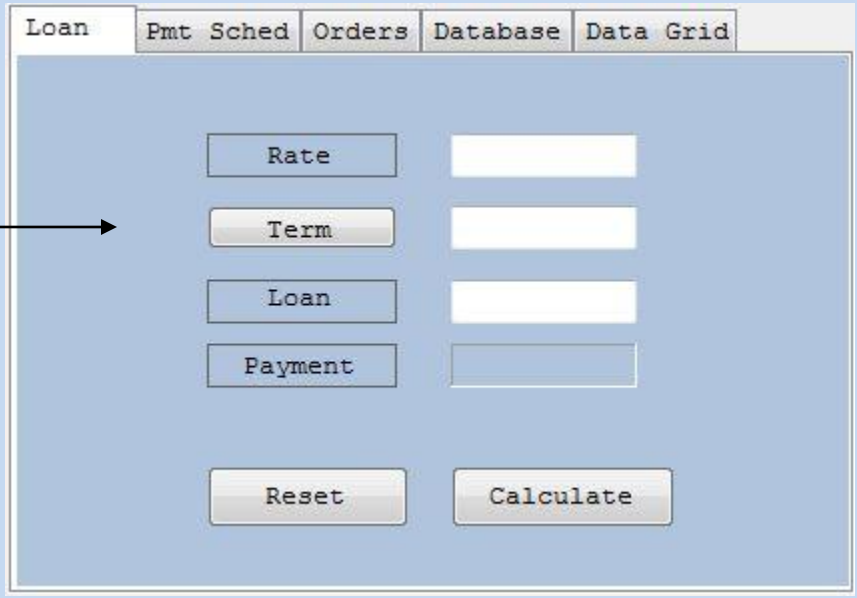
12. **Clear:** Will clear all the fields, input boxes and labels if the user needs to start over.
13. **Reminder:** is formatted to display an invisible text field reminding the user to save their work.
14. **Message:** Causes a pop up message to appear. The pop up message can be manipulated through a text editor in the program so users can leave a message for the next person.
15. **Status Strip:** Displays a message during certain functions like clicking the save button.



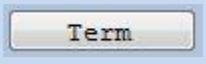
This Group Box contains radial buttons that can be checked to indicate whether or not the parts have been inspected by Quality Assurance. The buttons are formatted so that if the "Not QA Inspected" radial button is checked then it will clear the Total in the invoice inhibiting a transaction to be executed.

This Button labeled "Side Notes" will open Notepad for the user to enter any side notes as they fill out the invoice.



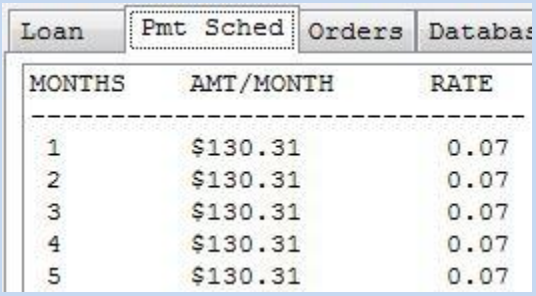


Above is a tab control that contains a variety of different tabs. The first tab shown here is the Loan tab which calculates the payment amount based on the loan amount, rate and term in months.



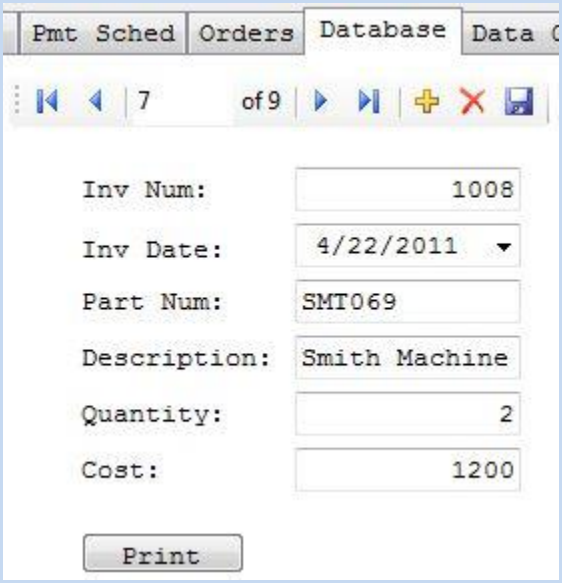
This Term button will calculate how many months are in "X" amount of years.

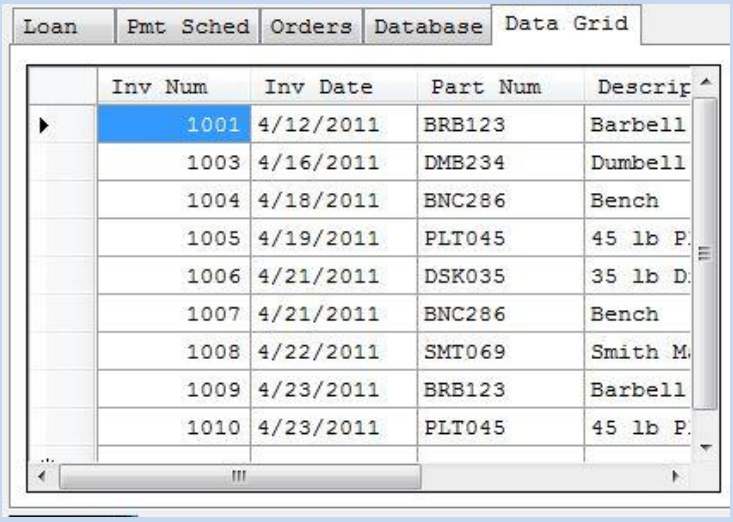
The next tab is the Payment Schedule tab which is directly influenced by the Loan tab. Once the loan is calculated, a payment schedule is displayed on the next tab as shown.



The Orders tab is simply a tab containing a list of prior orders including the customer's name, the order number and the date the order was made.

The Database tab is linked to an actual Access database. The Access database can be manipulated directly from the inventory program in this tab.

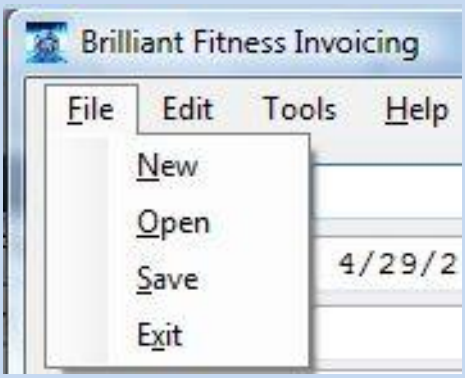




The Data Grid is linked to the same Access database as the Database tab is. In the Data Grid tab the database can be seen in "grid view."

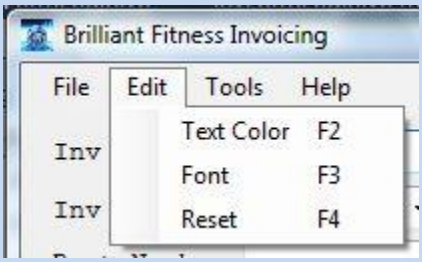
Form Menu

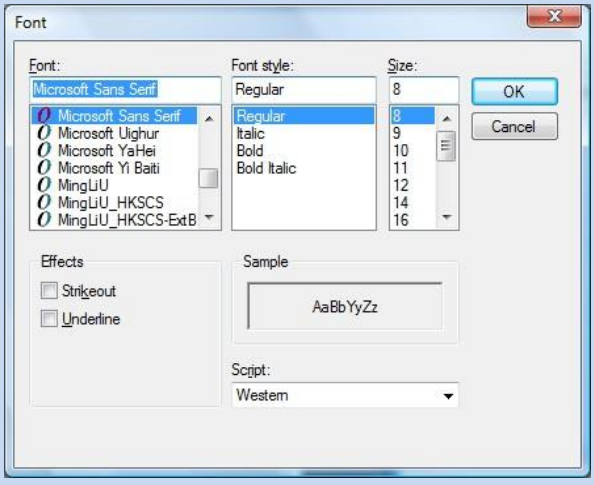
The Form Menu has the typical menu options along with a few program specific options as shown.



Here are the typical menu options you might see on most programs containing the New, Open, Save and Exit options.

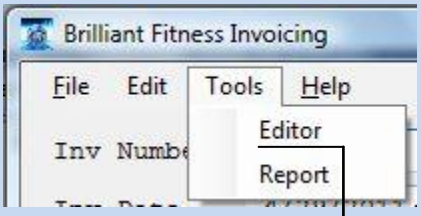
Under the Edit menu option, the Text Color option will prompt a color dialog box where the user can choose a color that may seem easier to read for them. The same type of dialog box will appear for the Font. The Reset option will restore the color and font to its original form.



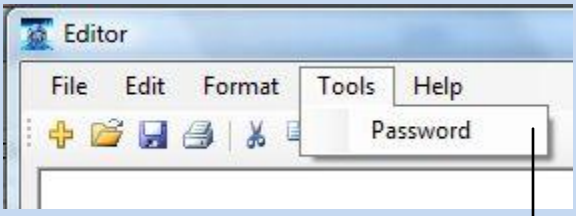


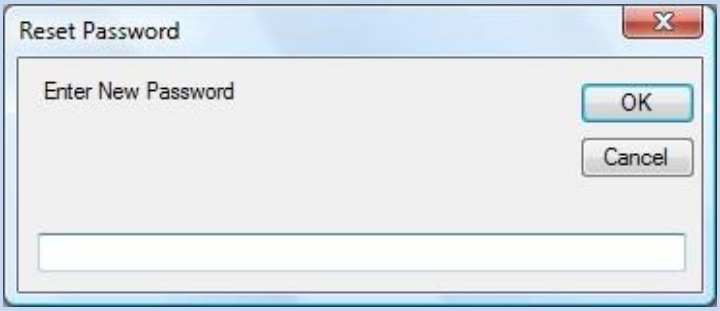


The Text Editor



Under the Tools menu the user has the Editor option which opens a text editor that has its own menu options, one of which the administrator can change the password to the form.



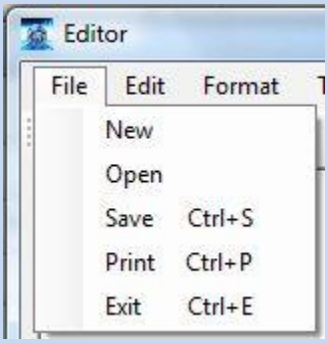


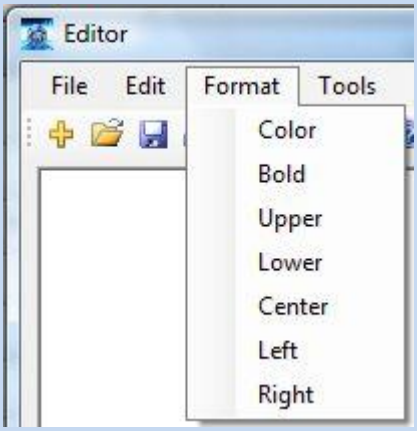
Once the password is changed in the text editor, the new password will be applied to the Main Form Login.

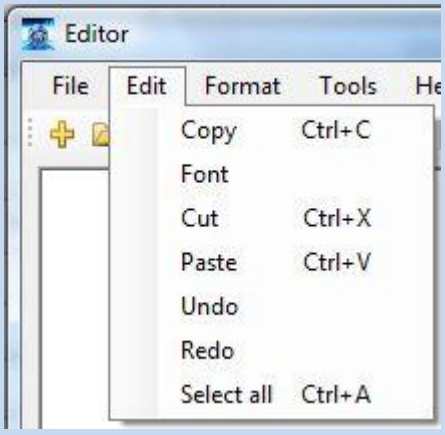
The text editor has menu options similar to the menus one would see in most word processing software. This editor also has a tool strip under the menu for easy access of some of the most common menu options used.



Below the menu options are displayed.

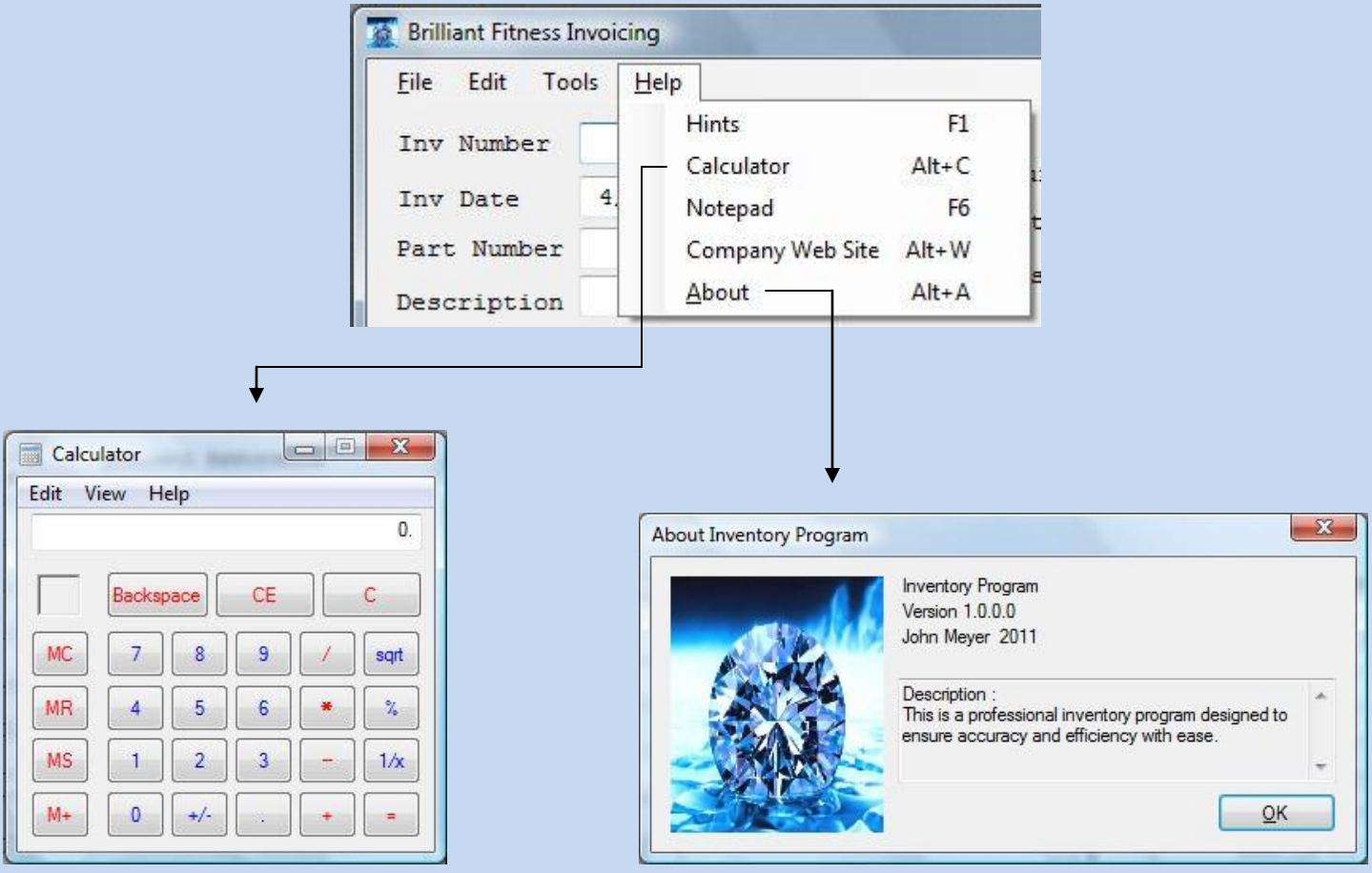






Although the menu options are similar to those of a word processing software, some keyboard shortcuts may vary. The programmer has the capability to designate shortcut keys to any particular function.

Under the Help menu option on the Main Entry Form, the user can utilize helpful items such as the Calculator, Notepad, the Company Web Site, the About box and even view some hints to make navigation through the inventory program easier for those who are learning the software.

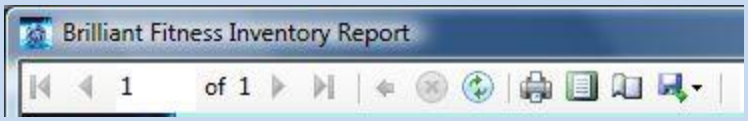


The Inventory Report Viewer

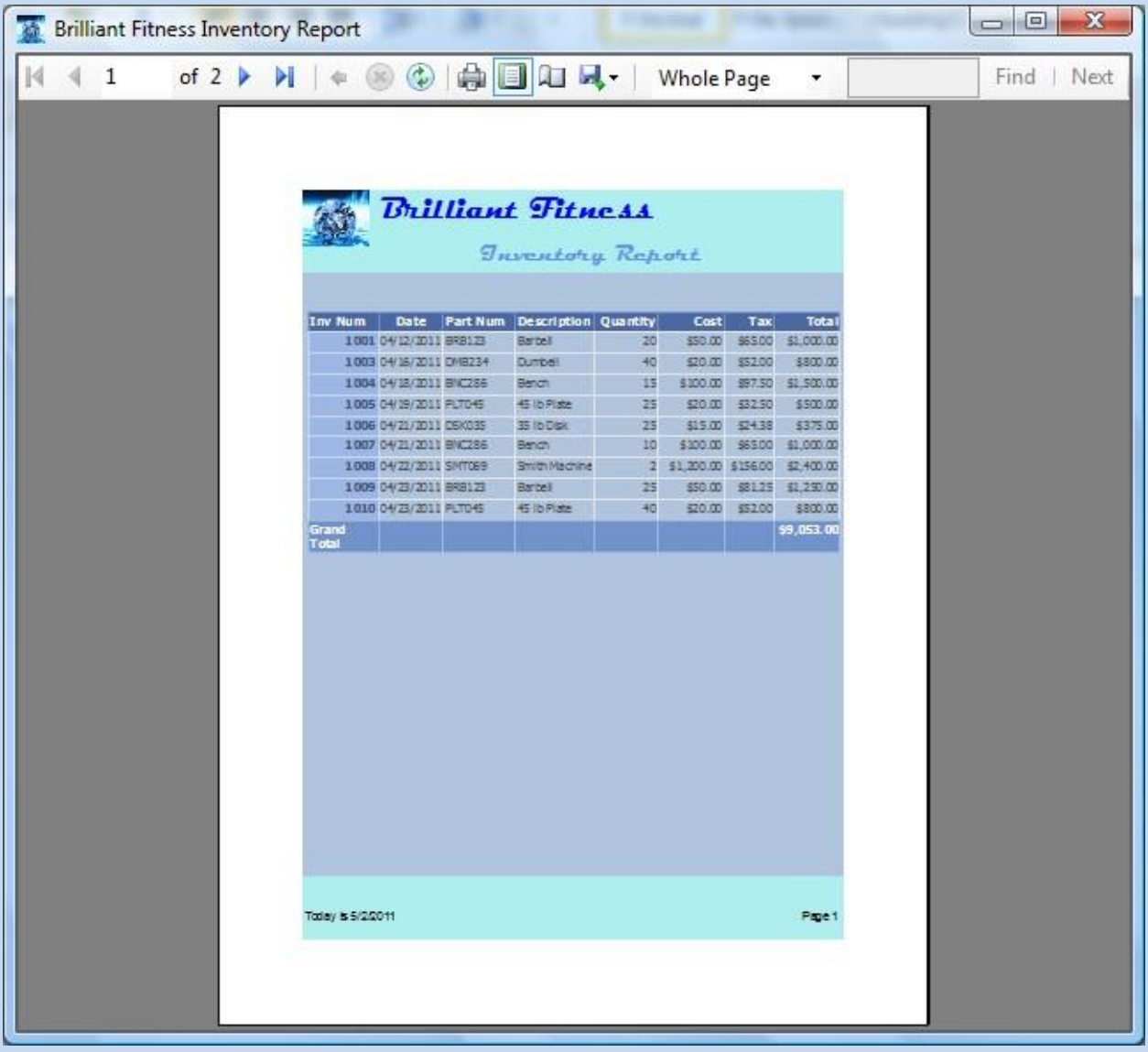
Below is a display of the Report Viewer in action. The Report Viewer allows the database to be viewed and printed in a more appealing manner than the data grid view.



Inv Num	Date	Part Num	Description	Quantity	Cost	Tax	Total
1001	04/12/2011	BRB123	Barbell	20	$50.00	$65.00	$1,000.00
1003	04/16/2011	DMB234	Dumbell	40	$20.00	$52.00	$800.00
1004	04/18/2011	BNC286	Bench	15	$100.00	$97.50	$1,500.00
1005	04/19/2011	PLT045	45 lb Plate	25	$20.00	$32.50	$500.00
1006	04/21/2011	DSK035	35 lb Disk	25	$15.00	$24.38	$375.00
1007	04/21/2011	BNC286	Bench	10	$100.00	$65.00	$1,000.00
1008	04/22/2011	SMT069	Smith Machine	2	$1,200.00	$156.00	$2,400.00
1009	04/23/2011	BRB123	Barbell	25	$50.00	$81.25	$1,250.00
1010	04/23/2011	PLT045	45 lb Plate	40	$20.00	$52.00	$800.00
Grand Total							$9,053.00



The Report Viewer menu allows the user to view all pages of the report, refresh, print, view the report in layout view and export the file in Excel, PDF and Word formats. Below is the report in print layout view.



Inv Num	Date	Part Num	Description	Quantity	Cost	Tax	Total
1001	04/12/2011	BRB123	Barbell	20	$50.00	$65.00	$1,000.00
1003	04/16/2011	DMB234	Dumbell	40	$20.00	$52.00	$800.00
1004	04/18/2011	BNC286	Bench	15	$100.00	$97.50	$1,500.00
1005	04/19/2011	PLT045	45 lb Plate	25	$20.00	$32.50	$500.00
1006	04/21/2011	DSK035	35 lb Disk	25	$15.00	$24.38	$375.00
1007	04/21/2011	BNC286	Bench	10	$100.00	$65.00	$1,000.00
1008	04/22/2011	SMT069	Smith Machine	2	$1,200.00	$156.00	$2,400.00
1009	04/23/2011	BRB123	Barbell	25	$50.00	$81.25	$1,250.00
1010	04/23/2011	PLT045	45 lb Plate	40	$20.00	$52.00	$800.00
Grand Total							$9,053.00

The Programmer's Reference

Divisions of Code Regions





Public Variable Declarations

```
Module Variables
    'Public Const dblmonths_Year As Double + 12

    Public GrandTotal As Decimal
    Public Newmessage As String = "Don't forget to save your work!!!"
    Public Cost As Decimal
    Public colFormLabels As New Collection() ' Key word New is very important
    Public ctrlFormLabel As Control ' Used to iterate the FormLabels collection
    Public TaxRate As Decimal = CDec(0.065)
    Public Tax As Decimal
    Public TaxGrandTotal As Decimal
    Public Limit As Integer = 5000
    Public PartNumber(Limit) As String
    Public Description(Limit) As String
    Public Amount(Limit) As Integer
    Public TotalAmount As Integer
    Public C As Integer 'Counter for Array

End Module
```

Load

```
#Region "Load"
    Private Property TotalSalesCount As Integer
    Private Sub invWeights_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'InvWeightsDataSet.InvWeights'
table. You can move, or remove it, as needed.
        Me.InvWeightsTableAdapter.Fill(Me.InvWeightsDataSet.InvWeights)
        txtDate.Text = Format(Today, "MM/dd/yyyy")
        lblSysDate.Text = (FormatDateTime(Today, DateFormat.ShortDate))
        colFormLabels.Add(lblDate)
        colFormLabels.Add(lblDescribe)
        colFormLabels.Add(lblState)
        colFormLabels.Add(lblCost)
        colFormLabels.Add(lblPart)
        colFormLabels.Add(lblQty)
        colFormLabels.Add(lblTax)
        colFormLabels.Add(lblTotal)
        colFormLabels.Add(lblMessage)
        colFormLabels.Add(lblNpNg)
        ssNote.Visible = False
        ssQuality.Visible = False
        'these do the same edit color of lables
    End Sub
#End Region
```

Buttons

```
#Region "Buttons"
    Private Sub btnReminder_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReminder.Click
        lblMessage.Visible = True
        btnReminder.Visible = False
    End Sub
    Private Sub btnMsg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnMsg.Click
        MessageBox.Show(Newmessage.Trim, " Be Advised")
        btnMsg.Visible = False
    End Sub
    Private Sub btnCalc_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Dim Qty As Integer
        Qty = CInt(txtQty.Text)
        Try
            lblCalcTotal.Text = CStr(Val(txtQty.Text) * Val(txtCost.Text))
        Catch ex As Exception
            MessageBox.Show("Unexpected Error" & vbCrLf & ex.Message, "Title goes here")
        End Try
    End Sub
    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
        ' Clear the Text Boxes
        txtDate.Clear()
        txtDescribe.Clear()
        txtPart.Clear()
        txtQty.Clear()
        txtCost.Clear()
        txtDiscount.Clear()
        txtInvNum.Clear()
        lblCalcTotal.Text = String.Empty
        ' Give the focus to txtDate.
        txtDate.Focus()
    End Sub
    Private Sub btnReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReset.Click
        ' Clear the Text Boxes
        txtRate.Clear()
        txtYrs.Clear()
        txtLoanAmt.Clear()
        lblPMT.Text = String.Empty
        lblTaxTotal.Text = String.Empty
        ' Give the focus to txtDate.
        txtRate.Focus()
    End Sub
    Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClose.Click
        End
    End Sub
    Private Sub btnlink_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnlink.Click
        Process.Start("notepad")
    End Sub
    Sub calcinfo(ByVal DataIn As Decimal) 'Data in is the name of what is received
```

```
        Tax = CDec(CDbl(lblCalcTotal.Text) * TaxRate)
    End Sub
    Private Sub btnGrandTotal_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGrandTotal.Click
        Call calcinfo(Cost) 'Price will be sent to procedure called calcinfo
        lblTaxTotal.Visible = True
        lblTaxTotal.Text = FormatCurrency(Tax)
        lblTaxGrandTotal.Visible = True
        lblTaxGrandTotal.Text = FormatCurrency(Tax + CDbl(lblCalcTotal.Text))
        'lblTaxGrandTotal.Text = FormatCurrency(TaxGrandTotal)
    End Sub

    Private Sub rbQAno_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles rbQAno.CheckedChanged
        If rbQAno.Checked = True Then
            lblCalcTotal.Text = "Check QA"
            ssQuality.Visible = True
            ssQuality.Text = "NOT INSPECTED BY QA"
        End If
    End Sub
    Private Sub rbQAyes_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles rbQAyes.CheckedChanged
        lblCalcTotal.Text = String.Empty
        ssQuality.Visible = False
    End Sub
#End Region
```

Calcs

```
#Region "Calcs"
    Private Sub Calcs()
        Debug.WriteLine("txtQty.Text= " & txtQty.Text & " txtCost.Text= " & txtCost.Text)
        If txtQty.Text <> "" And txtCost.Text <> "" Then
            Try
                Dim MarkUp As Integer
                If cbDiscount.Checked Then MarkUp = 50
                txtDiscount.Text = Discount(0).ToString & "%"
                lblCalcTotal.Text = CStr((CDbl(txtQty.Text) * CDbl(txtCost.Text) * (100 -
CDbl(Discount(0))) / 100) + MarkUp)
                lblCalcTotal.Text = FormatCurrency(lblCalcTotal.Text)
            Catch ex As Exception
                'add vbcrlf & ex.message
                MessageBox.Show("you must enter only numbers in Quantity field" & vbCrLf
& ex.Message, "Data entry error")
            End Try
        End If
    End Sub
    Private Function Discount(ByVal Cost As Decimal) As Decimal
        'Calculates the Discount Amount
        If CDbl(txtCost.Text) < 100 Then
            Return CDec("0")
        ElseIf CDbl(txtCost.Text) <= 200 Then
            Return CDec("5")
        ElseIf CDbl(txtCost.Text) <= 400 Then
            Return CDec("10")
```

```
        Else
            Return CDec("15")
        End If
    End Function
#End Region
```

Labels

```
#Region "Labels"
    Private Sub lblCalcTotal_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles lblCalcTotal.LostFocus
        lblCalcTotal.Text = CStr(CDec(lblCalcTotal.Text))
    End Sub
#End Region
```

Input

```
#Region "Input"
    Private Sub txtInvNum_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtInvNum.Leave
        If txtInvNum.Text = "" Then Exit Sub
        If txtInvNum.Text = "" Then
            MessageBox.Show("You must enter an invoice number", "Data entry required",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
            'InvNum = CInt(txtInvNum.Text)
        ElseIf Integer.TryParse(txtInvNum.Text, InvNum) Then
            MessageBox.Show("Data is valid - do your next step")
        Else
            MessageBox.Show("You must enter only whole numbers", "Invalid Entry",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        End If

    End Sub
    Private Sub txtDate_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtDate.LostFocus
    End Sub
    Dim intQty As Integer
    Private Sub txtQty_Leave(ByVal sender As Object, ByVal e As System.EventArgs) Handles
txtQty.Leave
        Select Case intQty
            Case 500
                lblCalcTotal.Text = "Inop"
        End Select
    End Sub
    Private Sub txtQty_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtQty.LostFocus
        If txtQty.Text = "" Then Exit Sub

        If IsNumeric(txtQty.Text) Then
```

```
            If CInt(txtQty.Text) < 20 Then 'Or CDbl(txtQty.Text) < 5 Then

                response = MessageBox.Show("Is Quantity too low?", "Out of range test",
MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation)
                If response = Windows.Forms.DialogResult.Yes Then
                    txtQty.Focus()
                Else
                End If
            End If
            If Integer.TryParse(txtQty.Text, intQty) Then
                'MessageBox.Show("Data is valid – do your next step")
            Else
                MessageBox.Show("You must enter a whole number", "Invalid Entry",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
                txtQty.Focus()
            End If
            Call Calcs()
        Else
            MessageBox.Show("You must enter only numbers in Quantity field", "Data entry
error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
            txtQty.Focus()
        End If
    End Sub
    'Private Sub txtCost_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtCost.LostFocus
    '    txtCost.Text = CStr(CDec(Val(txtCost.Text)))
    'End Sub

    Private Sub txtPart_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtPart.LostFocus
        If txtPart.Text = "" Then Exit Sub
        txtPart.Text = txtPart.Text.ToUpper.Trim.Substring(0, 6)

    End Sub
    Private Sub txtCost_Leave(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtCost.Leave
        Dim Cost As Decimal
        If IsNumeric(txtCost.Text) Then
            Cost = CDec(txtCost.Text)

            'txtDiscount.Text = CStr(Discount(Cost)) ' Perform the function Discount on
price
            Call Calcs()
            txtCost.Text = FormatCurrency(Cost)
        End If
    End Sub
#End Region
```

Timer

```
#Region "Timer"
    Private Sub btnCountDown_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCountDown.Click
        ' This Button will begin Rocket count down and launch
        Timer1.Enabled = True
        ' Once Timer1 is enabled, the commands in Private Sub Timer1_Tick
        'will execute on each tick
    End Sub
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
        Static C As Integer = 5
        ' Static means remember current value as other parts of program are run
        lblCount.Text = CStr(C)
        C -= 1 ' Decrement by 1 each timer click
        If C = -1 Then
            lblCount.Text = "Get Pumped!" : lblCount.ForeColor = Color.Red
        End If
        If C <= -2 Then
            lblCount.Text = "Move Weight!" : lblCount.ForeColor = Color.Blue
            Timer1.Interval = 100
            ' 100 = 100 ms. - reset Interval to move rocket faster
            picRocket.Left -= 20
            ' Move picture up by decrementing the top position
        End If
    End Sub
#End Region
```

Loan

```
#Region "Loan"
    Private Sub btnTerm_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnTerm.Click
        Dim Years, Months As Integer
        Years = CInt(InputBox("Enter term in Years", "Loan Term"))
        ' InputBox("Prompt", "Dialog box title")
        Months = Years * 12
        MessageBox.Show("Years: " & Years & " Months: " & Months, " Loan Term")
    End Sub
    Private Sub btnCallLoan_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCallLoan.Click
        Call loantotal()
        lblPMT.Text = FormatCurrency(lblPMT.Text)
        For I As Integer = 1 To CInt(txtYrs.Text) * 12
            lbPmt.Items.Add(" " & I & "        " & lblPMT.Text & "         " &
txtRate.Text)
        Next
    End Sub
    Private Sub loantotal()
        lblPMT.Text = CStr(Pmt(CDbl(txtRate.Text) / 12, CDbl(txtYrs.Text) * 12, -
CDbl(txtLoanAmt.Text))) '* CDbl(txtYrs.Text) * 12)
        If lblPMT.Text = "" Then Exit Sub
    End Sub
    'Private Sub loantotal()
```

```
    '    lblPMT.Text = CStr(Pmt(CDbl(txtRate.Text) / 12, Val(txtYrs.Text) * 12, -
Val(txtLoanAmt.Text))) '* CDbl(txtYrs.Text) * 12)
    '    If lblPMT.Text = "" Then Exit Sub
    'End Sub

    Private Sub txtRate_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtRate.TextChanged
        If txtRate.Text = "" Then Exit Sub
        If CDbl(txtRate.Text) > 1 Then
            txtRate.Text = CStr(CDbl(txtRate.Text) / 100)
        End If
    End Sub
    Private Sub txtLoanAmt_Leave(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtLoanAmt.Leave
        txtLoanAmt.Text = FormatCurrency(txtLoanAmt.Text)
    End Sub
    'Private Sub txtLoanAmt_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtLoanAmt.TextChanged
    'Call loantotal()
    'End Sub
#End Region
```

Loop

```
Private Sub Whileloop()
        lblWhileLoop.Text = ""
        Dim Count As Integer = 0 ' Notice Count is Initialized at 0
        Do While Count < 10 ' Pretest - Evaluate before execute
            Count += 1 ' This is a Counter, notice it is placed before the main code
            lblWhileLoop.Text += Count & vbCrLf ' display new X value & Carriage
Return/Line Feed
        Loop
    End Sub
```

Accumulator

```
    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnSave.Click
        If lblCalcTotal.Text <> "" Then
            'GrandTotal += CDec(lblCalcTotal.Text)
            GrandTotal = CDec(GrandTotal + (CDec(lblCalcTotal.Text) +
CDbl(lblTaxTotal.Text)))
            lblGrandTotal.Text = FormatCurrency(GrandTotal)
            lblTaxGrandTotal.Text = FormatCurrency(TaxGrandTotal)
        End If
        C += 1
        TotalAmount = CInt(TotalAmount + CDbl(lblGrandTotal.Text)) 'was lblCalcTotal
        PartNumber(C) = CStr(txtPart.Text)
        Amount(C) = CInt(lblCalcTotal.Text)
        Description(C) = txtDescribe.Text
        Debug.WriteLine("Part Number(C)= " & C & " " & PartNumber(C))
        txtCost.Clear()
        txtPart.Clear()
```

```
        txtDescribe.Clear()
        txtQty.Clear()
        txtInvNum.Clear()
        'txtAmount.Text = 0
        'txtName.Text = ""
        ssNote.Visible = True
        ssNote.Text = "Saved,"
    End Sub
```

Check Box

```
Private Sub cbDiscount_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cbDiscount.CheckedChanged
        Call Calcs()
    End Sub
```

Menu

```
Private Sub mnuEditColor_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditColor.Click
        'ColorDialog1.ShowDialog()
        'lblCost.ForeColor = ColorDialog1.Color '//changes color of lbl
        'Me.BackColor = ColorDialog1.Color  '//changes the color of background
        ColorDialog1.ShowDialog()
        For Each Me.ctrlFormLabel In colFormLabels
            ctrlFormLabel.ForeColor = ColorDialog1.Color
        Next
    End Sub
    'Sub mnuEditColorCode()
    '    ColorDialog1.ShowDialog()
    '    Me.BackColor = ColorDialog1.Color
    'End Sub
    Private Sub mnuEditFont_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditFont.Click
        FontDialog1.ShowDialog()
        For Each Me.ctrlFormLabel In colFormLabels
            ctrlFormLabel.Font = FontDialog1.Font
        Next
    End Sub
    Private Sub mnuEditReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditReset.Click
        For Each Me.ctrlFormLabel In colFormLabels
            ctrlFormLabel.ForeColor = Color.Black
        Next
        For Each Me.ctrlFormLabel In colFormLabels
            ctrlFormLabel.Font = btnClear.Font
        Next
        'For Each Me.BackColor
        'Next
    End Sub
    Private Sub CompanyWebSiteToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuHelpCompany.Click
        Process.Start("http://john-meyer.info")
    End Sub
```

```
    Private Sub mnuHelpHints_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuHelpHints.Click
        Process.Start("notepad", "../Hints.txt")
    End Sub
    Private Sub mnuHelpCalculator_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuHelpCalculator.Click
        Process.Start("Calc")
    End Sub
    Private Sub mnuHelpNotepad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuHelpNotepad.Click
        Process.Start("Notepad")
    End Sub
    Private Sub mnuFileExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFileExit.Click
        Me.Close()
    End Sub
    Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuHelpAbout.Click
        frmAbout.ShowDialog()
    End Sub
    Private Sub ColorToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ColorToolStripMenuItem.Click
        'Call mnuEditColorCode()
        ColorDialog1.ShowDialog()
        For Each Me.ctrlFormLabel In colFormLabels
            ctrlFormLabel.ForeColor = ColorDialog1.Color
        Next
    End Sub
    Private Sub HintToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles HintToolStripMenuItem.Click
        Process.Start("NotePad", "../Hints.txt")

    End Sub
    Private Sub FontToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FontToolStripMenuItem.Click
        FontDialog1.ShowDialog()
        For Each Me.ctrlFormLabel In colFormLabels
            ctrlFormLabel.Font = FontDialog1.Font
        Next
    End Sub
    Private Sub ResetToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ResetToolStripMenuItem.Click
        For Each Me.ctrlFormLabel In colFormLabels
            ctrlFormLabel.ForeColor = Color.Black
        Next
        For Each Me.ctrlFormLabel In colFormLabels
            ctrlFormLabel.Font = Font
        Next
    End Sub
    Private Sub mnuToolsEditor_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuToolsEditor.Click
        txtEditor.ShowDialog()
    End Sub
```

Link

```
Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles LinkLabel1.LinkClicked
        Process.Start("http://john-meyer.info")
    End Sub
```

Array

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnDisplay.Click
        ListBox1.Items.Clear()
        ListBox1.Items.Add("Part Number" & vbTab & "Description" & vbTab & "Amount")
        ListBox1.Items.Add("---------------------------------------")
        For x = 1 To C
            ListBox1.Items.Add(PartNumber(x) & vbTab & vbTab & Description(x) & vbTab &
vbTab & FormatCurrency(Amount(x)))
        Next
        ListBox1.Items.Add("---------------------------------------")
        ListBox1.Items.Add("Grand Total: " & FormatCurrency(TotalAmount))
    End Sub
    Private Sub tbArray_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Me.Load
        Customer(1, 1) = "Brady"   ' Row 1, Column 1
        Customer(1, 2) = "134"     ' Row 1, Column 2
        Customer(1, 3) = "3/24/11"
        Customer(2, 1) = "Jones"
        Customer(2, 2) = "135"
        Customer(2, 3) = "3/28/11"
        Customer(3, 1) = "Smith" ' Row 3, Column 1
        Customer(3, 2) = "136"     ' Row 3, Column 2
        Customer(3, 3) = "4/01/11"
        Customer(4, 1) = "Thomas"
        Customer(4, 2) = "137"
        Customer(4, 3) = "4/03/11"
    End Sub
    Private Sub btnDisplayOrders_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDisplayOrders.Click
        Array.Sort(arrayCustomerSort)
        ListBoxOrders.Items.Clear()
        ListBoxOrders.Items.Add("Name" & vbTab & Chr(32) & Chr(32) & Chr(32) & "Order#" &
vbTab & "Date")
        ListBoxOrders.Items.Add("--------------------------------")
        For Count1 = 1 To 4
            ListBoxOrders.Items.Add(Customer(Count1, 1) & vbTab & Chr(32) & Chr(32) &
Chr(32) & Customer(Count1, 2) & vbTab & vbTab & Customer(Count1, 3))
        Next Count1
    End Sub
```

Database

```
Private Sub InvWeightsBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles InvWeightsBindingNavigatorSaveItem.Click
```

```
        Me.Validate()
        Me.InvWeightsBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.InvWeightsDataSet)
    End Sub
    Private Sub Inv_NumTextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Inv_NumTextBox.TextChanged
        If Inv_NumTextBox.Text = "" Then Exit Sub
    End Sub
    Private Sub Part_NumTextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Part_NumTextBox.Leave
        If Part_NumTextBox.Text = "" Then Exit Sub
        Part_NumTextBox.Text = Part_NumTextBox.Text.ToUpper.Trim.Substring(0, 6)
    End Sub
    Private Sub DescriptionTextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles DescriptionTextBox.TextChanged
        If DescriptionTextBox.Text = "" Then Exit Sub
    End Sub
    Private Sub QuantityTextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QuantityTextBox.Leave
        If QuantityTextBox.Text = "" Then Exit Sub
        If IsNumeric(QuantityTextBox.Text) Then
            If CInt(QuantityTextBox.Text) < 20 Then 'Or CDbl(QuantityTextBox.Text) < 5
Then
                Dim response As DialogResult
                response = MessageBox.Show("Is Quantity too low?", "Out of range test",
MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation)
                If response = Windows.Forms.DialogResult.Yes Then
                    QuantityTextBox.Focus()
                Else
                End If
            End If
            'If Integer.TryParse(QuantityTextBox.Text, intQty) Then
            '    MessageBox.Show("Data is valid - do your next step")
            '    'Else
            '    'MessageBox.Show("You must enter a whole number", "Invalid Entry")
            'End If
            Call Calcs()
        Else
            MessageBox.Show("you must enter only numbers in Quantity field", "Data entry
error")
            QuantityTextBox.Focus()
        End If
    End Sub
    Private Sub PrintDocument2_PrintPage(ByVal sender As System.Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument2.PrintPage
        'For Each row As System.Windows.Forms.DataGridViewRow In
InvWeightsDataGridView.Rows
        e.Graphics.DrawString(InvWeightsDataGridView.Rows(0).Cells(0).Value.ToString & "
" & InvWeightsDataGridView.Rows(0).Cells(1).Value.ToString & "  " &
InvWeightsDataGridView.Rows(0).Cells(2).Value.ToString & "  " &
InvWeightsDataGridView.Rows(0).Cells(3).Value.ToString & "  " &
InvWeightsDataGridView.Rows(0).Cells(4).Value.ToString & "  " &
InvWeightsDataGridView.Rows(0).Cells(5).Value.ToString, New Font("arial", 10),
Brushes.Black, New Point(0, 0))
        'Next row
    End Sub
    Private Sub btnPrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrint.Click
```

```
        PrintDocument2.PrinterSettings = PrintDialog2.PrinterSettings
        If PrintDialog2.ShowDialog = vbOK Then
            PrintDocument2.Print()
        Else
        End If
        Exit Sub
    End Sub
```

File

```
Private Sub mnuFileNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles mnuFileNew.Click
        RichTextBox1.Clear()
    End Sub
    Private Sub mnuFileOpen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFileOpen.Click
        Dim fileName = String.Empty ' Define and Clear Contents of text box
        Dim inputFile As StreamReader
        OpenFileDialog1.Filter = "Rich Text files (*.rtf)|*.rtf|All Files (*.*)|*.*"
        ' Set to default to .rtf files
        OpenFileDialog1.ShowDialog()
        fileName = OpenFileDialog1.FileName
        inputFile = File.OpenText(fileName)
        RichTextBox1.Rtf = inputFile.ReadToEnd ' You will need more logic to also open
.txt files
    End Sub
    Private Sub mnuFileSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFileSave.Click
        Dim outputFileName As System.IO.StreamWriter
        SaveFileDialog1.Filter() = "Rich Text File (*.rtf)|*.rtf|Text File
(*.txt)|*.txt|All files (*.*)|*.*"
        Try
            SaveFileDialog1.ShowDialog()
            outputFileName = File.CreateText(SaveFileDialog1.FileName)
            ' Without a path, fileDemo will be saved in the bin/debug folder
            outputFileName.Write(RichTextBox1.Rtf)
            outputFileName.Close()
        Catch
            MessageBox.Show("Sorry, the file cannot be created.")
        End Try
    End Sub
    Private Sub mnuFilePrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFilePrint.Click
        PrintDocument1.PrinterSettings = PrintDialog1.PrinterSettings
        If PrintDialog1.ShowDialog = vbOK Then
            PrintDocument1.Print()
        Else
        End If
        Exit Sub
    End Sub

    Private Sub mnuFileExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFileExit.Click
        Me.Close()
    End Sub
```

Edit

```
Private Sub mnuEditCopy_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles mnuEditCopy.Click
        RichTextBox1.Copy()
        ' Follow this intellisense pattern for Cut, Paste, Undo, Redo, Select All
    End Sub
    Private Sub mnuEditCut_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditCut.Click
        RichTextBox1.Cut()
    End Sub
    Private Sub mnuEditPaste_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditPaste.Click
        RichTextBox1.Paste()
    End Sub
    Private Sub mnuEditUndo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditUndo.Click
        RichTextBox1.Undo()
    End Sub
    Private Sub mnuEditRedo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditRedo.Click
        RichTextBox1.Redo()
    End Sub
    Private Sub mnuEditSelectAll_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditSelectAll.Click
        RichTextBox1.SelectAll()
    End Sub
```

Format

```
Private Sub mnuFormatColor_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFormatColor.Click
        ColorDialog1.ShowDialog()
        RichTextBox1.SelectionColor = ColorDialog1.Color
        ' Add a menu item and follow this pattern for formatting the Font
    End Sub
    Private Sub mnuEditFont_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuEditFont.Click
        FontDialog1.ShowDialog()
        RichTextBox1.SelectionFont = FontDialog1.Font
    End Sub
    Private Sub mnuFormatBold_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFormatBold.Click
        RichTextBox1.SelectionFont = New Font(RichTextBox1.SelectionFont, FontStyle.Bold)
    End Sub
    Private Sub mnuFormatUpper_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFormatUpper.Click
        RichTextBox1.SelectedText = RichTextBox1.SelectedText.ToUpper
        ' Follow this intellisense pattern for .ToLower
    End Sub
    Private Sub mnuFormatLower_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFormatLower.Click
        RichTextBox1.SelectedText = RichTextBox1.SelectedText.ToLower
    End Sub
```

```
    Private Sub mnuFormatCenter_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFormatCenter.Click
        RichTextBox1.SelectionAlignment = HorizontalAlignment.Center
        ' Follow this intellisense pattern for Left and Right
    End Sub
    Private Sub mnuFormatLeft_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFormatLeft.Click
        RichTextBox1.SelectionAlignment = HorizontalAlignment.Left
    End Sub
    Private Sub mnuFormatRight_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFormatRight.Click
        RichTextBox1.SelectionAlignment = HorizontalAlignment.Right
    End Sub
```

Tools

```
Private Sub mnuToolsPassword_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuToolsPassword.Click
        Try
            txtEditor = File.CreateText("FileDemo.txt")
            ' Without a path, fileDemo will be saved in  the bin/debug folder
            ' Or for a Password file consider a fake name like ("sys.dll") in the
bin/debug folder
            ' Start a loop here for multiple input lines
            InputLine = InputBox("Enter New Password", "Reset Password")
            ' Or InputLine = TextBox1.Text (This is another way to get data for file)
            txtEditor.WriteLine(InputLine)
            ' End a loop here for multiple lines
            txtEditor.Close()
        Catch
            MessageBox.Show("Sorry, the file cannot be created.")
        End Try
    End Sub
```

Toolstrip

```
Private Sub btnNew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnNew.Click
        RichTextBox1.Clear()
    End Sub
    Private Sub btnOpen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnOpen.Click
        Dim fileName = String.Empty ' Define and Clear Contents of text box
        Dim inputFile As StreamReader
        OpenFileDialog1.Filter = "Rich Text files (*.rtf)|*.rtf|All Files (*.*)|*.*"
        ' Set to default to .rtf files
        OpenFileDialog1.ShowDialog()
        fileName = OpenFileDialog1.FileName
        inputFile = File.OpenText(fileName)
        RichTextBox1.Rtf = inputFile.ReadToEnd
    End Sub
    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnSave.Click
        Dim outputFileName As System.IO.StreamWriter
```

```
        SaveFileDialog1.Filter() = "Rich Text File (*.rtf)|*.rtf|Text File
(*.txt)|*.txt|All files (*.*)|*.*"
        Try
            SaveFileDialog1.ShowDialog()
            outputFileName = File.CreateText(SaveFileDialog1.FileName)
            ' Without a path, fileDemo will be saved in the bin/debug folder
            outputFileName.Write(RichTextBox1.Rtf)
            outputFileName.Close()
        Catch
            MessageBox.Show("Sorry, the file cannot be created.")
        End Try
    End Sub
    Private Sub PrintDocument1_Click(ByVal sender As System.Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
        e.Graphics.DrawString(RichTextBox1.Text, New Font("MS Sans Serif", 12,
FontStyle.Regular), Brushes.Black, 10, 10)
    End Sub
    Private Sub btnPrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrint.Click
        PrintDocument1.PrinterSettings = PrintDialog1.PrinterSettings
        If PrintDialog1.ShowDialog = vbOK Then
            PrintDocument1.Print()
        Else
        End If
        Exit Sub
    End Sub
    Private Sub btnCut_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnCut.Click
        RichTextBox1.Cut()
    End Sub
    Private Sub btnCopy_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnCopy.Click
        RichTextBox1.Copy()
    End Sub
    Private Sub btnPaste_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPaste.Click
        RichTextBox1.Paste()
    End Sub
    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLink.Click
        Process.Start("http://john-meyer.info")
    End Sub
    Private Sub ToolStripButton1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ToolStripButton1.Click
        Me.Close()
    End Sub
```

Splash Screen Code

```
Public NotInheritable Class SplashScreen1
    'TODO: This form can easily be set as the splash screen for the application by going
to the "Application" tab
    '  of the Project Designer ("Properties" under the "Project" menu).
    Private Sub SplashScreen1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
```

```
        'Set up the dialog text at runtime according to the application's assembly
information.
        'TODO: Customize the application's assembly information in the "Application" pane
of the project
        '  properties dialog (under the "Project" menu).
        'Application title
        If My.Application.Info.Title <> "" Then
            ApplicationTitle.Text = My.Application.Info.Title
        Else
            'If the application title is missing, use the application name, without the
extension
            ApplicationTitle.Text =
System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
        End If
        'Format the version information using the text set into the Version control at
design time as the
        '  formatting string.  This allows for effective localization if desired.
        '  Build and revision information could be included by using the following code
and changing the
        '  Version control's designtime text to "Version {0}.{1:00}.{2}.{3}" or something
similar.  See
        '  String.Format() in Help for more information.
        '
        '    Version.Text = System.String.Format(Version.Text,
My.Application.Info.Version.Major, My.Application.Info.Version.Minor,
My.Application.Info.Version.Build, My.Application.Info.Version.Revision)
        Version.Text = System.String.Format(Version.Text,
My.Application.Info.Version.Major, My.Application.Info.Version.Minor)
        'Copyright info
        Copyright.Text = My.Application.Info.Copyright
    End Sub


End Class
```

Login Form Code

```
Imports System.IO
Public Class invFormLogin
    Dim bad As Integer
    Dim FileDemo As System.IO.StreamReader
    Dim InputLine As String
    ' TODO: Insert code to perform custom authentication using the provided username and
password
    ' (See http://go.microsoft.com/fwlink/?LinkId=35339).
    ' The custom principal can then be attached to the current thread's principal as
follows:
    '     My.User.CurrentPrincipal = CustomPrincipal
    ' where CustomPrincipal is the IPrincipal implementation used to perform
authentication.
    ' Subsequently, My.User will return identity information encapsulated in the
CustomPrincipal object
    ' such as the username, display name, etc.
    Private Sub OK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles OK.Click
```

```
        bad += 1
        If bad >= 3 Then End
        If txtPassword.Text = InputLine Then
            invWeights.Show()
            Me.Close()
        Else
            MessageBox.Show("Incorrect password", "Invaled entry", MessageBoxButtons.OK,
MessageBoxIcon.Hand)
        End If
    End Sub
    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Cancel.Click
        Me.Close()
    End Sub
    Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
        invWeights.Show()
        Me.Close()
    End Sub
    Private Sub invFormLogin_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        FileDemo = New StreamReader("FileDemo.txt")
        InputLine = FileDemo.ReadLine
        FileDemo.Close()
    End Sub
End Class
```

About Box Code

```
Public NotInheritable Class frmAbout
    Private Sub AboutBox1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ' Set the title of the form.
        Dim ApplicationTitle As String
        If My.Application.Info.Title <> "" Then
            ApplicationTitle = My.Application.Info.Title
        Else
            ApplicationTitle =
System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
        End If
        Me.Text = String.Format("About {0}", ApplicationTitle)
        ' Initialize all of the text displayed on the About Box.
        ' TODO: Customize the application's assembly information in the "Application"
pane of the project
        '    properties dialog (under the "Project" menu).
        Me.lblProductName.Text = My.Application.Info.ProductName
        Me.lblVersion.Text = String.Format("Version {0}",
My.Application.Info.Version.ToString)
        Me.lblCopyright.Text = My.Application.Info.Copyright
        Me.lblCompanyName.Text = My.Application.Info.CompanyName
        Me.txtDescription.Text = My.Application.Info.Description
    End Sub
    Private Sub OKButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OKButton.Click
        Me.Close()
```

```
    End Sub
End Class
```

Database SQL Code

