**Case One**

**The Downtown Department Store**

**19 February 2015**



**Pradip Pathak**

Additional Charges Form

Monogramming Form

Singing Telegram Form

Gift Wrapping Form

Testing

****

**Ricky Genz**

Main Form

Shipping Form

Publication



**Natalie Wright**

Broadway Shows From

Enhancements

Testing

Developers

Table of Contents

[Part A – Test Data 3](#_Toc412028633)

[Set One – Ana Anderson 4](#_Toc412028634)

[Set Two – Bob Branson 8](#_Toc412028635)

[Set Three – Carla Crowley 12](#_Toc412028636)

[Set Four – Diane Donaldson 16](#_Toc412028637)

[Part B – Technical Documentation 20](#_Toc412028638)

[Module & Main Form 21](#_Toc412028639)

[Shipping Form 29](#_Toc412028640)

[Additional Charges Form 33](#_Toc412028641)

[Monogramming Form 35](#_Toc412028642)

[Telegram Form 38](#_Toc412028643)

[Gift Wrapping Form 41](#_Toc412028644)

[Broadway Form 44](#_Toc412028645)

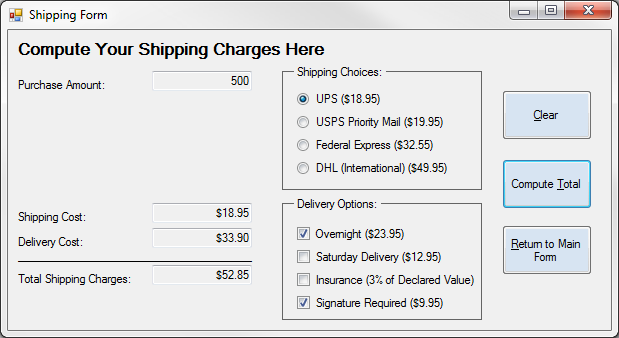
[Part C – Enhancements 50](#_Toc412028646)

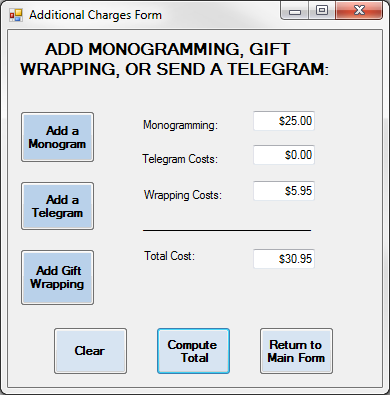
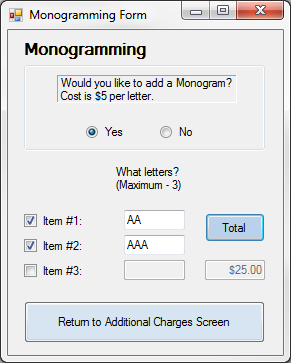
[Enhancement 1 – Zip Code Validation 51](#_Toc412028647)

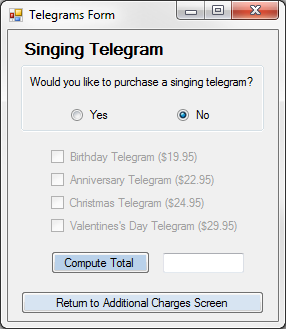
[Enhancement 2 – About Form 52](#_Toc412028648)

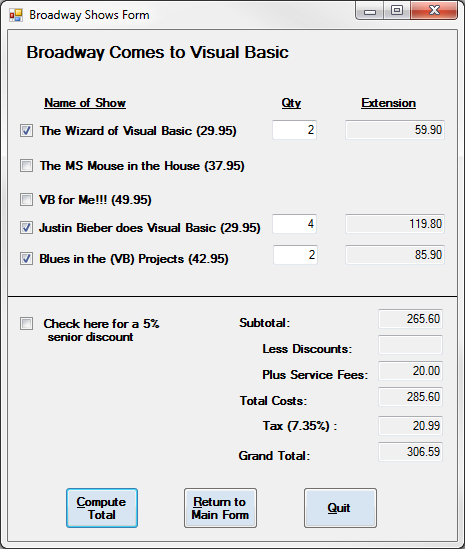
# Part A – Test Data

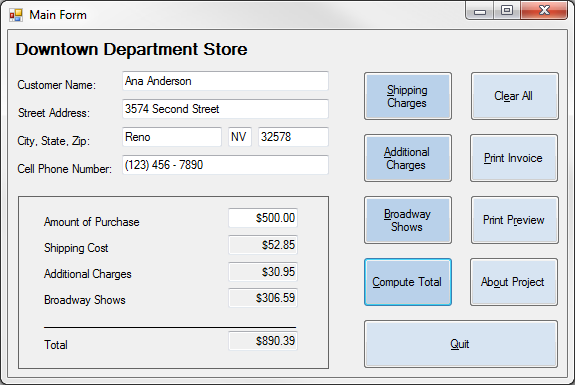
## Set One – Ana Anderson

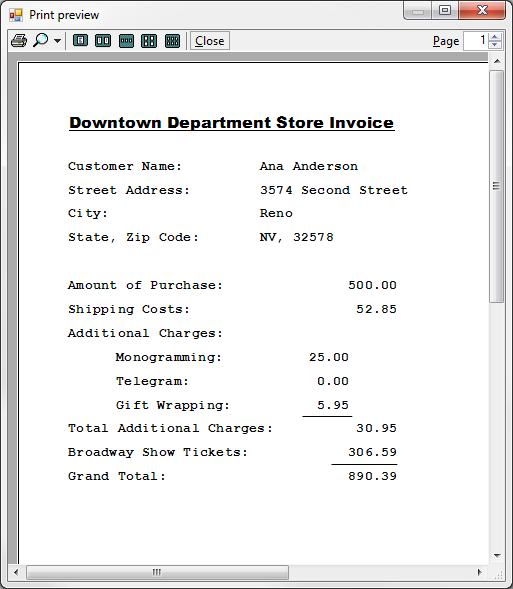




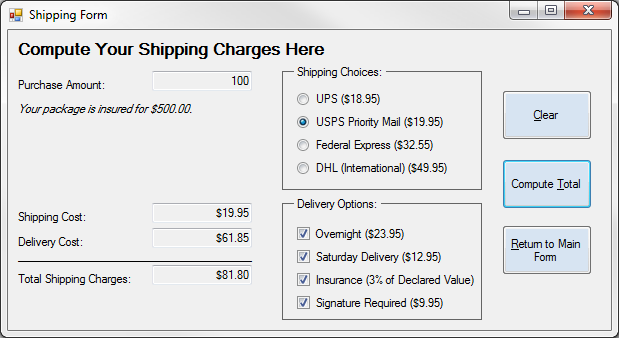


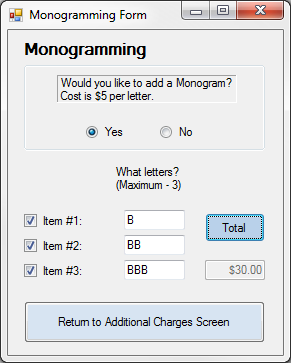


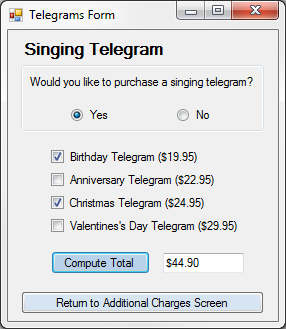


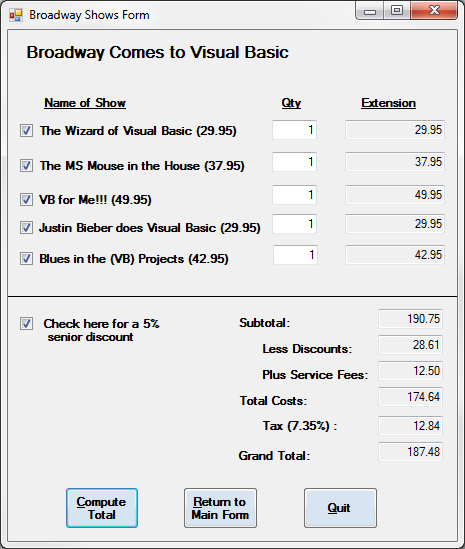


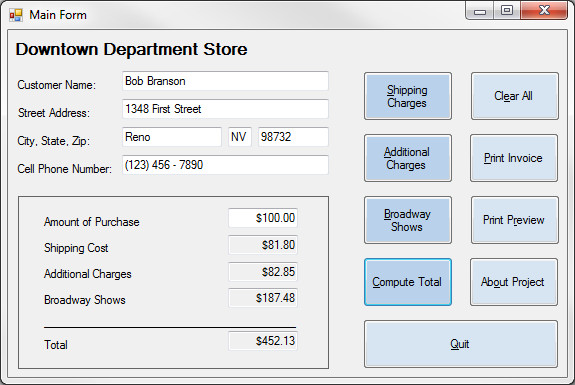
## Set Two – Bob Branson

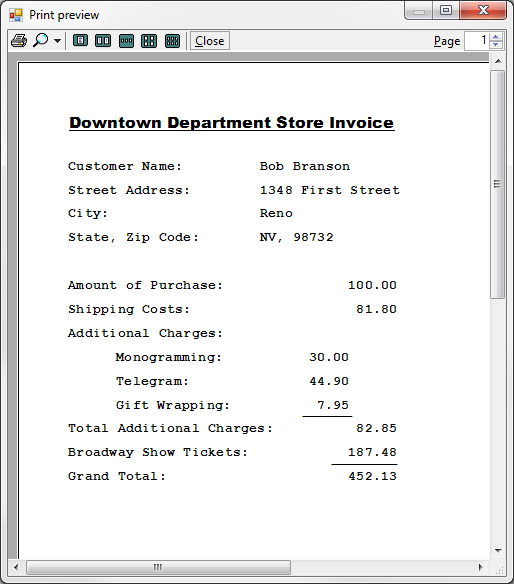




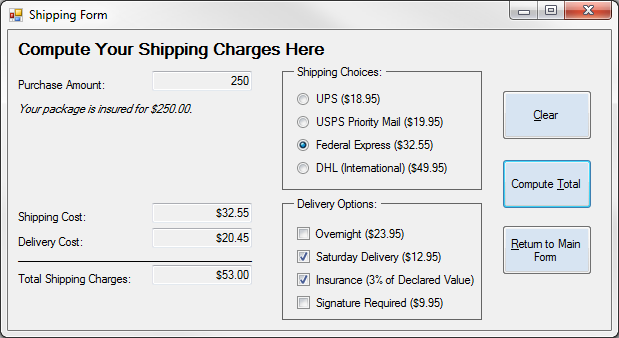


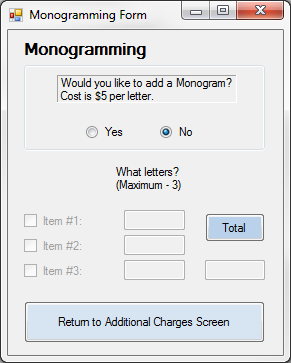


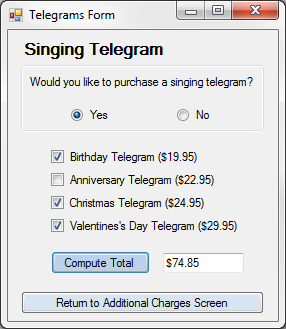
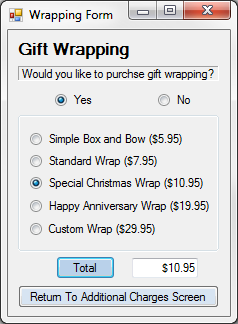


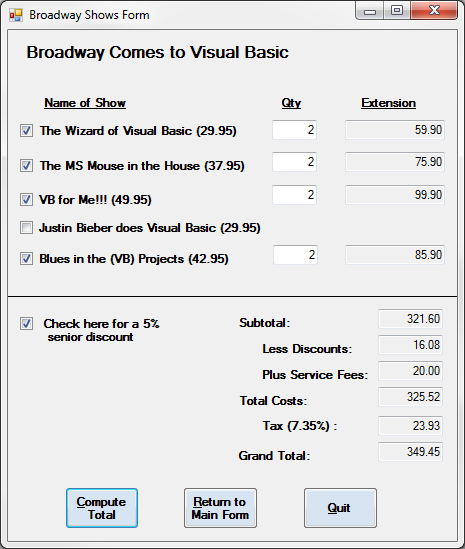


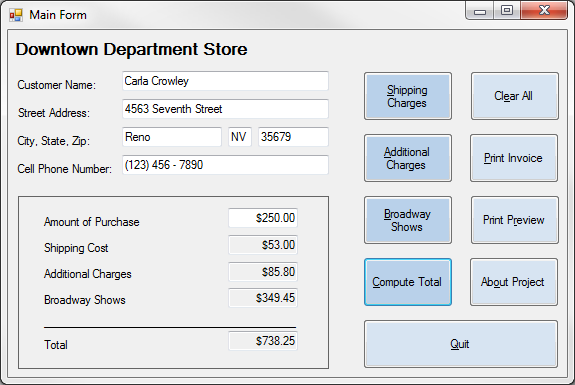
## Set Three – Carla Crowley

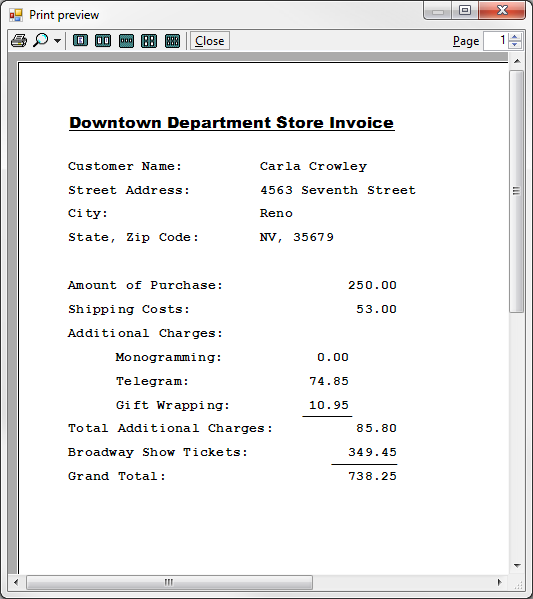




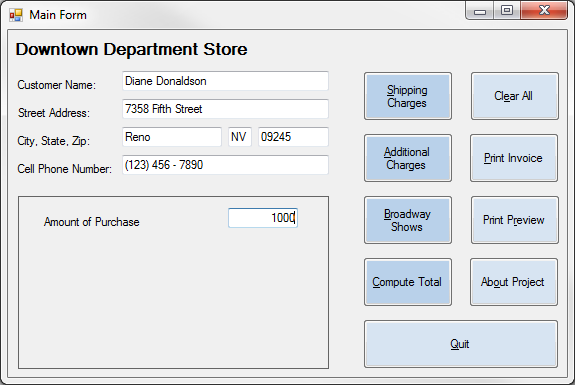


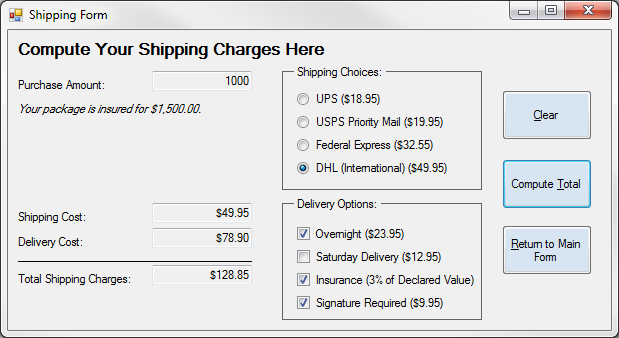


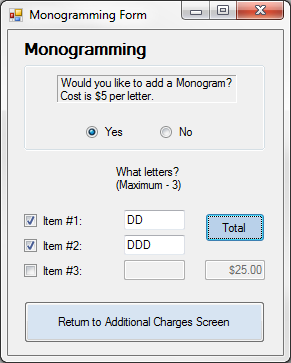


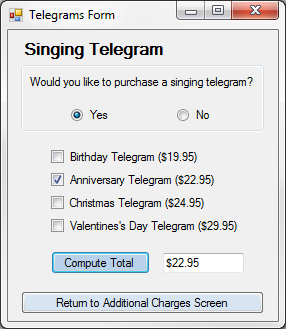
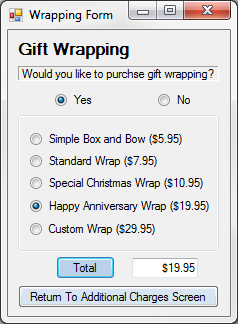


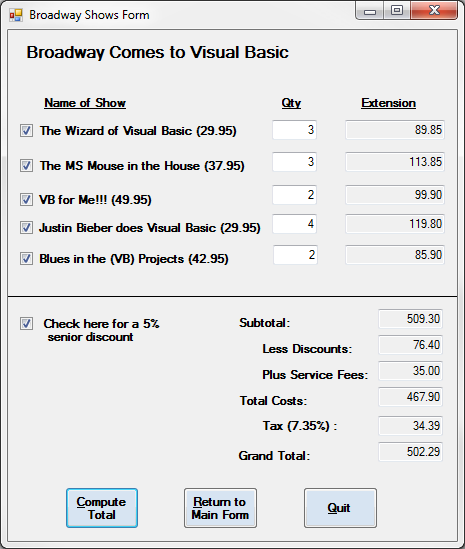
## Set Four – Diane Donaldson

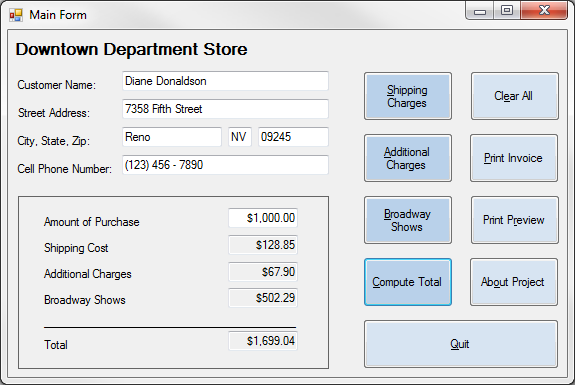


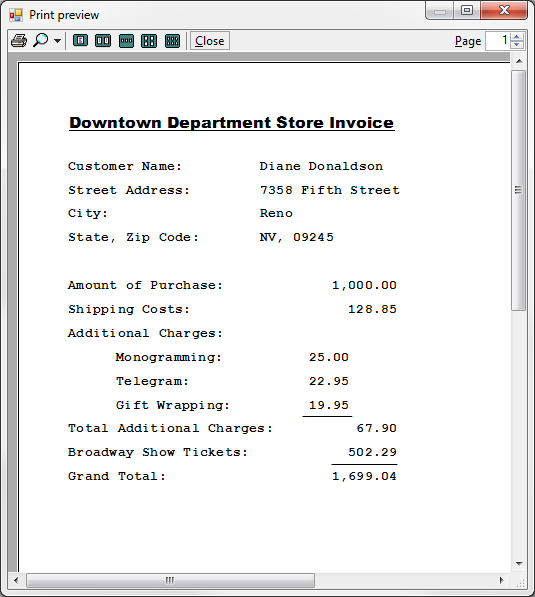






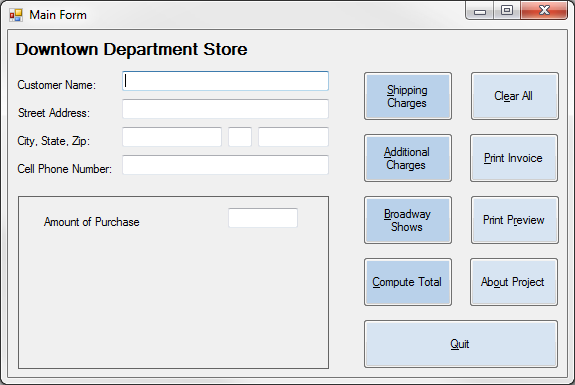


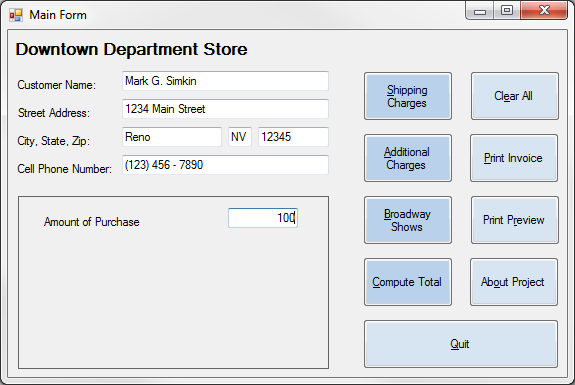




# Part B – Technical Documentation

## Module & Main Form





Module modMain

'Declare global variables.

Public Name, Address, City, StateZip As String

Public PurchaseAmount, TotalShippingCharges, TotalAdditionalCharges, TotalBroadwayShows, GrandTotal As Single

Public Insurance As String

Public ShippingCost, DeliveryCost As Single

Public monogrammingsum, telegramssum, wrappingsum As Single

End Module

Public Class frmMain

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

'Test for valid password when main form loads. User gets three attempts.

Dim Password As String

Dim attempts As Single = 0

While attempts < 5

attempts += 1

If attempts = 1 Then

Password = InputBox("Please provide your password.", "Password Attempt One")

If Password.ToLower = "a" Then

attempts = 5

ElseIf Password = "" Then

MsgBox("No Password Found.", MsgBoxStyle.Information, "Error")

End If

ElseIf attempts = 2 Then

Password = InputBox("Please provide your password.", "Password Attempt Two")

If Password.ToLower = "a" Then

attempts = 5

ElseIf Password = "" Then

MsgBox("No Password Found.", MsgBoxStyle.Information, "Error")

End If

ElseIf attempts = 3 Then

Password = InputBox("Please provide your password.", "Password Attempt Three")

If Password.ToLower = "a" Then

attempts = 5

ElseIf Password = "" Then

MsgBox("No Password Found.", MsgBoxStyle.Information, "Error")

End If

ElseIf attempts = 4 Then

MsgBox("Entered the wrong password too many times!", MsgBoxStyle.Critical, "Termination")

Me.Close()

End If

End While

End Sub

Private Sub btnClearAll\_Click(sender As Object, e As EventArgs) Handles btnClearAll.Click

'Clear all fields.

txtName.Clear()

txtAddress.Clear()

txtCity.Clear()

txtState.Clear()

txtZip.Clear()

txtPhone.Clear()

txtPurchaseAmount.Clear()

txtShippingCost.Clear()

txtAdditionalCharges.Clear()

txtBroadwayShows.Clear()

txtGrandTotal.Clear()

End Sub

Private Sub btnComputeTotal\_Click(sender As Object, e As EventArgs) Handles btnComputeTotal.Click

'Display labels and textboxes.

lblShippingCost.Visible = True

txtShippingCost.Visible = True

lblAdditionalCharges.Visible = True

txtAdditionalCharges.Visible = True

lblBroadwayShows.Visible = True

txtBroadwayShows.Visible = True

lblSpace.Visible = True

lblGrandTotal.Visible = True

txtGrandTotal.Visible = True

'Compute GrandTotal.

GrandTotal = PurchaseAmount + TotalShippingCharges + TotalAdditionalCharges + TotalBroadwayShows

'Format totals as currency.

txtPurchaseAmount.Text = FormatCurrency(PurchaseAmount)

txtShippingCost.Text = FormatCurrency(TotalShippingCharges)

txtAdditionalCharges.Text = FormatCurrency(TotalAdditionalCharges)

txtBroadwayShows.Text = FormatCurrency(TotalBroadwayShows)

txtGrandTotal.Text = FormatCurrency(GrandTotal)

End Sub

Private Sub btnAbout\_Click(sender As Object, e As EventArgs) Handles btnAbout.Click

'Show about form.

frmAbout.ShowDialog()

End Sub

Private Sub btnQuit\_Click(sender As Object, e As EventArgs) Handles btnQuit.Click

'Close the form after user ensures they wanted to quit.

If MsgBox("Are you sure you want to quit?", MsgBoxStyle.YesNo, "Quit Confirmation") = MsgBoxResult.Yes Then

Me.Close()

End If

End Sub

Private Sub btnShippingCharges\_Click(sender As Object, e As EventArgs) Handles btnShippingCharges.Click

'Test input boxes for valid content.

If txtName.Text = "" Then

MsgBox("No Customer Name Found.", MsgBoxStyle.Information, "Error")

txtName.Focus()

Exit Sub

ElseIf txtAddress.Text = "" Then

MsgBox("No Street Address Found.", MsgBoxStyle.Information, "Error")

txtAddress.Focus()

Exit Sub

ElseIf txtCity.Text = "" Then

MsgBox("No City Found.", MsgBoxStyle.Information, "Error")

txtCity.Focus()

Exit Sub

ElseIf txtState.Text = "" Then

MsgBox("No State Found.", MsgBoxStyle.Information, "Error")

txtState.Focus()

Exit Sub

ElseIf txtZip.Text = "" Then

MsgBox("No Zip Found.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf IsNumeric(txtZip.Text) = False Then

MsgBox("Zip must be numeric.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf txtZip.Text Like "#####" = False Then

MsgBox("Zip must be like 00000.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf txtPhone.Text = "" Then

MsgBox("No Phone Number Found.", MsgBoxStyle.Information, "Error")

txtPhone.Focus()

Exit Sub

ElseIf txtPurchaseAmount.Text = "" Then

MsgBox("No Purchase Amount Found.", MsgBoxStyle.Information, "Error")

txtPurchaseAmount.Focus()

Exit Sub

ElseIf IsNumeric(txtPurchaseAmount.Text) = False Then

MsgBox("Purchase Amount must be numeric.", MsgBoxStyle.Information, "Error")

txtPurchaseAmount.Focus()

Exit Sub

End If

'Save purchase amount as global variable.

PurchaseAmount = txtPurchaseAmount.Text

'Show shipping form.

frmShip.ShowDialog()

'Display global variable.

txtShippingCost.Text = TotalShippingCharges

lblShippingCost.Visible = True

txtShippingCost.Visible = True

End Sub

Private Sub btnAdditionalCharges\_Click(sender As Object, e As EventArgs) Handles btnAdditionalCharges.Click

'Test input boxes for valid content.

If txtName.Text = "" Then

MsgBox("No Customer Name Found.", MsgBoxStyle.Information, "Error")

txtName.Focus()

Exit Sub

ElseIf txtAddress.Text = "" Then

MsgBox("No Street Address Found.", MsgBoxStyle.Information, "Error")

txtAddress.Focus()

Exit Sub

ElseIf txtCity.Text = "" Then

MsgBox("No City Found.", MsgBoxStyle.Information, "Error")

txtCity.Focus()

Exit Sub

ElseIf txtState.Text = "" Then

MsgBox("No State Found.", MsgBoxStyle.Information, "Error")

txtState.Focus()

Exit Sub

ElseIf txtZip.Text = "" Then

MsgBox("No Zip Found.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf IsNumeric(txtZip.Text) = False Then

MsgBox("Zip must be numeric.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf txtZip.Text Like "#####" = False Then

MsgBox("Zip must be like 00000.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf txtPhone.Text = "" Then

MsgBox("No Phone Number Found.", MsgBoxStyle.Information, "Error")

txtPhone.Focus()

Exit Sub

ElseIf txtPurchaseAmount.Text = "" Then

MsgBox("No Purchase Amount Found.", MsgBoxStyle.Information, "Error")

txtPurchaseAmount.Focus()

Exit Sub

ElseIf IsNumeric(txtPurchaseAmount.Text) = False Then

MsgBox("Purchase Amount must be numeric.", MsgBoxStyle.Information, "Error")

txtPurchaseAmount.Focus()

Exit Sub

End If

'Save purchase amount as global variable.

PurchaseAmount = txtPurchaseAmount.Text

'Show additional charges form.

frmAdd.ShowDialog()

'Display global variable.

txtAdditionalCharges.Text = TotalAdditionalCharges

lblAdditionalCharges.Visible = True

txtAdditionalCharges.Visible = True

End Sub

Private Sub btnBroadwayShows\_Click(sender As Object, e As EventArgs) Handles btnBroadwayShows.Click

'Test input boxes for valid content.

If txtName.Text = "" Then

MsgBox("No Customer Name Found.", MsgBoxStyle.Information, "Error")

txtName.Focus()

Exit Sub

ElseIf txtAddress.Text = "" Then

MsgBox("No Street Address Found.", MsgBoxStyle.Information, "Error")

txtAddress.Focus()

Exit Sub

ElseIf txtCity.Text = "" Then

MsgBox("No City Found.", MsgBoxStyle.Information, "Error")

txtCity.Focus()

Exit Sub

ElseIf txtState.Text = "" Then

MsgBox("No State Found.", MsgBoxStyle.Information, "Error")

txtState.Focus()

Exit Sub

ElseIf txtZip.Text = "" Then

MsgBox("No Zip Found.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf IsNumeric(txtZip.Text) = False Then

MsgBox("Zip must be numeric.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf txtZip.Text Like "#####" = False Then

MsgBox("Zip must be like 00000.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

ElseIf txtPhone.Text = "" Then

MsgBox("No Phone Number Found.", MsgBoxStyle.Information, "Error")

txtPhone.Focus()

Exit Sub

ElseIf txtPurchaseAmount.Text = "" Then

MsgBox("No Purchase Amount Found.", MsgBoxStyle.Information, "Error")

txtPurchaseAmount.Focus()

Exit Sub

ElseIf IsNumeric(txtPurchaseAmount.Text) = False Then

MsgBox("Purchase Amount must be numeric.", MsgBoxStyle.Information, "Error")

txtPurchaseAmount.Focus()

Exit Sub

End If

'Save purchase amount as global variable.

PurchaseAmount = txtPurchaseAmount.Text

'Show broadway form.

frmBroad.ShowDialog()

'Display global variable.

txtBroadwayShows.Text = TotalBroadwayShows

lblBroadwayShows.Visible = True

txtBroadwayShows.Visible = True

End Sub

Private Sub btnPrintInvoice\_Click(sender As Object, e As EventArgs) Handles btnPrintInvoice.Click

'Print Invoice.

PrintInvoice.Print()

End Sub

Private Sub btnPrintPreview\_Click(sender As Object, e As EventArgs) Handles btnPrintPreview.Click

'Preview print invoice.

PreviewInvoice.Document = PrintInvoice

PreviewInvoice.ShowDialog()

End Sub

Private Sub PrintInvoice\_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintInvoice.PrintPage

'Set information variables

Name = txtName.Text

Address = txtAddress.Text

City = txtCity.Text

StateZip = txtState.Text & ", " & txtZip.Text

'Declare and set printing variables.

Dim g As Graphics = e.Graphics

Dim x0 As Single = 50

Dim x1 As Single = 275

Dim x2 As Single = 100

Dim x3 As Single = 325

Dim x4 As Single = 250

Dim y0 As Single = 50

Dim y1 As Single = 25

Dim y2 As Single = 50

Dim y3 As Single = 5

Dim h As New Font("Arial Black", 12, FontStyle.Underline)

Dim b As New Font("Courier New", 10, FontStyle.Bold)

Dim fmt As String = "{0,8:n2}"

'Draw heading.

g.DrawString("Downtown Department Store Invoice", h, Brushes.Black, x0, y0)

y0 += y2

'Draw customer information.

g.DrawString("Customer Name:", b, Brushes.Black, x0, y0)

g.DrawString(Name, b, Brushes.Black, x4, y0)

y0 += y1

g.DrawString("Street Address:", b, Brushes.Black, x0, y0)

g.DrawString(Address, b, Brushes.Black, x4, y0)

y0 += y1

g.DrawString("City:", b, Brushes.Black, x0, y0)

g.DrawString(City, b, Brushes.Black, x4, y0)

y0 += y1

g.DrawString("State, Zip Code:", b, Brushes.Black, x0, y0)

g.DrawString(StateZip, b, Brushes.Black, x4, y0)

y0 += y2

'Draw invoice amounts.

g.DrawString("Amount of Purchase:", b, Brushes.Black, x0, y0)

g.DrawString(String.Format(fmt, PurchaseAmount), b, Brushes.Black, x3, y0)

y0 += y1

g.DrawString("Shipping Costs:", b, Brushes.Black, x0, y0)

g.DrawString(String.Format(fmt, TotalShippingCharges), b, Brushes.Black, x3, y0)

y0 += y1

g.DrawString("Additional Charges:", b, Brushes.Black, x0, y0)

y0 += y1

g.DrawString("Monogramming:", b, Brushes.Black, x2, y0)

g.DrawString(String.Format(fmt, monogrammingsum), b, Brushes.Black, x1, y0)

y0 += y1

g.DrawString("Telegram:", b, Brushes.Black, x2, y0)

g.DrawString(String.Format(fmt, telegramssum), b, Brushes.Black, x1, y0)

y0 += y1

g.DrawString("Gift Wrapping:", b, Brushes.Black, x2, y0)

g.DrawString(String.Format(fmt, wrappingsum), b, Brushes.Black, x1, y0)

g.DrawString("\_\_\_\_\_\_", b, Brushes.Black, x1 + 20, y0 + y3)

y0 += y1

g.DrawString("Total Additional Charges:", b, Brushes.Black, x0, y0)

g.DrawString(String.Format(fmt, TotalAdditionalCharges), b, Brushes.Black, x3, y0)

y0 += y1

g.DrawString("Broadway Show Tickets:", b, Brushes.Black, x0, y0)

g.DrawString(String.Format(fmt, TotalBroadwayShows), b, Brushes.Black, x3, y0)

g.DrawString("\_\_\_\_\_\_\_\_", b, Brushes.Black, x3, y0 + y3)

y0 += y1

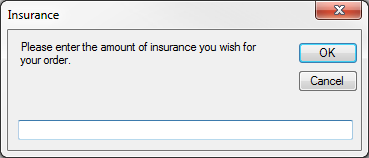
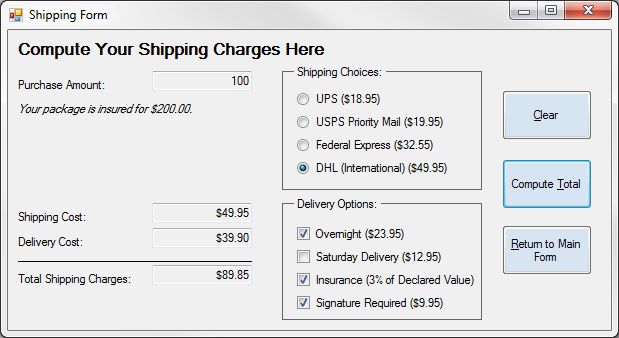
g.DrawString("Grand Total:", b, Brushes.Black, x0, y0)

g.DrawString(String.Format(fmt, GrandTotal), b, Brushes.Black, x3, y0)

End Sub

End Class

## Shipping Form



Public Class frmShip

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

'Display purchase amount global variable locally.

txtPurchaseAmount.Text = PurchaseAmount

End Sub

Private Sub btnClear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click

'Clear all fields and local variables.

rdoUPS.Checked = False

rdoUSPS.Checked = False

rdoFedEx.Checked = False

rdoDHL.Checked = False

chkOvernight.Checked = False

chkSaturday.Checked = False

chkSignature.Checked = False

chkInsurance.Checked = False

shippingCost = 0

txtShippingCost.Clear()

deliveryCost = 0

txtDeliveryCost.Clear()

txtTotalShippingCharges.Clear()

End Sub

Private Sub btnComputeTotal\_Click(sender As Object, e As EventArgs) Handles btnComputeTotal.Click

'Compute the total shipping costs.

txtShippingCost.Text = FormatCurrency(ShippingCost)

txtDeliveryCost.Text = FormatCurrency(DeliveryCost)

TotalShippingCharges = ShippingCost + DeliveryCost

txtTotalShippingCharges.Text = FormatCurrency(TotalShippingCharges)

End Sub

Private Sub btnReturn\_Click(sender As Object, e As EventArgs) Handles btnReturn.Click

'Save shipping charges as a global variable and then close.

TotalShippingCharges = Math.Round(TotalShippingCharges, 2)

Me.Close()

End Sub

Private Sub rdoDHL\_rdoFedEx\_rdoUSPS\_rdoUPS\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoDHL.CheckedChanged, rdoFedEx.CheckedChanged, rdoUSPS.CheckedChanged, rdoUPS.CheckedChanged

'Recalculate the shipping costs each time a radio button is clicked.

txtShippingCost.Clear()

ShippingCost = 0

If rdoDHL.Checked Then

ShippingCost = 49.95

End If

If rdoFedEx.Checked Then

ShippingCost = 32.55

End If

If rdoUSPS.Checked Then

ShippingCost = 19.95

End If

If rdoUPS.Checked Then

ShippingCost = 18.95

End If

txtShippingCost.Text = ShippingCost

End Sub

Private Sub chkInsurance\_CheckedChanged(sender As Object, e As EventArgs) Handles chkInsurance.CheckedChanged

'Check insurance and set the insurance variable.

If chkInsurance.Checked Then

Insurance = InputBox("Please enter the amount of insurance you wish for your order.", "Insurance")

If Insurance = "" Then

MsgBox("Insurance was not found.", MsgBoxStyle.Information, "Error")

chkInsurance.Checked = False

Exit Sub

ElseIf IsNumeric(Insurance) = False Then

MsgBox("Insurance is not numeric.", MsgBoxStyle.Information, "Error")

chkInsurance.Checked = False

Exit Sub

ElseIf Insurance < 1 Or Insurance > 100000 Then

MsgBox("Insurance must be between 1 and 100000.", MsgBoxStyle.Information, "Error")

chkInsurance.Checked = False

Exit Sub

End If

lblInsurance.Text = "Your package is insured for " & FormatCurrency(Insurance) & "."

lblInsurance.Visible = True

Else

lblInsurance.Visible = False

Insurance = 0

End If

End Sub

Private Sub chkSignature\_chkSaturday\_chkOvernight\_CheckedChanged(sender As Object, e As EventArgs) Handles chkSignature.CheckedChanged, chkSaturday.CheckedChanged, chkInsurance.CheckedChanged, chkOvernight.CheckedChanged

'Recalculate the delivery costs each time a delivery option is clicked.

txtDeliveryCost.Clear()

DeliveryCost = 0

If chkSignature.Checked Then

DeliveryCost += 9.95

End If

If chkSaturday.Checked Then

DeliveryCost += 12.95

End If

If chkInsurance.Checked Then

DeliveryCost += Insurance \* 0.03

End If

If chkOvernight.Checked Then

DeliveryCost += 23.95

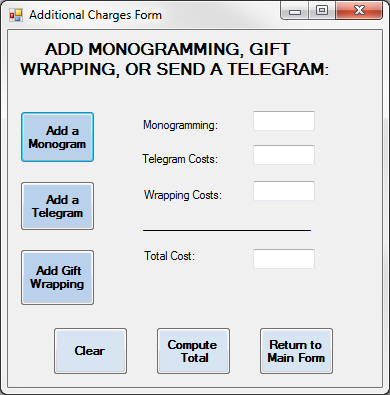
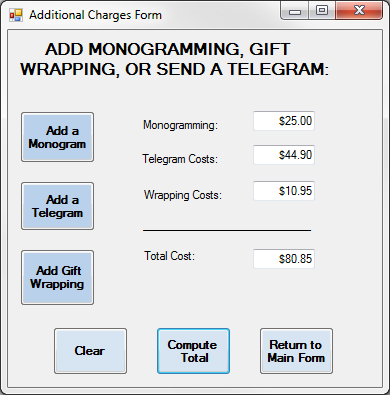
End If

txtDeliveryCost.Text = DeliveryCost

End Sub

End Class

## Additional Charges Form



Public Class frmAdd

Private Sub btnMonogram\_Click(sender As Object, e As EventArgs) Handles btnMonogram.Click

'Show monogram form.

frmMono.ShowDialog()

txtBoxMono.Text = FormatCurrency(monogrammingsum)

End Sub

Private Sub btnTelegram\_Click(sender As Object, e As EventArgs) Handles btnTelegram.Click

'Show telegram form.

frmTele.ShowDialog()

txtBoxTele.Text = FormatCurrency(telegramssum)

End Sub

Private Sub btnWrapping\_Click(sender As Object, e As EventArgs) Handles btnWrapping.Click

'Show wrapping form.

frmWrap.ShowDialog()

txtBoxWrap.Text = FormatCurrency(wrappingsum)

End Sub

Private Sub btnClear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click

'Clear all fields.

txtBoxMono.Clear()

txtBoxTele.Clear()

txtBoxWrap.Clear()

txtBoxTotalCost.Clear()

End Sub

Private Sub btnComputeTotal\_Click(sender As Object, e As EventArgs) Handles btnComputeTotal.Click

'Compute the total shipping costs and format as currency.

TotalAdditionalCharges = monogrammingsum + telegramssum + wrappingsum

txtBoxTotalCost.Text = FormatCurrency(TotalAdditionalCharges)

End Sub

Private Sub btnReturn\_Click(sender As Object, e As EventArgs) Handles btnReturn.Click

'Save additional charges as a global variable and then close.

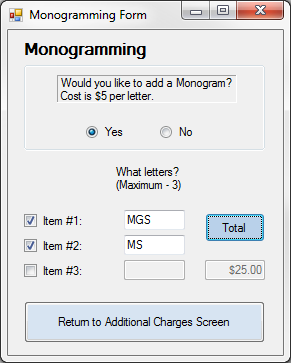
TotalAdditionalCharges = Math.Round(TotalAdditionalCharges, 2)

Me.Close()

End Sub

End Class

### Monogramming Form



Public Class frmMono

Private Sub Monogramming\_Load\_radButtonNo\_CheckedChanged(sender As Object, e As EventArgs) Handles MyBase.Load, radButtonNo.CheckedChanged

'Disable checkboxes and textboxes.

chkBox1.Enabled = False

chkBox2.Enabled = False

chkBox3.Enabled = False

chkBox1.Checked = False

txtBoxItem1.Enabled = False

txtBoxItem2.Enabled = False

txtBoxItem3.Enabled = False

txtBoxMonogramming.Enabled = False

End Sub

Private Sub radButtonYes\_CheckedChanged(sender As Object, e As EventArgs) Handles radButtonYes.CheckedChanged

'Enable the checkboxes and check the first checkbox.

chkBox1.Enabled = True

chkBox2.Enabled = True

chkBox3.Enabled = True

txtBoxItem1.Enabled = True

chkBox1.Checked = True

End Sub

Private Sub chkBox2\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBox2.CheckedChanged

'Enable textbox 2 when checkbox 2 is checked.

txtBoxItem2.Enabled = True

End Sub

Private Sub chkBox3\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBox3.CheckedChanged

'Enable textbox 3 when checkbox 3 is checked.

txtBoxItem3.Enabled = True

End Sub

Private Sub btnTotal\_Click(sender As Object, e As EventArgs) Handles btnTotal.Click

'Compute the total price based on each items length.

Dim length1, price1 As Double

Dim length2, price2 As Double

Dim length3, price3 As Double

If chkBox1.Checked Then

'Test input box for valid content.

If txtBoxItem1.Text = "" Then

MsgBox("No Item #1 Monogram Found.", MsgBoxStyle.Information, "Error")

txtBoxItem1.Focus()

Exit Sub

ElseIf txtBoxItem1.Text.Trim.Length <= 3 Then

MsgBox("Item #1 Monogram is 3 or less characters.", MsgBoxStyle.Information, "Error")

End If

length1 = txtBoxItem1.Text.Trim.Length

price1 = length1 \* 5

End If

If chkBox2.Checked Then

'Test input box for valid content.

If txtBoxItem2.Text = "" Then

MsgBox("No Item #2 Monogram Found.", MsgBoxStyle.Information, "Error")

txtBoxItem2.Focus()

Exit Sub

ElseIf txtBoxItem2.Text.Trim.Length <= 3 Then

MsgBox("Item #2 Monogram is 3 or less characters.", MsgBoxStyle.Information, "Error")

End If

length2 = txtBoxItem2.Text.Trim.Length

price2 = length2 \* 5

End If

If chkBox3.Checked Then

'Test input box for valid content.

If txtBoxItem3.Text = "" Then

MsgBox("No Item #3 Monogram Found.", MsgBoxStyle.Information, "Error")

txtBoxItem3.Focus()

Exit Sub

ElseIf txtBoxItem3.Text.Trim.Length <= 3 Then

MsgBox("Item #3 Monogram is 3 or less characters.", MsgBoxStyle.Information, "Error")

End If

length3 = txtBoxItem3.Text.Trim.Length

price3 = length3 \* 5

End If

monogrammingsum = price1 + price2 + price3

txtBoxMonogramming.Text = FormatCurrency(monogrammingsum)

End Sub

Private Sub btnReturn\_Click(sender As Object, e As EventArgs) Handles btnReturn.Click

'Save monogrammingsum as a global variable and close.

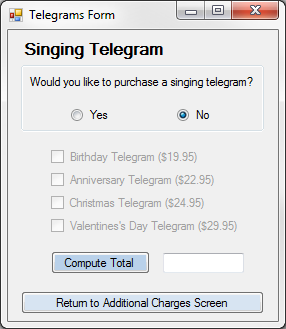
monogrammingsum = Math.Round(monogrammingsum, 2)

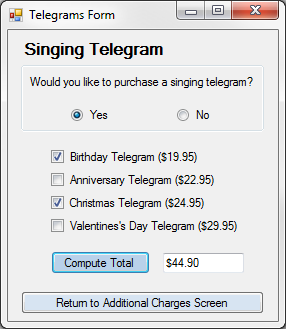
Me.Close()

End Sub

End Class

### Telegram Form





Public Class frmTele

Private Sub Telegrams\_Load\_radBtnNo\_CheckedChanged(sender As Object, e As EventArgs) Handles MyBase.Load, radBtnNo.CheckedChanged

'Disable checkboxes.

chkBoxAnniv.Enabled = False

chkBoxBirth.Enabled = False

chkBoxChrist.Enabled = False

chkBoxValen.Enabled = False

End Sub

Private Sub radBtnYes\_CheckedChanged(sender As Object, e As EventArgs) Handles radBtnYes.CheckedChanged

'Enable checkboxes when yes button is clicked.

chkBoxAnniv.Enabled = True

chkBoxBirth.Enabled = True

chkBoxChrist.Enabled = True

chkBoxValen.Enabled = True

End Sub

Private Sub btnCmpt\_Click(sender As Object, e As EventArgs) Handles btnCmpt.Click

'Compute total cost.

Dim Birth, Anniv, Christ, Valen As Double

txtBoxResult.Clear()

If chkBoxBirth.Checked Then

Birth = 19.95

End If

If chkBoxAnniv.Checked Then

Anniv = 22.95

End If

If chkBoxChrist.Checked Then

Christ = 24.95

End If

If chkBoxValen.Checked Then

Valen = 29.95

End If

telegramssum = Birth + Anniv + Christ + Valen

txtBoxResult.Text = FormatCurrency(telegramssum)

End Sub

Private Sub chkBoxBirth\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBoxBirth.CheckedChanged, chkBoxAnniv.CheckedChanged, chkBoxChrist.CheckedChanged, chkBoxValen.CheckedChanged

'Reset total cost when checkbox is checked.

If chkBoxBirth.Checked = True Then txtBoxResult.Clear()

If chkBoxAnniv.Checked = True Then txtBoxResult.Clear()

If chkBoxChrist.Checked = True Then txtBoxResult.Clear()

If chkBoxValen.Checked = True Then txtBoxResult.Clear()

txtBoxResult.Text = "0"

End Sub

Private Sub btnRetrn\_Click(sender As Object, e As EventArgs) Handles btnRetrn.Click

'Save telegrammssum as global variable and close.

telegramssum = Math.Round(telegramssum, 2)

Me.Close()

End Sub

End Class

### Gift Wrapping Form





Public Class frmWrap

Private Sub Wrapping\_Load\_radButtonNo\_CheckedChanged(sender As Object, e As EventArgs) Handles MyBase.Load, radButtonNo.CheckedChanged

'Disable radio buttons on load.

radButton1.Enabled = False

radButton2.Enabled = False

radButton3.Enabled = False

radButton4.Enabled = False

radButton5.Enabled = False

End Sub

Private Sub radButtonYes\_CheckedChanged(sender As Object, e As EventArgs) Handles radButtonYes.CheckedChanged

'Enable radio buttons when yes button is clicked.

radButton1.Enabled = True

radButton2.Enabled = True

radButton3.Enabled = True

radButton4.Enabled = True

radButton5.Enabled = True

End Sub

Private Sub btnTotal\_Click(sender As Object, e As EventArgs) Handles btnTotal.Click

'Compute the total based on which radio button is checked.

Dim z As Double

If radButton1.Checked = True Then txtBoxResult.Text = 5.95

If radButton2.Checked = True Then txtBoxResult.Text = 7.95

If radButton3.Checked = True Then txtBoxResult.Text = 10.95

If radButton4.Checked = True Then txtBoxResult.Text = 19.95

If radButton5.Checked = True Then txtBoxResult.Text = 29.95

z = txtBoxResult.Text

txtBoxResult.Text = FormatCurrency(z)

End Sub

Private Sub radButton1\_CheckedChanged(sender As Object, e As EventArgs) Handles radButton1.CheckedChanged, radButton2.CheckedChanged, radButton3.CheckedChanged, radButton4.CheckedChanged, radButton5.CheckedChanged

'Clear the total each time a radio button is clicked.

If radButton1.Checked = True Then txtBoxResult.Clear()

If radButton2.Checked = True Then txtBoxResult.Clear()

If radButton3.Checked = True Then txtBoxResult.Clear()

If radButton4.Checked = True Then txtBoxResult.Clear()

If radButton5.Checked = True Then txtBoxResult.Clear()

End Sub

Private Sub btnReturn\_Click(sender As Object, e As EventArgs) Handles btnReturn.Click

'Save wrappingsum as a global variable and close.

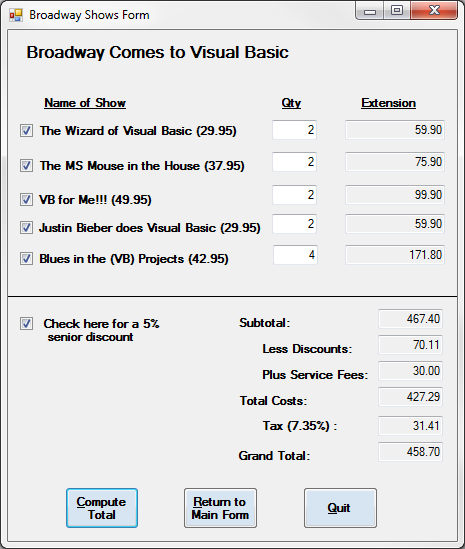
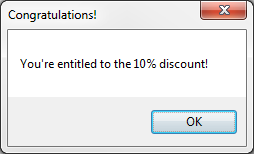
wrappingsum = txtBoxResult.Text

Me.Close()

End Sub

End Class

## Broadway Form



Public Class frmBroad

Private Sub chkBox1\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBox1.CheckedChanged

'If CheckBox 1 is checked then it becomes visible, if it is unchecked it is hidden.

If chkBox1.Checked = True Then

txtBoxQTY1.Visible = True

txtBoxExtension1.Visible = True

Else

txtBoxQTY1.Visible = False

txtBoxExtension1.Visible = False

txtBoxQTY1.Clear()

txtBoxExtension1.Clear()

End If

End Sub

Private Sub btnQuit\_Click(sender As Object, e As EventArgs) Handles btnQuit.Click

'Close the form after user ensures they wanted to quit.

If MsgBox("Are you sure you want to quit?", MsgBoxStyle.YesNo, "Quit Confirmation") = MsgBoxResult.Yes Then

Me.Close()

End If

End Sub

Private Sub chkBox2\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBox2.CheckedChanged

'If CheckBox 2 is checked then it becomes visible, if it is unchecked it is hidden.

If chkBox2.Checked = True Then

txtBoxQTY2.Visible = True

txtBoxExtension2.Visible = True

Else

txtBoxQTY2.Visible = False

txtBoxExtension2.Visible = False

txtBoxQTY2.Clear()

txtBoxExtension2.Clear()

End If

End Sub

Private Sub chkBox3\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBox3.CheckedChanged

'If CheckBox 3 is checked then it becomes visible, if it is unchecked it is hidden.

If chkBox3.Checked = True Then

txtBoxQTY3.Visible = True

txtBoxExtension3.Visible = True

Else

txtBoxQTY3.Visible = False

txtBoxExtension3.Visible = False

txtBoxQTY3.Clear()

txtBoxExtension3.Clear()

End If

End Sub

Private Sub chkBox4\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBox4.CheckedChanged

'If CheckBox 4 is checked then it becomes visible, if it is unchecked it is hidden.

If chkBox4.Checked = True Then

txtBoxQTY4.Visible = True

txtBoxExtension4.Visible = True

Else

txtBoxQTY4.Visible = False

txtBoxExtension4.Visible = False

txtBoxQTY4.Clear()

txtBoxExtension4.Clear()

End If

End Sub

Private Sub chkBox5\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBox5.CheckedChanged

'If CheckBox 5 is checked then it becomes visible, if it is unchecked it is hidden.

If chkBox5.Checked = True Then

txtBoxQTY5.Visible = True

txtBoxExtension5.Visible = True

Else

txtBoxExtension5.Visible = False

txtBoxQTY5.Visible = False

txtBoxQTY5.Clear()

txtBoxExtension5.Clear()

End If

End Sub

Private Sub btnComputeTotal\_Click(sender As Object, e As EventArgs) Handles btnComputeTotal.Click

'Calculates total price of tickets in extension box 1 by multiplying quantity times price.

If chkBox1.Checked Then

If IsNumeric(txtBoxQTY1.Text) = False Then

MsgBox("Quantity must be a number.", , "Error")

txtBoxQTY1.Clear()

Exit Sub

Else

txtBoxExtension1.Text = FormatNumber(txtBoxQTY1.Text \* 29.95)

End If

End If

'Calculates total price of tickets in extension box 2 by multiplying quantity times price.

If chkBox2.Checked Then

If IsNumeric(txtBoxQTY2.Text) = False Then

MsgBox("Quantity must be a number.", , "Error")

txtBoxQTY2.Clear()

Exit Sub

Else

txtBoxExtension2.Text = FormatNumber(txtBoxQTY2.Text \* 37.95)

End If

End If

'Calculates total price of tickets in extension box 3 by multiplying quantity times price.

If chkBox3.Checked Then

If IsNumeric(txtBoxQTY3.Text) = False Then

MsgBox("Quantity must be a number.", , "Error")

txtBoxQTY3.Clear()

Exit Sub

Else

txtBoxExtension3.Text = FormatNumber(txtBoxQTY3.Text \* 49.95)

End If

End If

'Calculates total price of tickets in extension box 4 by multiplying quantity times price.

If chkBox4.Checked Then

If IsNumeric(txtBoxQTY4.Text) = False Then

MsgBox("Quantity must be a number.", , "Error")

txtBoxQTY4.Clear()

Exit Sub

Else

txtBoxExtension4.Text = FormatNumber(txtBoxQTY4.Text \* 29.95)

End If

End If

'Calculates total price of tickets in extension box 5 by multiplying quantity times price.

If chkBox5.Checked Then

If IsNumeric(txtBoxQTY5.Text) = False Then

MsgBox("Quantity must be a number.", , "Error")

txtBoxQTY5.Clear()

Exit Sub

Else

txtBoxExtension5.Text = FormatNumber(txtBoxQTY5.Text \* 42.95)

End If

End If

'Calculates total price of all tickets in Subtotal Textbox.

Dim total As Double

Dim x1 As Double = Val(txtBoxExtension1.Text)

Dim x2 As Double = Val(txtBoxExtension2.Text)

Dim x3 As Double = Val(txtBoxExtension3.Text)

Dim x4 As Double = Val(txtBoxExtension4.Text)

Dim x5 As Double = Val(txtBoxExtension5.Text)

total = x1 + x2 + x3 + x4 + x5

txtBoxSubtotal.Text = FormatNumber(total)

'Gives 10% discount if customer buys at least one of each ticket.

If IsNumeric(txtBoxQTY1.Text) = True And

IsNumeric(txtBoxQTY2.Text) = True And

IsNumeric(txtBoxQTY3.Text) = True And

IsNumeric(txtBoxQTY4.Text) = True And

IsNumeric(txtBoxQTY5.Text) = True Then

txtBoxDiscounts.Text = (0.1 \* txtBoxSubtotal.Text)

MsgBox("You're entitled to the 10% discount!", , "Congratulations!")

Else : txtBoxDiscounts.Text = ""

End If

'Gives 5% Senior discount if checkbox is checked.

If chkBox6.Checked = True Then

txtBoxDiscounts.Text = FormatNumber(0.05 \* txtBoxSubtotal.Text)

Else

txtBoxDiscounts.Clear()

End If

'Gives 15% discount if customer checks senior discount box and orders at least one of each ticket.

If IsNumeric(txtBoxQTY1.Text) = True And

IsNumeric(txtBoxQTY2.Text) = True And

IsNumeric(txtBoxQTY3.Text) = True And

IsNumeric(txtBoxQTY4.Text) = True And

IsNumeric(txtBoxQTY5.Text) = True And

chkBox6.Checked = True Then

txtBoxDiscounts.Text = FormatNumber(0.15 \* txtBoxSubtotal.Text)

End If

'Gives 10% Discount if Senior Discount is unchecked and customer orders at least one of each ticket.

If chkBox6.Checked = False And

IsNumeric(txtBoxQTY1.Text) = True And

IsNumeric(txtBoxQTY2.Text) = True And

IsNumeric(txtBoxQTY3.Text) = True And

IsNumeric(txtBoxQTY4.Text) = True And

IsNumeric(txtBoxQTY5.Text) = True Then

txtBoxDiscounts.Text = FormatNumber(0.1 \* txtBoxSubtotal.Text)

End If

'Adds service fees of $2.50 times quantity of tickets selected.

Dim boxQ1, boxQ2, boxQ3, boxQ4, boxQ5 As Double

boxQ1 = Val(txtBoxQTY1.Text)

boxQ2 = Val(txtBoxQTY2.Text)

boxQ3 = Val(txtBoxQTY3.Text)

boxQ4 = Val(txtBoxQTY4.Text)

boxQ5 = Val(txtBoxQTY5.Text)

txtBoxServiceFees.Text = FormatNumber((boxQ1 + boxQ2 + boxQ3 + boxQ4 + boxQ5) \* 2.5)

'Total cost of tickets after adding service fees and subtracting discounts.

Dim y As Double = Val(txtBoxSubtotal.Text)

Dim z As Double = Val(txtBoxServiceFees.Text)

Dim q As Double = Val(txtBoxDiscounts.Text)

txtBoxTotalCost.Text = FormatNumber(y + (z - q))

'Calculates cost of tax at 7.35%.

Dim n As Double = Val(txtBoxTotalCost.Text)

txtBoxTax.Text = FormatNumber(0.0735 \* n)

'Grand total is total costs plus tax.

Dim l As Double = Val(txtBoxTax.Text)

Dim o As Double = Val(txtBoxTotalCost.Text)

txtBoxGrandTotal.Text = FormatNumber(o + l)

TotalBroadwayShows = txtBoxGrandTotal.Text

End Sub

Private Sub btnReturn\_Click(sender As Object, e As EventArgs) Handles btnReturn.Click

'Alerts customer if they try to return to main form without having computed grand total.

If IsNumeric(txtBoxGrandTotal.Text) = False Then

MsgBox("Grand Total is blank!", , "Return")

Else

TotalBroadwayShows = Math.Round(TotalBroadwayShows, 2)

Close()

End If

End Sub

Private Sub chkBox6\_CheckedChanged(sender As Object, e As EventArgs) Handles chkBox6.CheckedChanged

'Clears grand total textbox if senior discount is unchecked or checked after already being computed.

If chkBox6.Checked = False Then

txtBoxGrandTotal.Clear()

End If

If chkBox6.Checked = True Then

txtBoxGrandTotal.Clear()

End If

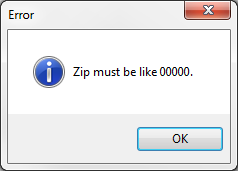
End Sub

End Class

# Part C – Enhancements

## Enhancement 1 – Zip Code Validation

For our first enhancement, we decided to create and additional zip code validation. This validation tests if the zip code entered matches a string of five integers. If the test fails the user is notified that the zip code is not in the proper format and then the sub will terminate execution. The validation was inserted into the existing validation rules for the Shipping, Additional Charges, and Broadway Click events.



ElseIf txtZip.Text Like "#####" = False Then

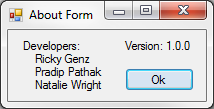
MsgBox("Zip must be like 00000.", MsgBoxStyle.Information, "Error")

txtZip.Focus()

Exit Sub

## Enhancement 2 – About Form

For our second enhancement, we decided to create an about form. This form informs the user of the developers who built the application and the current version number of the application. The form performs one task. If the user clicks on the “Ok” button, the about form will terminate execution.



Public Class frmAbout

Private Sub btnOk\_Click(sender As Object, e As EventArgs) Handles btnOk.Click

'Close about form after user clicks ok

Me.Close()

End Sub

End Class