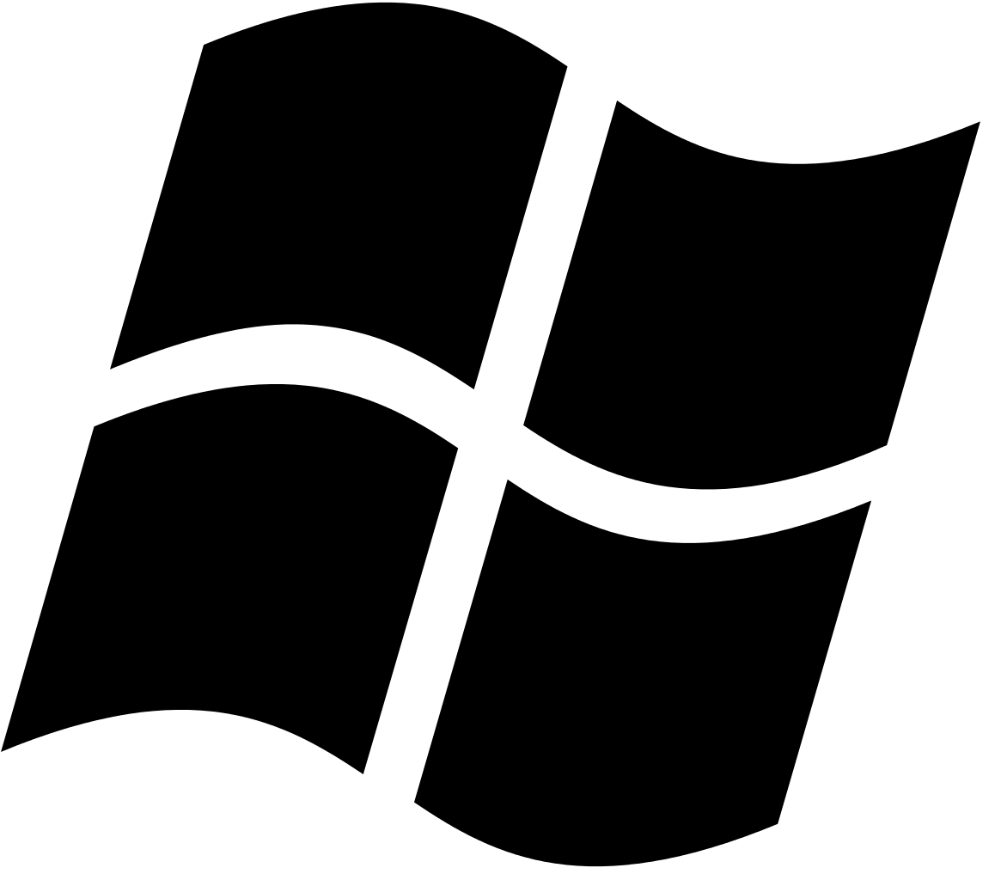
|  |  |  |  |
| --- | --- | --- | --- |
| Page  Number | Problem | Max  Points | Points  Scored |
| 2 | Main Screen Interface | 2 |  |
| 3 | Supporting Screen Interface | 2 |  |
| 4 | A | 4 |  |
| 4 | B | 3 |  |
| 4 | C | 8 |  |
| 6 | D | 3 |  |
| 6 | E | 7 |  |
| 6 | F | 4 |  |
| 6 | G | 7 |  |
| 7 | H | 2 |  |
| 5 | I | 1 |  |
| 5 | J | 5 |  |
| 7 | K | 2 |  |
|  | Total | 50 |  |

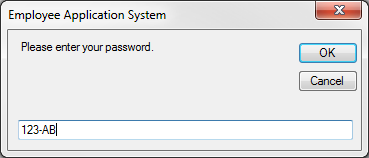


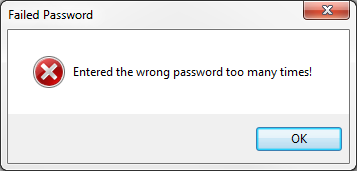
Kenneth R Genz III

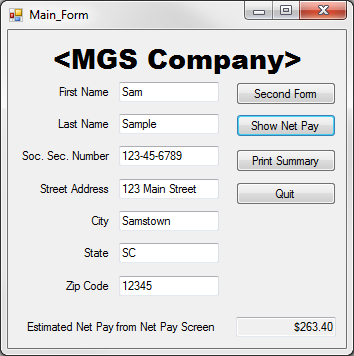
IS 350 1001

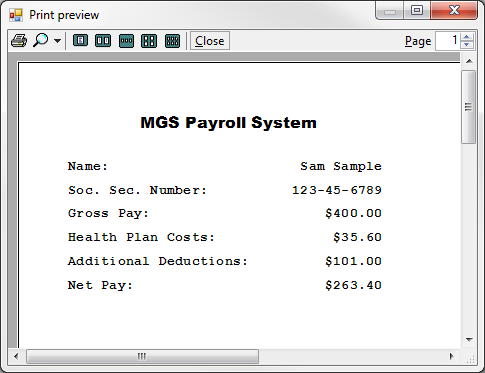
Spring 2015

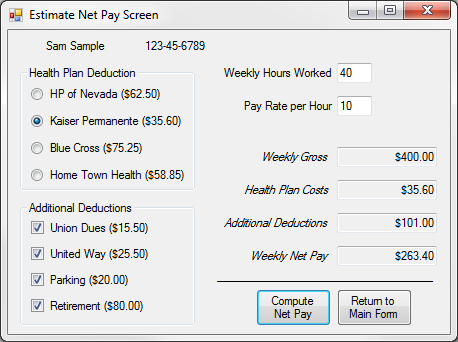
Midterm Part II

Figure 1.0 Password Input

Figure 1.1 Failed Password

Figure 2.0 Main Screen

Figure 3.0 Print Summary

Figure 4.0 Supporting Screen

A. Password Validation

Public Class frmMain

Function IsPassword() As Boolean

'Test password input for valid data.

Password = InputBox("Please enter your password.", "Employee Application System")

If Password Like "###-[A-Z][A-Z]" Then

Attempts = 5

Return True

ElseIf Password = "" Then

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

End If

Return False

End Function

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

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

While Attempts < 5

Attempts += 1

If Attempts = 1 Then IsPassword()

If Attempts = 2 Then IsPassword()

If Attempts = 3 Then IsPassword()

If Attempts = 4 Then

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

End

End If

End While

End Sub

B. Social Security Number Validation & \_

C. Data Entry Validation

Function IsInput(ByVal s As String) As Boolean

'Test input fields for valid data.

Dim Controls = From c In Me.Controls Order By c.TabIndex

For Each c In Controls

If c.AccessibleRole = Windows.Forms.AccessibleRole.Text Then

If Len(Trim(c.Text)) <= 0 Then

MsgBox(c.AccessibleName & " is missing required info.", MsgBoxStyle.Information, "Error")

c.Focus()

Return False

End If

End If

If c.AccessibleName Like s Then

If Not c.Text Like "###-##-####" Then

MsgBox(c.AccessibleName & " is not valid.", MsgBoxStyle.Critical, "Error")

c.Focus()

Return False

End If

End If

Next

Return True

End Function

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

'Test for valid SSN, First Name, Last Name, Address, City, State, and Zip Code.

If IsInput("SSN") Then

FullName = txtFirstName.Text & " " & txtLastName.Text

SSN = txtSSN.Text

frmNetPay.ShowDialog()

End If

End Sub

I. Show Net Pay

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

'Display Net Pay

txtNetPay.Text = FormatCurrency(NetPay)

End Sub

J. Prepare a Printed Copy of the Weekly Gross Computations

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

'Print summary.

PreviewSummary.Document = PrintSummary

PreviewSummary.ShowDialog()

End Sub

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

'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 = 125

Dim y0 As Single = 50

Dim y1 As Single = 25

Dim y2 As Single = 50

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

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

Dim fmt As String = "{0,12:c2}"

'Draw heading.

g.DrawString("MGS Payroll System", h, Brushes.Black, x2, y0)

y0 += y2

'Draw information.

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

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

y0 += y1

g.DrawString("Soc. Sec. Number:", b, Brushes.Black, x0, y0)

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

y0 += y1

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

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

y0 += y1

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

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

y0 += y1

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

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

y0 += y1

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

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

End Sub

The Quit Button

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

'End execution

End

End Sub

End Class

D. Display Employee’s Name and Social Security Number in the New Screen

Public Class frmNetPay

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

'Display Name and SSN on form load.

lblName.Text = FullName

lblSSN.Text = SSN

End Sub

E. Computing Estimated Weekly Gross & \_

G. Computing Estimated Net Pay

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

'Calculate Gross and Net Weekly Pay. Format all values to currency.

Hours = CInt(txtHours.Text)

Rate = CDec(txtRate.Text)

Select Case Hours

Case 1 To 40

Gross = Hours \* Rate

Case 41 To 60

Gross = (40 \* Rate) + ((Hours - 40) \* Rate \* 1.5)

Case Is > 60

Gross = (40 \* Rate) + (20 \* Rate \* 1.5) + ((Hours - 60) \* Rate \* 2)

End Select

txtGross.Text = FormatCurrency(Gross)

txtHealthPlanCosts.Text = FormatCurrency(HPCosts)

txtAdditionalDeductions.Text = FormatCurrency(ADCosts)

NetPay = Gross - HPCosts - ADCosts

txtNetPay.Text = FormatCurrency(NetPay)

End Sub

F. Computing Health Plan and Additional Deductions

Private Sub HealthPlanDeductions(sender As Object, e As EventArgs) Handles rdoHPOfNevada.CheckedChanged, rdoKaiserPermanente.CheckedChanged, rdoBlueCross.CheckedChanged, rdoHomeTownHealth.CheckedChanged

'Recalculate the Health Plan Deduction costs each time a radio button is clicked.

HPCosts = 0

If rdoHPOfNevada.Checked Then

HPCosts = 62.5

End If

If rdoKaiserPermanente.Checked Then

HPCosts = 35.6

End If

If rdoBlueCross.Checked Then

HPCosts = 75.25

End If

If rdoHomeTownHealth.Checked Then

HPCosts = 58.85

End If

End Sub

Private Sub AdditionalDeductions(sender As Object, e As EventArgs) Handles chkUnionDues.CheckedChanged, chkUnitedWay.CheckedChanged, chkParking.CheckedChanged, chkRetirement.CheckedChanged

'Recalculate the Additional Deductions costs each time a checkbox is clicked.

ADCosts = 0

If chkUnionDues.Checked Then

ADCosts += 15.5

End If

If chkUnitedWay.Checked Then

ADCosts += 25.5

End If

If chkParking.Checked Then

ADCosts += 20.0

End If

If chkRetirement.Checked Then

ADCosts += 80.0

End If

If chkUnionDues.Checked And chkUnitedWay.Checked And chkParking.Checked And chkRetirement.Checked Then ADCosts += -40

End Sub

H. Returning to the Main Form

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

'Return to Main Form

Me.Close()

End Sub

End Class

K. Document Global Variables

Module modMain

'Declare global variables.

Public Password, FullName, SSN As String

Public Attempts, Hours As Integer

Public Rate, Gross, HPCosts, ADCosts, NetPay As Decimal

End Module