

'Solution name: MesaPayroll  
'Solution folder name: MesaPayroll  
'Project name: MesaPayroll  
'Project folder name: MesaPayroll  
'Programmer name: Kimberly Beswick  
'Date created: 11/2/2013  
'Project description: Calculates the gross and net salary of Mesa Employees and deducts for medical insurance, 401K plans, and misc. medical extras.  
', Sends calculated totals to be displayed on a listbox. Includes a feature which displays a total gross payment for all listed  
', employees and displays the calculated average salary.

Option Strict On

Public Class MesaPayroll

'Declares module-level constants and variables.

Public Const AlliedRate As Decimal = 21.5D  
Public Const BlueCrossRate As Decimal = 12.25D  
Public Const KaiserRate As Decimal = 17.5D  
Public Const ScrippsRate As Decimal = 23.75D  
Public Const FedTaxRate As Decimal = 0.15D

```

Public Const StateTaxRate As Decimal = 0.04D
Public Const FicaTaxRate As Decimal = 0.025D
Public Const MinimumWage As Decimal = 8.25D
Public Const OnePercent401K As Decimal = 0.01D
Public Const TwoPercent401K As Decimal = 0.02D
Public Const ThreePercent401K As Decimal = 0.03D
Public Const FourPercent401K As Decimal = 0.04D
Public Const FivePercent401K As Decimal = 0.05D
Public Const DentalDeduction As Decimal = 6D
Public Const LifeDeduction As Decimal = 6.5D
Public Const LongTermCareDeduction As Decimal = 10D
Public Const VisionDeduction As Decimal = 4.5D
Public Const PercentOvertimeRate As Decimal = 1.5D
Public strMesaEmployee As String
Public intEmployeeNum As Integer
Public decEmployeeGrossTotal As Decimal
Public dtmListTitleDate As Date

```

```

Private Sub MesaPayroll_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

```

```

    'Prompts the user to enter his/her name.

```

```

    Do While String.IsNullOrEmpty(strMesaEmployee)
        strMesaEmployee = InputBox("Enter the name of the employee preparing payroll",
"Employee")
    Loop
    DisplayTitle()
End Sub

```

```

Private Sub btnPayCalc_Click(sender As Object, e As EventArgs) Handles
btnPayCalc.Click

```

```

    'Calculates the weekly payment as well as amounts owed for insurance and taxes, lastly
displaying net pay. Displays totals on listbox.

```

```

Dim intHoursWorked As Integer = 0
Dim decPayRate As Decimal = 0
Dim decGrossPay As Decimal = 0
Dim decBenefits As Decimal = 0
Dim decStateTax As Decimal = 0

```

```
Dim decFederalTax As Decimal = 0
Dim decFICATax As Decimal = 0
Dim decNetPay As Decimal = 0
Dim intOverTimeHours As Integer = 0
```

```
Try
```

```
'Catches user input for hours worked if not a positive real number.
```

```
intHoursWorked = Integer.Parse(txtNumberOfHours.Text)
Catch exNumHours As FormatException
    MessageBox.Show("Hours worked must be a positive real number.")
    txtNumberOfHours.SelectionStart = 0
    With txtNumberOfHours
        .Focus()
        .SelectAll()
    End With
Return
End Try
```

```
Try
```

```
'Catches user input for payrate if not a positive real number.
```

```
decPayRate = Decimal.Parse(txtPayRate.Text)
Catch exPayRate As FormatException
    MessageBox.Show("Pay rate must be a positive real number.")
    txtPayRate.SelectionStart = 0
    With txtPayRate
        .Focus()
        .SelectAll()
    End With
Return
End Try
```

```
Select Case intHoursWorked
```

```
'Multiplies pay rate by hours worked, and overtime is taken into account as "pay and a
half"
```

```
Case 0 To 40
```

```

        decGrossPay = intHoursWorked * decPayRate
    Case 41 To 60
        intOverTimeHours = intHoursWorked - 40
        intHoursWorked = 40
        decGrossPay = (40 * decPayRate) + (intOverTimeHours * decPayRate *
PercentOverTimeRate)
    Case Else
        MessageBox.Show("Hours worked must be between 0 and 60.")
        With txtNumberOfHours
            .Focus()
            .SelectAll()
        End With
        Return
    End Select

```

If (decPayRate < 8.25) Then

'If payrate is under 8.25, displays error message.

```

    MessageBox.Show("Pay rate must equal or exceed $8.25.")
    With txtPayRate
        .Focus()
        .SelectAll()
    End With
    Return
End If

```

```

decEmployeeGrossTotal += decGrossPay
intEmployeeNum += 1

```

'Adds individual's gross pay to total gross pay for all employees. Adds one to number of employees.

```

decBenefits = MedicalChoice() + (Percentage401KCost() * decGrossPay)
decStateTax = decGrossPay * StateTaxRate
decFederalTax = decGrossPay * FedTaxRate
decFICATax = decGrossPay * FicaTaxRate
decNetPay = decGrossPay - decBenefits - decStateTax - decFederalTax - decFICATax -
MiscDeductable()

```

```

lstPayroll.Items.Add("      Employee Number " & intEmployeeNum)
lstPayroll.Items.Add("Gross pay:      " & decGrossPay.ToString("c"))

```

```

lstPayroll.Items.Add("Benefits: ..... " & decBenefits.ToString("c"))
lstPayroll.Items.Add("State Tax: ..... " & decStateTax.ToString("c"))
lstPayroll.Items.Add("Federal tax: ..... " & decFederalTax.ToString("c"))
lstPayroll.Items.Add("FICA tax: ..... " & decFICATax.ToString("c"))
lstPayroll.Items.Add("Net pay: ..... " & decNetPay.ToString("c"))

```

```

lstPayroll.Items.Add("-----
-----")

```

'Sends calculated totals to the listbox.

End Sub

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

'Clears all the employee information on the form, except on the listbox.

```

MessageBox.Show("Are you sure?", "A yes response resets the employee information.",
MessageBoxButtons.YesNo, MessageBoxIcon.Question)
ResetMedicalType()
ResetPercentMedical()
ResetMiscellaneous()
ResetHoursAndPayrate()
btnPayCalc.Focus()

```

End Sub

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

'Displays the gross employee wages for all employees entered, as well as the average wage.

```

MessageBox.Show("Total Gross Pay = " & decEmployeeGrossTotal.ToString("c") & ",
with an Average Gross Pay = " & (decEmployeeGrossTotal / intEmployeeNum).ToString("c"))

```

End Sub

Function MedicalChoice() As Decimal

'Specifies and returns which of the medical insurance options was selected.

```

Dim decMedChoice As Decimal
If radAlliedMedical.Checked Then
decMedChoice = AlliedRate
ElseIf radBlueCrossMedical.Checked Then
decMedChoice = BlueCrossRate
ElseIf radKaiserMedical.Checked Then
decMedChoice = KaiserRate
ElseIf radScrippsMedical.Checked Then
decMedChoice = ScrippsRate
ElseIf radNoneMedical.Checked Then
decMedChoice = 0
End If
Return decMedChoice
End Function

```

Public Function Percentage401KCost() As Decimal

'Specifies and returns which of the 401K plans was selected.

```

Dim dec401KChoice As Decimal
If radOnePercent401K.Checked Then
dec401KChoice = OnePercent401K
ElseIf radTwoPercent401K.Checked Then
dec401KChoice = TwoPercent401K
ElseIf radThreePercent401K.Checked Then
dec401KChoice = ThreePercent401K
ElseIf radFourPercent401K.Checked Then
dec401KChoice = FourPercent401K
ElseIf radFivePercent401K.Checked Then
dec401KChoice = FivePercent401K
End If
Return dec401KChoice
End Function

```

Function MiscDeductable() As Decimal

'Specifies and returns the total of all miscellaneous insurance add-ons.

```

Dim totalDeductables As Decimal
totalDeductables = 0
If chkDental.Checked Then
totalDeductables += DentalDeduction

```

```
End If
If chkLifeInsurance.Checked Then
totalDeductables += LifeDeduction
End If
If chkLongTermCare.Checked Then
totalDeductables += LongTermCareDeduction
End If
If chkVision.Checked Then
totalDeductables += VisionDeduction
End If
Return totalDeductables
End Function
```

```
Sub ResetMedicalType()
```

```
' This procedure resets the medical type selection.
```

```
radAlliedMedical.Checked = True
radBlueCrossMedical.Checked = False
radKaiserMedical.Checked = False
radScrippsMedical.Checked = False
radNoneMedical.Checked = False
End Sub
```

```
Sub ResetPercentMedical()
```

```
' This procedure resets the percent selection.
```

```
radOnePercent401K.Checked = True
radTwoPercent401K.Checked = False
radThreePercent401K.Checked = False
radFourPercent401K.Checked = False
radFivePercent401K.Checked = False
```

```
End Sub
```

```
Sub ResetMiscellaneous()
```

```
' This procedure resets the miscellaneous selection.
```

```
chkDental.Checked = False
chkLifeInsurance.Checked = False
chkLongTermCare.Checked = False
```

```

chkVision.Checked = False

End Sub

Sub ResetHoursAndPayrate()

' This procedure resets the price.

txtNumberOfHours.Text = String.Empty
txtPayRate.Text = String.Empty
End Sub

Public Sub DisplayTitle()

'This procedure displays the title of the listbox.

dtmListTitleDate = Now
lstPayroll.Items.Add("Payroll Report - prepared " & dtmListTitleDate & " by " &
strMesaEmployee)

lstPayroll.Items.Add("-----
-----")

End Sub

Private Sub Tabstop_Click(sender As Object, e As EventArgs) Handles Tabstop.Click

'Closes the form.

MessageBox.Show("Are you sure?", "A yes response ends the program",
MessageBoxButtons.YesNo, MessageBoxIcon.Question)
Me.Close()

End Sub
End Class

```