

## Payroll Application [Element 3]

Payroll Application [ACME Corporation]

Enter Employee Name:  Add  Remove

Select Day Worked:

Enter Rate of Pay:

Enter Number of Hours:

Add Pay to Continue

Calculations for day:

Overtime Hours:

Overtime Rate:

Basic Pay:

Overtime Pay:

Gross Pay:

Taxes:

Net Pay:

Total Calculations:

Basic Pay:

Overtime Pay:

Gross Pay:

Taxes:

Net Pay:

Emergency Taxes is 40%

Exit

### Brief Description

This is beginning to be a bit complicated. So I would like to start by defying the variables that I have declared at the beginning of the program:

First of all I declared Global Variables to index the Employees and the Dates lists. Then, I added the Global Variables that are focused only on the calculations stored in memory.

The first subroutine of the program ensures that the values in the memory are set to zero.

The second and third subroutine add employee's name to the list and validates that a blank name cannot be entered. Then, it indexes the employee value on memory and display the date entered in the list box. To finalises this block on code, I added a 'remove' button that ensure an employee name have to be selected before removal.

On the subroutine that follows I indexed the dates of the week on a table.

What follows next are the different calculations that the program performs to calculate the Basic Pay, Overtime and different payments, with the correct validation of data as necessary.

I have included as well lots of 'green' comments that help me understand what I am doing as I read through the program's code.

## Programming listing of Payroll Application [Element 3]

```
Public Class frmMain
    'Declare global variables needed for the program
    Dim intIndexEmployee, intIndexDate As Integer

    'Declar global variables needed for the total calculations stored in Memory
    Dim intMemBasicPay, intMemOverTimePay, intMemGrossPay, intMemNetPay As Integer
    Dim sngMemTaxesPay As Single

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Set to zero the value of the global variables stored in Memory
        intMemBasicPay = 0
        intMemOverTimePay = 0
        intMemGrossPay = 0
        sngMemTaxesPay = 0
        intMemNetPay = 0
    End Sub

    Private Sub btnAddEmployee_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddEmployee.Click
        'Add and validates that blank employee cannot be entered
        If txtEmployee.Text = "" Then
            MsgBox("Please enter the name of the employee")
            txtEmployee.Focus()
        Else
            lstEmployees.Items.Add(txtEmployee.Text)
            txtEmployee.Clear()
        End If
    End Sub

    Private Sub lstEmployees_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles lstEmployees.SelectedIndexChanged
        Dim strEmployee As String
        'Display values on the employee list
        intIndexEmployee = lstEmployees.SelectedIndex
        If intIndexEmployee > -1 Then
            strEmployee = lstEmployees.Items.Item(intIndexEmployee)
            txtEmployee.Text = strEmployee
        End If
        txtEmployee.Clear()
    End Sub

    Private Sub btnRemove_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRemove.Click
        'Remove and validate that employee is selected
        With lstEmployees
            If .SelectedIndex < 0 Then
                MsgBox("Please select employee to remove")
            End If
            If .SelectedIndex >= 0 Then
                If MsgBox("Are you sure you want to delete this employee?",
MsgBoxStyle.OkCancel, "Yes") = MsgBoxResult.Ok Then
                    .Items.RemoveAt(lstEmployees.SelectedIndex)
                End If
            End If
        End With
    End Sub

    Private Sub lstDates_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles lstDates.SelectedIndexChanged
        Dim strDate As String

        intIndexDate = lstDates.SelectedIndex
        If intIndexDate >= 0 Then
            strDate = lstDates.Items.Item(intIndexDate)
            txtDate.Text = strDate
        End If
    End Sub

    Private Sub btnNetPay_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNetPay.Click
        Dim sngRatePay, sngOverTimeRate, sngTaxesPay As Single
    End Sub
End Class
```

```

    Dim intNoHours, intOverTimeHours, intBasicPay, intOverTimePay, intGrossPay,
intNetPay As Integer

    Dim intTotalBasicPay, intTotalOverTimePay, intTotalGrossPay, intTotalNetPay As
Integer
    Dim sngTotalTaxesPay As Single

    'Start validation of input data
    If lstEmployees.SelectedItem = "" Then
        MsgBox("Please enter an Employee and select it from the list to continue")
        lstEmployees.Focus()
    ElseIf lstDates.SelectedItem = "" Then
        MsgBox("Please select a Date")
        txtDate.Focus()
    ElseIf IsNumeric(txtRateOfPay.Text) = False Then
        MsgBox("Please input only numbers in the Rate of Pay field")
        txtRateOfPay.Focus()
    ElseIf IsNumeric(txtNoHours.Text) = False Then
        MsgBox("Please input only numbers in the Number of Hours field")
        txtNoHours.Focus()
    ElseIf Fix(CSng(txtNoHours.Text)) <> (CSng(txtNoHours.Text)) Then
        MsgBox("Please DO NOT put decimals in the Number of Hours field")
        txtNoHours.SelectAll()
        txtNoHours.Focus()

    Else
        'Start to display input values
        sngRatePay = CSng(txtRateOfPay.Text)
        intNoHours = CInt(txtNoHours.Text)

        'Start validating overtime and hours worked
        If intNoHours <= 8 Then
            intOverTimeHours = 0
            sngOverTimeRate = 0
        ElseIf intNoHours > 16 Then
            MsgBox("It is illegal to work more than 16 hours a day. Please check
the hours and try again")
            intNoHours = 0
            sngOverTimeRate = 0
        Else
            intOverTimeHours = intNoHours - 8
            sngOverTimeRate = sngRatePay * 1.5
        End If

        'Start calculating Basic Pay
        intBasicPay = intNoHours * sngRatePay

        'Start calculating Overtime
        intOverTimePay = sngOverTimeRate * intOverTimeHours

        'Start calculating Gross Pay, Taxes and Net Pay
        intGrossPay = intBasicPay + intOverTimePay
        sngTaxesPay = intGrossPay * 0.4
        intNetPay = intGrossPay - sngTaxesPay

        'Display overtime calculations
        txtOverTimeHours.Text = CStr(intOverTimeHours)
        txtOverTimeRate.Text = CStr(sngOverTimeRate)

        'Display the payments calculations
        txtBasicPay.Text = CStr(intBasicPay)
        txtOverTimePay.Text = CStr(intOverTimePay)
        txtGrossPay.Text = CStr(intGrossPay)
        txtTaxesPay.Text = CStr(sngTaxesPay)
        txtNetPay.Text = CStr(intNetPay)

        'Clear the input fields and focus on Rate of Pay
        txtRateOfPay.Clear()
        txtNoHours.Clear()
        txtRateOfPay.Focus()

        'Start calculating the Totals by adding the memory values
        intTotalBasicPay = CInt(intBasicPay)
        intMemBasicPay = intMemBasicPay + intTotalBasicPay
        intTotalOverTimePay = CInt(intOverTimePay)
        intMemOverTimePay = intMemOverTimePay + intTotalOverTimePay
        intTotalGrossPay = CInt(intGrossPay)

```

```

intMemGrossPay = intMemGrossPay + intTotalGrossPay
sngTotalTaxesPay = CSng(sngTaxesPay)
sngMemTaxesPay = sngMemTaxesPay + sngTotalTaxesPay
intTotalNetPay = CInt(intNetPay)
intMemNetPay = intMemNetPay + intTotalNetPay

'Display the Totals values
txtTotalBasicPay.Text = CStr(intMemBasicPay)
txtTotalOverTimePay.Text = CStr(intMemOverTimePay)
txtTotalGrossPay.Text = CStr(intMemGrossPay)
txtTotalTaxesPay.Text = CStr(sngMemTaxesPay)
txtTotalNetPay.Text = CStr(intMemNetPay)
End If

End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
Dim msg As String
Dim style As MsgBoxStyle
Dim response As MsgBoxResult
msg = "Are you sure you want to to EXIT the Application?"
style = MsgBoxStyle.YesNo
response = MsgBox(msg, style)
If response = MsgBoxResult.Yes Then
Me.Close() 'Close the program if YES
Else
'Do nothing if NO
End If
End Sub

End Class

```