10:

Public Class frm10

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

Dim oldBalance, charges, credits, newBalance, minPayment As Double

InputData(oldBalance, charges, credits)

CalculateNewValues(oldBalance, charges, credits, newBalance, minPayment)

DisplayData(newBalance, minPayment)

End Sub

Sub InputData(ByRef oldBalance As Decimal, ByRef charges As Decimal, ByRef credits As Decimal)

oldBalance = txtOldBalance.Text

charges = txtCharges.Text

credits = txtCredits.Text

End Sub

Sub CalculateNewValues(ByVal oldBalance As Decimal, ByVal charges As Decimal, ByVal credits As Decimal, ByRef newBalance As Decimal, ByRef minPayment As Decimal)

newBalance = oldBalance + charges - credits + (oldBalance \* 0.015)

If newBalance < 20 Then minPayment = newBalance Else minPayment = 20 + ((newBalance - 20) \* 0.1)

End Sub

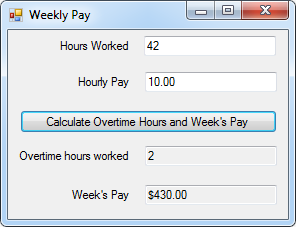
Sub DisplayData(ByVal newBalance As Decimal, ByVal minPayment As Decimal)

txtNewBalance.Text = FormatCurrency(newBalance)

txtMinPayment.Text = FormatCurrency(minPayment)

End Sub

End Class

12:

Public Class frm12

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

Dim hours, payPerHour, overtimeHours, pay As Double

InputData(hours, payPerHour)

CalculateValues(hours, payPerHour, overtimeHours, pay)

DisplayData(overtimeHours, pay)

End Sub

Sub InputData(ByRef hours As Decimal, ByRef payPerHour As Decimal)

hours = txtHours.Text

payPerHour = txtPayPerHour.Text

End Sub

Sub CalculateValues(ByVal hours As Decimal, ByVal payPerHour As Decimal, ByRef overtimeHours As Decimal, ByRef pay As Decimal)

If hours > 40 Then overtimeHours = hours - 40 Else overtimeHours = 0

If overtimeHours = 0 Then pay = hours \* payPerHour Else pay = (40 \* payPerHour) + (overtimeHours \* (payPerHour \* 1.5))

End Sub

Sub DisplayData(ByVal overtimeHours As Decimal, ByVal pay As Decimal)

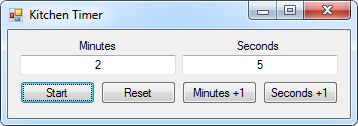
txtOvertimeHours.Text = overtimeHours

txtPay.Text = FormatCurrency(pay)

End Sub

End Class

Kitchen Timer:



Public Class frmTimer

Dim Minutes, Seconds As Integer

Sub SyncTime(ByVal Minutes, ByVal Seconds)

txtMinutes.Text = Minutes

txtSeconds.Text = Seconds

End Sub

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

SyncTime(Minutes, Seconds)

End Sub

Private Sub Timer\_Tick(sender As Object, e As EventArgs) Handles Timer.Tick

If Seconds <= 0 And Minutes <= 0 Then End

If Seconds <= 0 And Minutes > 0 Then Minutes += -1 : Seconds = 60

If Seconds > 60 Then Minutes += Int(Seconds / 60) : Seconds = Seconds - (60 \* Int(Seconds / 60))

Seconds += -1

SyncTime(Minutes, Seconds)

End Sub

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

Timer.Enabled = True

End Sub

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

Timer.Enabled = False

Minutes = 0 : Seconds = 0

SyncTime(Minutes, Seconds)

End Sub

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

If txtMinutes.Text = "" Then Minutes = 1 Else Minutes += 1

SyncTime(Minutes, Seconds)

End Sub

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

If txtSeconds.Text = "" Then Seconds = 1 Else Seconds += 1

If Seconds > 59 Then Minutes += 1 : Seconds = 0

SyncTime(Minutes, Seconds)

End Sub

Private Sub txtMinutes\_txtSeconds\_Enter(sender As Object, e As EventArgs) Handles txtMinutes.Enter, txtSeconds.Enter

Timer.Enabled = False

End Sub

Private Sub txtMinutes\_txtSeconds\_Leave(sender As Object, e As EventArgs) Handles txtMinutes.Leave, txtSeconds.Leave

If txtMinutes.Text = "" Then Minutes = 0 Else Minutes = txtMinutes.Text

If txtSeconds.Text = "" Then Seconds = 0 Else Seconds = txtSeconds.Text

SyncTime(Minutes, Seconds)

Timer.Enabled = True

End Sub

End Class