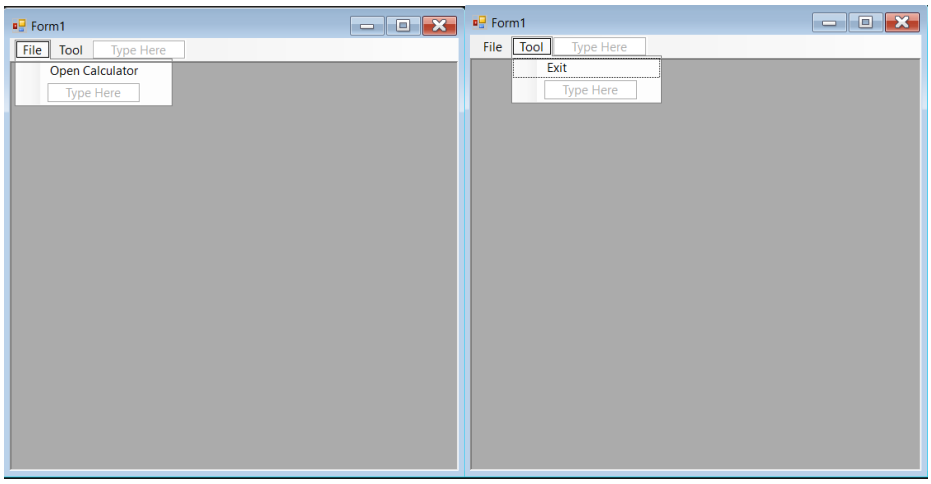
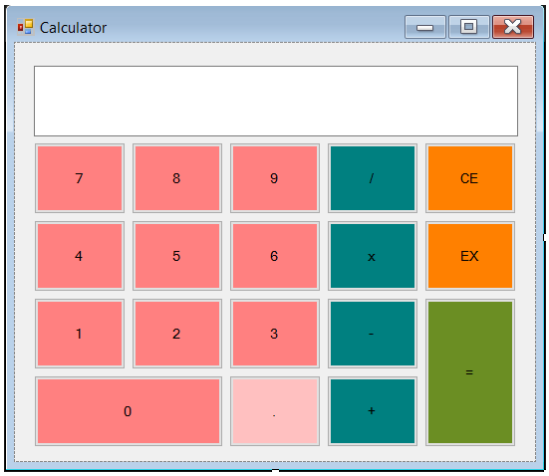
LAB 1




```
Public Class Form1
    Private Sub OpenCalculatorToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles OpenCalculatorToolStripMenuItem.Click
        Calculator.MdiParent = Me
        Calculator.Show()
    End Sub

    Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExitToolStripMenuItem.Click
        Me.Close()
    End Sub
End Class
```

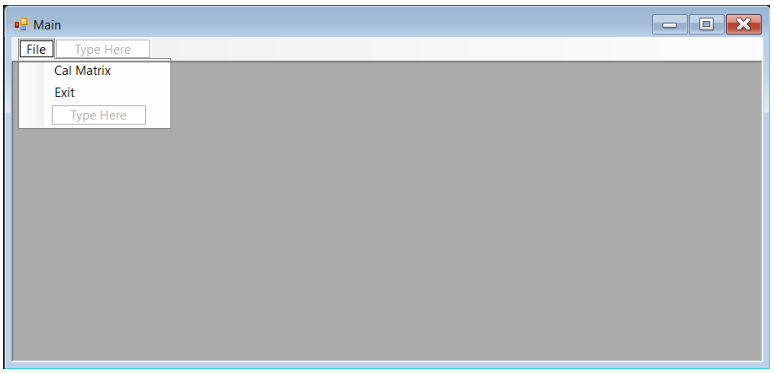



```
Public Class Calculator
    Dim strOpertors As String = ""
    Dim intFirstNumber As Double
    Dim intLastNumber As Double

    Private Sub Btn_num0_Click(sender As Object, e As EventArgs) Handles Btn_num0.Click
        screen.Text += "0"
    End Sub
```

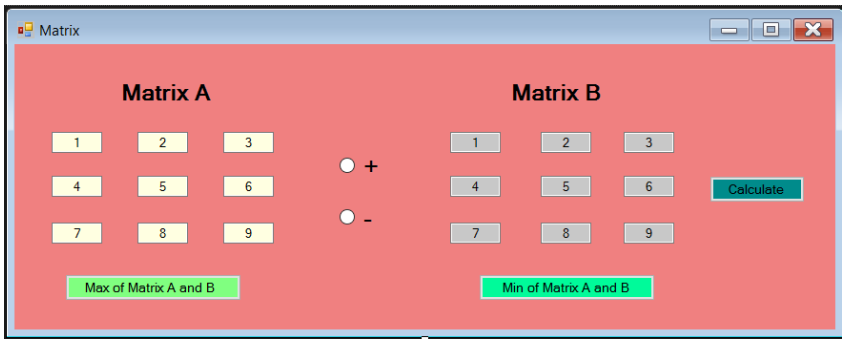
```
    Private Sub Btn_num1_Click(sender As Object, e As EventArgs) Handles Btn_num1.Click
        screen.Text += "1"
    End Sub

    Private Sub Btn_num2_Click(sender As Object, e As EventArgs) Handles Btn_num2.Click
        screen.Text += "2"
    End Sub

    Private Sub Btn_num3_Click(sender As Object, e As EventArgs) Handles Btn_num3.Click
        screen.Text += "3"
    End Sub

    Private Sub Btn_num4_Click(sender As Object, e As EventArgs) Handles Btn_num4.Click
        screen.Text += "4"
    End Sub

    Private Sub Btn_num5_Click(sender As Object, e As EventArgs) Handles Btn_num5.Click
        screen.Text += "5"
    End Sub

    Private Sub Btn_num6_Click(sender As Object, e As EventArgs) Handles Btn_num6.Click
        screen.Text += "6"
    End Sub

    Private Sub Btn_num7_Click(sender As Object, e As EventArgs) Handles Btn_num7.Click
        screen.Text += "7"
    End Sub

    Private Sub Btn_num8_Click(sender As Object, e As EventArgs) Handles Btn_num8.Click
        screen.Text += "8"
    End Sub

    Private Sub Btn_num9_Click(sender As Object, e As EventArgs) Handles Btn_num9.Click
        screen.Text += "9"
    End Sub

    Private Sub Btn_adds_Click(sender As Object, e As EventArgs) Handles Btn_adds.Click
        intFirstNumber = CDbl(screen.Text)
        strOpertors = "+"
        screen.Text = ""
    End Sub

    Private Sub Btn_subtracts_Click(sender As Object, e As EventArgs) Handles
Btn_subtracts.Click
        intFirstNumber = CDbl(screen.Text)
        strOpertors = "-"
        screen.Text = ""
    End Sub

    Private Sub Btn_multiplys_Click(sender As Object, e As EventArgs) Handles
Btn_multiplys.Click
        intFirstNumber = CDbl(screen.Text)
        strOpertors = "x"
        screen.Text = ""
    End Sub

    Private Sub Btn_divides_Click(sender As Object, e As EventArgs) Handles
Btn_divides.Click
```

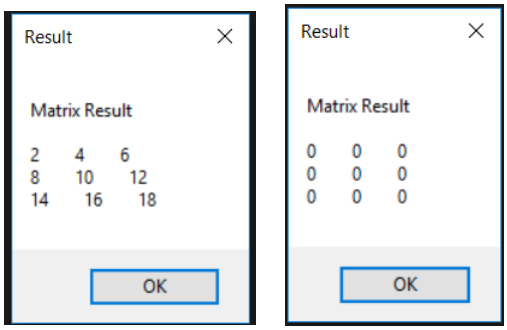
```
        intFirstNumber = CDbl(screen.Text)
        strOpertors = "/"
        screen.Text = ""
    End Sub

    Private Sub Btn_sumall_Click(sender As Object, e As EventArgs) Handles
Btn_sumall.Click
        intLastNumber = screen.Text
        If strOpertors = "+" Then
            screen.Text = intFirstNumber + intLastNumber
        ElseIf strOpertors = "-" Then
            screen.Text = intFirstNumber - intLastNumber
        ElseIf strOpertors = "x" Then
            screen.Text = intFirstNumber * intLastNumber
        Else : strOpertors = "/"
            screen.Text = intFirstNumber / intLastNumber
        End If
    End Sub

    Private Sub Btn_clearall_Click(sender As Object, e As EventArgs) Handles
Btn_clearall.Click
        screen.Text = ""
    End Sub

    Private Sub Btn_addpoint_Click(sender As Object, e As EventArgs) Handles
Btn_addpoint.Click
        If screen.Text Like "*.*" Then
            screen.Text += ""
        Else : screen.Text += "."
        End If
    End Sub

    Private Sub Btn_exits_Click(sender As Object, e As EventArgs) Handles Btn_exits.Click
        Me.Close()
    End Sub

End Class
```

LAB 2




```
Public Class Main

    Private Sub CalMatrixToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles CalMatrixToolStripMenuItem.Click
        MatrixForm.MdiParent = Me
        MatrixForm.Show()
    End Sub

    Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExitToolStripMenuItem.Click
        Me.Close()
    End Sub
End Class
```



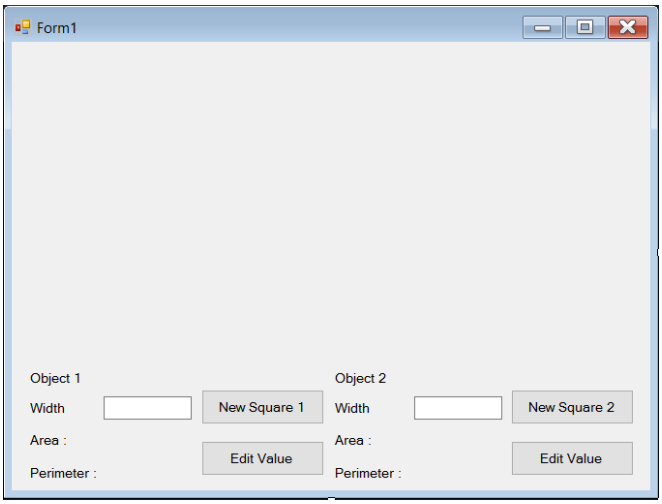


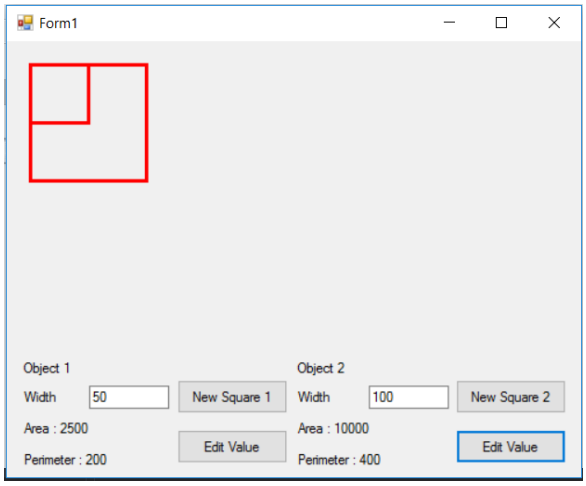

```
Public Class MatrixForm
    Dim Array_A(2, 2), Array_B(2, 2) As Integer
    Sub inputArrayA()
        Array_A = {{TxtB_A1.Text, TxtB_A2.Text, TxtB_A3.Text},
```

```
                {TxtB_A4.Text, TxtB_A5.Text, TxtB_A6.Text},
                {TxtB_A7.Text, TxtB_A8.Text, TxtB_A9.Text}}
    End Sub
    Sub inputArrayB()
        Array_B = {{TxtB_B1.Text, TxtB_B2.Text, TxtB_B3.Text},
                {TxtB_B4.Text, TxtB_B5.Text, TxtB_B6.Text},
                {TxtB_B7.Text, TxtB_B8.Text, TxtB_B9.Text}}
    End Sub
    Function AddMatrix(ByVal A(,) As Integer, ByVal B(,) As Integer)
        Dim sumAB(2, 2) As Integer
        For i = 0 To A.GetLength(1) - 1
            For j = 0 To B.GetLength(1) - 1
                sumAB(i, j) = A(i, j) + B(i, j)
            Next
        Next
        Return sumAB
    End Function
    Function SubtractMatrix(ByVal A(,) As Integer, ByVal B(,) As Integer)
        Dim sumAB(2, 2) As Integer
        For i = 0 To A.GetLength(1) - 1
            For j = 0 To B.GetLength(1) - 1
                sumAB(i, j) = A(i, j) - B(i, j)
            Next
        Next
        Return sumAB
    End Function
    Sub Show_matrix(ByVal result(,) As Integer)
        Dim resultMatrix As String = ""
        resultMatrix &= "Matrix Result" & vbCrLf & vbCrLf
        For i = 0 To result.GetLength(1) - 1
            For j = 0 To result.GetLength(1) - 1
                resultMatrix &= result(i, j) & "        "
            Next
            resultMatrix &= vbCrLf
        Next
        MessageBox.Show(resultMatrix, "Result", MessageBoxButtons.OK)
    End Sub

    Private Sub btn_calculate_Click(sender As Object, e As EventArgs) Handles
btn_calculate.Click
        inputArrayA()
        inputArrayB()
        If RadioBtn_add.Checked Then
            Show_matrix(AddMatrix(Array_A, Array_B))
        ElseIf RadioBtn_sub.Checked Then
            Show_matrix(SubtractMatrix(Array_A, Array_B))
        Else : MessageBox.Show("Please select : + RadioButton OR - RadioButton", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
        End If
    End Sub
    Function MatrixMin(ByVal A(,) As Integer, ByVal B(,) As Integer) As String
        Dim minA = A(0, 0)
        Dim minB = B(0, 0)
        Dim ia = 0
        Dim ja = 0
        Dim ib = 0
        Dim jb = 0
        For i = 0 To A.GetLength(0) - 1
```

```
            For j = 0 To A.GetLength(0) - 1
                If A(i, j) < minA Then
                    minA = A(i, j)
                    ia = i + 1
                    ja = j + 1
                End If
                If B(i, j) < minB Then
                    minB = B(i, j)
                    ib = i + 1
                    jb = j + 1
                End If
            Next
        Next
        Dim str = "Min value of Matrix A: A[" & ia & "," & ja & "] = " & minA
        str += vbCrLf & "Min value of Matrix B: B[" & ib & "," & jb & "] = " & minB
        Return str
    End Function
    Function MatrixMax(ByVal A(,) As Integer, ByVal B(,) As Integer) As String
        Dim maxA = A(0, 0)
        Dim maxB = B(0, 0)
        Dim ia = 0
        Dim ja = 0
        Dim ib = 0
        Dim jb = 0
        For i = 0 To A.GetLength(0) - 1
            For j = 0 To A.GetLength(0) - 1
                If A(i, j) > maxA Then
                    maxA = A(i, j)
                    ia = i + 1
                    ja = j + 1
                End If
                If B(i, j) > maxB Then
                    maxB = B(i, j)
                    ib = i + 1
                    jb = j + 1
                End If
            Next
        Next
        Dim str = "Max value of Matrix A: A[" & ia & "," & ja & "] = " & maxA
        str += vbCrLf & "Max value of Matrix B: B[" & ib & "," & jb & "] = " & maxB
        Return str
    End Function

    Private Sub btn_min_Click(sender As Object, e As EventArgs) Handles btn_min.Click
        inputArrayA()
        inputArrayB()
        MessageBox.Show(MatrixMin(Array_A, Array_B), "Result", MessageBoxButtons.OK)
    End Sub

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

    Private Sub btn_max_Click(sender As Object, e As EventArgs) Handles btn_max.Click
        inputArrayA()
        inputArrayB()
        MessageBox.Show(MatrixMax(Array_A, Array_B), "Result", MessageBoxButtons.OK)
    End Sub
End Class
```

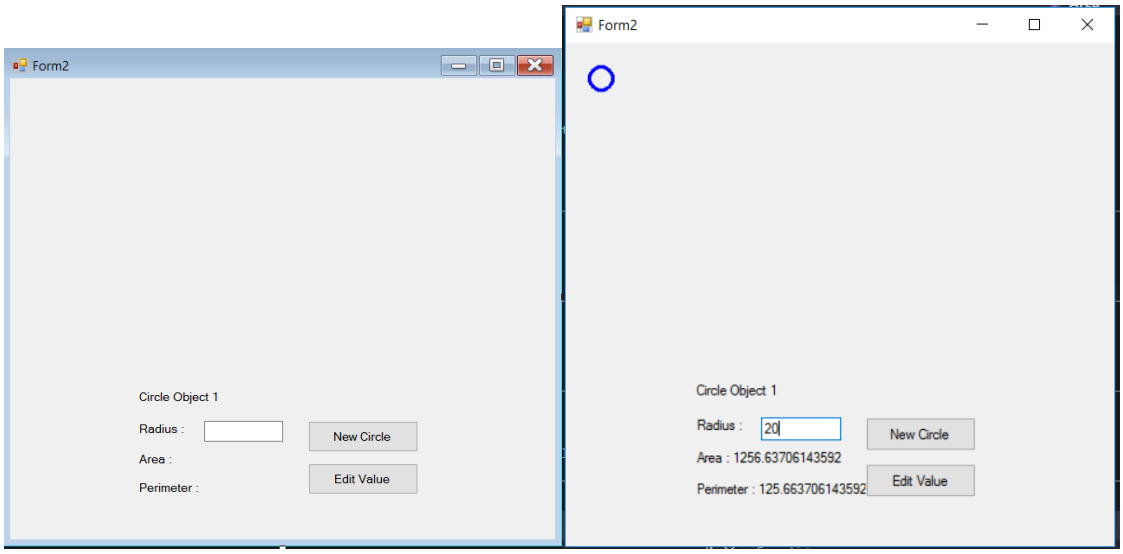
LAB 3






```
Public Class Form1
    Dim MySquare1 As Square_class
    Dim MySquare2 As Square_class

    Private Sub btn_square1_Click(sender As Object, e As EventArgs) Handles
btn_square1.Click
        MySquare1 = New Square_class()
        MySquare1.mywidth() = TxtB_width1.Text
        Lb_square_area1.Text = "Area : " + CStr(MySquare1.Area(MySquare1.mywidth))
        Lb_square_perimeter1.Text = "Perimeter : " + CStr(MySquare1.Perimeter())
        MySquare1.Draw_square()
    End Sub


    Private Sub btn_edit1_Click(sender As Object, e As EventArgs) Handles btn_edit1.Click
        MySquare1.Clear_square()
        MySquare1.mywidth = TxtB_width1.Text
        Lb_square_area1.Text = "Area : " + CStr(MySquare1.Area(MySquare1.mywidth))
        Lb_square_perimeter1.Text = "Perimeter : " + CStr(MySquare1.Perimeter())
        MySquare1.Draw_square()
    End Sub

    Private Sub btn_square2_Click(sender As Object, e As EventArgs) Handles
btn_square2.Click
        MySquare2 = New Square_class()
        MySquare2.mywidth() = TxtB_width2.Text
        Lb_square_area2.Text = "Area : " + CStr(MySquare2.Area(MySquare2.mywidth))
        Lb_square_perimeter2.Text = "Perimeter : " + CStr(MySquare2.Perimeter())
        MySquare2.Draw_square()
    End Sub

    Private Sub btn_edit2_Click(sender As Object, e As EventArgs) Handles btn_edit2.Click
        MySquare2.Clear_square()
        MySquare2.mywidth = TxtB_width2.Text
        Lb_square_area2.Text = "Area : " + CStr(MySquare2.Area(MySquare2.mywidth))
        Lb_square_perimeter2.Text = "Perimeter : " + CStr(MySquare2.Perimeter())
        MySquare2.Draw_square()
    End Sub
End Class
```

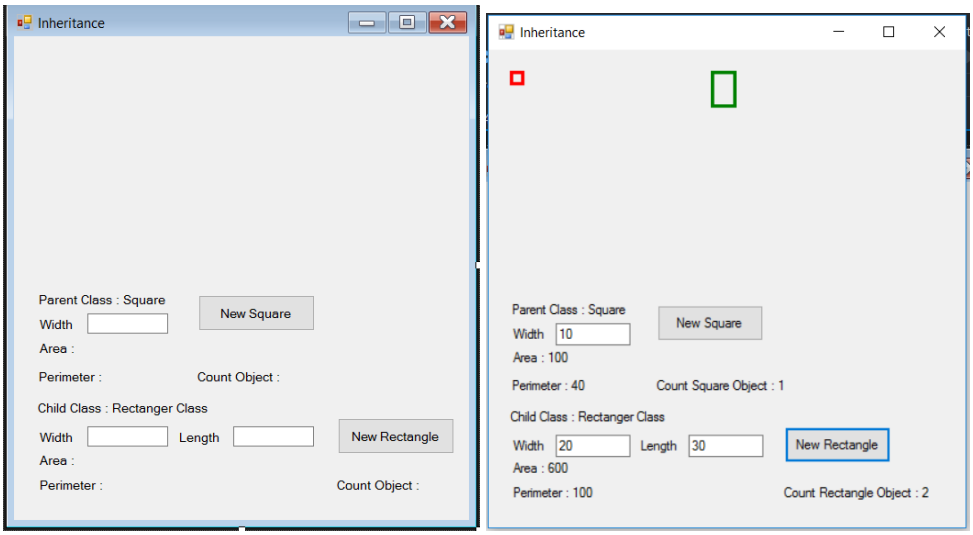
```
Public Class Square_class
    Private width As Double
    Public Property mywidth() As String
        Get
            Return width
        End Get
        Set(ByVal value As String)
            value = value.Trim()
            If value <> String.Empty And IsNumeric(value) Then
                width = value
            Else
                MessageBox.Show("Width Value is 0", "Error !!!", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                width = 0
            End If
        End Set
    End Property
    Public Function Area(ByVal w As Double) As Double
        Dim area_Square As Double
        area_Square = w * w
        Return area_Square
    End Function
    Public Function Perimeter() As Double
        Dim perimeter_Square As Double
        perimeter_Square = (width * 4)
        Return perimeter_Square
    End Function
    Public Sub Draw_square()
        Dim g_Draw As Graphics = Form1.CreateGraphics
        Dim Pen As Pen = New Pen(Color.Red, 3)
        g_Draw.DrawRectangle(Pen, 20, 20, Convert.ToInt32(mywidth),
Convert.ToInt32(mywidth))
    End Sub
    Public Sub Clear_square()
        Dim g_Clear As Graphics = Form1.CreateGraphics
        Dim Pen2 As Pen = New Pen(Color.FromArgb(Form1.BackColor.ToArgb), 3)
        g_Clear.DrawRectangle(Pen2, 20, 20, Convert.ToInt32(mywidth),
Convert.ToInt32(mywidth))
    End Sub
End Class
```




```
Public Class Form2
    Dim MyCircle As Circle_class

    Private Sub btn_circle_Click(sender As Object, e As EventArgs) Handles
btn_circle.Click
        MyCircle = New Circle_class()
        MyCircle.myradius() = TxtB_radius.Text
        Lb_circle_area.Text = "Area : " + CStr(MyCircle.Area(MyCircle.myradius))
        Lb_circle_perimeter.Text = "Perimeter : " + CStr(MyCircle.Perimeter())
        MyCircle.Draw_circle()
    End Sub

    Private Sub btn_edit_Click(sender As Object, e As EventArgs) Handles btn_edit.Click
        MyCircle.Clear_circle()
        MyCircle.myradius = TxtB_radius.Text
        Lb_circle_area.Text = "Area : " + CStr(MyCircle.Area(MyCircle.myradius))
        Lb_circle_perimeter.Text = "Perimeter : " + CStr(MyCircle.Perimeter())
        MyCircle.Draw_circle()
    End Sub
End Class


Public Class Circle_class
    Private radius As Double
    Public Property myradius() As String
        Get
            Return radius
        End Get
        Set(ByVal value As String)
            value = value.Trim()
            If value <> String.Empty And IsNumeric(value) Then
                radius = value
            Else
                MessageBox.Show("Width Value is 0", "Error !!!", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                radius = 0
            End If
```

```
        End Set
    End Property
    Public Function Area(ByVal radius As Double) As Double
        Dim area_Ciecle As Double
        area_Ciecle = Math.PI * Math.Pow(radius, 2)
        Return area_Ciecle
    End Function
    Public Function Perimeter() As Double
        Dim perimeter_Circle As Double
        perimeter_Circle = 2 * Math.PI * radius
        Return perimeter_Circle
    End Function
    Public Sub Draw_circle()
        Dim g_Draw As Graphics = Form2.CreateGraphics
        Dim Pen As Pen = New Pen(Color.Blue, 3)
        g_Draw.DrawEllipse(Pen, 20, 20, Convert.ToInt32(myradius),
Convert.ToInt32(myradius))
    End Sub
    Public Sub Clear_circle()
        Dim g_Clear As Graphics = Form2.CreateGraphics
        Dim Pen2 As Pen = New Pen(Color.FromArgb(Form2.BackColor.ToArgb), 3)
        g_Clear.DrawRectangle(Pen2, 20, 20, Convert.ToInt32(myradius),
Convert.ToInt32(myradius))
    End Sub
End Class
```

LAB 4

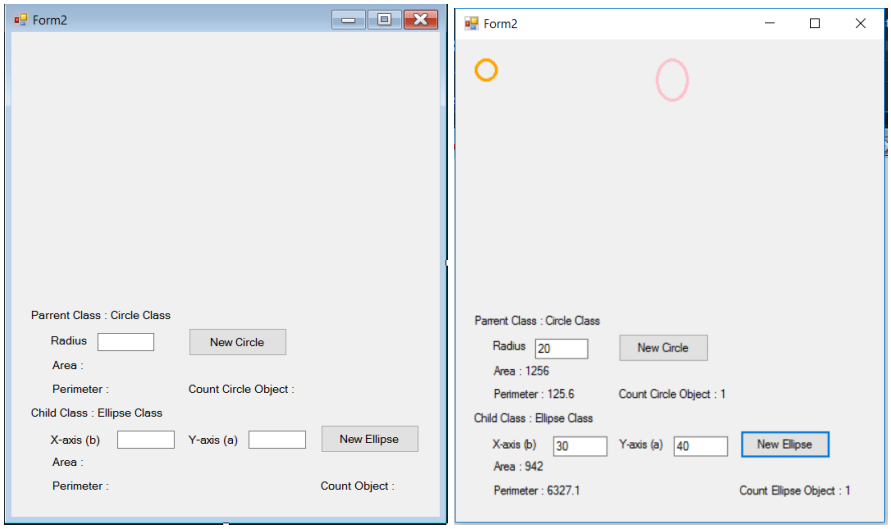



```
Public Class Form1
    Private Sub Btn_square_Click(sender As Object, e As EventArgs) Handles
Btn_square.Click
        Dim MySquare As Square_class
        MySquare = New Square_class()
        MySquare.width_p = TxtB_square_width.Text
        Lb_square_area.Text = "Area : " + CStr(MySquare.Area(MySquare.width_p))
        Lb_square_perimeter.Text = "Perimeter : " + CStr(MySquare.Perimeter())
        MySquare.Draw_square()
        Lb_count_square.Text = "Count Square Object : " + CStr(MySquare.squareCount_p)
    End Sub

    Private Sub Btn_rectangle_Click(sender As Object, e As EventArgs) Handles
Btn_rectangle.Click
        Dim MyRectangle As Rectangle_class
        MyRectangle = New Rectangle_class()
        MyRectangle.width_p = TxtB_rectangle_width.Text
        MyRectangle.length_p = TxtB_rectangle_length.Text
        Lb_rectangle_area.Text = "Area : " + CStr(MyRectangle.Area(MyRectangle.width_p,
MyRectangle.length_p))
        Lb_rectangle_perimeter.Text = "Perimeter : " + CStr(MyRectangle.Perimeter())
        MyRectangle.Draw_Rectangle()
        Lb_count_rectangle.Text = "Count Rectangle Object : " +
CStr(MyRectangle.squareCount_p)
    End Sub
End Class
```

```
Public Class Square_class
    Private width As Double
    Private Shared squareCount As Integer

    Public Property width_p() As String
        Get
            Return width
        End Get
```

```
        Set(value As String)
            value = value.Trim
            If value <> String.Empty And IsNumeric(value) Then
                width = value
            Else
                MessageBox.Show("Width Value is 0", "Error!!!", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                width = 0
            End If
        End Set
    End Property

    Public Property squareCount_p() As Integer
        Get
            Return squareCount
        End Get
        Set(value As Integer)
            squareCount = value
        End Set
    End Property

    Public Overridable Function Area(ByVal w As Double) As Double
        Dim area_square = w * w
        Return area_square
    End Function

    Public Overridable Function Perimeter() As Double
        Dim perimeter_square As Double
        perimeter_square = (4 * width)
        Return perimeter_square
    End Function

    Public Sub Draw_square()
        Dim g_Draw As Graphics = Form1.CreateGraphics
        Dim Pen As Pen = New Pen(Color.Red, 3)
        g_Draw.DrawRectangle(Pen, 20, 20, Convert.ToInt32(width_p),
Convert.ToInt32(width_p))
    End Sub

    Public Sub New()
        squareCount_p += 1
    End Sub
End Class




Public Class Rectangle_class
    Inherits Square_class
    Private length As Double
    Private Shared rectangleCount As Integer
    Public Property length_p() As String
        Get
            Return length
        End Get
        Set(value As String)
            value = value.Trim
            If value <> String.Empty And IsNumeric(value) Then
```

```
                length = value
            Else
                MessageBox.Show("Length Value is 0", "Error!!!", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                length = 0
            End If
        End Set
    End Property

    Public Property rectangleCount_p() As Integer
        Get
            Return rectangleCount
        End Get
        Set(value As Integer)
            rectangleCount = value
        End Set
    End Property
    Public Overloads Function Area(w As Double, ByVal l As Double) As Double
        Dim area_rectangle As Double
        area_rectangle = w * l
        Return area_rectangle
    End Function

    Public Overrides Function Perimeter() As Double
        Dim perimeter_rectangle As Double
        perimeter_rectangle = (2 * width_p) + (2 * length)
        Return perimeter_rectangle
    End Function

    Public Sub Draw_Rectangle()
        Dim g_DrawR As Graphics = Form1.CreateGraphics
        Dim Pen_green = New Pen(Color.Green, 3)
        g_DrawR.DrawRectangle(Pen_green, 200, 20, Convert.ToInt32(width_p),
Convert.ToInt32(length_p))
    End Sub

    Public Sub New()
        rectangleCount += 1
    End Sub
End Class
```

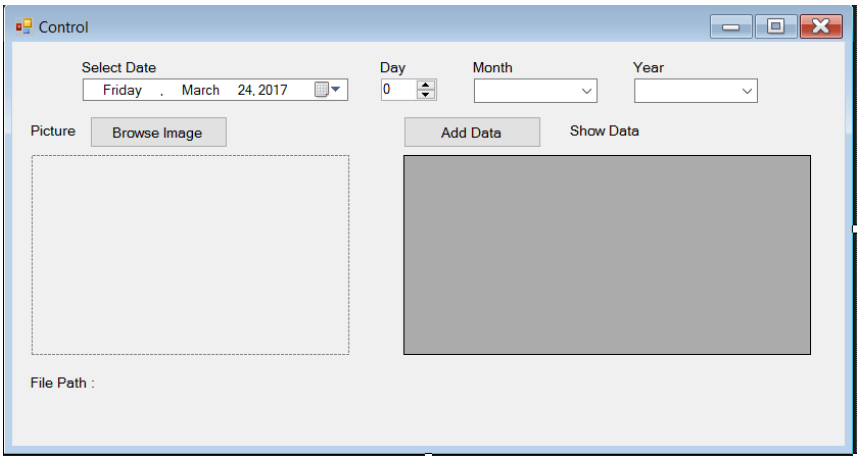


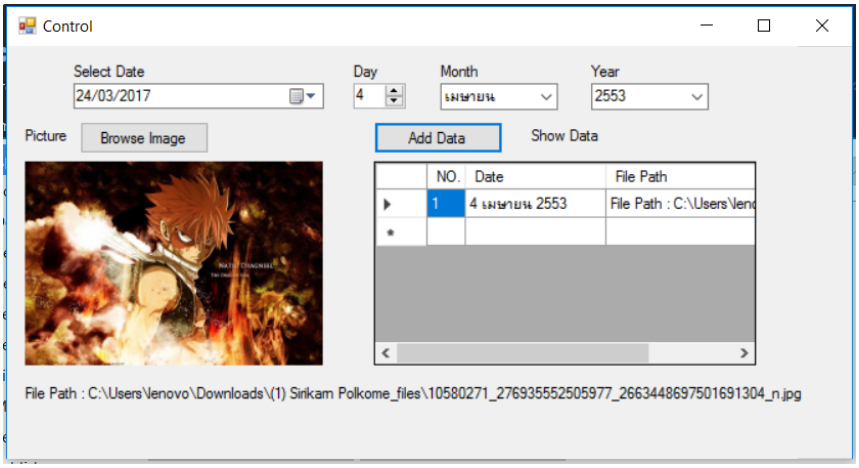

```
Public Class Form2
    Private Sub Btn_circle_Click(sender As Object, e As EventArgs) Handles
Btn_circle.Click
        Dim MyCircle As New Circle()
        MyCircle.radius_p = TxtB_circle_width.Text
        Lb_circle_area.Text = "Area : " + CStr(MyCircle.Area(MyCircle.radius_p))
        Lb_circle_perimeter.Text = "Perimeter : " + CStr(MyCircle.Perimeter())
        MyCircle.Draw_Circle()
        Lb_count_circle.Text = "Count Circle Object : " + CStr(MyCircle.circleCount_p)
    End Sub

    Private Sub Btn_ellipse_Click(sender As Object, e As EventArgs) Handles
Btn_ellipse.Click
        Dim MyEllipse As New Ellipse()
        MyEllipse.xB_p = TxtB_x.Text
        MyEllipse.yA_p = TxtB_y.Text
        Lb_ellipse_area.Text = "Area : " + CStr(MyEllipse.Area(MyEllipse.xB_p,
MyEllipse.yA_p))
        Lb_ellipse_perimeter.Text = "Perimeter : " + CStr(MyEllipse.Perimeter())
        MyEllipse.Draw_Circle()
        Lb_count_ellipse.Text = "Count Ellipse Object : " +
CStr(MyEllipse.EllipseCount_p)
    End Sub
End Class

Public Class Circle
    Private radius As Double
    Private PI As Double = 3.14
    Private Shared circleCount As Integer

    Public Property radius_p() As String
        Get
            Return radius
        End Get
        Set(value As String)
            value = value.Trim
            If value <> String.Empty And IsNumeric(value) Then
                radius = value
            Else
```

```
                MessageBox.Show("Radius Value is 0", "Error!!", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                radius = 0
            End If
        End Set
    End Property

    Public Property circleCount_p() As Integer
        Get
            Return circleCount
        End Get
        Set(value As Integer)
            circleCount = value
        End Set
    End Property

    Public Overridable Function Area(ByVal rad As Double) As Double
        Dim area_circle As Double
        area_circle = PI * rad ^ 2
        Return area_circle
    End Function

    Public Overridable Function Perimeter() As Double
        Dim perimeter_circle As Double
        perimeter_circle = 2 * PI * radius
        Return perimeter_circle
    End Function

    Public Sub Draw_Circle()
        Dim c_Draw As Graphics = Form2.CreateGraphics
        Dim pen As Pen = New Pen(Color.Orange, 3)
        c_Draw.DrawEllipse(pen, 20, 20, Convert.ToInt32(radius_p),
Convert.ToInt32(radius_p))
    End Sub

    Public Sub New()
        circleCount_p += 1
    End Sub
End Class




Public Class Ellipse
    Inherits Circle
    Private xB As Double
    Private yA As Double
    Private PI As Double = 3.14
    Private Shared EllipseCount As Integer

    Public Property xB_p() As String
        Get
            Return xB
        End Get
        Set(value As String)
            value = value.Trim
            If value <> String.Empty And IsNumeric(value) Then
                xB = value
```

```
            Else
                MessageBox.Show("x (b) Value is 0", "Error!!", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                xB = 0
            End If
        End Set
    End Property

    Public Property yA_p() As String
        Get
            Return yA
        End Get
        Set(value As String)
            value = value.Trim
            If value <> String.Empty And IsNumeric(value) Then
                yA = value
            Else
                MessageBox.Show("y (a) Value is 0", "Error!!", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                yA = 0
            End If
        End Set
    End Property

    Public Property EllipseCount_p() As Integer
        Get
            Return EllipseCount
        End Get
        Set(value As Integer)
            EllipseCount = value
        End Set
    End Property

    Public Overloads Function Area(ByVal a As Double, ByVal b As Double) As Double
        Dim area_Ellipse As Double
        area_Ellipse = (PI / 4) * a * b
        Return area_Ellipse
    End Function

    Public Overrides Function Perimeter() As Double
        Dim perimeter_Ellipse As Double
        perimeter_Ellipse = (PI * (yA_p + xB_p)) / 2
        Return perimeter_Ellipse
    End Function

    Public Overloads Sub Draw_Circle()
        Dim c_DrawR As Graphics = Form2.CreateGraphics
        Dim Pen_green As Pen = New Pen(Color.Pink, 3)
        c_DrawR.DrawEllipse(Pen_green, 200, 20, Convert.ToInt32(xB_p),
Convert.ToInt32(yA_p))
    End Sub

    Public Sub New()
        EllipseCount += 1
    End Sub
End Class
```

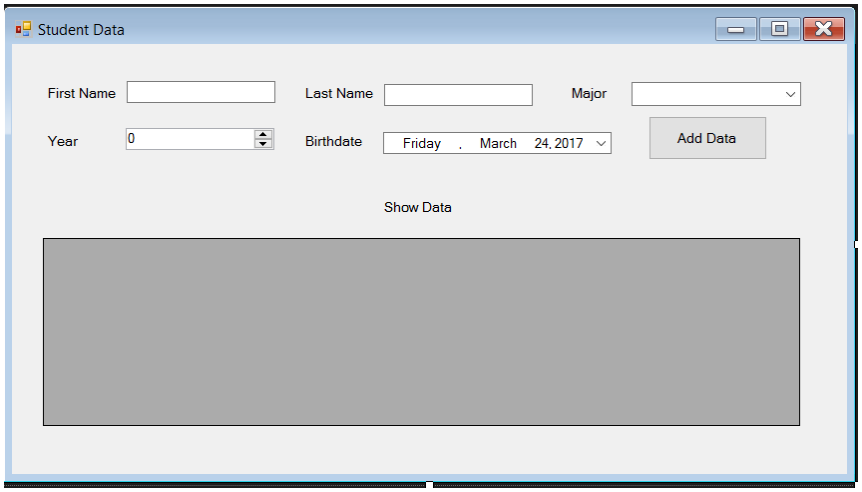
LAB 5

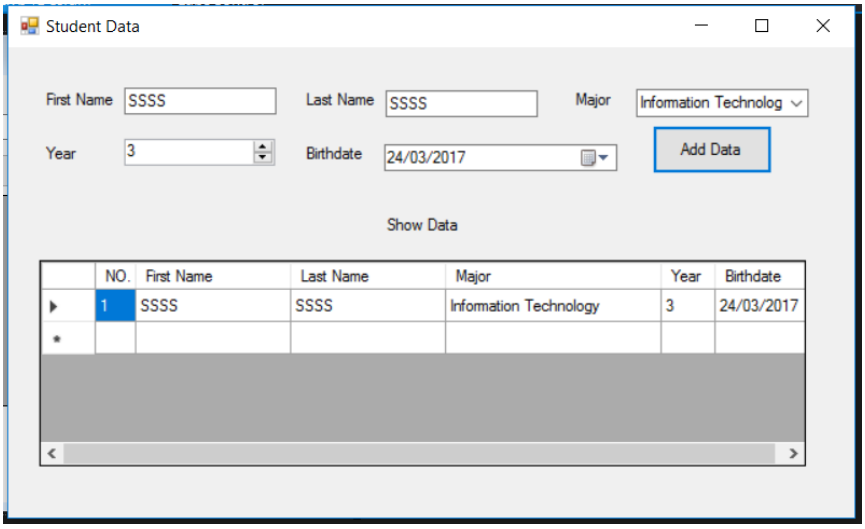





```
Public Class Form1

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        MyDateTimePicker.Format = DateTimePickerFormat.Custom
        MyDateTimePicker.CustomFormat = ("dd/MM/yyyy")

        NumericUpDown_Day.Minimum = 1
        NumericUpDown_Day.Maximum = 31

        Dim localDate = DateTime.Now.Year
        For i = -10 To 10

            With ComboBox_year
                .Items.Add(CStr(localDate + 543 + i))
            End With
        Next

        With ComboBox_month
            .Items.Insert(0, ("มกราคม"))
            .Items.Insert(1, ("กุมภาพันธ์"))
            .Items.Insert(2, ("มีนาคม"))
            .Items.Insert(3, ("เมษายน"))
```

```
            .Items.Insert(4, ("พฤษภาคม"))
            .Items.Insert(5, ("มิถุนายน"))
            .Items.Insert(6, ("กรกฎาคม"))
            .Items.Insert(7, ("สิงหาคม"))
            .Items.Insert(8, ("กันยายน"))
            .Items.Insert(9, ("ตุลาคม"))
            .Items.Insert(10, ("พฤศจิกายน"))
            .Items.Insert(11, ("ธันวาคม"))
        End With

        DataGridView_data.ColumnCount = 3

        DataGridView_data.Columns(0).Width = 30
        DataGridView_data.Columns(1).Width = 110
        DataGridView_data.Columns(2).Width = 200

        DataGridView_data.Columns(0).Name = "NO."
        DataGridView_data.Columns(1).Name = "Date"
        DataGridView_data.Columns(2).Name = "File Path"
    End Sub

    Private Sub MyDateTimePicker_ValueChanged(sender As Object, e As EventArgs) Handles
MyDateTimePicker.ValueChanged
        NumericUpDown_Day.Value = MyDateTimePicker.Value.Day
        ComboBox_month.SelectedIndex() = MyDateTimePicker.Value.Month - 1
        ComboBox_year.Text = CStr(CDbl(MyDateTimePicker.Value.Year) + 543)

    End Sub

    Private Sub Btn_OpenFile_Click(sender As Object, e As EventArgs) Handles
Btn_OpenFile.Click
        Dim img As String = ""
        OpenImageDialog.Filter = "Picture |*.bmp;*.jpg;*.gif| All Files|*.*"
        OpenImageDialog.FileName = ""
        If OpenImageDialog.ShowDialog(Me) = DialogResult.OK Then
            img = OpenImageDialog.FileName
            MyPictureBox.SizeMode = PictureBoxSizeMode.StretchImage
            MyPictureBox.Image = System.Drawing.Bitmap.FromFile(img)
        End If
        Lb_filePath.Text = "File Path : " + img
    End Sub

    Private Sub Btn_AddData_Click(sender As Object, e As EventArgs) Handles
Btn_AddData.Click
        Dim num As Integer
        Dim dateT As String
        num = DataGridView_data.RowCount
        dateT = NumericUpDown_Day.Value.ToString + " " + ComboBox_month.Text + " " +
ComboBox_year.Text
        Dim row As String() = New String() {num, dateT, Lb_filePath.Text}
        DataGridView_data.Rows.Add(row)

    End Sub
End Class
```






```
Public Class Form2
    Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        MyDateTimePicker.Format = DateTimePickerFormat.Custom
        MyDateTimePicker.CustomFormat = ("dd/MM/yyyy")

        NumericUpDown_Year.Minimum = 1
        NumericUpDown_Year.Maximum = 4

        With ComboBox_Major
            .Items.Add("Computer Science")
            .Items.Add("Information Technology")
            .Items.Add("Geographic Information System")
        End With

        DataGridView_data.ColumnCount = 6

        DataGridView_data.Columns(0).Width = 30
        DataGridView_data.Columns(1).Width = 115
        DataGridView_data.Columns(2).Width = 115
        DataGridView_data.Columns(3).Width = 160
        DataGridView_data.Columns(4).Width = 40
        DataGridView_data.Columns(5).Width = 70
```

```
        DataGridView_data.Columns(0).Name = "NO."
        DataGridView_data.Columns(1).Name = "First Name"
        DataGridView_data.Columns(2).Name = "Last Name"
        DataGridView_data.Columns(3).Name = "Major"
        DataGridView_data.Columns(4).Name = "Year"
        DataGridView_data.Columns(5).Name = "Birthdate"
    End Sub

    Private Sub Btn_AddData_Click(sender As Object, e As EventArgs) Handles
Btn_AddData.Click
        Dim num As Integer
        num = DataGridView_data.RowCount
        Dim row As String() = New String() {num, Tbx_Fname.Text, Tbx_Lname.Text,
ComboBox_Major.Text,
            NumericUpDown_Year.Value.ToString, MyDateTimePicker.Text}
        DataGridView_data.Rows.Add(row)
    End Sub
End Class
```