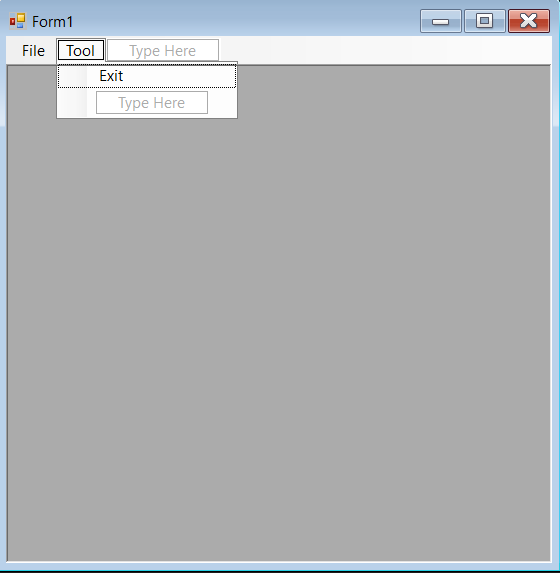
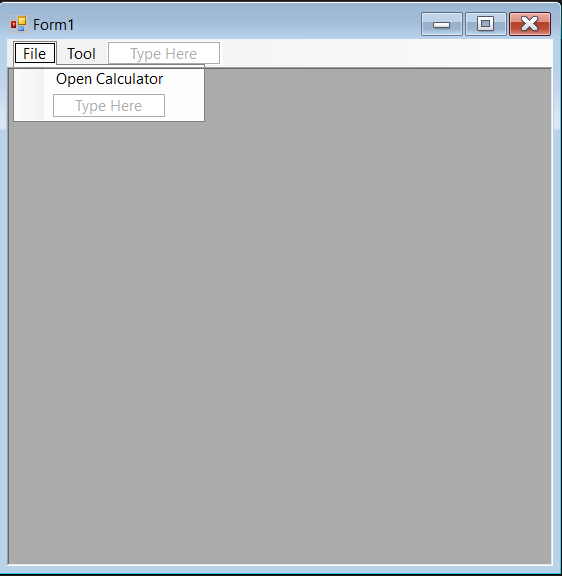
LAB 1



Public Class Form1

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

Calculator.MdiParent = Me

Calculator.Show()

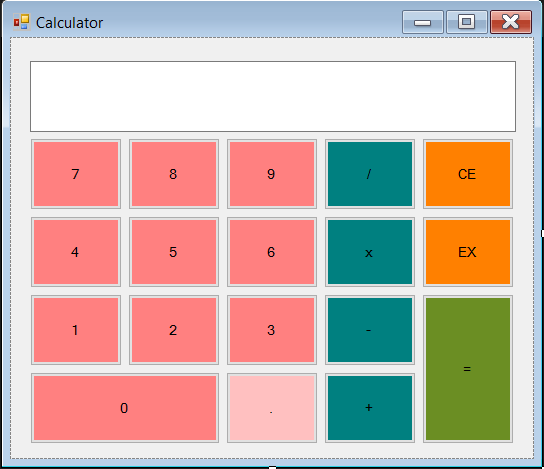
End Sub

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

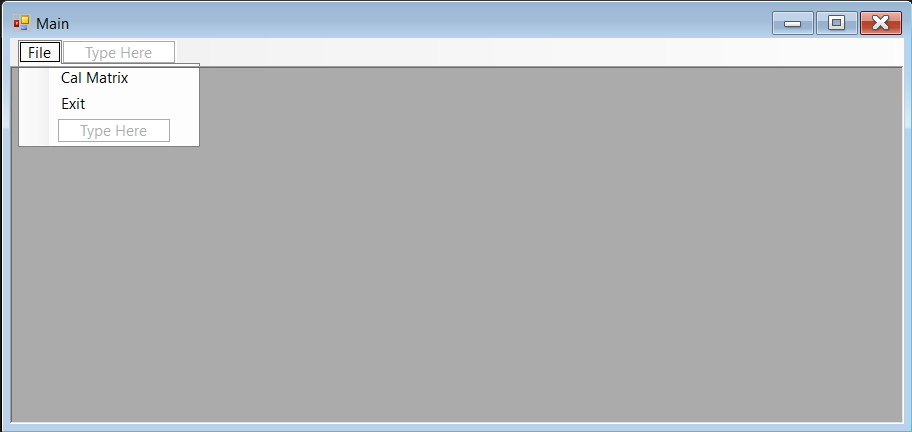
Private Sub Btn\_exits\_Click(sender As Object, e As EventArgs) Handles Btn\_exits.Click

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()

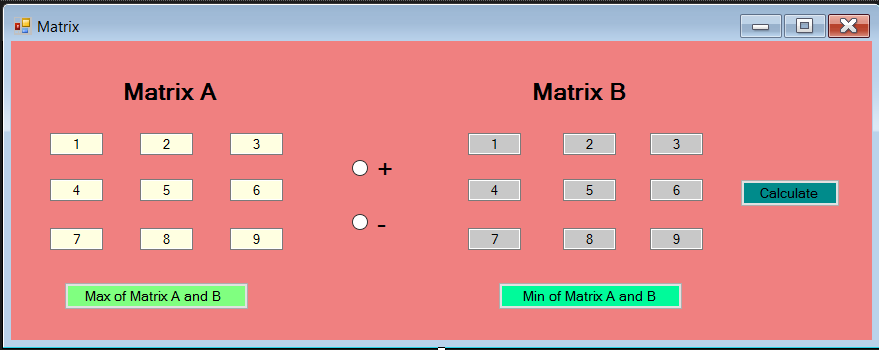
End Sub

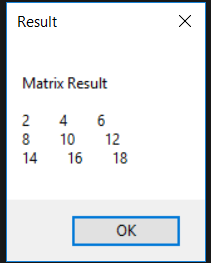
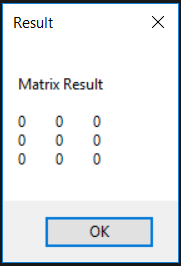
Private Sub ExitToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles ExitToolStripMenuItem.Click

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()

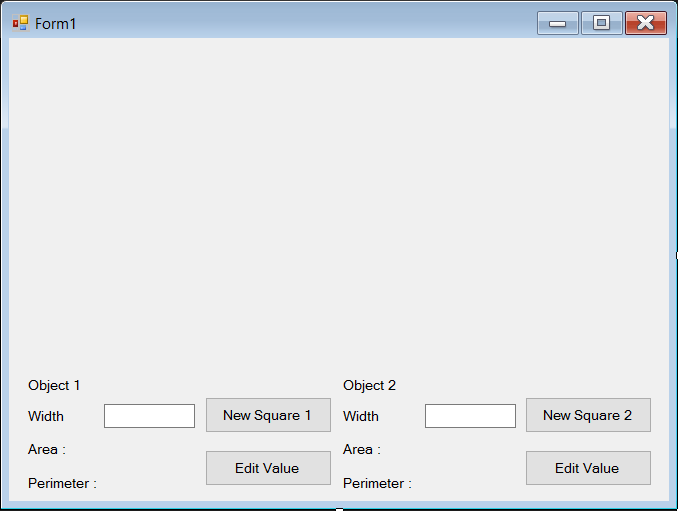
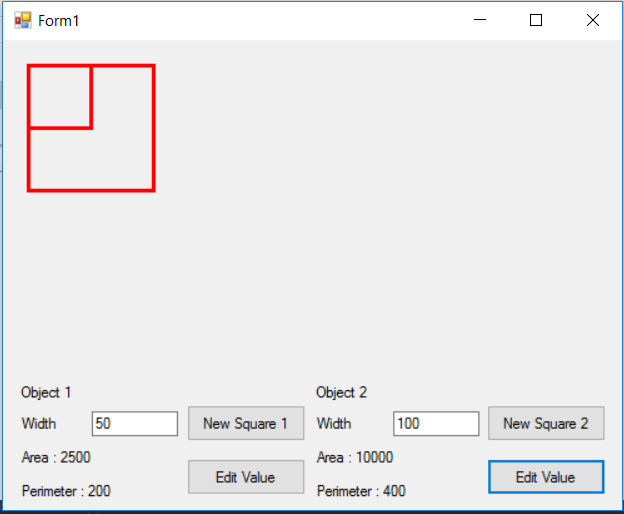
inputArrayB()

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()

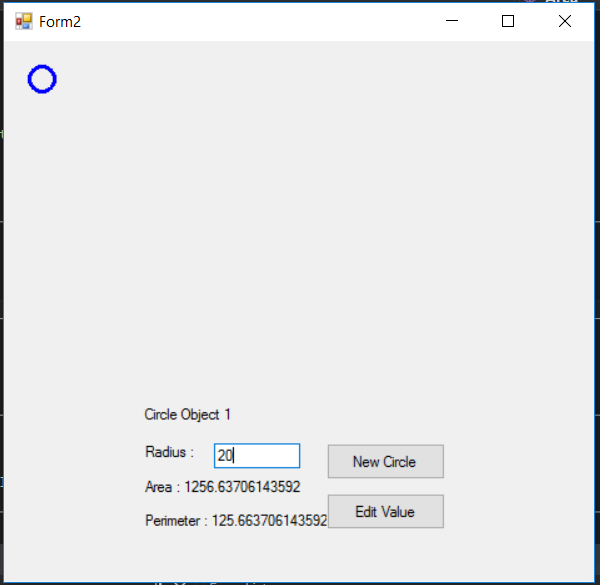
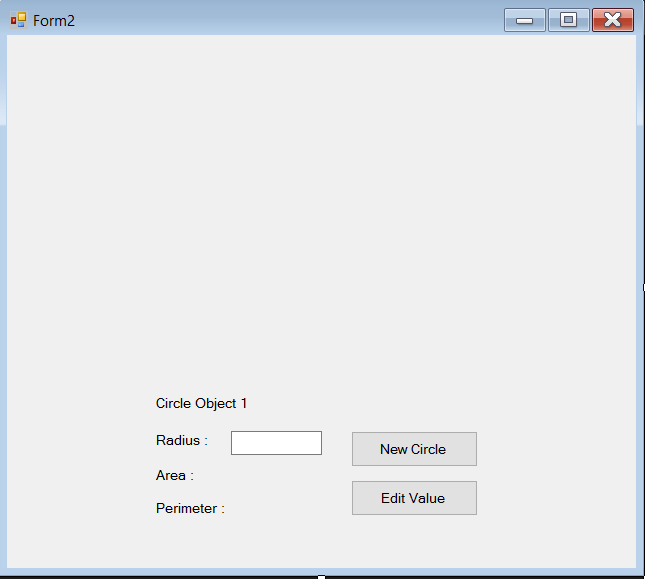
Dim g\_Clear As Graphics = Form1.CreateGraphics

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

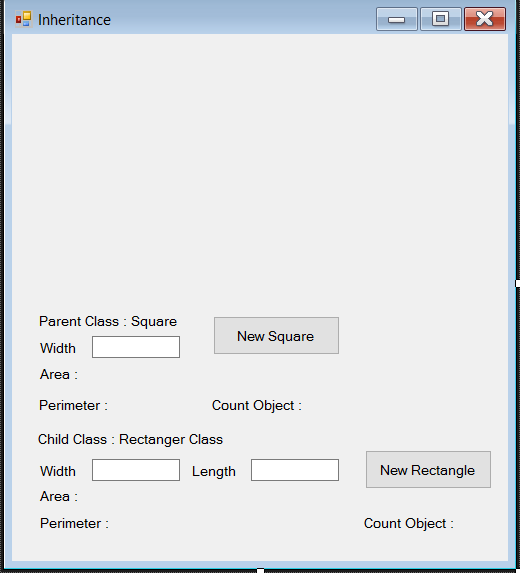
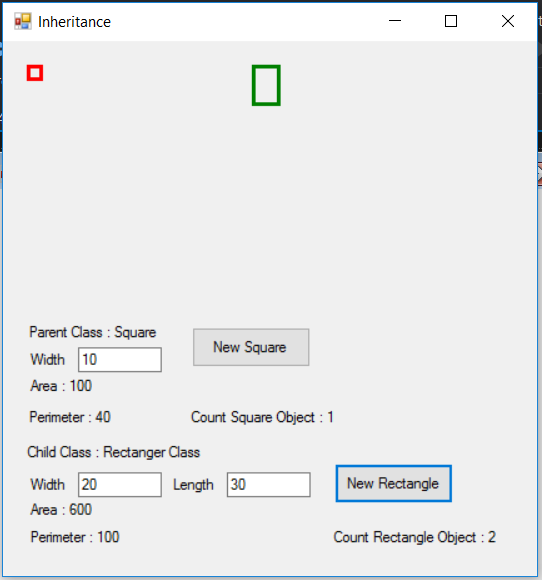
Dim Pen2 As Pen = New Pen(Color.FromArgb(Form2.BackColor.ToArgb), 3)

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))

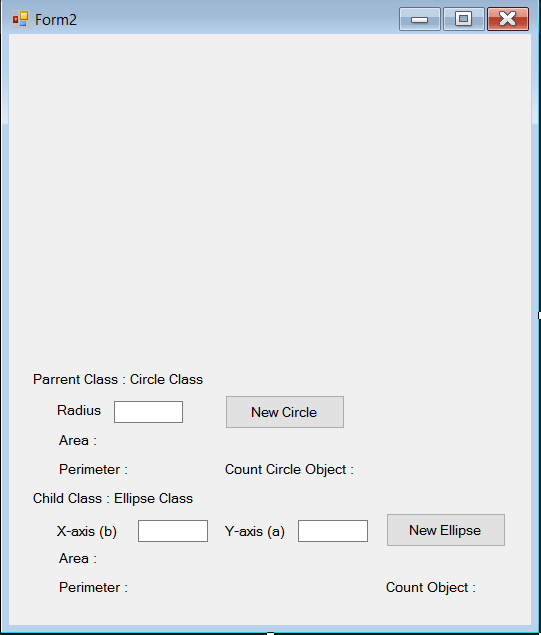
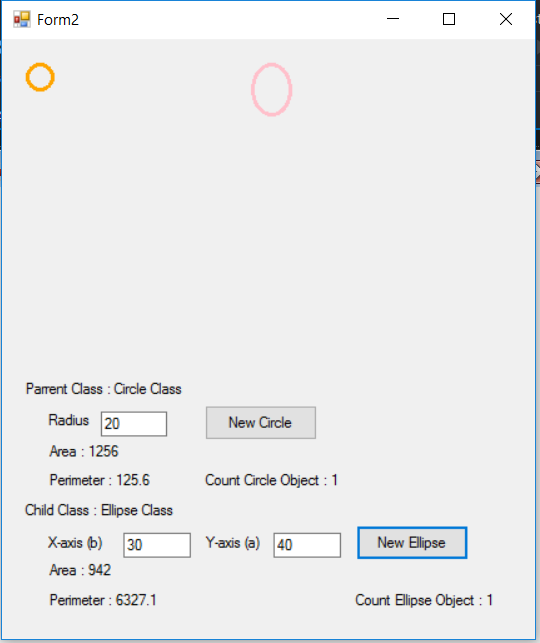
End Sub

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

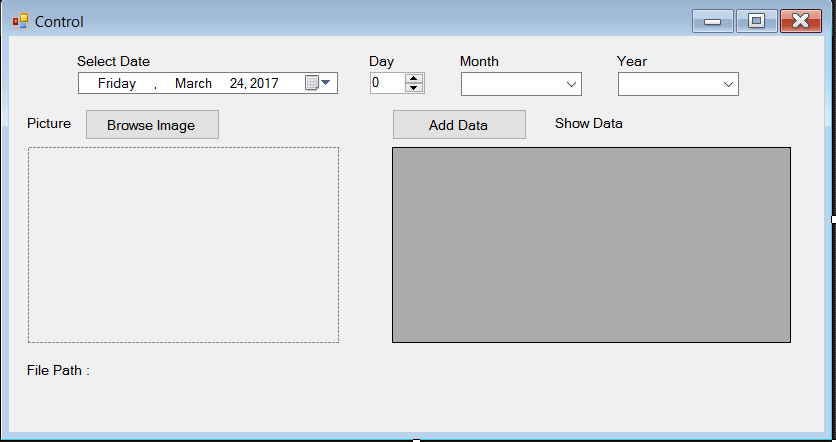
Public Sub New()

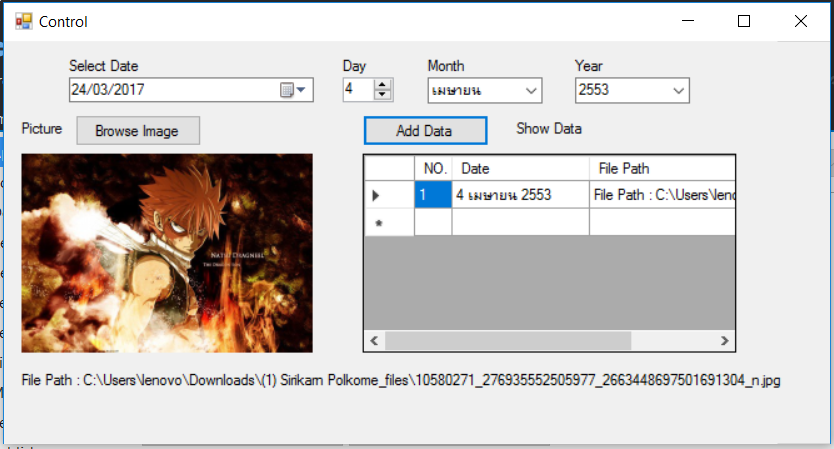
EllipseCount += 1

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

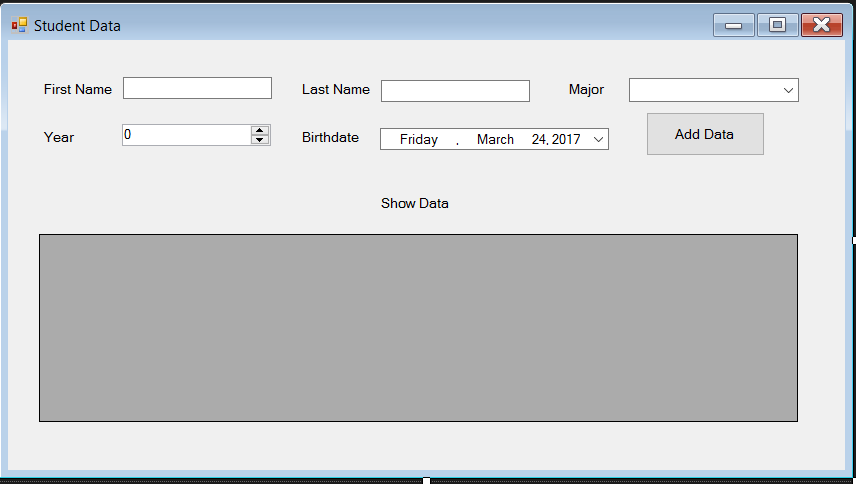
dateT = NumericUpDown\_Day.Value.ToString + " " + ComboBox\_month.Text + " " + ComboBox\_year.Text

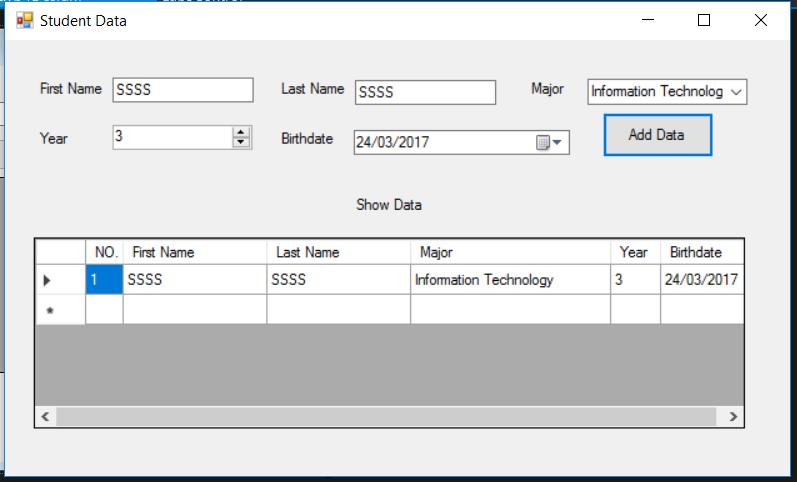
Dim row As String() = New String() {num, dateT, Lb\_filePath.Text}

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