Chapter 2

การออกแบบหน้าจอและการใช้คอนโทรลพื้นฐาน

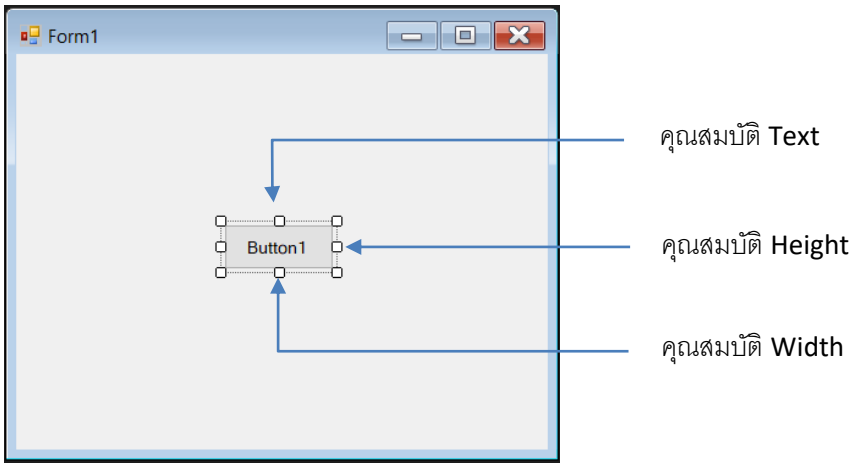
การสร้างโปรแกรมด้วย VB ในขั้นตอนแรกๆ จะเป็นการออกแบบหน้าจอที่ติดต่อกับผู้ใช้ โดยนำคอนโทรลที่มีอยู่มาออกแบบฟอร์มให้เหมาะสม จำเห็นได้ว่าทั้งฟอร์มและคอนโทรลนั้นเป็นเครื่องมือพื้นฐานที่สำคัญในการเขียนโปรแกรม VB สำหรับบนนี้จะกล่าวถึงหลักการพื้นฐานเกี่ยวกับฟอร์ม และคอนโทรลที่ควรทราบ เพื่อให้เข้าใจหลักการออกแบบหน้าจอมากยิ่งขึ้น

2.1 คุณสมบัติ เมธอด และอีเวนต์

ในการทำงานกับฟอร์มและคอนโทรล เราจะต้องทำความคุ้นเคยกับ 3 คำต่อไปนี้ คือ คุณสมบัติ (Properties), เมธอด (Method) และ อีเวนต์ (Event)

2.1.1 คุณสมบัติ (Properties)

เรากำหนดลักษณะต่างๆของ ฟอร์มและคอนโทรล เช่น ปุ่มคำสั่ง ชื่อ Button1 มีคุณสมบัติที่เรากำหนดได้



ภาพที่ 2.1

2.1.2 เมธอด (Method)

ความสามารถของ Object เป็นคำสั่งให้ฟอร์มและคอนโทรลทำงานตามที่เราต้องการ

2.1.3 อีเวนต์ (Event)

เป็นเหตุการณ์ที่เกิดขึ้นกับฟอร์ม หรือคอนโทรลที่เราสามารถใส่คำสั่งเพื่อตอบสนองได้ เช่น ถ้าเราต้องการตอบสนองต่ออีเวนต์ Click ของปุ่มคำสั่งชื่อ Button1 ให้เราดับเบิลคลิกปุ่มเพื่อเข้าสู่หน้าต่างการเขียนโค้ด และใส่คำสั่ง Button1.Text = “Click” จากนั้นเมื่อรันโปรแกรมให้คลิกปุ่มคำสั่งบนหน้าต่างโปรแกรม สังเกตเห็นคำว่า “Button1” บนปุ่มคำสั่ง จะเป็นเป็น “Click” ดังภาพที่ 2.2

```
    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Button1.Text = "Click"
    End Sub
End Class
```

ภาพที่ 2.2

2.2 เนมสเปซ (Namespaces)

หลักการที่สำคัญมากอย่างหนึ่งในการเขียนโปรแกรม VB ก็คือ เนมสเปซ ซึ่งจะช่วยให้การจัดการไลบรารีคลาส และออบเจ็กต์ต่างๆ ได้ง่ายขึ้น ทำให้ไม่เกิดความกำกวมเมื่ออ้างอิงถึงออบเจ็กต์ และระบบขอบเขตการใช้งานของออบเจ็กต์ รายละเอียดของเนมสเปซจะกล่าวในบทการเขียนโปรแกรมเชิงวัตถุ ส่วนบทนี้จะกล่าวถึงวิธีการอ้างอิงถึงเนมสเปซ

การอ้างอิงเนมสเปซ ทำให้เราเรียกใช้คลาสได้ง่าย เช่น คลาส System.WinForms.Button เป็นการเรียกชื่อแบบเต็มๆ ซึ่งยาวและยุ่งยาก แต่ถ้าเราอ้างอิงด้วยคำสั่ง import ดังตัวอย่าง

```
Imports System.WinForms
```

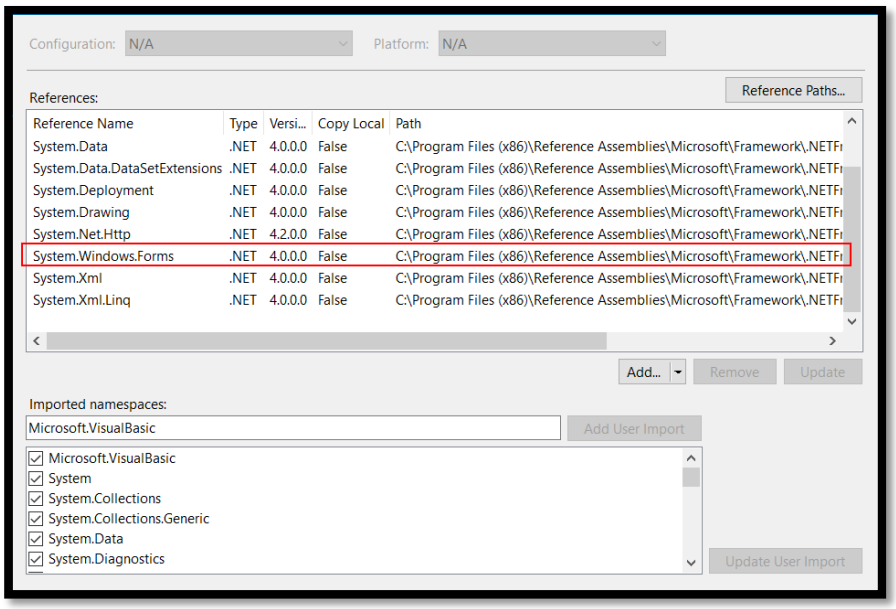
ทำให้เราสามารถเรียกใช้คลาสภายใน WinForms ทั้งหมดได้ในทันที เช่น เรียกใช้คลาส Button แทนการใช้คำว่า System.WinForms.Button ซึ่งจะสั้นกว่าและสะดวกกว่ามาก

2.3 การใช้งานฟอร์ม (Windows Forms)

2.3.1 ฟอร์ม หรือที่เรียกว่า WinForms เป็นเครื่องมือที่ใช้บ่อยมากในการสร้างโปรแกรมด้วย VB โดยผู้ใช้จะติดต่อทำงานผ่านทางคอนโทรลต่างๆ ที่เราวางบนฟอร์ม

ฟอร์มและคอนโทรลต่างๆ ที่มีใน VB นั้น เป็นคลาสที่มีอยู่ในเนมสเปซชื่อ System.Windows.Forms ซึ่งจะมีคลาสต่างๆ ที่ช่วยเราในการสร้างแอพพลิเคชันที่รันบนวินโดวส์ (สำหรับความหมายของคลาสและออบ

เจ็กต์ เราจะกล่าวถึงรายละเอียดเพิ่มเติม ในบทเรื่องการเขียนโปรแกรมเชิงวัตถุ) โดยเนมสเปซ System.Windows.Forms จะมีใช้ในโปรเจ็กต์เราอยู่แล้วตั้งแต่แรก โดยสามารถเข้าไปดูได้ที่หน้าต่าง Solution คลิกขวาที่ชื่อโปรเจคแล้วเลือก Properties จากนั้นไปที่เมนู References ดังภาพที่ 2.3



ภาพที่ 2.3

2.3.2 คุณสมบัติที่สำคัญของฟอร์ม

ฟอร์มมีคุณสมบัติต่างๆ มากมาย แต่ที่เราจำเป็นต้องรู้จัก มีดังต่อไปนี้

ชื่อคุณสมบัติ	คำอธิบาย
Name *	กำหนดชื่อของฟอร์มที่เราใช้อ้างอิงถึงในโปรแกรม
FormBorderStyle	กำหนดลักษณะการเปลี่ยนขนาดขอบของฟอร์ม
MinimizeBox หรือ MaximizeBox	กำหนดว่าให้ขยายหรือย่อฟอร์ม
ControlBox	กำหนดให้ฟอร์มมีเมนูระบบทางมุมซ้ายบนสุดด้วยหรือไม่
Text *	กำหนดข้อความที่แสดงบนไตเติลบาร์ของฟอร์ม
Icon	กำหนดรูปไอคอนของฟอร์มเมื่อย่อ (Minimize) ฟอร์ม
Size *	กำหนด Height : ความสูง และ Width : ความกว้างของฟอร์ม
Location *	กำหนดตำแหน่งของฟอร์มโดยคิดจากตำแหน่งบนซ้ายของหน้าจอ

ชื่อคุณสมบัติ	คำอธิบาย
WindowState	กำหนดว่าเมื่อเริ่มต้นเรียกฟอร์ม จะให้อยู่ในแบบขยาย (Maximize), แบบย่อ (Minimize) หรือแบบธรรมดา
Enabled *	กำหนดให้ฟอร์มสามารถตอบสนองต่ออีเวนต์ได้หรือไม่
TopMost	กำหนดว่าฟอร์มนี้จะแสดงอยู่เหนือฟอร์มอื่นๆ เสมอหรือไม่

หมายเหตุ * เป็นคุณสมบัติที่ใช้กับคอนโทรลอื่นได้ โดยมีความหมายใกล้เคียงกัน

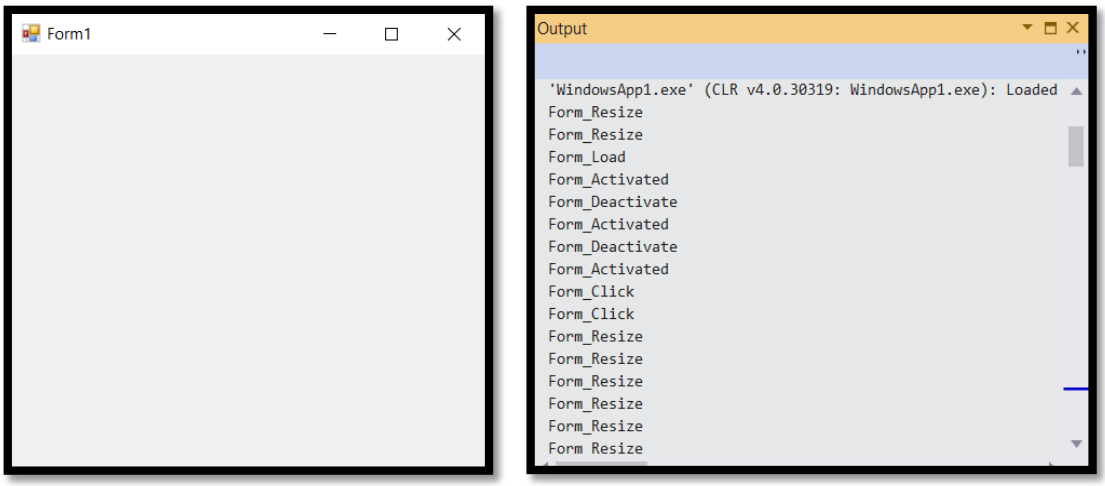
2.3.3 เมธอดและอีเวนต์ที่สำคัญของฟอร์ม

ฟอร์มมีเมธอดและอีเวนต์ที่เราจำเป็นต้องรู้จัก ดังตารางต่อไปนี้

ชื่อเมธอด, อีเวนต์	คำอธิบาย
Load	เป็นอีเวนต์ที่เกิดเมื่อเรียกฟอร์มขึ้นมาครั้งแรก
Resize	เป็นอีเวนต์ที่เกิดเมื่อมีการเปลี่ยนขนาดของฟอร์ม
Activated	เป็นอีเวนต์ที่เกิดขึ้นเมื่อฟอร์มนั้นเป็นฟอร์มที่เรากำลังทำงานด้วย
Deactivate	เป็นอีเวนต์ที่เกิดเมื่อฟอร์มอื่นๆ กลายเป็นฟอร์มที่แอคทีฟแทน
Click	เป็นอีเวนต์ที่เกิดเมื่อมีการใช้เมาส์คลิกบนฟอร์ม
DoubleClick	เป็นอีเวนต์ที่เกิดเมื่อมีการใช้เมาส์ดับเบิ้ลคลิกบนฟอร์ม
Show	เป็นเมธอดที่ใช้แสดงฟอร์มขึ้นมา
Hide	เป็นเมธอดที่ใช้ซ่อนฟอร์มไว้
Close	เป็นเมธอดที่ใช้ปิดฟอร์ม
Activate	เป็นเมธอดที่ใช้ทำให้ฟอร์มถูกเลือกใช้งานในขณะนั้น

2.3.4 ตัวอย่างโปรแกรมแสดงการทำงานของฟอร์ม

ตัวอย่างนี้เป็นเป็นโปรแกรมที่ช่วยให้เราเข้าใจสิ่งต่างๆ เกี่ยวกับฟอร์มมากยิ่งขึ้น ซึ่งเป็นการสร้างฟอร์มหนึ่งฟอร์ม และแสดงอีเวนต์ต่างๆ ทีเกิดขึ้นบนฟอร์มว่าเกิดขึ้นในตอนใด ดูผลลัพธ์ได้จากหน้าต่าง Output โดยโปรแกรมจะเรียกเมธอด WriteLine ของออบเจ็กต์ Console คำสั่ง Console.WriteLine เพื่อพิมพ์ชื่ออีเวนต์ที่เกิดขึ้นบนฟอร์มออกมา โดยมีการทำงานดังนี้

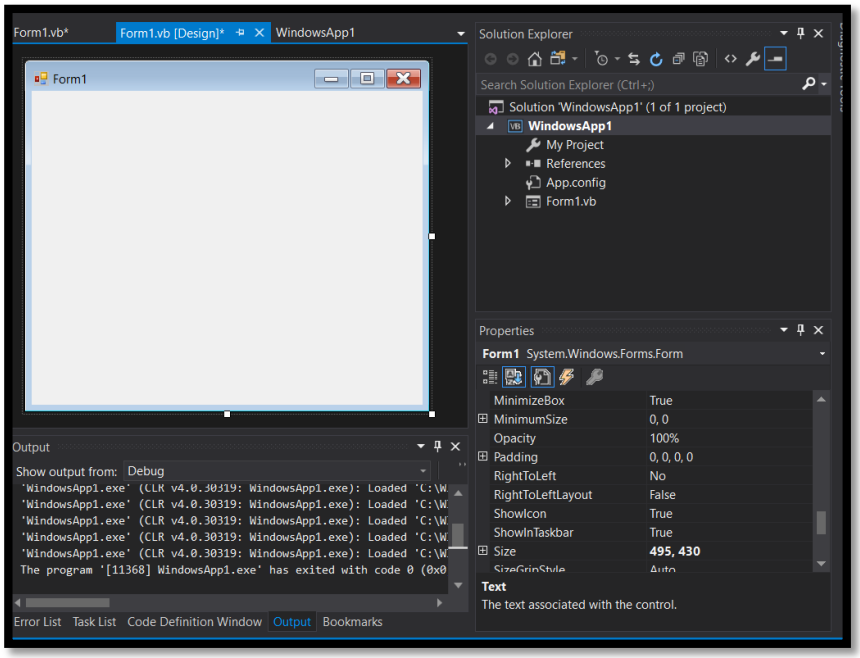


ภาพที่ 2.4

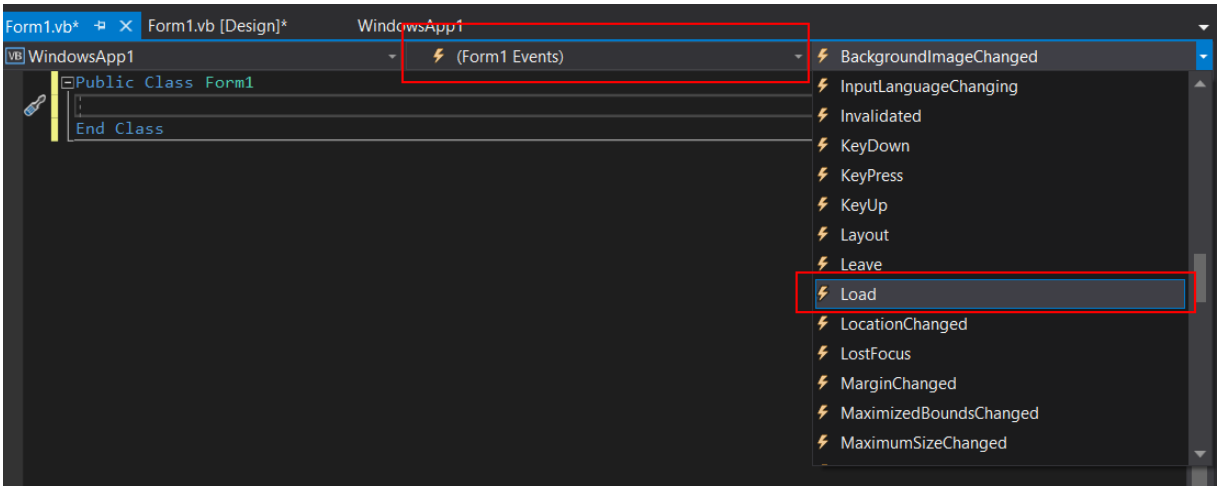
ขั้นตอนการสร้างโปรเจ็กต์

1) สร้างโปรเจ็กต์ใหม่ชนิด Windows Forms App

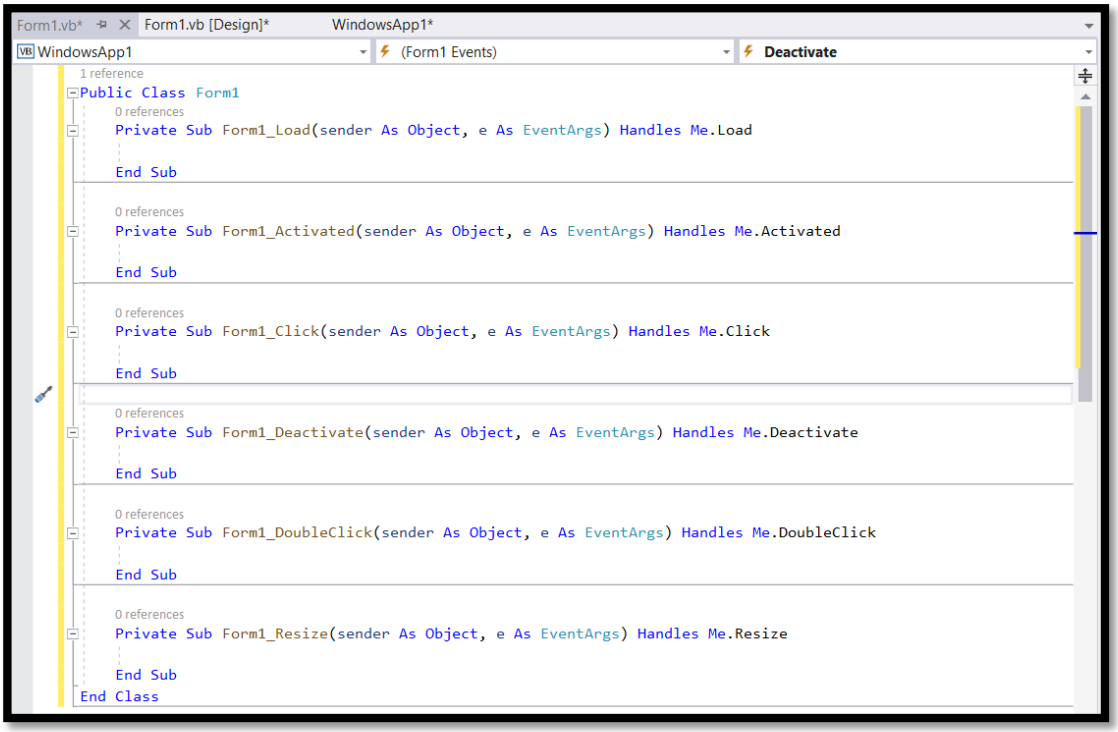
2) เลือก View -> Code หรือ กด F7 เพื่อเปิดหน้าต่างโค้ดขึ้นมา จากนั้นให้เลือกอีเวนต์ต่างๆ ของฟอร์มตามคำสั่งในโปรแกรม โดยเลือกจาก (Form1 Events) และเลือกอีเวนต์ของฟอร์มที่ต้องการตอบสนอง ในที่นี้เลือก 6 อีเวนต์ คือ Load, Activated, Click, Deactivate, DoubleClick, Resize



ภาพที่ 2.5 สร้างโปรเจ็กต์ใหม่



ภาพที่ 2.6 เลือก (Form1 Events) และเลือกอีเวนต์ตอบสนอง 6 ตัว



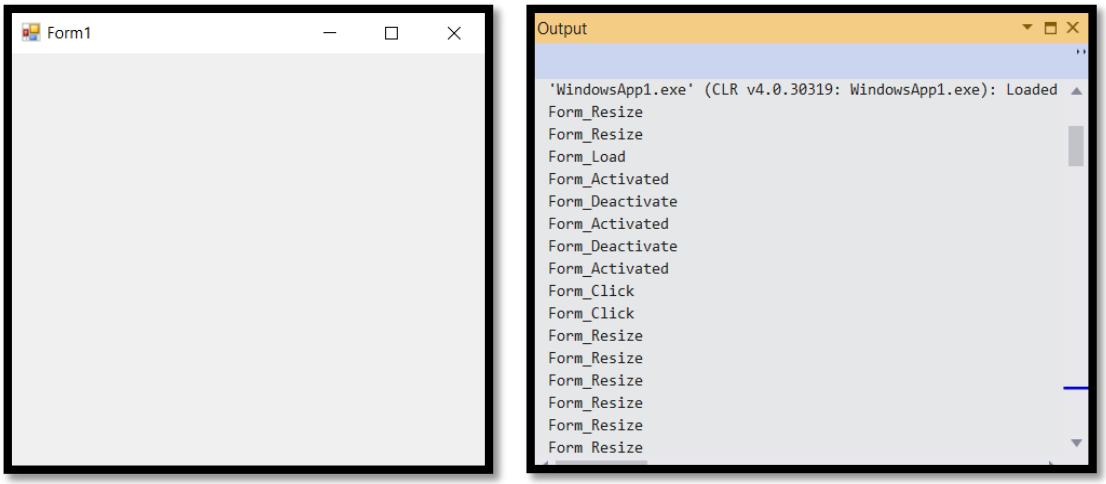
ภาพที่ 2.7 โปรแกรมย่อยหลังจากที่เลือกอีเวนต์

3) จากนั้นในหน้าต่างโค้ดโปรแกรม ให้เราใส่คำสั่งโปรแกรมต่อไปนี้ลงไปในอีเวนต์ต่างๆ ดังนี้

```
Public Class Form1
    0 references
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
        Console.WriteLine("Form_Load")      'ตอบสนองต่ออีเวนต์ Load
    End Sub
    0 references
    Private Sub Form1_Activated(sender As Object, e As EventArgs) Handles Me.Activated
        Console.WriteLine("Form_Activated")    'ตอบสนองต่ออีเวนต์ Activated
    End Sub
    0 references
    Private Sub Form1_Click(sender As Object, e As EventArgs) Handles Me.Click
        Console.WriteLine("Form_Click")        'ตอบสนองต่ออีเวนต์ Click
    End Sub
    0 references
    Private Sub Form1_Deactivate(sender As Object, e As EventArgs) Handles Me.Deactivate
        Console.WriteLine("Form_Deactivate")     'ตอบสนองต่ออีเวนต์ Deactivate
    End Sub
    0 references
    Private Sub Form1_DoubleClick(sender As Object, e As EventArgs) Handles Me.DoubleClick
        Console.WriteLine("Form_DoubleCLick")     'ตอบสนองต่ออีเวนต์ DoubleClick
    End Sub
    0 references
    Private Sub Form1_Resize(sender As Object, e As EventArgs) Handles Me.Resize
        Console.WriteLine("Form_Resize")        'ตอบสนองต่ออีเวนต์ Resize
    End Sub
End Class
```

ภาพที่ 2.8

4) บันทึกโปรเจ็กต์และรันทดสอบโปรแกรม โดยเลือก Debug และคลิก Start โดยผลลัพธ์การรันโปรแกรมจะปรากฏชื่อ Form1 ขึ้นมา และเมื่อมีอีเวนต์เกิดขึ้น โปรแกรมจะพิมพ์อีเวนต์ที่เกิดขึ้นบนหน้าต่าง Output ดังรูป



ภาพที่ 2.9

<u>เพิ่มเติม</u> คำสั่ง Console.WriteLine จะมีประโยชน์มากในการแสดงค่าข้อมูลในโปรแกรมของเราออกมาทางหน้าต่าง Output สามารถเรียกขึ้นมาแสดงได้โดยไปที่เมนู View-> Output หรือกดปุ่ม <Ctrl+Alt+O>

2.3.5 คอนโทรลและคอมโพเนนต์พื้นฐาน

ทูลบ็อกซ์ ประกอบด้วยเครื่องมือที่ใช้นำมาวางบนฟอร์มเพื่อออกแบบส่วนติดต่อผู้ใช้งาน มีเครื่องมือ 2 ประเภท คือ **คอนโทรล** ออบเจ็กต์ที่วางบนฟอร์มที่ผู้ใช้สามารถมองเห็น เช่น ปุ่ม กล่องข้อความ เป็นต้น และ **คอมโพเนนต์** ออบเจ็กต์ที่วางบนฟอร์มแต่ผู้ใช้มองไม่เห็น เช่น การนับเวลา เป็นต้น

คอนโทรล	คำอธิบาย
Button	ให้ผู้ใช้คลิกสั่งงานมายังโปรแกรม
Label	แสดงข้อมูลให้ผู้ใช้เห็นเพียงอย่างเดียว
Textbox	รับข้อมูลจากผู้ใช้ที่ป้อนเข้ามาในโปรแกรมและแสดงผลข้อมูล
CheckBox	ให้ผู้ใช้เลือกได้มากกว่า 1 ตัวเลือก
RadioButton	ให้ผู้ใช้เลือกได้ตัวใดตัวหนึ่งจากกลุ่มตัวเลือกทั้งหมด
ListBox	ให้ผู้ใช้เลือกโดยการเลื่อนดูรายการที่ต้องการได้
ComboBox	ให้ผู้ใช้เลือกคล้ายกับลิสต์บ็อกซ์ แต่สามารถพิมพ์ข้อมูลเข้าไปได้ด้วย
CheckedListBox	ให้ผู้ใช้เลือกรายการได้โดยเช็กที่หน้ารายการนั้น
GroupBox	รวมคอนโทรลให้อยู่ในกลุ่มเดียวกัน
PictureBox	ใส่ภาพประกอบและตกแต่งให้สื่อความหมายและสวยงาม
ScrollBar	เลื่อนดูข้อมูลทั้งหมดที่ไม่สามารถแสดงในคราวเดียวบนพื้นที่ที่จำกัด

คอมโพเนนต์	คำอธิบาย
Timer	ใช้นับเวลาตามช่วงเวลาที่กำหนดเพื่อควบคอมการทำงานบางอย่าง

2.3.5.1 คอนโทรลปุ่มคำสั่ง (Button)

ปุ่มคำสั่งมีหน้าที่หลักคือ ตอบสนองต่อการคลิกเมาส์ของผู้ใช้งานที่สั่งงานมายังโปรแกรมว่าต้องการให้โปรแกรมทำอะไรต่อ

ตัวอย่างโปรแกรมที่ใช้ปุ่มคำสั่ง

ในตัวอย่างนี้ จะแสดงการใช้งานปุ่มคำสั่ง ถ้าเราไม่ใส่ข้อความในเท็กซ์บ็อกซ์ ปุ่มคำสั่งจะไม่ตอบสนองการทำงาน แต่ถ้าเราใส่ข้อความในเท็กซ์บ็อกซ์ ปุ่มคำสั่งจะตอบสนองต่อการคลิกเมาส์ โดยแสดงชื่อที่พิมพ์ในเท็กซ์บ็อกซ์บนกล่องข้อความ

1) สร้างโปรเจคใหม่ชนิด Windows Forms App จากนั้นให้วางคอนโทรล และเปลี่ยนชื่อปุ่มกับลาเบล ดังภาพที่ 2.10

ภาพที่ 2.10

2) สังเกตได้ว่าที่ปุ่มมีรูปภาพไอคอนเพิ่มขึ้นมา โดยโปรแกรม Visual Studio มีคลังภาพให้นักพัฒนาโปรแกรมสามารถดาวน์โหลดมาใช้ประกอบในโปรแกรมได้ฟรี เรียกว่า Visual Studio Image Library ในตัวอย่างเราจะใช้ไฟล์ภาพ UserProfile_16x.png มาตกแต่งบนปุ่ม มีขั้นตอนดังนี้

- เปิดเว็บ Google แล้ว Search คำว่า Microsoft Visual Studio Image Library
- เข้าเว็บ https://www.microsoft.com/en-us/download/details.aspx?id=35825
- คลิกที่ Download
- เลือก VS2017 Image Library.zip จากนั้นคลิกปุ่ม Next
- ทำการ Save บันทึกไว้ในเครื่องคอมพิวเตอร์
- จะได้ไฟล์ VS2017 Image Library.zip ที่เก็บภาพไอคอนและกราฟฟิกจำนวนมาก

- จากนั้นแตกไฟล์ .zip ออกมา แล้วหาโฟลเดอร์ UserProfile

- ก็อปปี้ไฟล์ UserProfile_16x.png มาใส่ในโฟลเดอร์โปรเจ็กต์ของเรา

- คลิกเลือกคอนโทรลปุ่มที่เราต้องการวางภาพประกอบ

- ที่หน้าต่าง Properties ให้คลิกกำหนดคุณสมบัติ Image ที่  เพื่อใส่ภาพปุ่ม

- จะปรากฏหน้าต่าง Select Resource ที่ Project resource file ให้คลิก Import เพื่อเข้าไปเลือกไฟล์ภาพที่ต้องการนำมาวางบนปุ่ม

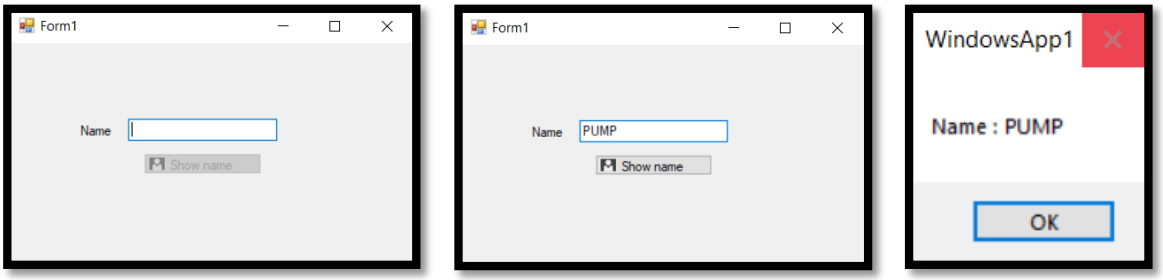
- ปรากฏหน้าต่าง Open ให้เลือกไฟล์ภาพที่จัดเตรียมเพื่อนำมาวางบนปุ่มและคลิก Open

- กลับมาที่หน้าต่าง Select Resource ที่ Project resource file ให้เลือก UserProfil_16x และคลิกปุ่ม OK จากนั้นภาพจะถูกวางบนปุ่ม ทำการจัดตำแหน่งให้เรียบร้อยที่หน้าต่าง Properties กำหนดคุณสมบัติที่ ImageAlign

3) ดับเบิลคลิกบนปุ่มเพื่อเข้าสู่ Code Editor จากนั้นสร้างโค้ดอีเวนต์ TextBox1_TextChanged แล้วให้ใส่คำสั่งภายในอีเวนต์ต่างๆ ดังนี้

```
1 reference
Public Class Form1
    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        'สร้างกล่องข้อความแสดงข้อความของ TextBox1
        MsgBox("Name : " & TextBox1.Text, MsgBoxStyle.OkOnly)
    End Sub
    0 references
    Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles TextBox1.TextChanged
        'ตรวจสอบข้อความใน TextBox1 เป็นข้อความว่างหรือไม่
        If TextBox1.Text <> "" Then
            Button1.Enabled = True      'ถ้าไม่ว่าง ทำให้ปุ่มคำสั่งสามารถทำงานได้
        Else
            Button1.Enabled = False     'ถ้าว่าง ทำให้ปุ่มคำสั่งทำงานไม่ได้
        End If
    End Sub
End Class
```

ภาพที่ 2.10



ภาพที่ 2.11

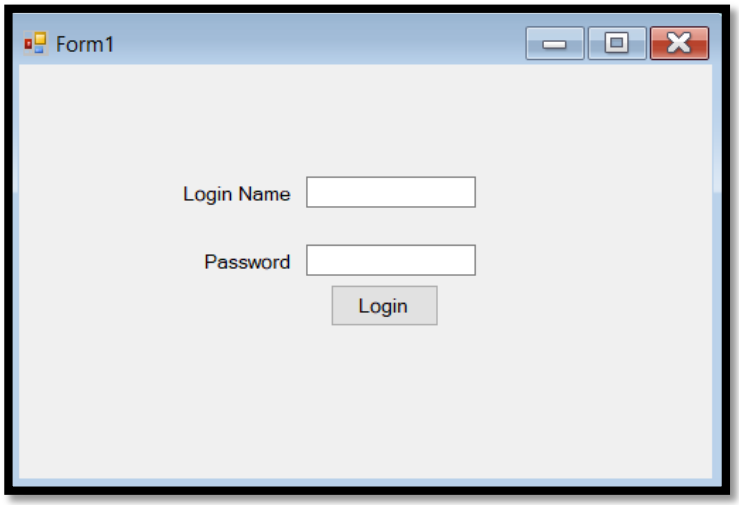
2.3.5.2 คอนโทรลเท็กซ์บ็อกซ์ (TextBox)

เป็นคอนโทรลที่เราใช้บ่อยมากในการรับข้อมูลจากผู้ใช้ที่ป้อนเข้ามาในโปรแกรม รวมทั้งสามารถแสดงผลและให้ผู้ใช้แก้ไขข้อมูลได้ด้วย การปรับแต่งจะอยู่ในส่วยของ Properties

<u>ตัวอย่างโปรแกรมแสดงการใช้งานเลเบล เท็กซ์บ็อกซ์ และปุ่มกด</u>

<u>ขั้นตอนการสร้างโปรแกรม</u>

1) สร้างโปรเจคใหม่เป็นชนิด Windows Forms App จากนั้นวางคอนโทรลต่างๆ บนฟอร์ม และกำหนดคุณสมบัติ ดังนี้



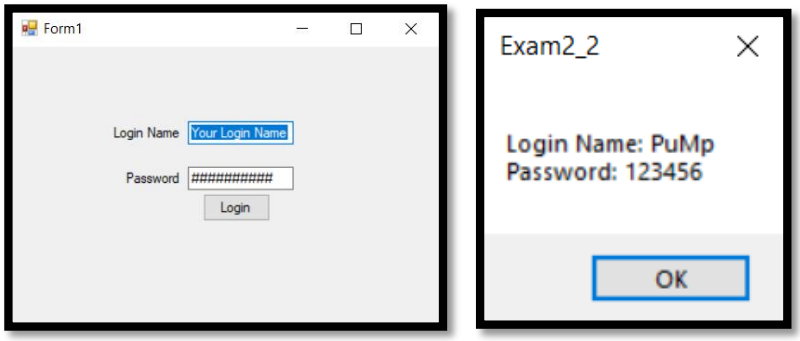
ภาพที่ 2.12

2) ดับเบิ้ลคลิกที่ปุ่ม Login เพื่อเข้าสู่หน้า Code Editor ให้ใส่คำสั่งใน Event Button1_Click จากนั้นดับเบิลคลิกที่ว่างบนฟอร์มสร้างโค้ดอีเวนต์ Form1_Load และใส่โค้ดการทำงานดังนี้

```
Public Class Form1
    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        'เมื่อคลิกปุ่มกดจะปรากฏกล่องข้อความแสดงชื่อล็อกอินและรหัสผ่านที่ใส่เข้าไป
        'vbnewline เป็นคำสั่งที่ใช้ขึ้นบรรทัดใหม่ใช้คู่กับ &
        MsgBox("Login Name: " & TextBox1.Text & vbNewLine & "Password: " & TextBox2.Text)
    End Sub
    0 references
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'เมื่อโหลดฟอร์มขึ้นมาให้แสดงข้อความเริ่มต้นดังนี้
        TextBox1.Text = "Your Login Name"
        TextBox2.Text = "##########"
    End Sub
End Class
```

ภาพที่ 2.13

3) รันโปรแกรมที่สร้างมา จะได้ผลลัพธ์ดังรูป



ภาพที่ 2.14

2.3.5.3 คอนโทรลเช็กบ็อกซ์ (CheckBox)

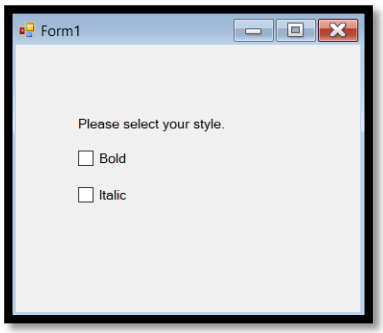
☑ คอนโทรลเช็กบ็อกซ์จะให้ผู้ใช้เลือกว่าจะเช็ก หรือไม่เช็ก เพียงอย่างใดอย่างหนึ่งเท่านั้น ค่าของคอนโทรลเช็กบ็อกซ์จะอยู่ในคุณสมบัติ Checked ที่มีค่าเป็น True หรือ False หรือคุณสมบัติ CheckState

<u>ตัวอย่างโปรแกรมแสดงการใช้งานเช็กบ็อกซ์</u>

ตัวอย่างต่อไปนี้เป็นโปรแกรมแสดงการทำงานของเช็กบ็อกซ์ โดยเราจะใช้เช็กบ็อกซ์ในการเลือกว่าต้องการตัวอักษรแบบตัวเน้น (Bold) หรือแบบตัวเอียง (Italic) จากนั้นจะปรากฏกล่องข้อความแสดงสิ่งที่เราเลือก

<u>ขั้นตอนการสร้างโปรแกรม</u>

1) สร้างโปรเจคชนิด Windows Forms App ใหม่ขึ้นมา และให้เราวางคอนโทรลต่างๆ ตามรูป



ภาพที่ 2.15

2) ไปที่หน้าต่าง Code Editor จากนั้นใส่คำสั่งโปรแกรมดังนี้

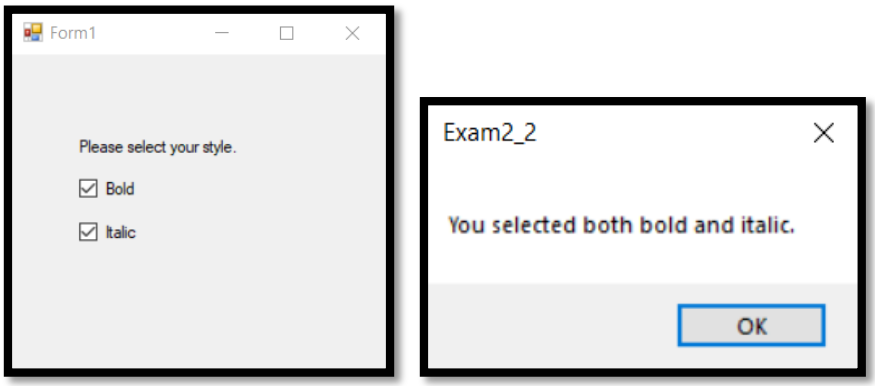
```
1 reference
Public Class Form1
    0 references
    Private Sub CheckBox1_CheckedChanged(sender As Object,
        e As EventArgs) Handles CheckBox1.CheckedChanged
        If CheckBox1.Checked = True Then 'ตรวจสอบว่า CheckBox1 ถูกเลือกหรือไม่
            If CheckBox2.Checked = True Then 'ตรวจสอบว่า CheckBox2 ถูกเลือกหรือไม่
                'ถ้าถูกเลือก
                MsgBox("You selected both bold and italic.", MsgBoxStyle.OkOnly)
            Else
                'ถ้าไม่ถูกเลือก
                MsgBox("You selected bold.", MsgBoxStyle.OkOnly)
            End If
        End If

    End Sub
```

```
    0 references
    Private Sub CheckBox2_CheckedChanged(sender As Object,
        e As EventArgs) Handles CheckBox2.CheckedChanged
        If CheckBox2.Checked = True Then 'ตรวจสอบว่า CheckBox2 ถูกเลือกหรือไม่
            If CheckBox1.Checked = True Then 'ตรวจสอบว่า CheckBox1 ถูกเลือกหรือไม่
                'ถ้าถูกเลือก
                MsgBox("You selected both bold and italic.", MsgBoxStyle.OkOnly)
            Else
                'ถ้าไม่ถูกเลือก
                MsgBox("You selected italic.", MsgBoxStyle.OkOnly)
            End If
        End If
    End Sub
End Class
```

ภาพที่ 2.16

3) รันโปรแกรมที่สร้างมาจะได้ผลลัพธ์ดังรูป



ภาพที่ 2.17

2.3.5.4 คอนโทรลเรดิโอบัตทอน (RadioButton)

 คอนโทรลเรดิโอบัตทอน ทำหน้าที่คล้ายกับเช็กบ็อกซ์ แต่คอนโทรลนี้จะสามารถเลือกได้เพียงตัวเลือกเดียวเท่านั้นในกลุ่มหนึ่ง

Example จากโจทย์เดิมข้างต้น ให้เปลี่ยนจาก CheckBox เป็น RadioButton โดยให้เลือก Bold กับ Italic เหมือนเดิม แล้วรันให้ได้ผลลัพธ์ออกมาตามที่เลือก

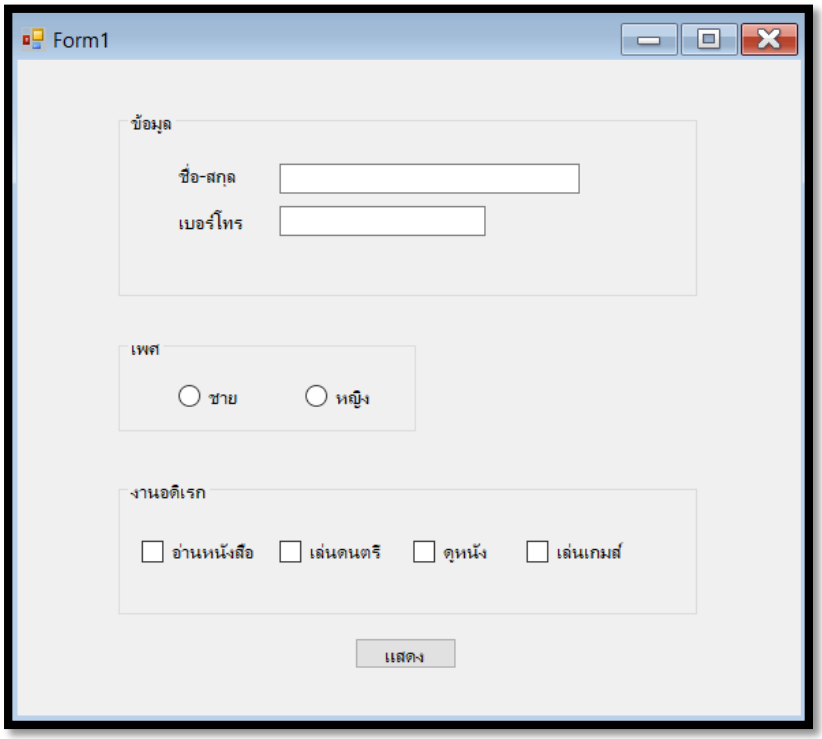
2.3.5.5 คอนโทรลกรุ๊ปบ็อกซ์ (GroupBox)

 คอนโทรลกรุ๊ปบ็อกซ์ใช้สำหรับจัดกลุ่มคอนโทรลที่ใช้ร่วมกัน ช่วยเพิ่มความเรียนร้อยและความสวยงามของหน้าตาโปรแกรมมากขึ้น เช่น ใช้แบ่งเรดิโอบัตทอน บนฟอร์มออกเป็นกลุ่มๆ เป็นต้น

ตัวอย่างโปรแกรมการใช้งานกรุ๊ปบ็อกซ์

เป็นการใช้กรุ๊ปบ็อกซ์จัดกกลุ่ม เรดิโอบัตทอน เช็กบ็อกซ์ และคอนโทรลอื่นๆ ออกแบบหน้าจอเป็นแบบฟอร์มสำหรับผู้ใช้กรอกข้อมูล และเมื่อคลิกปุ่ม Ok จะแสดงข้อมูลทั้งหมดบนกล่องข้อควา

1) สร้างโปรเจคใหม่ขึ้นมาชนิด Windows Forms App ชื่อ GroupBox และให้คอนโทรลต่างๆ ลงบนฟอร์ม พร้อมทั้งกำหนดคุณสมบัติต่างๆดังนี้



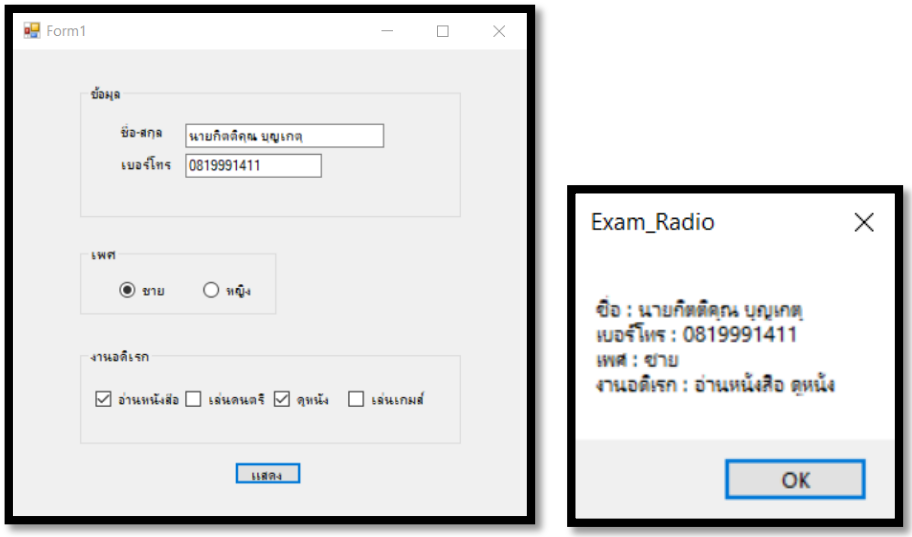
ภาพที่ 2.18

2) ดับเบิ้ลคลิ๊กที่ปุ่ม Button1 จะแสดงหน้าต่าง Code Editor จากนั้นใส่โค้ดโปรแกรม ดังนี้

```
Public Class Form1
    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim sex, hobby As String            'สร้างตัวแปร sex, hobby เก็บข้อความ
        If RadioButton1.Checked Then        'ถ้าเช็คเพศชายที่ RadioButton1
            sex = RadioButton1.Text         'เก็บข้อความ เพศชาย ในตัวแปร sex
        End If
        If RadioButton2.Checked Then        'ถ้าเช็คเพศหญิงที่ RadioButton2
            sex = RadioButton2.Text         'เก็บข้อความ เพศหญิง ในตัวแปร sex
        End If
        If CheckBox1.Checked Then           'ถ้าเช็กอ่านหนังสือที่ CheckBox1
            hobby = CheckBox1.Text          'เก็บข้อความอ่านหนังสือในตัวแปร hobby
        End If
        If CheckBox2.Checked Then           'ถ้าเช็กเล่นดนตรีที่ CheckBox2
            hobby += " " + CheckBox2.Text   'เก็บข้อความเพิ่มเติมในตัวแปร hobby
        End If
        If CheckBox3.Checked Then
            hobby += " " + CheckBox3.Text
        End If
        If CheckBox4.Checked Then
            hobby += " " + CheckBox4.Text
        End If
        MsgBox("ชื่อ : " & TextBox1.Text & vbNewLine _
            & "เบอร์โทร : " & TextBox2.Text & vbNewLine _
            & "เพศ : " & sex & vbNewLine _
            & "งานอดิเรก : " & hobby)
    End Sub
End Class
```

ภาพที่ 2.19

3) รันโปรแกรมที่สร้างมาจะได้ผลลัพธ์ ดังภาพ



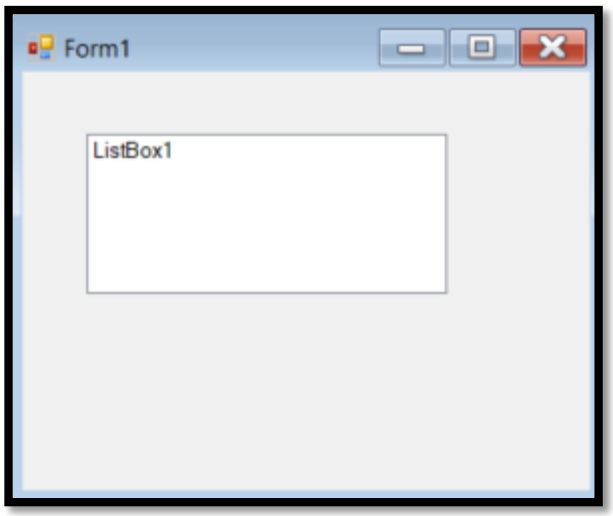
ภาพที่ 2.20

2.3.5.6 คอนโทรลลิสต์บ็อกซ์ (ListBox)

เราจะสังเกตได้ว่าทั้งเรดิโอบัตทอนและเช็กบ็อกซ์นั้น ถ้านำมาใช้แสดงตัวเลือกหลายๆตัว จะใช้เนื้อที่บนฟอร์มพอสมควร ในกรณีนี้เราสามารถนำเอาคอนโทรลลิสต์บ็อกซ์มาใช้แทนที่การแสดงตัวเลือกหลายๆ ตัวได้ในแบบของรายการ

<u>ตัวอย่างโปรแกรมการใช้งานลิสต์บ็อกซ์</u>

1) สร้างโปรเจ็กต์ใหม่ชนิด Windows Forms App ขึ้นมา กดคอนโทรลลิสต์บ็อกซ์มาวางบนฟอร์ม



ภาพที่ 2.21

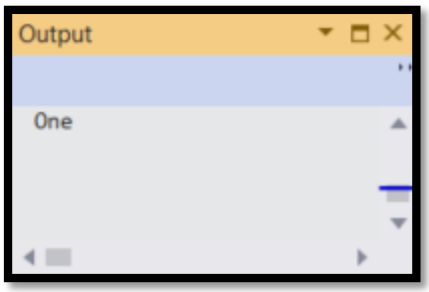
2) ไปที่หน้าต่าง Code Editor และใส่โค้ดโปรแกรมดงนี้

```
Public Class Form1
    0 references
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
        ListBox1.Items.Add("One")
        ListBox1.Items.Add("Two")
        ListBox1.Items.Add("Three")
        ListBox1.Items.Add("Four")
        ListBox1.Items.Add("Five")
        Console.WriteLine(ListBox1.Items(0))
    End Sub
End Class
```

ภาพที่ 2.22

เมื่อเปิดแสดงหน้าจอโปรแกรมจะเริ่มทำงานอีเวนต์ Form1_Load โดยเพิ่มรายการเข้าไปในลิสต์บอกซ์ด้วยคำสั่ง Item.Add() จำนวน 5 รายการ สุดท้ายจะหาค่าในรายการที่ตำแหน่ง Index(0) โดยให้พิมพ์แสดงค่าผลลัพธ์ในหน้าต่าง Output

3) คลิก Start เพื่อรันโปรแกรม จะแสดงรายการแรกสุดออกมาทางหน้าต่าง Output คือ One



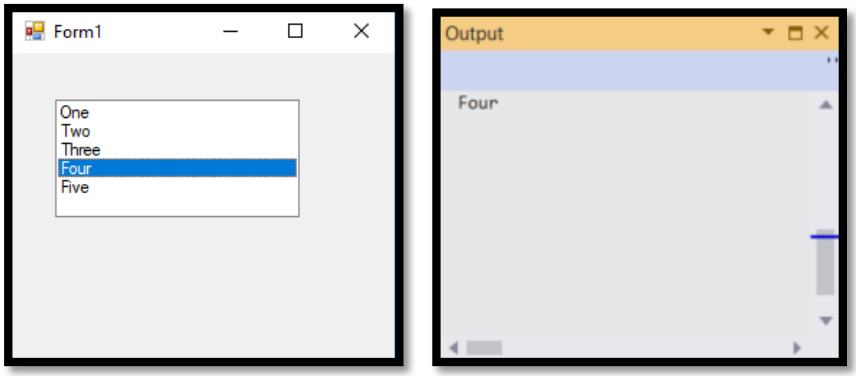
ภาพที่ 2.23

4) ให้เพิ่มโค้ดในส่วนอีเวนต์ ListBox1_SelectedIndexChanged เพื่อให้ทำงานเมื่อมีการเลือกรายการ โดยให้แสดงค่าจากรายการที่ถูกเลือกด้วยคุณสมบัติ Text เป็นผลลัพธ์หน้าต่าง Output ดังนี้

```
Public Class Form1
    0 references
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
        ListBox1.Items.Add("One")
        ListBox1.Items.Add("Two")
        ListBox1.Items.Add("Three")
        ListBox1.Items.Add("Four")
        ListBox1.Items.Add("Five")
    End Sub
    0 references
    Private Sub ListBox1_SelectedIndexChanged(sender As Object,
        e As EventArgs) Handles ListBox1.SelectedIndexChanged
        Console.WriteLine(ListBox1.Text)      'แสดงข้อความที่เลือกจาก ListBox1
    End Sub
End Class
```

ภาพที่ 2.24

5) เมื่อรันโปรแกรม จะแสดงรายการบนลิสต์บ็อกซ์ จากนั้นให้ผู้ใช้เลือกรายการ ก็จะนำค่าจากคุณสมบัติ Text ออกมาแสดงในหน้าต่าง Output



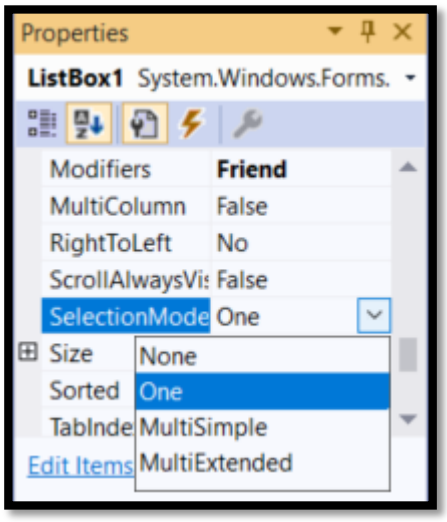
ภาพที่ 2.25

6) เราสามารถหาจำนวนของรายการในลิสต์บ็อกซ์ โดยใช้คุณสมบัติ Count ของลิสต์บ็อกซ์ เช่น

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
    ListBox1.Items.Add("One")
    ListBox1.Items.Add("Two")
    ListBox1.Items.Add("Three")
    ListBox1.Items.Add("Four")
    ListBox1.Items.Add("Five")
    Console.WriteLine(ListBox1.Items.Count)
End Sub
```

ภาพที่ 2.26

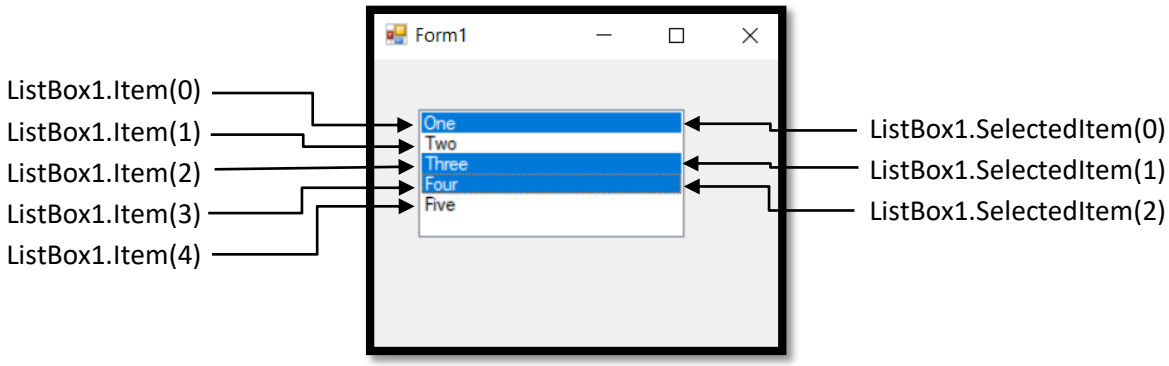
7) เราสามารถกำหนดว่าจะให้ผู้ใช้เลือกรายการในลิสต์บ็อกซ์ได้ทีละรายการหรือหลายรายการ โดยกำหนดที่คุณสมบัติ SelectionMode ซึ่งแต่ละค่าจะมีความหมายดังนี้



ภาพที่ 2.27

- None ไม่สามารถเลือกได้ หมายถึง ลิสต์บ็อกซ์ที่แสดงอย่างเดียว ไม่สามารถเลือกได้
- One เลือกได้รายการเดียว เป็นลิสต์บ็อกซ์แบบทั่วไป
- MultiSimple เลือกได้หลายรายการแบบ 1 กด <Spacebar> จะเป็นการเลือกรายการในลิสต์บ็อกซ์
- MultiExtended เลือกได้หลายรายการแบบ 2 โดยมีวิธีการกดปุ่มเลือกอยู่ 2 แบบ คือ
  - ✓ กดปุ่ม <Shift> พร้อมกับการคลิก จะเป็นการเลือกรายการที่อยู่ระหว่างที่เลือกไว้ก่อนหน้ากับรายการที่เลือกไว้ในปัจจุบัน
  - ✓ กดปุ่ม <Ctrl> พร้อมกับการคลิกรายการที่ต้องการ จะเป็นการเลือกรายการต่างๆ ในลิสต์บ็อกซ์

สำหรับลิสต์บ็อกซ์ที่เลือกได้หลายรายการ เราสามารถทราบว่าผู้ใช้ได้เลือกรายการใด โดยการตรวจสอบคุณสมบัติ SelectedItems ซึ่งเป็นคอลเล็กชันที่เก็บออบเจ็กต์ Item ที่ผู้ใช้เลือกไว้ นอกจากนี้ยังมีคุณสมบัติ SelectedIndices ที่เก็บอินเด็กซ์รายการแต่ละตัวที่ผู้ใช้เลือกไว้ในลิสต์บ็อกซ์ ดังรูป



ภาพที่ 2.28

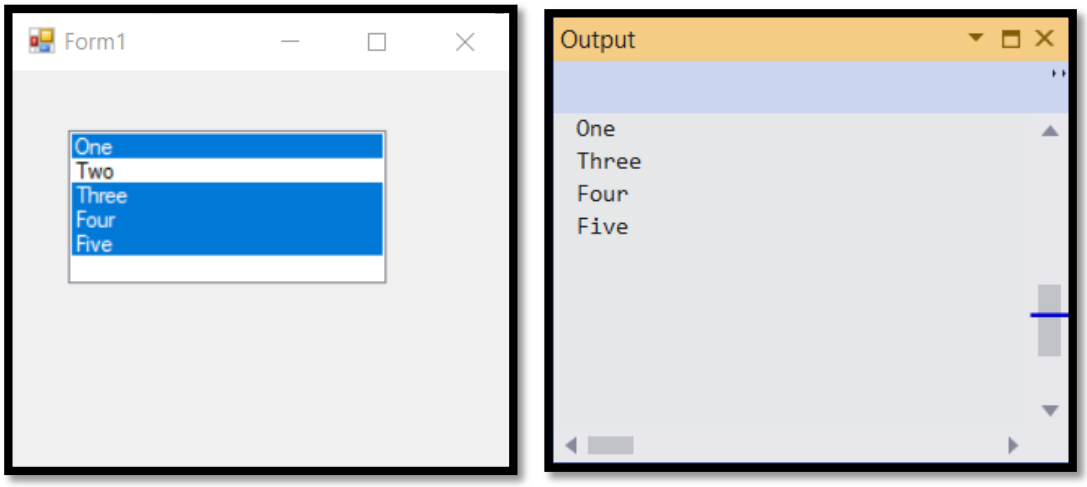
<u>สรุป</u> Items รายการตัวเลือก SelectedItems ก็คือ กลุ่มของ Items ที่ถูกเลือก ส่วน Items ที่ไม่เลือกจะไม่อยู่ในกลุ่มนี้ ดังนั้น SelectedItems จึงเป็นซับเซตของ Items เสมอ จากภาพ 2.28 จะเห็นว่า Items จะมี Index 5 ตำแหน่ง แต่ SelectedItems จะมี Index เพียง 3 ตำแหน่ง โดยค่าของ Index ใน Items และ SelectedItems จะอ้างอิงถึงรายการที่แตกต่างกัน เช่น Index ค่า 1 ใน Items คือรายการ Two แต่ค่าใน SelectedItems จะเป็นรายการ Three ซึ่งจุดนี้เป็นจุดที่ทำให้สับสนและงงกันมาก

<u>เพิ่มเติม</u> จาก Code เดิม ให้กำหนดคุณสมบัติของ SelectionMode ของลิสต์บ็อกซ์ในหน้าต่างคุณสมบัติเป็น MultiExtended จากนั้นไปที่หน้าต่าง Code Editor แล้วสร้างโค้ดอีเวนต์ SelectionIndexChanged ของ ListBox1 ให้ทำงานเมื่อผู้ใช้เลือกรายการโดยจะแสดงรายการที่ถูกเลือกด้วยคุณสมบัติ SelectedItem ดังนี้

```
Public Class Form1
    0 references
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
        ListBox1.Items.Add("One")
        ListBox1.Items.Add("Two")
        ListBox1.Items.Add("Three")
        ListBox1.Items.Add("Four")
        ListBox1.Items.Add("Five")
        Console.WriteLine(ListBox1.Items.Count)
    End Sub
    0 references
    Private Sub ListBox1_SelectedIndexChanged(sender As Object,
        e As EventArgs) Handles ListBox1.SelectedIndexChanged
        Dim i As Integer
        For i = 0 To ListBox1.SelectedItems.Count - 1
            Console.WriteLine(ListBox1.SelectedItems(i))
        Next i
    End Sub
End Class
```

ภาพที่ 2.29

เมื่อรันโปรแกรม จะแสดงรายการทั้งหมดที่ผู้ใช้เลือกบนหน้าต่าง Output



ภาพที่ 2.30

2.3.5.7 คอนโทรลคอมโบบ็อกซ์ (ComboBox)

คอมโบบ็อกซ์ จะรวมความสามารถของเท็กซ์บ็อกซ์และลิสต์บ็อกซ์เข้าไว้ด้วยกัน ทำให้เราสามารถเลือกรายการได้เช่นเดียวกับลิสต์บ็อกซ์ และแก้ไขรายการได้เหมือนกับเท็กซ์บ็อกซ์ แต่ใช้พื้นที่น้อยกว่าลิสต์บ็อกซ์ และใช้คำสั่งในการทำงานคล้ายกัน ดังนั้นเราสามารถนำความรู้เกี่ยวกับลิสต์บ็อกซ์มาใช้กับคอมโบบ็อกซ์ได้

การกำหนดค่ารูปแบบของคอมโบบ็อกซ์ เราจะกำหนดผ่านทางคุณสมบัติ DropDownStyle โดยคอมโบบ็อกซ์มีอยู่ 3 รูปแบบ ได้แก่

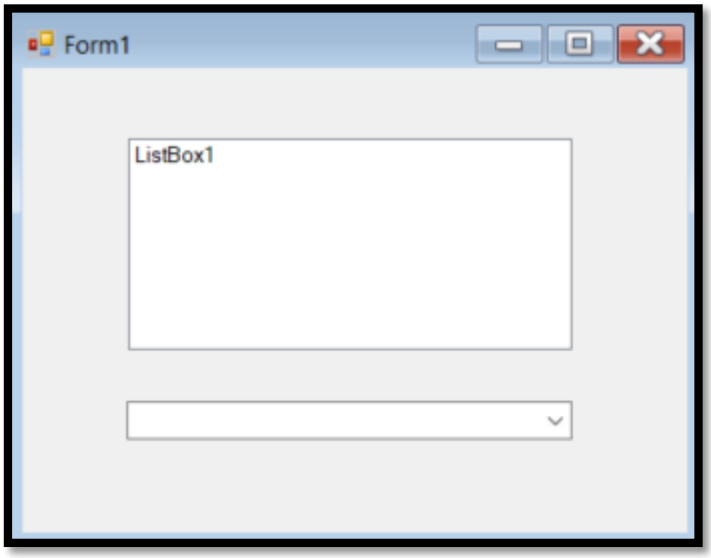
- ซิมเปิลคอมโบบ็อกซ์ (Simple ComboBox) เป็นคอมโบบ็อกซ์ที่แสดงรายการอยู่ตลอดเวลา และเมื่อจำนวนรายการเกินความสูงของคอนโทรล คอนโทรลจะแสดงสกอลบาร์แนวตั้ง เพื่อใช้เลื่อนดูรายการที่เหลือได้

- ดร็อปดาวน์คอมโบบ็อกซ์ (Dropdown ComboBox) เป็นคอมโบบ็อกซ์ที่ผู้ใช้สามารถใส่ค่าลงไปได้โดยตรง หรือเลือกจากรายการที่มีอยู่ก็ได้ โดยการคลิกที่ปุ่มลูกศรลง เมื่อเลือกตัวเลือกใดในรายการแล้ว ตัวเลือกนั้นจะแสดงอยู่ช่องบนสุดของคอนโทรล

- ดร็อปดาวน์ลิสต์บ็อกซ์ (Dropdown ListBox) เป็นคอมโบบ็อกซ์ที่คล้ายกับลิสต์บ็อกซ์ แต่จะประหยัดเนื้อที่ที่ใช้แสดงรายการมากกว่า โดยที่ผู้ใช้ไม่สามารถใส่ค่าเองได้ จะเลือกข้อมูลจากรายการที่มีอยู่ได้เท่านั้น

ตัวอย่างการใช้งานโปรแกรมลิสต์บ็อกซ์และคอมโบบ็อกซ์

1) สร้างโปรเจคใหม่ชนิด Windows Forms App ขึ้นมา และให้วางคอนโทรลต่างๆ ลงบนฟอร์ม พร้อมทั้งกำหนดคุณสมบัติต่างๆ ของฟอร์ม ดังภาพ



ภาพที่ 2.31

2) ดับเบิ้ลคลิกที่พื้นที่ว่างบนฟอร์ม จะได้ Events Load ขึ้นมา และเข้าสู่หน้า Code Editor จะทำการสร้างโค้ดอีเวนต์ Form1_Load เอาไว้ แล้วให้เราใส่ค่าเริ่มต้นของลิสต์บ็อกซ์และคอมโบบ็อกซ์เพิ่มเติม ดังนี้

```
    0 references
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ListBox1.Items.Add("One")
        ListBox1.Items.Add("Two")
        ListBox1.Items.Add("Three")
        ListBox1.Items.Add("Four")
        ListBox1.Items.Add("Five")
        ComboBox1.Items.Add("One")
        ComboBox1.Items.Add("Two")
        ComboBox1.Items.Add("Three")
        ComboBox1.Items.Add("Four")
        ComboBox1.Items.Add("Five")
    End Sub
End Class
```

ภาพที่ 2.32

3) คลิกที่ลิสต์บ็อกซ์ จากนั้นไปที่หน้าต่างคุณสมบัติและแท็บ Events ให้ดับเบิ้ลคลิ๊กที่อีเวนต์ SelectedIndexChanged จะเข้าสู่หน้าต่าง Code Editor สำหรับเขียนโปรแกรม จากนั้นให้ใส่โค้ดแสดงข้อความจากรายการที่ผู้ใช้เลือกในลิสต์บ็อกซ์ปรากฏในกล่องข้อความ

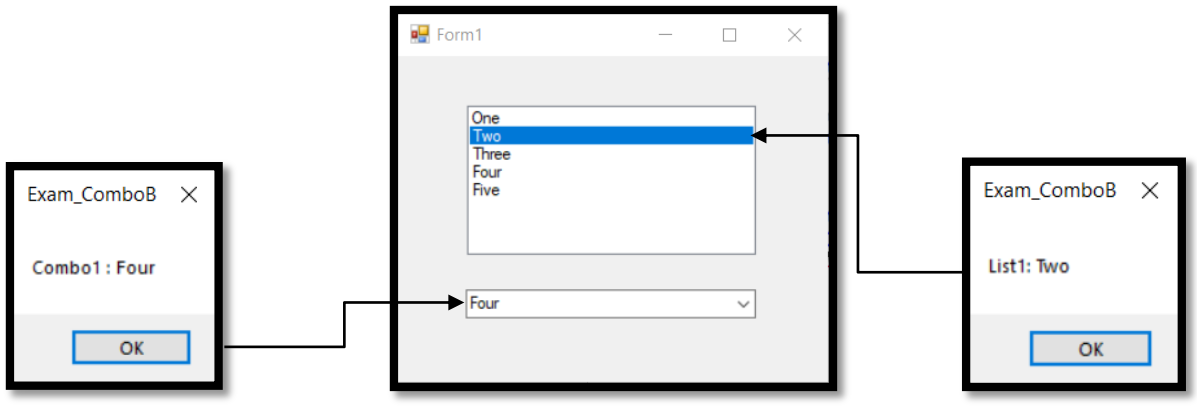
4) คลิกที่คอมโบบ็อกซ์ จากนั้นไปที่หน้าต่างคุณสมบัติและแท็บ Events ให้ดับเบิ้ลคลิกที่อีเวนต์ TextChanged จะเข้าสู่หน้าต่าง Code Editor สำหรับเขียนโปรแกรม จากนั้นให้ใส่โค้ดแสดงข้อความจากรายการที่ผู้ใช้เลือกในคอมโบบ็อกซ์ให้ปรากฏในกล่องข้อความ

```
    0 references
    Private Sub ListBox1_SelectedIndexChanged(sender As Object,
        e As EventArgs) Handles ListBox1.SelectedIndexChanged
        MsgBox("List1: " & ListBox1.Text, MsgBoxStyle.OkOnly)
    End Sub

    0 references
    Private Sub ComboBox1_TextChanged(sender As Object,
        e As EventArgs) Handles ComboBox1.TextChanged
        MsgBox("Combo1 : " & ComboBox1.Text, MsgBoxStyle.OkOnly)
    End Sub
End Class
```

ภาพที่ 2.33

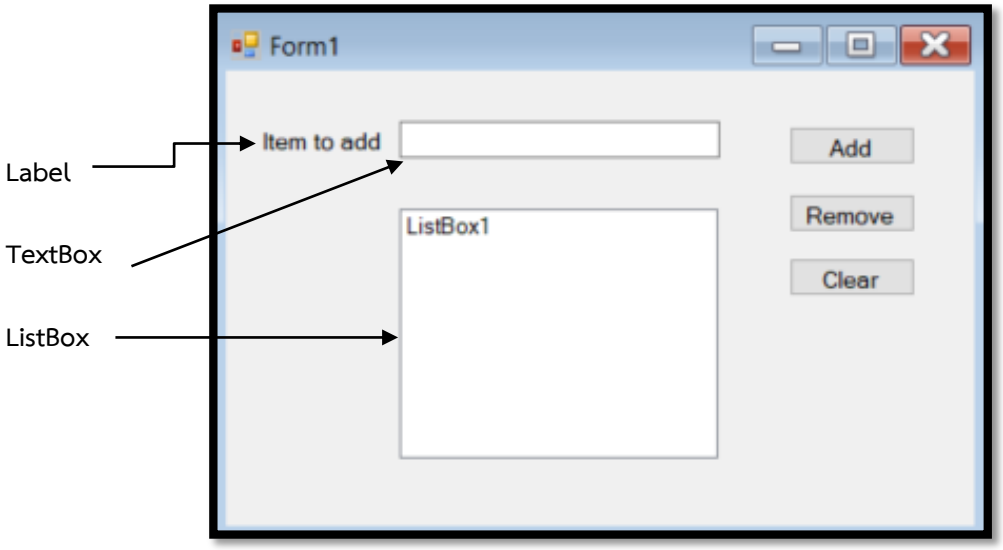
5) รันโปรแกรม จะแสดงผลลัพธ์ดังภาพ



ภาพที่ 2.34

ตัวอย่างโปรแกรมแสดงการเพิ่มและลบรายการในลิสต์บ็อกซ์/คอมโบบ็อกซ์

1) สร้างโปรเจคใหม่ชนิด Windows Forms App ขึ้นมา และให้วางคอนโทรลต่างๆ บนฟอร์ม พร้อมกำหนดคุณสมบัติ ดังภาพ



ภาพที่ 2.35

2) เขียนโปรแกรมให้กับปุ่ม Add โดยดับเบิลคลิกปุ่ม Add จะปรากฏหน้าต่าง Code Editor และเกิดอีเวนต์ Button1_Click จากนั้นให้ใส่โค้ดรับข้อมูลจากเท็กซ์บ็อกซ์มาเพิ่มเป็นรายการอยู่ในลิสต์บ็อกซ์

3) เขียนโปรแกรมให้กับปุ่ม Remove โดยดับเบิลคลิกปุ่ม Remove จะเกิดอีเวนต์ Button2_Click จากนั้นให้ใส่โค้ดตรวจสอบว่าผู้ใช้ได้เลือกรายการในลิสต์บ็อกซ์เพื่อจะลบออกหรือไม่ ถ้ามีจะถอดรายการที่เลือกออก ถ้าไม่มีจะแจ้งผ่านกล่องข้อความว่าไม่สามารถถอดรายการออกได้

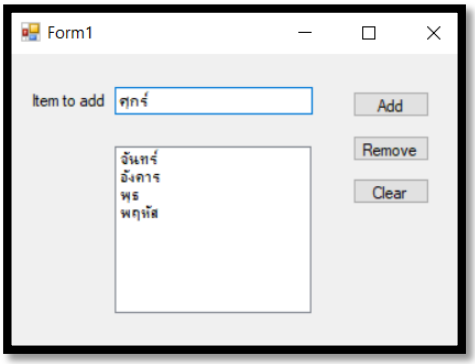
4) เขียนโปรแกรมให้กับปุ่ม Clear โดยดับเบิลคลิกปุ่ม Clear จะเกิดอีเวนต์ Button3_Click จากนั้นให้ใส่โค้ดลบรายการทั้งหมดออกจากลิสต์บ็อกซ์ ดังภาพ

```
Public Class Form1
    0 references
    Private Sub Button1_Click(sender As Object,
        e As EventArgs) Handles Button1.Click
        If TextBox1.Text <> "" Then         'เช็คว่ามีข้อความถูกพิมพ์หรือเปล่า
            ListBox1.Items.Add(TextBox1.Text) 'ถ้ามี เพิ่มข้อความจากเท็กซ์บ็อกซ์ไปที่ลิสต์บ็อกซ์
        Else
            MsgBox("Please input message!!!")  'ถ้าไม่มี ให้ขึ้นโชว์ข้อความ
        End If
        TextBox1.Text = ""          'ทำการเคลียร์ข้อความหลัง Add
        TextBox1.Focus()            'โฟกัสมายังTextBox
    End Sub
    0 references
    Private Sub Button2_Click(sender As Object,
        e As EventArgs) Handles Button2.Click
        If ListBox1.SelectedIndex > -1 Then
            'ตรวจสอบว่ามีรายการถูกเลือกอยู่หรือไม่
            'ถ้ามีรายการถูกเลือกอยู่ ลบรายการนั้นทิ้ง
            ListBox1.Items.RemoveAt(ListBox1.SelectedIndex)
        Else
            'ถ้าไม่มีรายการถูกเลือก แจ้งว่าลบไม่ได้
            MsgBox("Can't remove item from list.")
        End If
    End Sub

    0 references
    Private Sub Button3_Click(sender As Object,
        e As EventArgs) Handles Button3.Click
        ListBox1.Items.Clear()  'ลบรายการในลิสต์บ็อกซ์ทั้งหมด
    End Sub
End Class
```

ภาพที่ 2.36

5) ทำการคลิกรันโปรแกรม ทดสอบผลลัพธ์ที่ได้ดังภาพ



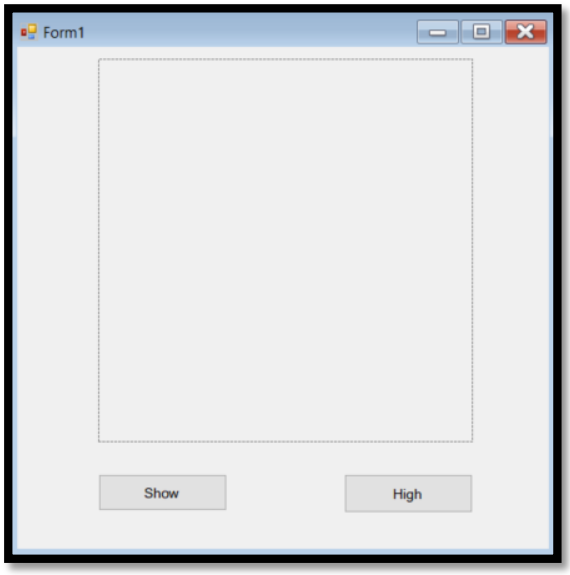
ภาพที่ 2.37

2.3.5.8 คอนโทรลพิกเจอร์บ็อกซ์ (PictureBox)

 เป็นคอนโทรลที่ใช้แสดงภาพและยังสร้างเป็นภาพแบ็กกราวด์ให้กับฟอร์มได้ด้วย ซึ่งการนำรูปภาพ ไอคอน หรือภาพสัญลักษณ์ต่างๆ มาใช้บนฟอร์มจะช่วยสื่อความหมายผู้ใช้งานได้ดียิ่งขึ้น

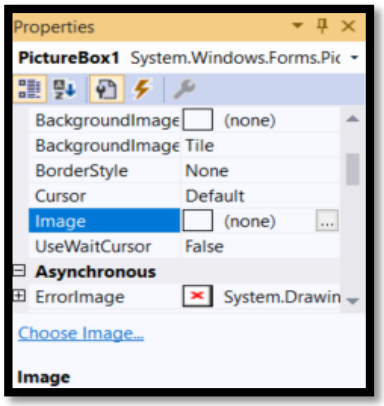
<u>ตัวอย่างโปรแกรมการเพิ่มและซ่อนภาพในพิกเจอร์บ็อกซ์</u>

1) สร้างโปรเจคชนิดใหม่เป็นชนิด Windows Forms App ขึ้นมา จากนั้นให้วางคอนโทรลพิกเจอร์บ็อกซ์และปุ่มบนฟอร์ม ตามภาพ



ภาพที่ 2.38

2) คลิกเลือกคอนโทรลพิกเจอร์ ไปที่ Properties และคลิกคุณสมบัติ Image เพื่อเข้าไปเลือกรูปภาพที่จะนำมาแสดงในคอนโทรลพิกเจอร์ โดยให้นำภาพมาเก็บไว้ใน Resources เพราะเมื่อเราบิลด์โปรแกรมจะแพ็กเกจไฟล์ภาพนี้ไปด้วย ทำให้สามารถรันโปรแกรมแดสงภาพบนเครื่องต่างๆ ได้อย่างถูกต้อง



ภาพที่ 2.39

3) เขียนโปรแกรมให้กับปุ่ม Show และ Hide

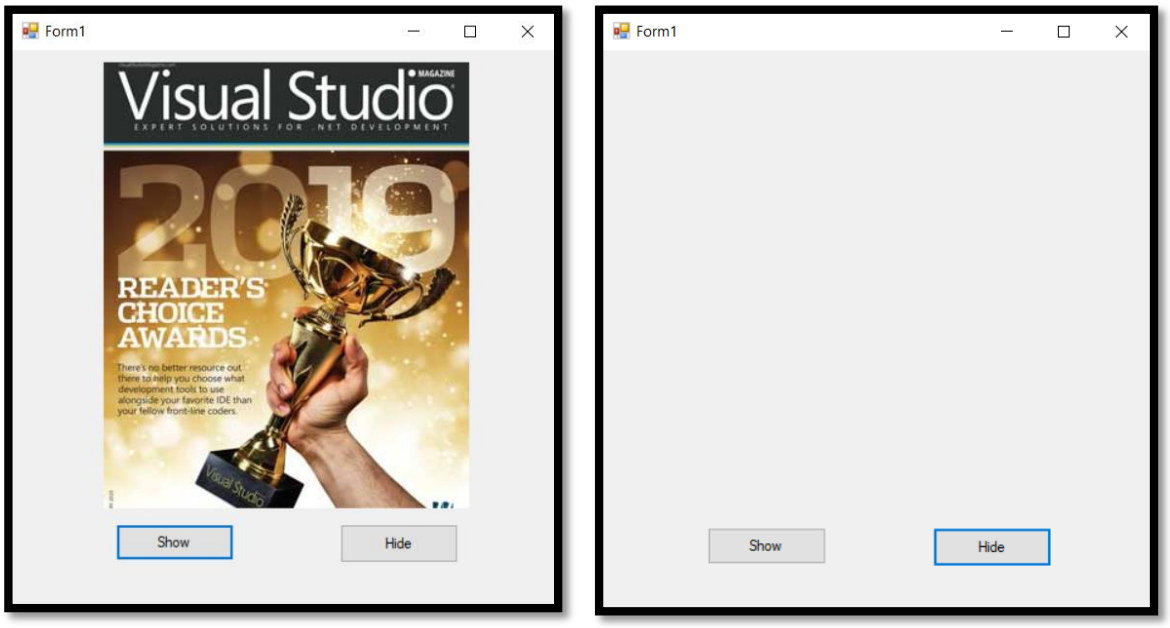
```
Public Class Form1

    0 references
    Private Sub Button1_Click(sender As Object, e As EventArgs) _
        Handles Button1.Click
        PictureBox1.Visible = True
    End Sub

    0 references
    Private Sub Button2_Click(sender As Object, e As EventArgs) _
        Handles Button2.Click
        PictureBox1.Visible = False
    End Sub
End Class
```

ภาพที่ 2.40

4) รันโปรแกรม จะได้ผลลัพธ์ตามภาพ



ภาพที่ 2.41

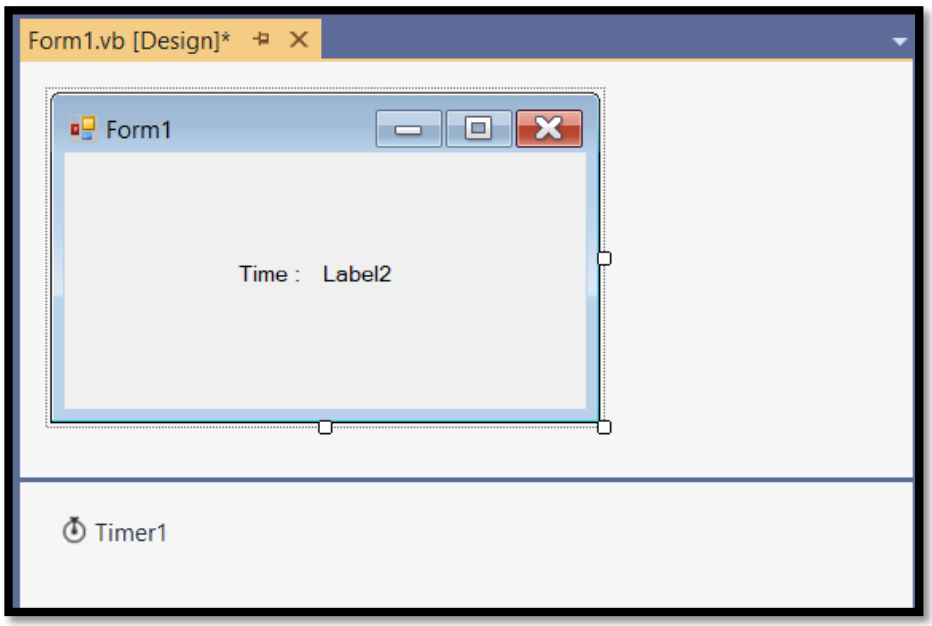
2.3.5.8 คอมโพเนนต์ไทเมอร์ (Timer)

การใช้คอมโพเนนต์ไทเมอร์จะทำให้มีการทำงานบางอย่างในทุกๆ ช่วงเวลาที่เรากำหนด ซึ่งจะช่วยให้เราสามารถทำอย่างอื่น โดยทิ้งโปรแกรมนี้ให้ทำงานตามเวลาที่ตั้งไว้ คอมโพเนนต์ไทเมอร์จัดเป็นคอนโทรลประเภทที่เรียกว่า Non Visual interface หมายถึง จะไม่เห็นตัวคอนโทรลในขณะที่รันโปรแกรมขึ้นมา ซึ่งจะพบได้บ่อยในการประยุกต์สร้างเกม แบบทดสอบ หรือการจำกัดเวลาในการแสดงหน้าจอการทำงานบางอย่าง

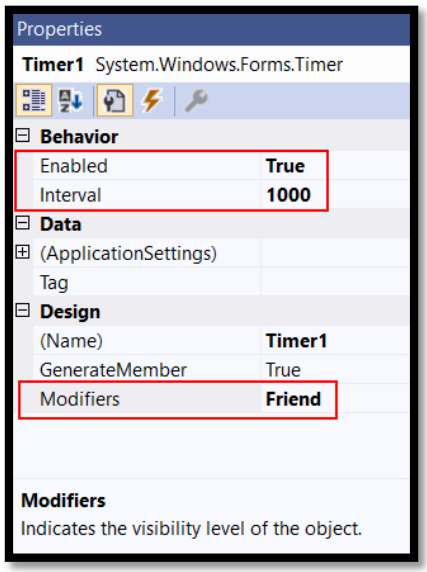
- อีเวนต์ที่สำคัญของ Timer คือ Tick เป็นอีเวนต์ที่เกิดขึ้นในทุกช่วงเวลาที่เราได้กำหนดไว้ในคุณสมบัติ Interval
- เมธอดของไทเมอร์ Start() คือเริ่มจับเวลา และ Stop() คือหยุดจับเวลา

<u>ตัวอย่างโปรแกรมการแสดงเวลาโดยใช้ไทเมอร์</u>

1) สร้างโปรเจ็กต์ใหม่ขึ้นมา จากนั้นวางคอนโทรลและคอมโพเนนต์บนฟอร์ม และกำหนดคุณสบัติของไทเมอร์ ดังภาพ



ภาพที่ 2.42



ภาพที่ 2.43

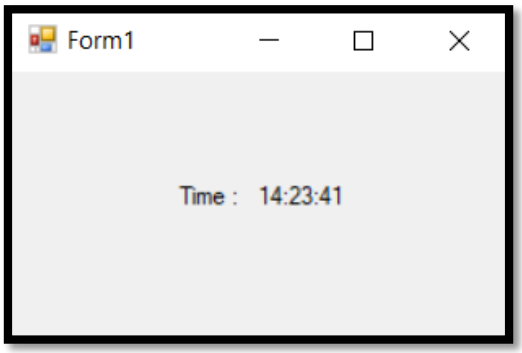
2) ดับเบิ้ลคลิกที่คอมโพเนนต์ Timer เข้าสู่หน้าต่าง Code Editor ใส่คำสั่งดังนี้

```
    0 references
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) _
        Handles Timer1.Tick    'จะทำงานทุกๆ 1 วินาทีตามค่าที่กำหนดใน Interval = 1000
        Label2.Text = My.Computer.Clock.LocalTime.ToLongTimeString
    End Sub
End Class
```

ภาพที่ 2.44

My.Computer.Clock.LocalTime.ToLongTimeString คือ การดึงเวลาปัจจุบันบนคอมพิวเตอร์เครื่องนั้นออกมาอยู่ในรูปของข้อความ

3) คลิกรันโปรแกรมจะได้ผลลัพธ์ตามภาพ

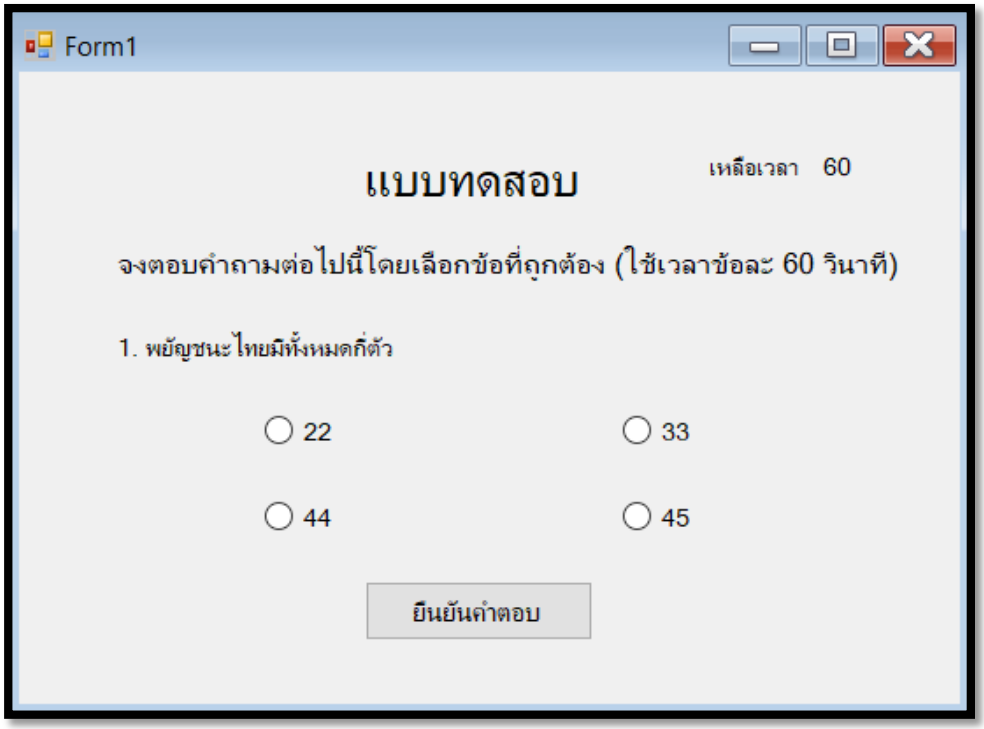


ภาพที่ 2.45

ตัวอย่างการทำแบบทดสอบแบบจับเวลา

โปรแกรมต่อไปนี้จะแสดงการใช้งานคอมโพเนนต์ไทเมอร์ โดยจะใช้เมธอด Start() เริ่มจับเวลา เมื่อมีการตอบคำถาม และเมธอด Stop() จะหยุดการจับเวลา และหากไม่สามารถตอบในเวลาที่กำหนด จะเรียกใช้เมธอด Stop() เพื่อหยุดการจับเวลา

1) สร้างโปรเจ็กต์ใหม่ขึ้นมา จากนั้นวางคอนโทรลและคอมโพเนนต์ต่างๆ ดังภาพ



ภาพที่ 2.46

2) ไปที่หน้าต่าง Code Editor กำหนดตัวแปร count = 60 เป็นเวลาในการทำแบบทดสอบหนึ่งข้อ และตัวแปล time เก็บจำนวนเวลาที่ใช้ในการทำแบบทดสอบ

3) คลิกพื้นที่ว่างบนฟอร์ม จากนั้นไปที่หน้าต่างคุณสมบัติและแท็บ Events ให้ดับเบิลคลิกี่อีเวนต์ Load จะเข้าสู่หน้าต่าง Code Editor สำหรับเขียนโค้ดการทำงานในหน้า Load ของ Form1 โดยให้กำหนด Interval ค่าการจับเวลาและเรียกใช้เมธอด Start() ให้ไทเมอร์เริ่มจับเวลา

```
Public Class Form1
    Private count As Integer = 60
    Private time As Integer
    0 references
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Timer1.Interval = 1000
        Timer1.Start()
    End Sub
End Class
```

ภาพที่ 2.47

4) ดับเบิลคลิกที่ปุ่มกด จะแสดงหน้าต่าง Code Editor จากนั้นแทรกโค้ดหยุดนับเวลา ตรวจคำตอบ แสดงผลคำตอบและเวลาที่ใช้ไป

5) ดับเบิลคลิกที่คอมโพเนนต์ไทเมอร์ในกรอบด้านล่าง จะแสดงหน้าต่าง Code Editor และเกิดอีเวนต์ Timer1_Tick จากนั้นให้แทรกโค้ดสำหรับแสดงการนับเวลาถอยหลัง หยุดจับเวลาเมื่อหมดเวลา และแสดงกล่องข้อความแจ้งว่าผู้ใช้ไม่สามารถตอบคำถามในเวลาที่กำหนด

```
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Timer1.Stop()
        If (RadioButton2.Checked) Then  'เมื่อตอบถูก
            time = 60 - count
            MsgBox("คำตอบถูกต้อง คุณใช้เวลาตอบ" & time & "วินาที")
        Else
            MsgBox("คุณตอบผิด")      'เมื่อตอบผิด
        End If
    End Sub

    0 references
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        Label5.Text = count     'แสดงตัวเลขการนับเวลาที่คอนโทรล Label5 เป็นตัวอักษร
        count -= 1              'ลดการนับเวลาลง 1
        If (count < 0) Then
            Timer1.Stop()        'หยุดการนับเวลา
            MsgBox("คุณไม่สามารถตอบในเวลาที่กำหนด")     'แจ้งเมื่อตอบไม่ทันในเวลาที่กำหนด
        End If
    End Sub
End Class
```

ภาพที่ 2.48

6) รันเพื่อทดสอบโปรแกรม

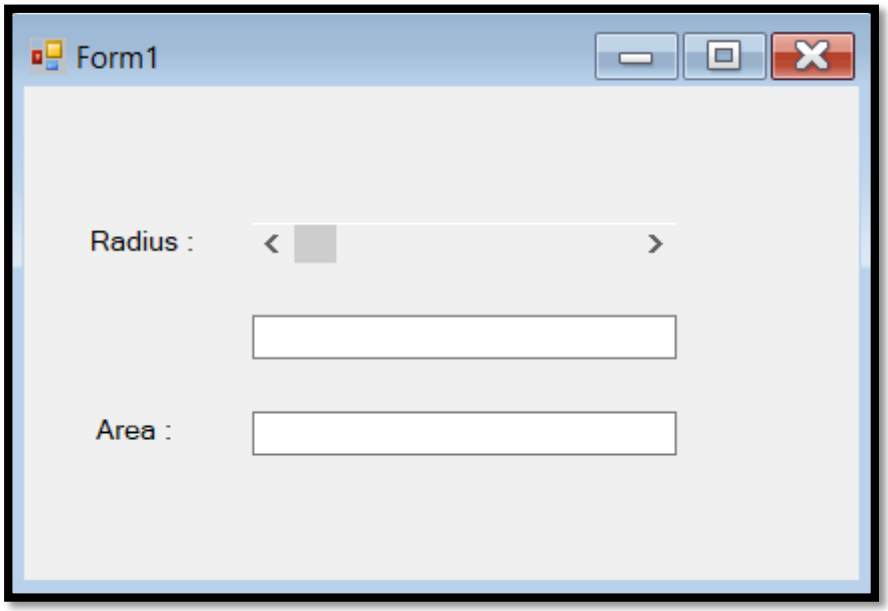
2.3.5.9 คอนโทรลสกรอลบาร์ (ScrollBar)

สกรอลบาร์จะใช้ในโอกาสที่ไม่สามารถแสดงผลข้อมูลทั้งหมดได้ในคราวเดียว เนื่องจากมีพื้นที่จำกัด หรืออาจจะใช้เป็นอินพุตในการปรับค่าข้อมูลก็ได้ ซึ่งคอนโทรลสกรอลบาร์จะมีทั้งในแนวนอนและแนวตั้ง ดังนี้

<u>ตัวอย่างโปรแกรมแสดงการใช้งานสกรอลบาร์</u>

ตัวอย่างโปรแกรมนี้จะแสดงการใช้งานคอนโทรลสกรอลบาร์ โดยจะรับค่ารัศมีวงกลมผ่านทางสกรอลบาร์แล้วแสดงค่ารัศมี และคำนวณพื้นที่วงกลมมาทางเท็กซ์บ็อกซ์

1) สร้างโปรเจ็กต์ใหม่ขึ้นมา จากนั้นวางคอนโทรลต่างๆ ดังต่อไปนี้



ภาพที่ 2.49

2) ดับเบิ้ลคลิกที่สกรอลบาร์ จะเข้าสู่หน้าต่าง Code Editor สำหรับเขียนโปรแกรม และแทรกโค้ดดังต่อไปนี้

```
Public Class Form1
    0 references
    Private Sub HScrollBar1_Scroll(sender As Object, e As ScrollEventArgs) _
        Handles HScrollBar1.Scroll
        TextBox1.Text = Format(HScrollBar1.Value, "###.00") 'แสดงรัศมีจากการเลื่อนสกรอบาร์
    End Sub
End Class
```

ภาพที่ 2.50

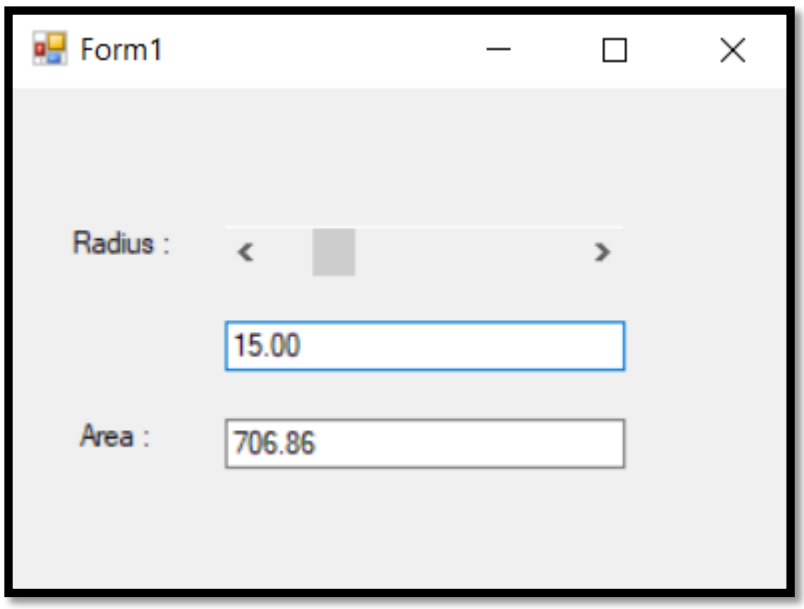
3) คลิกที่คอนโทรล HScrollBar1 จากนั้นไปที่ Properties และดูที่แท็บ Events ให้ดับเบิ้ลคลิกที่อีเวนต์ ValueChanged จะเข้าสู่หน้าต่าง Code Editor สำหรับเขียนโปรแกรม โดยให้แทรกโค้ดนำค่ารัศมีจากการปรับสกอรลบาร์มาคำนวณหาพื้นที่ของวงกลมและแสดงผลใน TextBox2

```
Public Class Form1
    0 references
    Private Sub HScrollBar1_Scroll(sender As Object, e As ScrollEventArgs) _
        Handles HScrollBar1.Scroll
        TextBox1.Text = Format(HScrollBar1.Value, "###.00") 'แสดงรัศมีจากการเลื่อนสกรอบาร์
    End Sub

    0 references
    Private Sub HScrollBar1_ValueChanged(sender As Object, e As EventArgs) _
        Handles HScrollBar1.ValueChanged 'เมื่อค่าถูกเปลี่ยน
        TextBox2.Text = Format((HScrollBar1.Value ^ 2) * Math.PI, "###.00") 'แสดงพื้นที่วงกลม
    End Sub
End Class
```

ภาพที่ 2.51

4) รันโปรแกรมจะได้ผลลัพธ์ดังภาพ



ภาพที่ 2.52