**Week 09 การเขียนโปรแกรมกราฟฟิกส์ด้วย GDI+**

**1. กล่าวนำ**

ใบงานนี้ มีวัตถุประสงค์ เพื่อให้นักศึกษา ได้รู้จักกับ GDI+ ซึ่งจะช่วยให้นักษาสามารถ อธิบายระบบพิกัดใน GDI+ ได้ สามารถใช้คำสั่งพื้นฐานของ GDI+ ในการวาดกราฟฟิกส์อย่างง่ายได้

**2. เนื้อหา**

**2.1. เกี่ยวกับ GDI+**

GDI+ (Graphics Device Interface Plus) เป็นกราฟฟิกส์ไลบรารี่ ซึ่ง Microsoft จัดเตรียมไว้ให้นักพัฒนาโปรแกรมได้ใช้ทำงานด้านกราฟฟิกส์ โดยที่ GDI+ จะมีลักษณะเป็น OOP เต็มตัว

ก่อนที่จะมี GDI+ ไมโครซอฟท์มีไลบรารี่ชื่อ GDI ซึ่งอยู่ในไฟล์ gdi32.dll ติดตั้งมาบน Windows ทุกตัวจนกระทั่งมาถึงยุค Windows XP ทางไมโครซอฟท์ก็ได้ปรับปรุง GDI ให้มีความสามารถมากขึ้น โดยให้ชื่อว่า GDI+ และยังคงใช้ต่อมาจนถึงวินโดวส์รุ่นปัจจุบัน

**2.2. ความสามารถของ GDI+**

ความสามารถของ GDI+ จะมีอยู่ 3 ส่วนหลักๆ คือ

* Vector graphics เป็นการสร้างกราฟฟิกส์ 2 มิติ ด้วยรูปทรงเรขาคณิตพื้นฐาน เช่น เส้นตรง(Line) เส้นโค้ง (Curve) สี่เหลี่ยม(Rectangle) วงรี (Ellipse) เป็นต้น
* Imaging นอกจากความสามารถในการวาดภาพด้วยรูปทรงเรขาคณิตแล้ว GDI+ ยังมีความสามารถในด้านการ แสดงผล, Load/Save ไฟล์ภาพชนิดต่างๆ ในรูปของ bitmap (bitmap ในที่นี้หมายถึงไฟล์ภาพทุกชนิด ที่แสดงเป็น pixel และไม่สามารถอธิบายด้วยรูปทรงเรขาคณิตอย่างง่ายๆ ได้ เช่น ภาพถ่าย ภาพสแกน เป็นต้น)
* Typology เป็นการจัดการเกี่ยวกับฟอนต์ ซึ่งมีความสามารถในการแสดงผลแบบ anti aliasing ทำให้ขอบของตัวอักษรดูเรียบขึ้นมากกว่าการเขียนโปรแกรมด้วย GDI

**2.3. ระบบพิกัดใน GDI+**

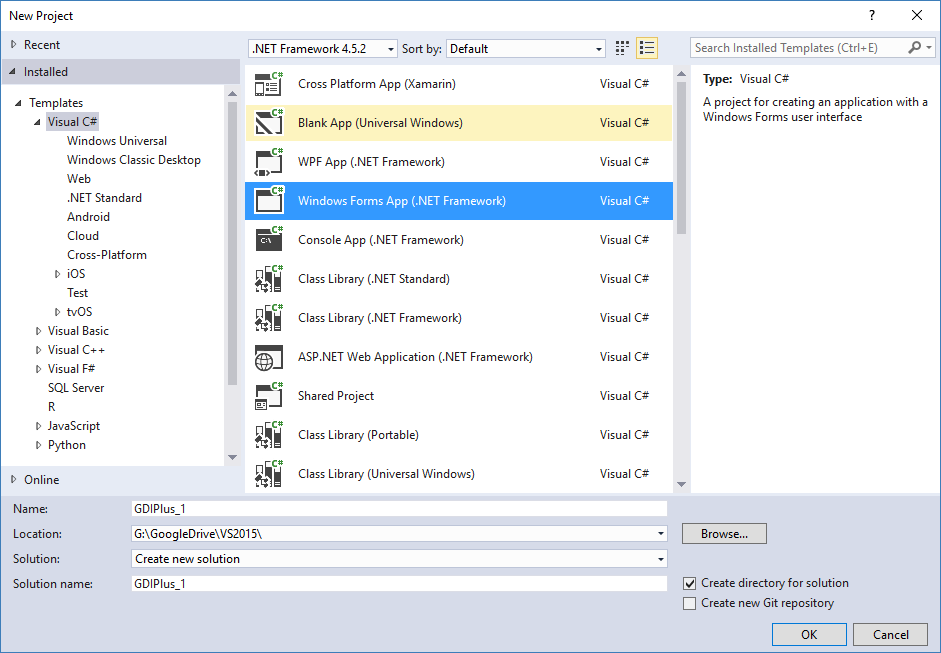
ใน GDI+ จะมีระบบพิกัดที่ใช้งานอยู่เป็นจำนวน 3 ระบบ คือ World coordinate, Page coordinate และ Device coordinate

* World coordinate จะทำงานกับระบบหน่วยที่ใช้ในชีวิตประจำวัน เช่น หน่วยมิลลิเมตร หรือ หน่วยนิ้ว เป็นต้น
* Page coordinate จะทำงานกับระบบ coordinate บนหน้ากระดาษ และจะสามารถพิมพ์ออกทางเครื่องพิมพ์ตามขนาดที่กำหนดบน page coordinate
* Device coordinate จะทำงานกับ pixel บนจอหรือบนเครื่องพิมพ์

**3. การทดลอง**

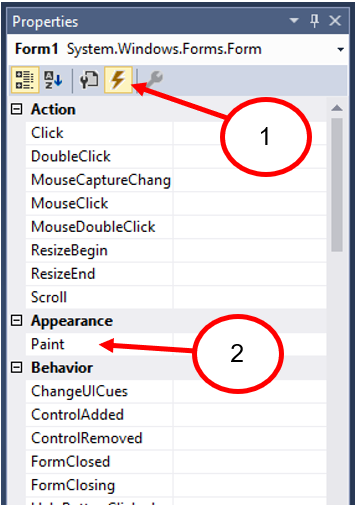
**เริ่มต้นกับ C# และ GDI+**

1. เรียกโปรแกรม Visual Studio
2. สร้าง Project ใหม่ เป็นชนิด C# โดยมีชื่อ project คือ GDIPlus\_1

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-01.png)

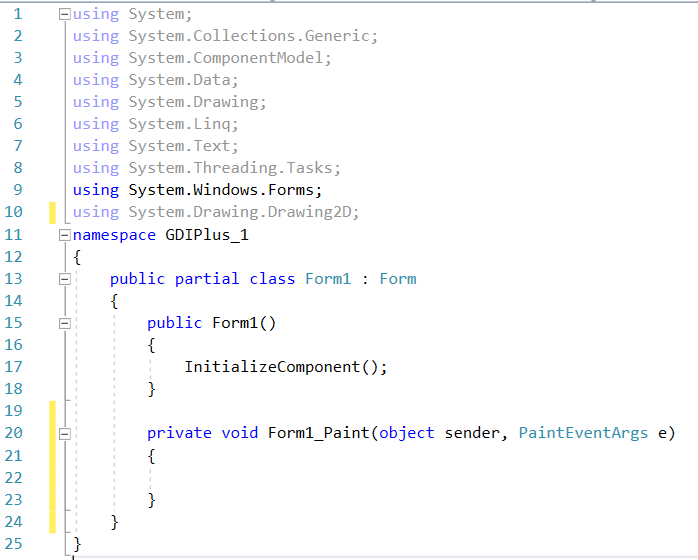
**รูปที่ 1** การสร้าง Project ชนิด Windows Form App (.NET Framework)

1. เมื่อ Wizard สร้าง Project เสร็จแล้ว จะนำเรามาที่หน้าต่าง Form1.cs[Design] ให้คลิกที่ปุ่ม Events ของ Properties pane ตาม (1) และ Double click ที่ Paint ตาม (2)

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-02.png)

**รูปที่ 2** การเพิ่ม event ให้กับ Paint

1. เพื่อให้โปรแกรมของเราสามารถใช้งาน GDI+ ในการวาดภาพ 2D ได้ ให้ทำการเพิ่ม “using System.Drawing.Drawing2D;” ลงในบรรทัดที่ 10 ดังรูป

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-03.png)

**รูปที่ 3** การเพิ่ม using System.Drawing.Drawing2D;

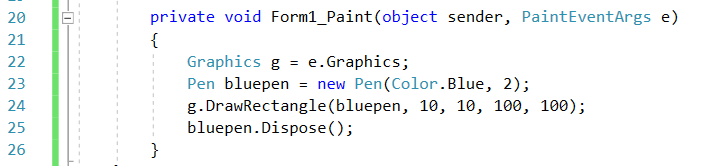
1. ให้เพิ่มบรรทัดต่อไปนี้ลงในฟังก์ชัน private void Form1\_Paint(object sender, PaintEventArgs e)

5.1 สร้าง Object ของกราฟิกส์ โดยคำสั่ง Graphics g = e.Graphics; ซึ่ง Object ชื่อ e ถูกส่งผ่านมาทาง argument ของฟังก์ชัน

5.2 เพิ่มออบเจกต์ของปากกา สีน้ำเงินขนาด 2 พิกเซล ด้วยคำสั่ง Pen = new Pen(Color.Blue, 2);

5.3 วาดสี่เหลี่ยมด้วยคำสั่ง g.DrawRectangle(bluepen, 10, 10, 100, 100);

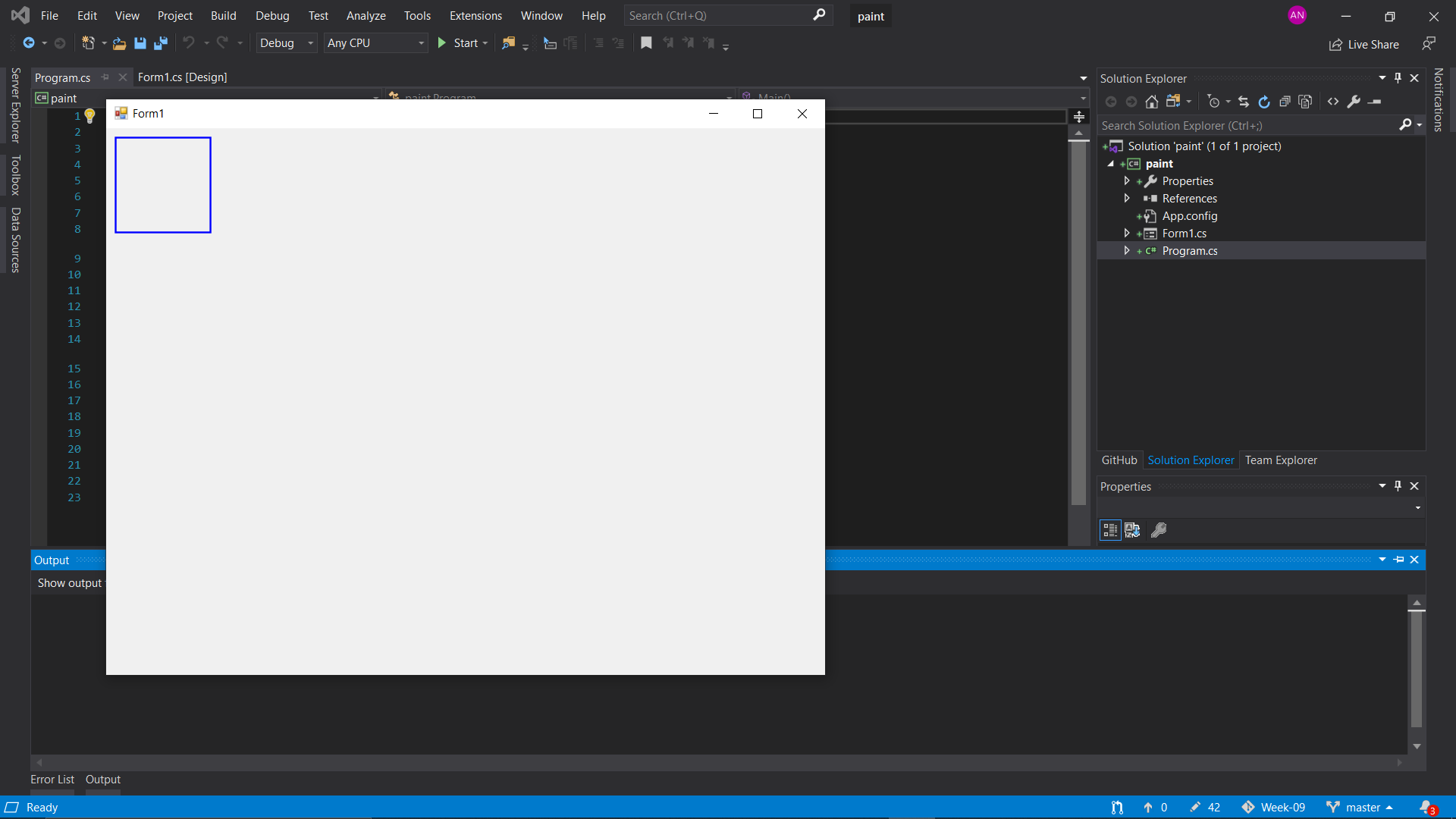
5.4 คืนหน่วยความจำให้ระบบโดยการลบออบเจ็กต์ปากกาสีน้ำเงิน โดยคำสั่ง bluepen.Dispose();

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-04.png)

**รูปที่ 4** ตัวอย่างการแก้ไขเมธอด Form1\_Paint

1. ทดลอง Build และ Run โปรแกรมและบันทึกผล

บันทึกผลการทดลอง

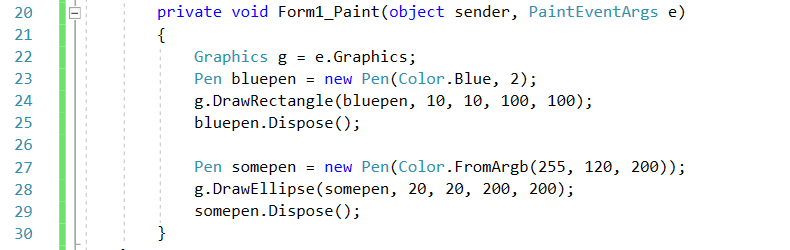


มีสี่เหลี่ยมสีฟ้าข้นมา

**3.1 การใช้สี**

**3.1.1 การใช้สีโดยการผสมค่าสี**

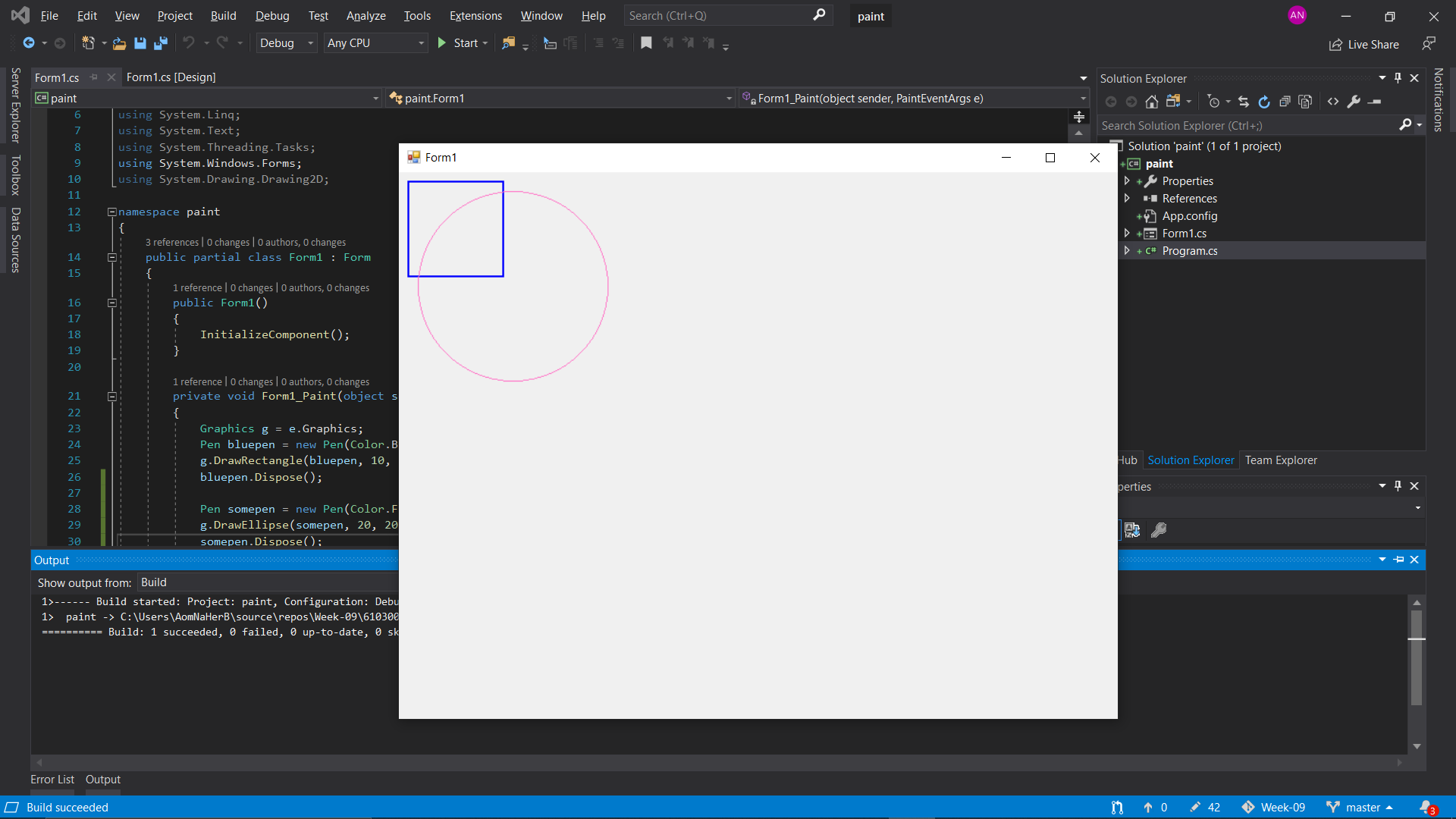
1. เพิ่ม Code ต่อไปนี้ลงในฟังก์ชัน private void Form1\_Paint(object sender, PaintEventArgs e)

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-05.png)

**รูปที่ 5** การใช้สีโดยการผสมค่าสี

1. รันโปรแกรมและบันทึกผล

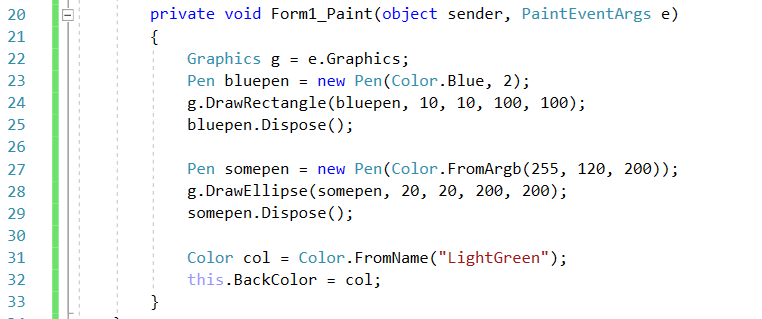
บันทึกผลการทดลอง



มีวงกลมขึ้นมาตรงสี่เหลี่ยมสีออกแดงๆ

**3.1.2 การใช้สี โดยการใช้เมธอด FromName**

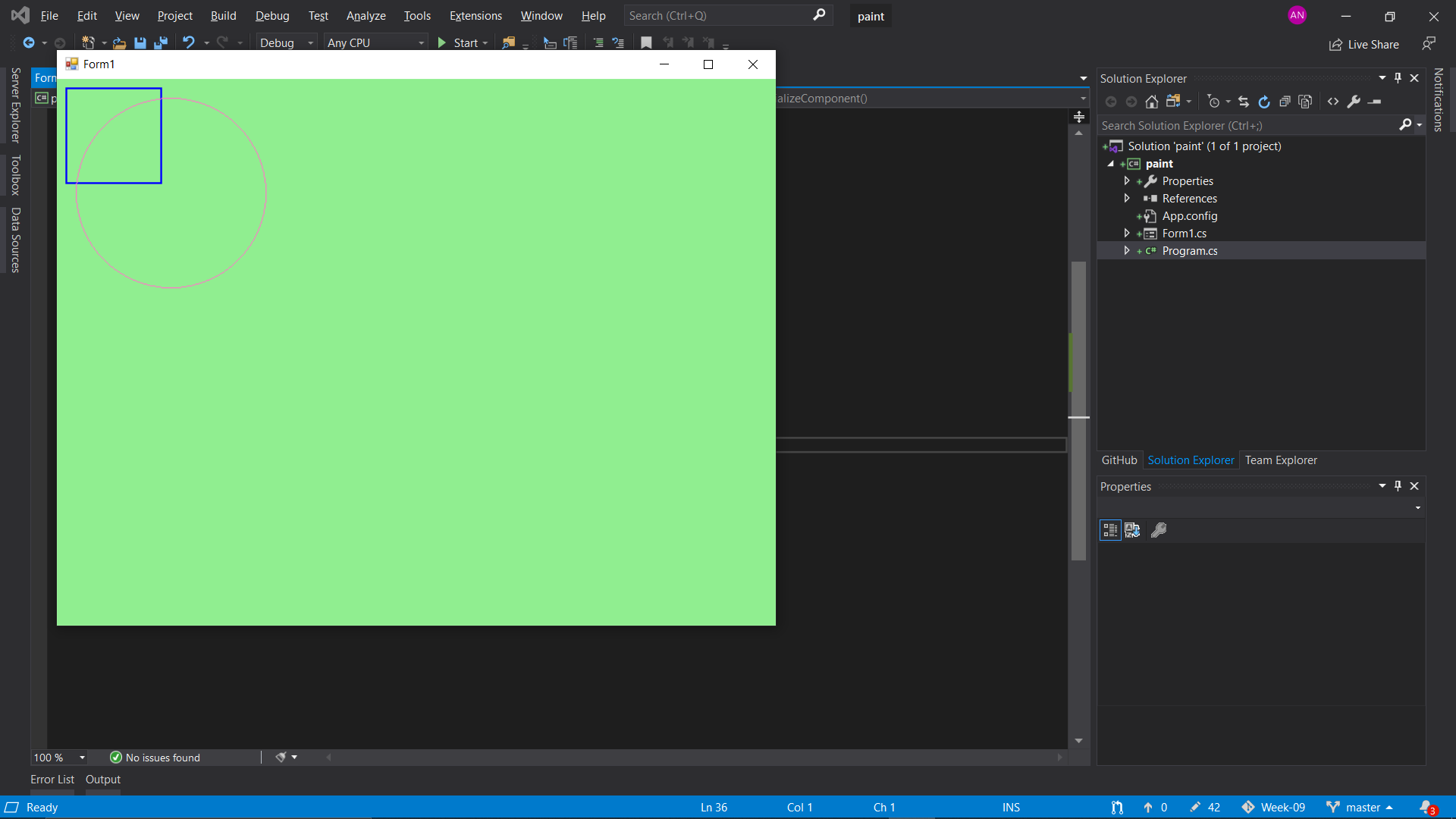
1. เพิ่ม Code ต่อไปนี้ลงในฟังก์ชัน private void Form1\_Paint(object sender, PaintEventArgs e)

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-06.png)

**รูปที่ 6** การใช้สี โดยการใช้เมธอด FromName

1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง

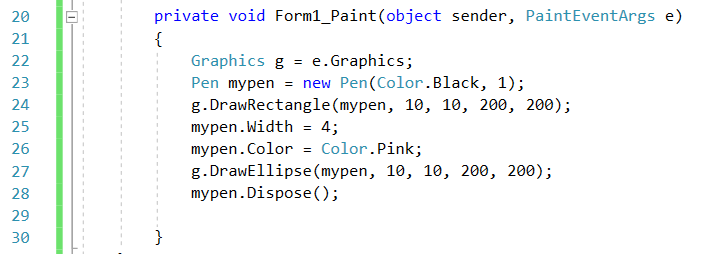


พื้นหลังเป็นสีเขียว

**3.2 การใช้ปากกา**

**3.2.1 ขนาดและสีของปากกา**

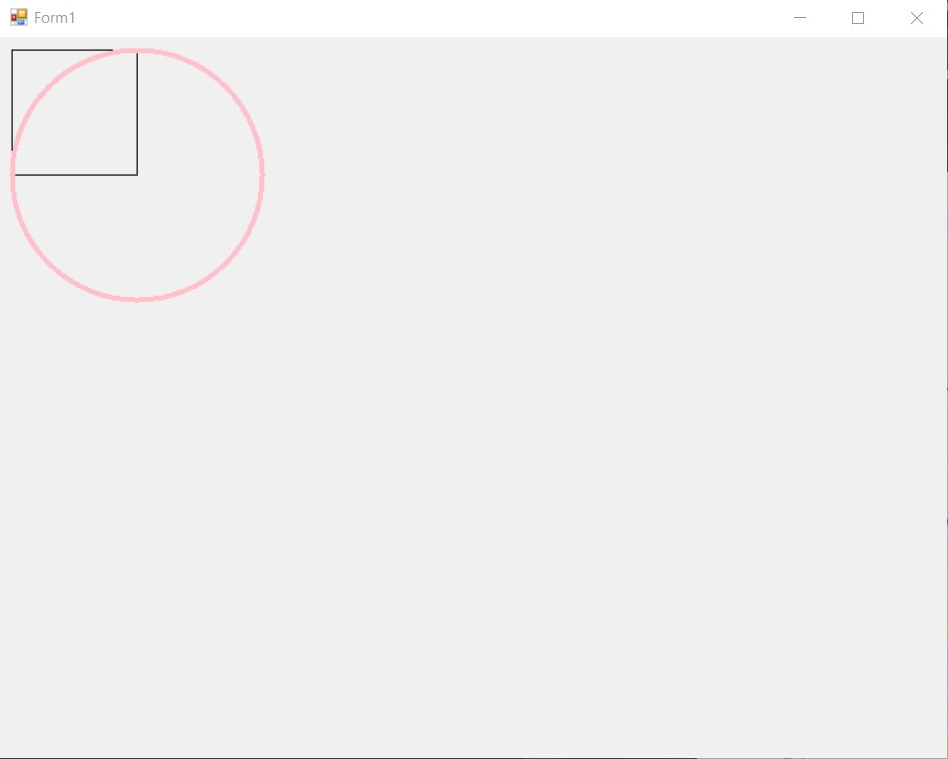
1. การทดลองเปลี่ยนขนาดและสีของปากกา โดยใช้ properties Width และ Color

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-07.png)

**รูปที่ 7** ขนาดและสีของปากกา

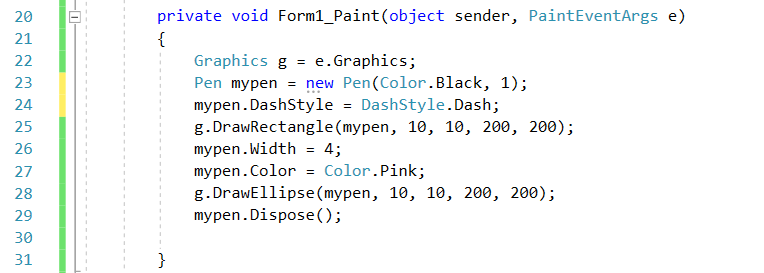
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.2.2 ชนิดของปากกาเป็นเส้นประ**

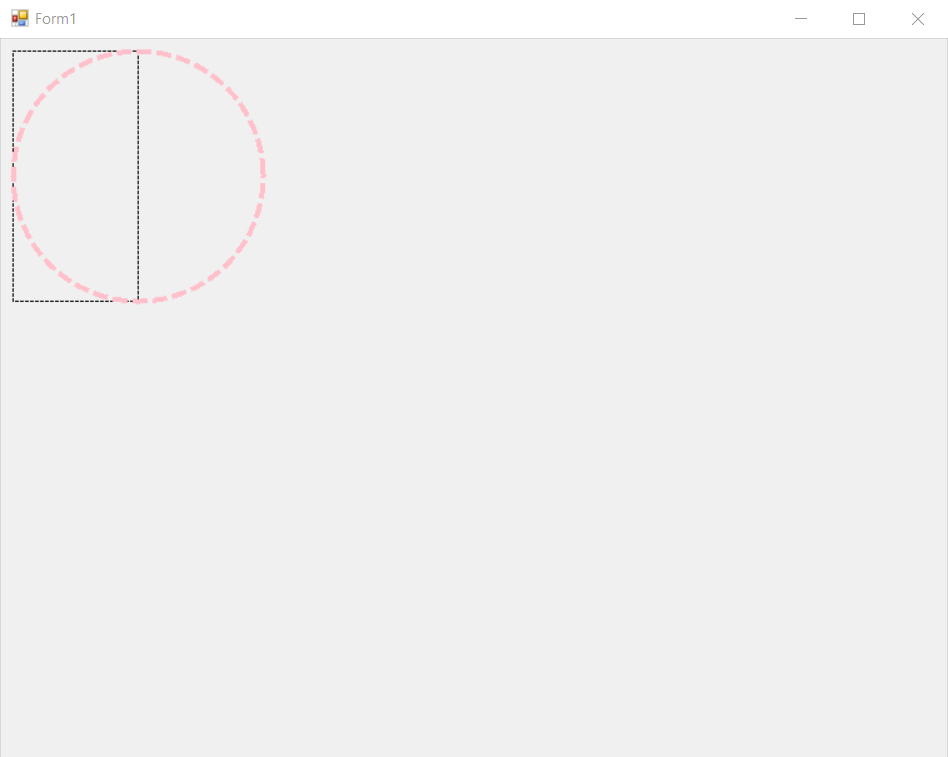
1. เปลี่ยนชนิดของปากกาเป็นเส้นประ

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-08.png)

**รูปที่ 8** เปลี่ยนชนิดของปากกาเป็นเส้นประ

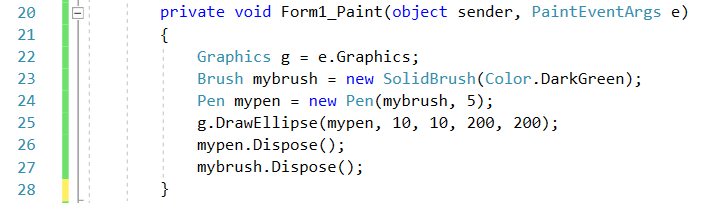
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.2.3 การใช้ปากกา (Pen) ร่วมกับแปรงทาสี (Brush)**

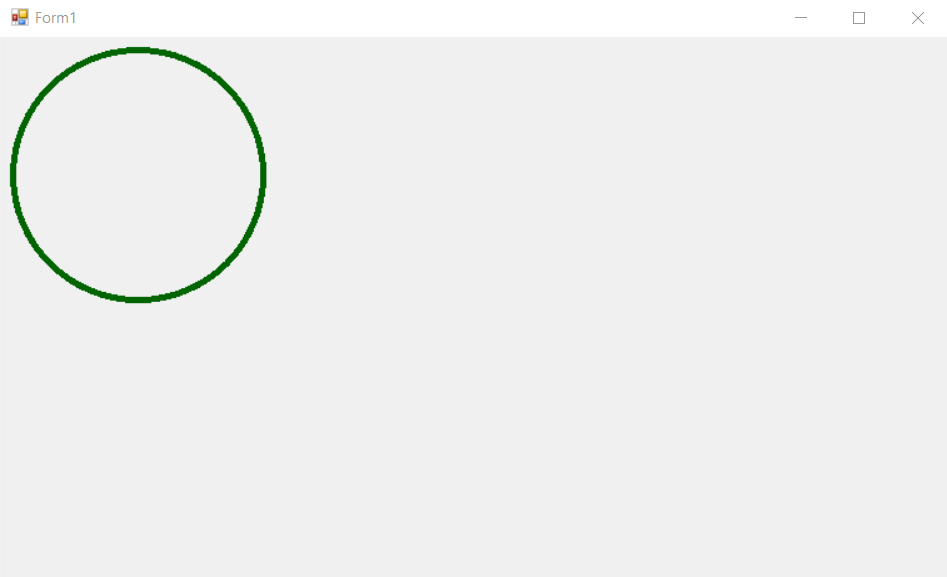
1. เขียนโปรแกรมใช้ปากการ่วมกับแปรงทาสี

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-09.png)

**รูปที่ 9** การใช้ปากการ่วมกับแปรงทาสี

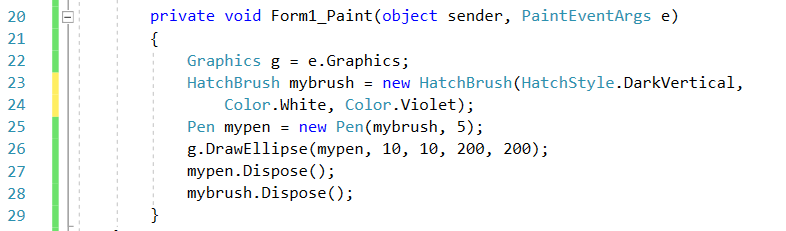
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.2.4 ใช้ Pen ร่วมกับ HatchBrush เพื่อสร้างลายเส้นแบบต่างๆ**

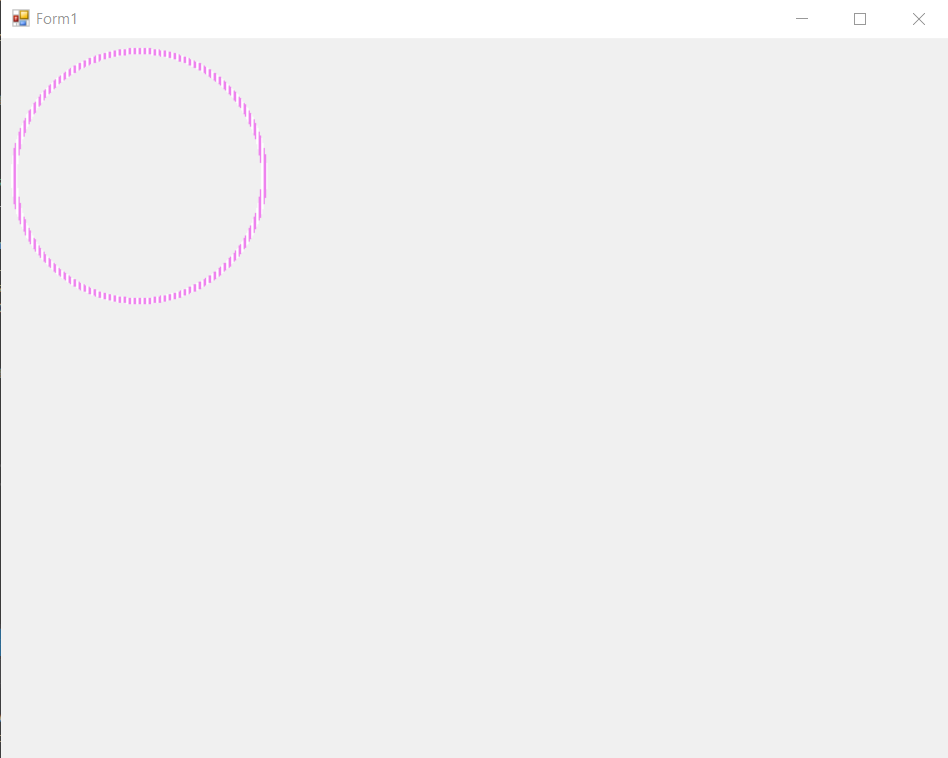
1. ใช้ Pen ร่วมกับ HatchBrush เพื่อสร้างลายเส้นแบบต่างๆ

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-10.png)

**รูปที่ 10** ใช้ Pen ร่วมกับ HatchBrush เพื่อสร้างลายเส้นแบบต่างๆ

1. รันโปรแกรมและบันทึกผล

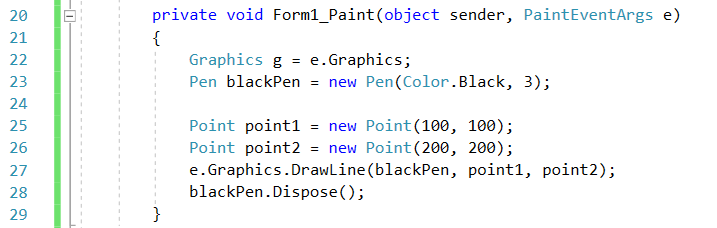
บันทึกผลการทดลอง



**3.3 การวาดเส้นตรง**

**3.3.1 การวาดเส้นตรงเชื่อมต่อระหว่างจุด จำนวน 2 จุด**

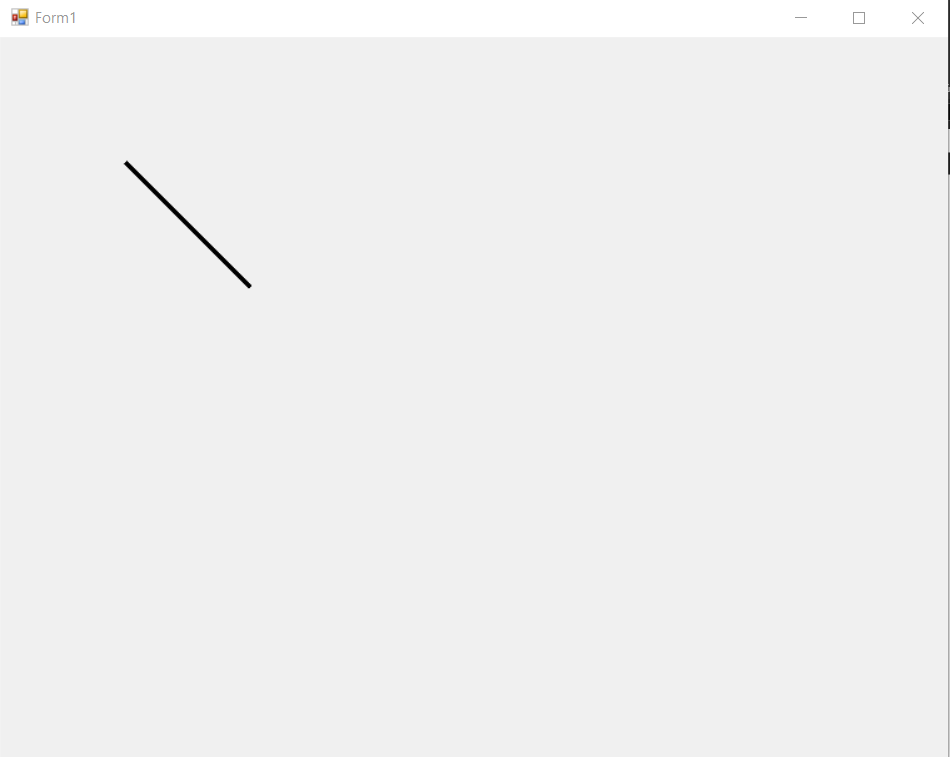
1. การวาดเส้นตรง เป็นการเชื่อมต่อระหว่างจุด จำนวน 2 จุด โดยใช้ออบเจกต์ Pen เป็นตัวกำหนดลักษณะของเส้น ให้แก้ไข code ต่อไปนี้ในฟังก์ชัน private void Form1\_Paint(object sender, PaintEventArgs e)

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-11.png)

**รูปที่ 11** การวาดเส้นตรงเชื่อมต่อระหว่างจุด จำนวน 2 จุด

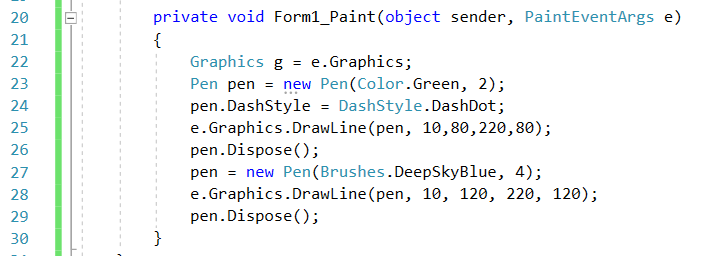
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.3.2 การวาดเส้นตรงด้วย pen style และ brush**

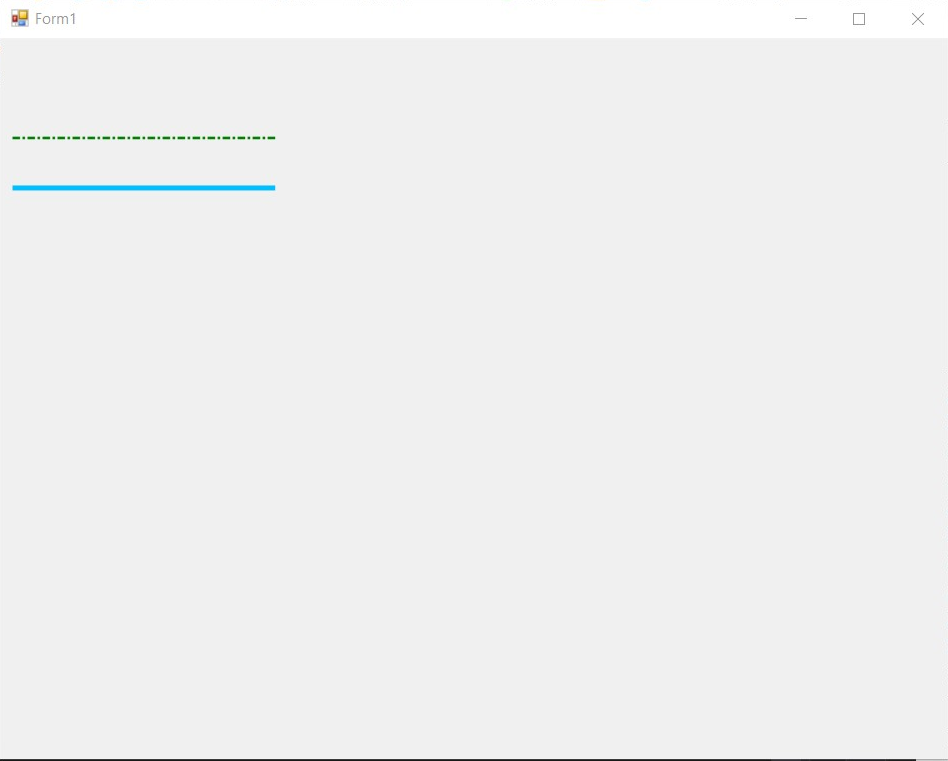
1. การวาดเส้นตรงด้วย pen style และ brush

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-12.png)

**รูปที่ 12** การวาดเส้นตรงด้วย pen style และ brush

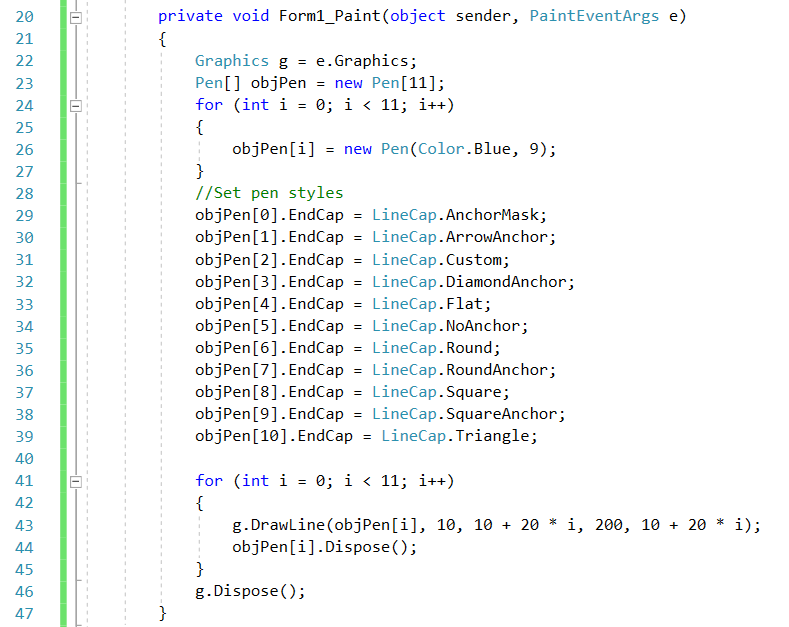
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.3.3 การกำหนดจุดปลายของเส้นตรงด้วย style แบบต่างๆ**

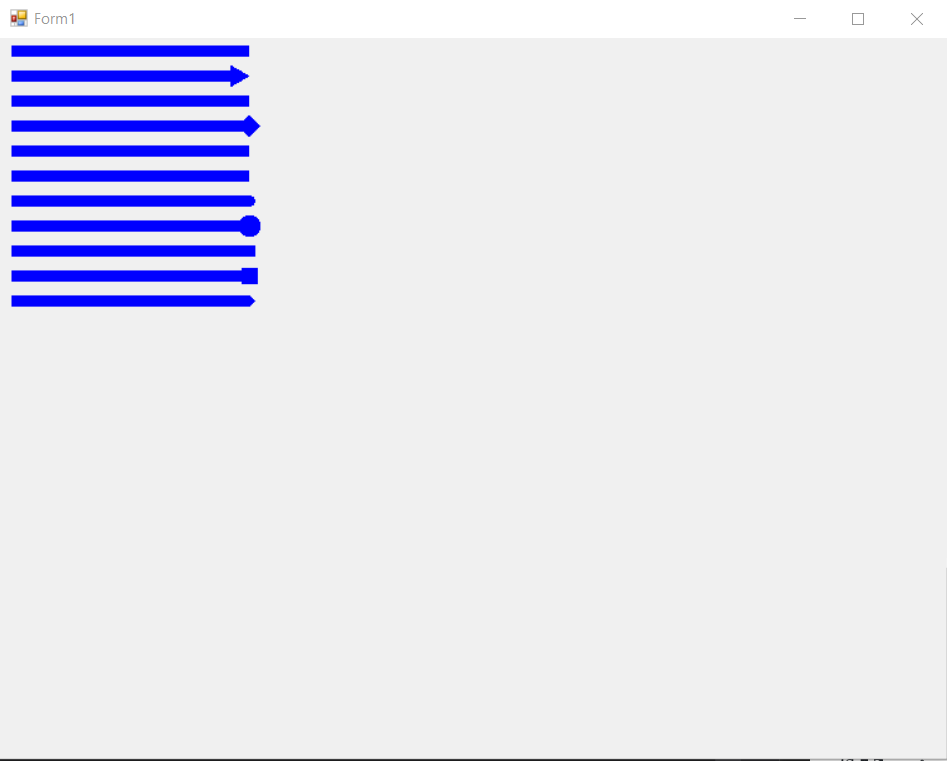
1. การกำหนดจุดปลายของเส้นตรงด้วย style แบบต่างๆ

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-13.png)

**รูปที่ 13** การกำหนดจุดปลายของเส้นตรงด้วย style แบบต่างๆ

1. รันโปรแกรมและบันทึกผล

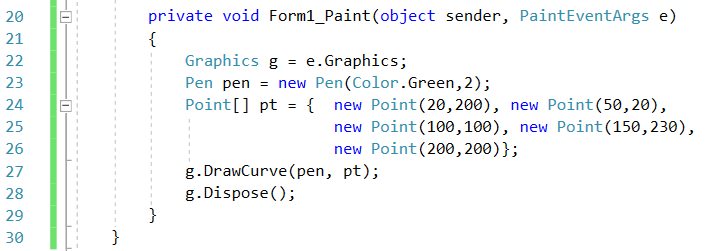
บันทึกผลการทดลอง



**3.4 การวาดเส้นโค้ง**

**3.4.1 การวาดเส้นโค้งด้วยเมธอด DrawCurve(...)**

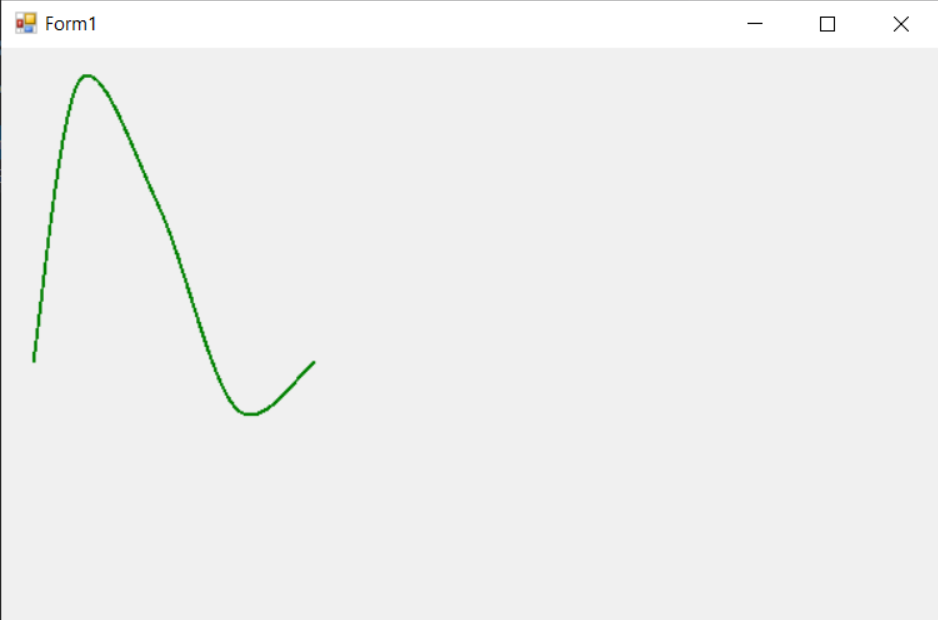
1. การวาดเส้นโค้งโดยการกำหนดจุดไว้ใน array ของ point แล้วส่งให้กับฟังก์ชัน DrawCurve(...)

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-14.png)

**รูปที่ 14** การวาดเส้นโค้งด้วยเมธอด DrawCurve(...)

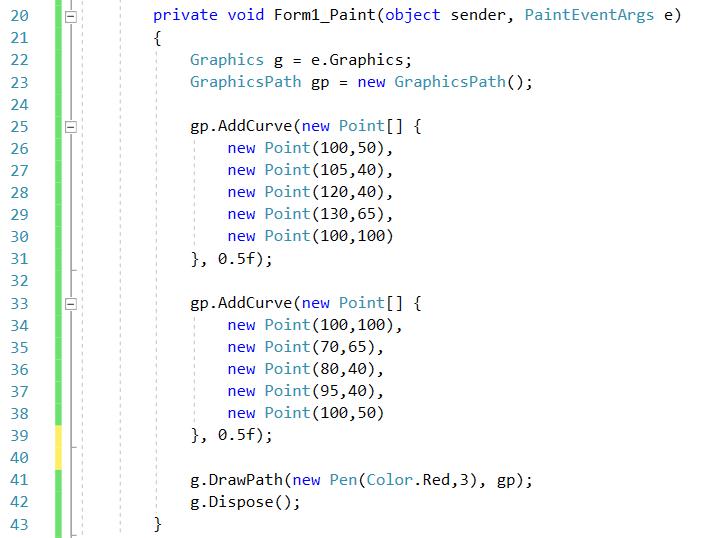
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.4.2 การวาดเส้นโค้งด้วย Graphics path**

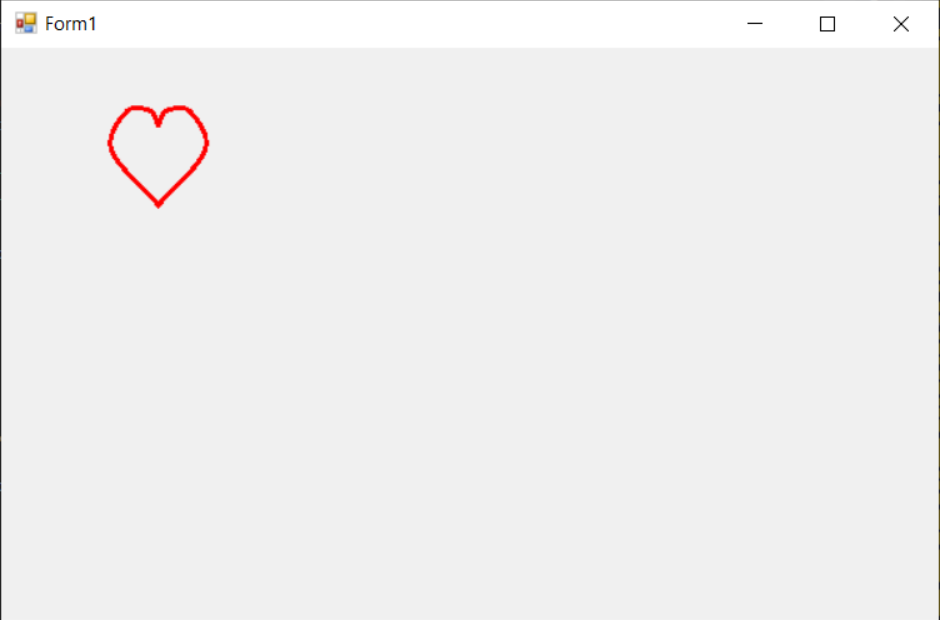
1. การวาดเส้นโค้งด้วย Graphics path

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-15.png)

**รูปที่ 15** การวาดเส้นโค้งด้วย Graphics path

1. รันโปรแกรมและบันทึกผล

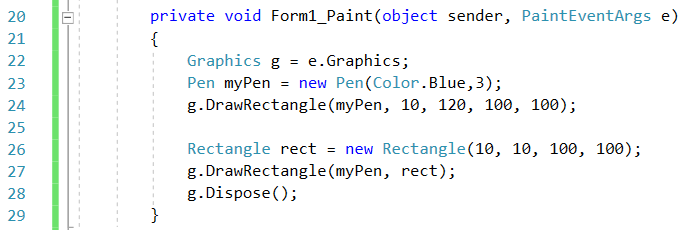
บันทึกผลการทดลอง



**3.5 การวาดรูปทรงต่างๆ**

**3.5.1 การวาดสี่เหลี่ยมครั้งละรูป**

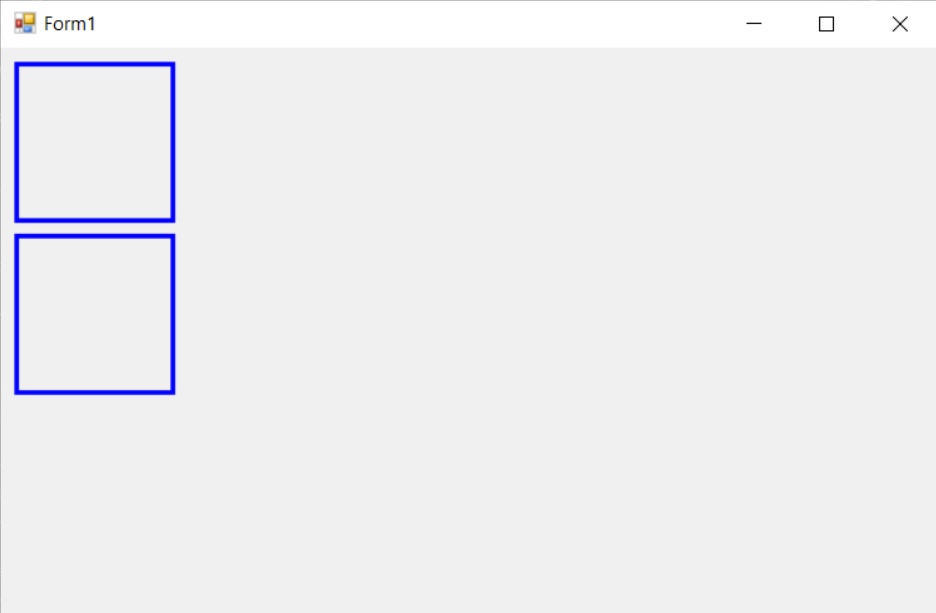
1. การวาดสี่เหลี่ยมครั้งละรูปเดียว

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-16.png)

**รูปที่ 16** การวาดสี่เหลี่ยมครั้งละรูปเดียว

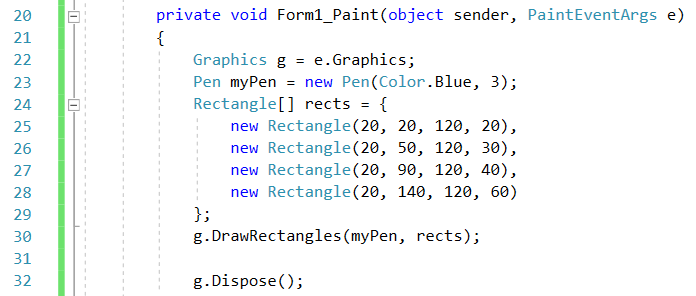
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.5.2 การวาดสี่เหลี่ยมพร้อมกันครั้งละหลายๆ รูป**

1. การวาดสี่เหลี่ยมพร้อมกันครั้งละหลายๆ รูป

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-17.png)

**รูปที่ 17** การวาดสี่เหลี่ยมพร้อมกันครั้งละหลายๆ รูป

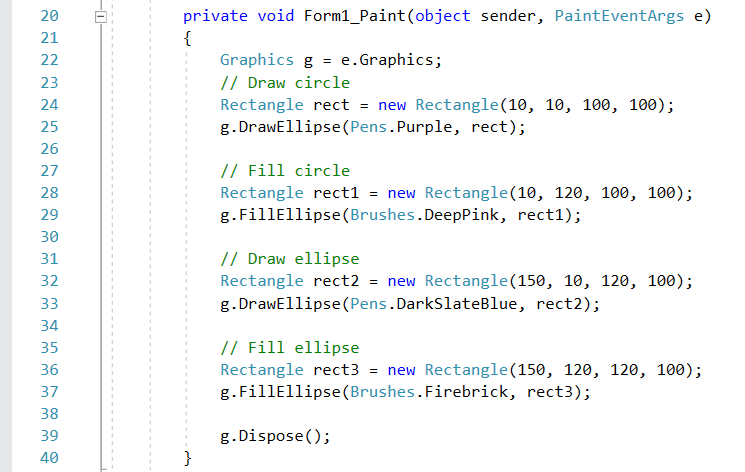
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.5.3 การวาดวงกลมและวงรี**

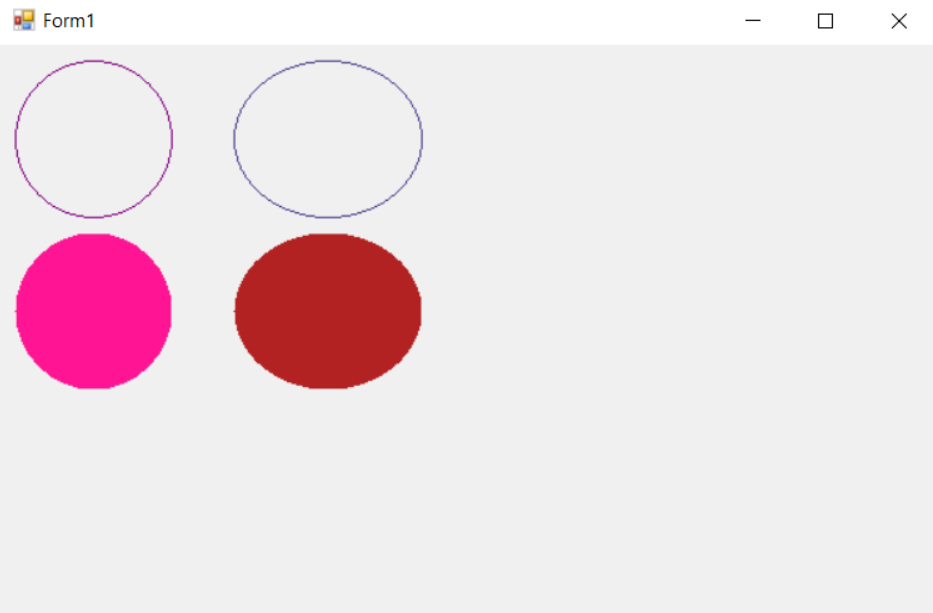
1. วงรีต่างจากวงกลมตรงที่เส้นผ่านศูนย์กลางในแกนตั้งและแกนนอนจะไม่เท่ากัน ในภาษาโปรแกรมส่วนใหญ่จะมีเฉพาะฟังก์ชันวาดวงรี ถ้าต้องการวาดวงกลม ให้กำหนดเส้นผ่านศูนย์กลางในแกนตั้งและแกนนอนให้เท่ากัน

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-18.png)

**รูปที่ 18** การวาดวงกลมและวงรี

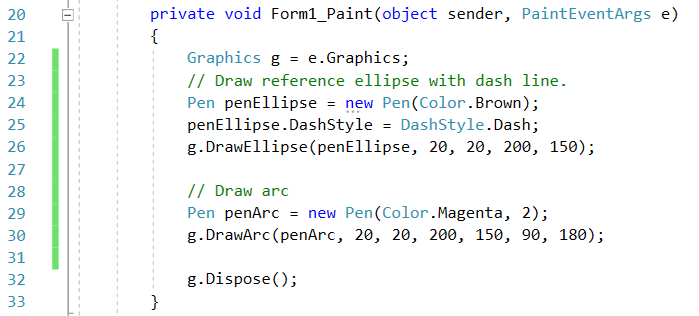
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.5.4 การวาดส่วนโค้ง (Arc)**

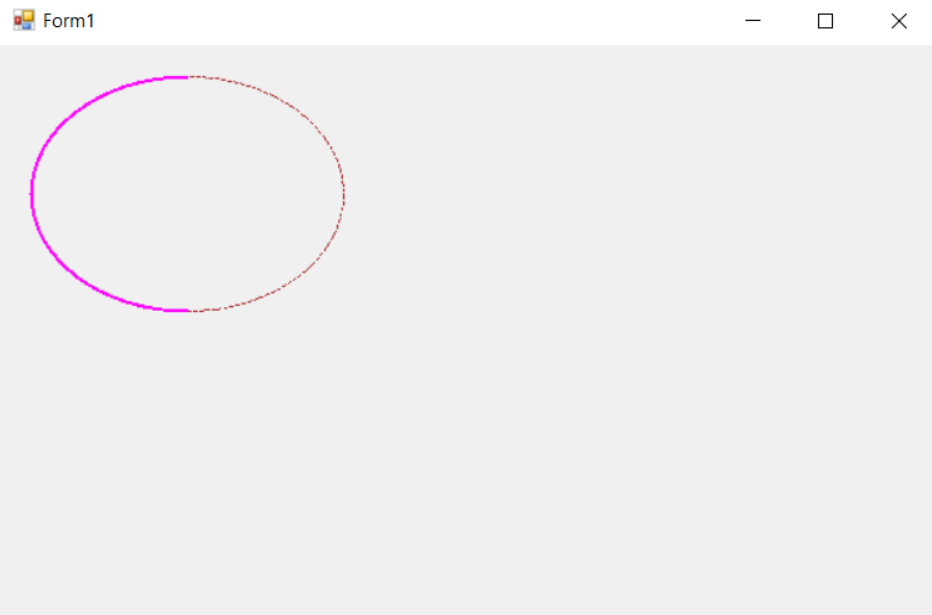
1. การวาดส่วนโค้ง (Arc)

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-19.png)

**รูปที่ 19** การวาดส่วนโค้ง (Arc)

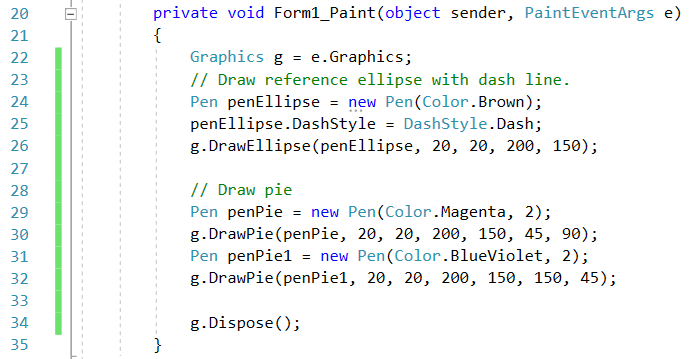
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.5.4 การวาดรูป Pie**

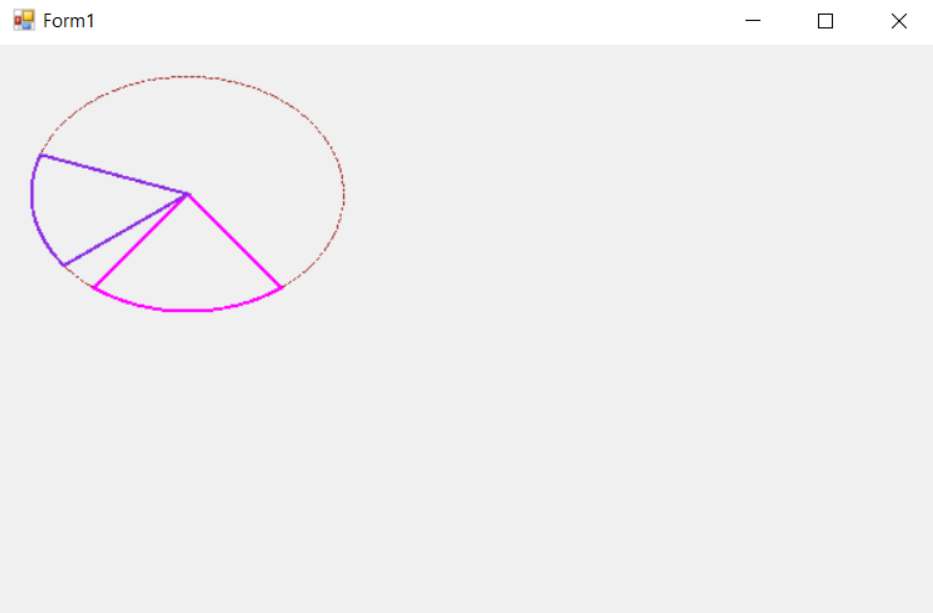
1. การวาดรูป Pie

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-20.png)

**รูปที่ 20** การวาดรูป Pie

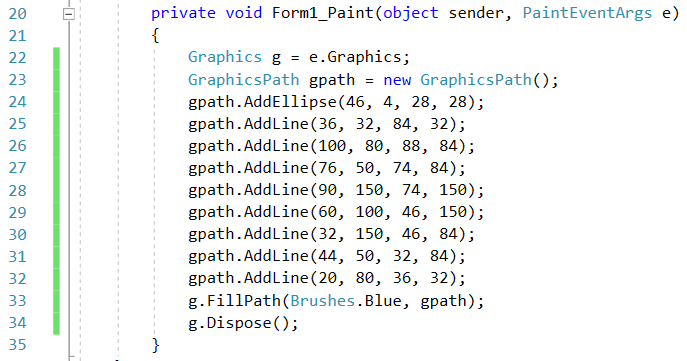
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.5.5 การสร้าง graphics path จากรูปต่างๆ**

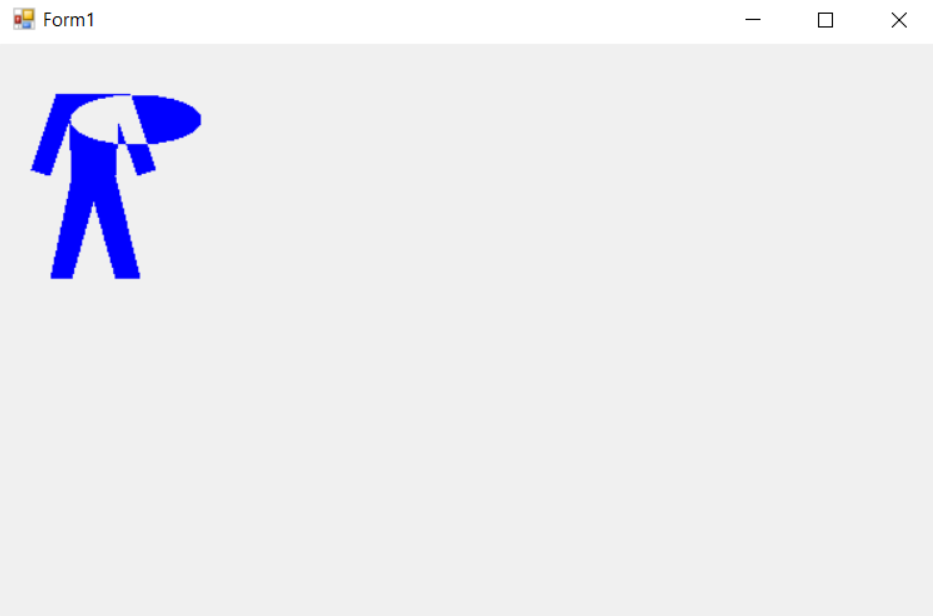
1. การสร้าง graphics path จากรูปต่างๆ

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-21.png)

**รูปที่ 21** การสร้าง graphics path จากรูปต่างๆ

1. รันโปรแกรมและบันทึกผล

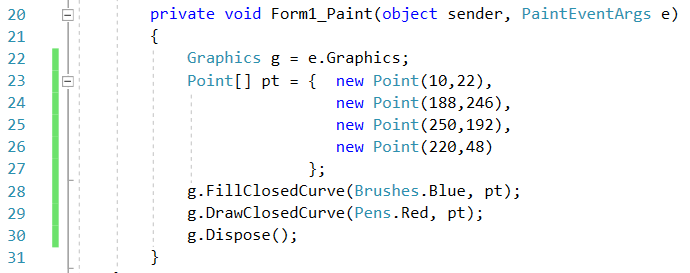
บันทึกผลการทดลอง



**3.6 การใช้งานแปรงระบายสี (Brush)**

**3.6.1 การระบายสีด้วย SolidBrush**

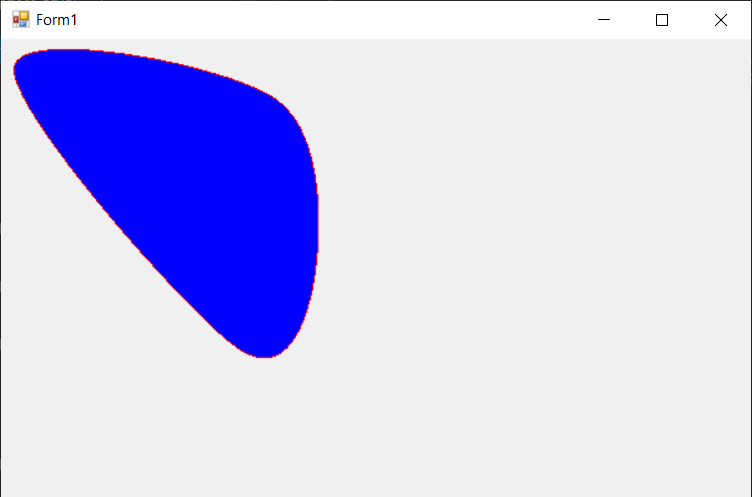
1. การระบายสีด้วย SolidBrush

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-22.png)

**รูปที่ 22** การระบายสีด้วย SolidBrush

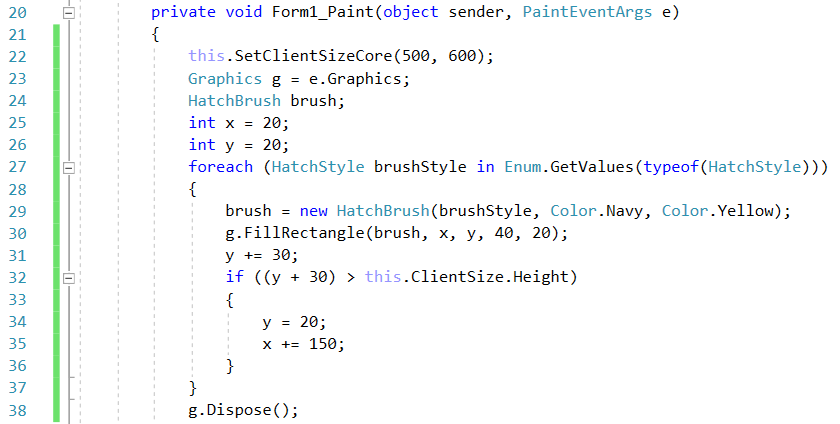
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.6.2 การระบายสีด้วย HatchBrush**

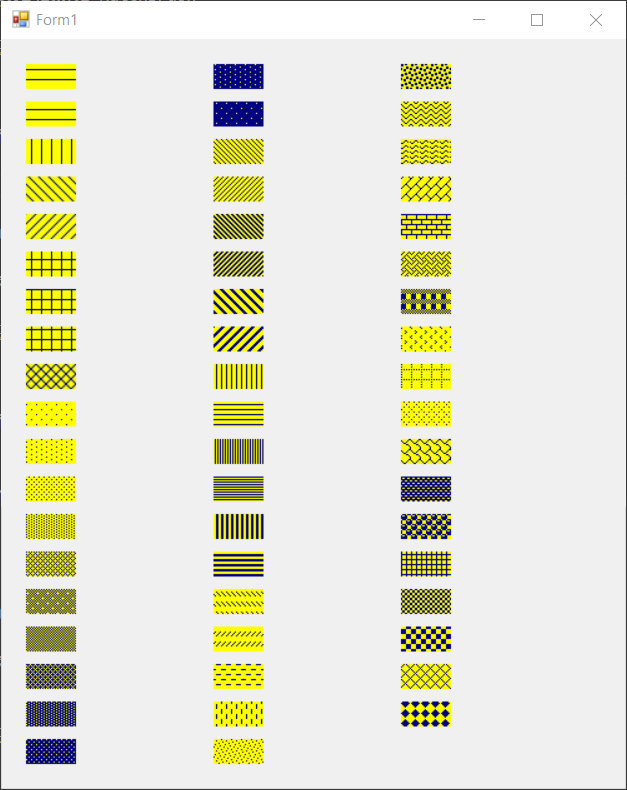
1. การระบายสีด้วย HatchBrush

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-23.png)

**รูปที่ 23** การระบายสีด้วย HatchBrush

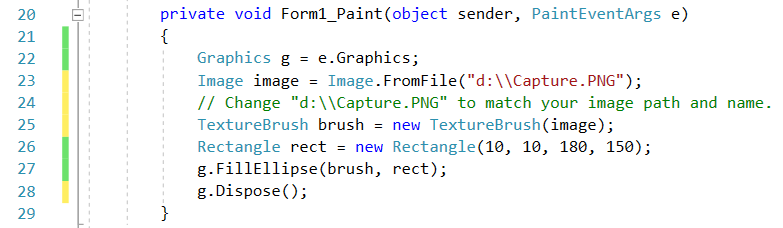
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.6.3 การระบายสีด้วย TextureBrush**

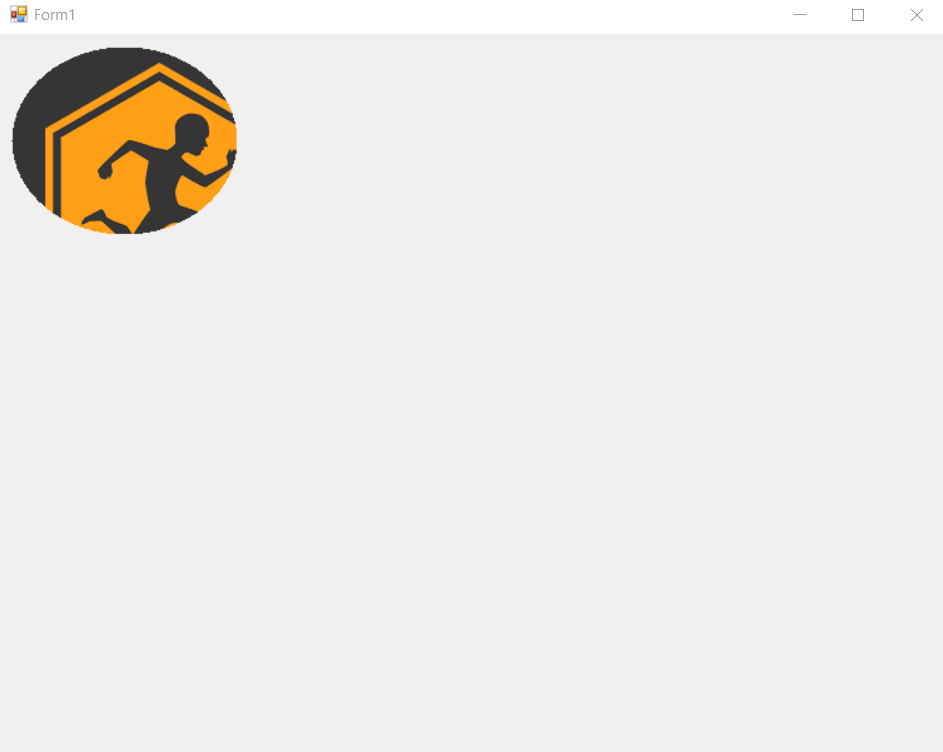
1. การระบายสีด้วย TextureBrush

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-24.png)

**รูปที่ 24** การระบายสีด้วย TextureBrush

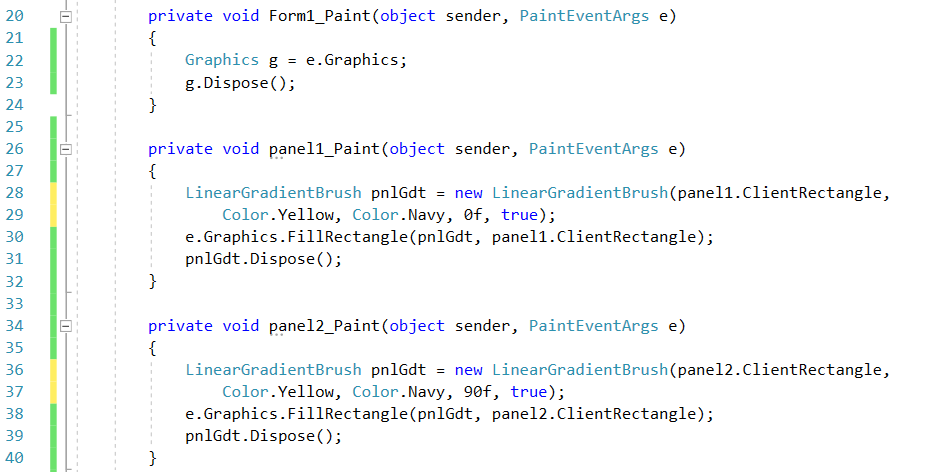
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.6.4 การระบายสีด้วย TextureBrush บน panel**

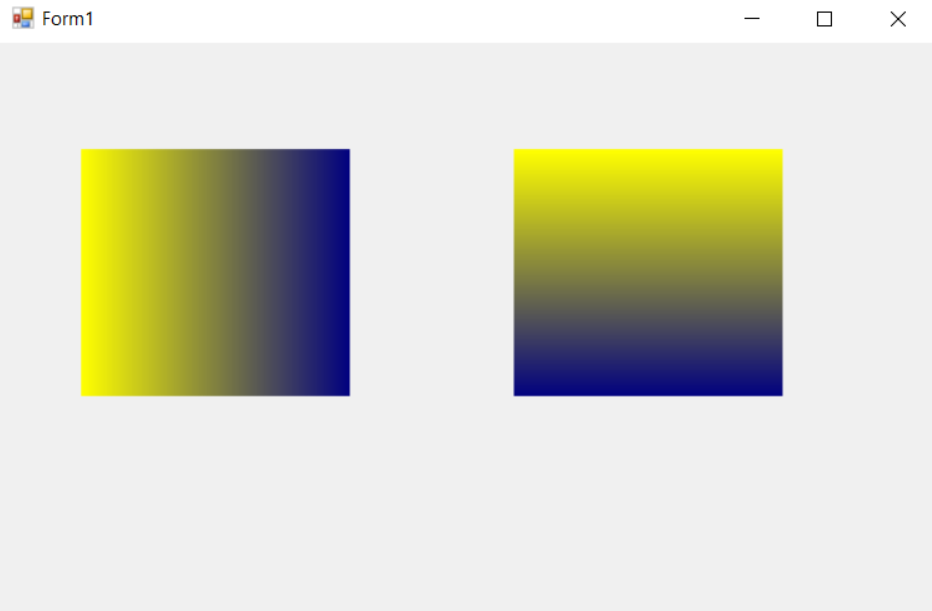
1. การระบายสีด้วย TextureBrush บน panel

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-25.png)

**รูปที่ 25** การระบายสีด้วย TextureBrush บน panel

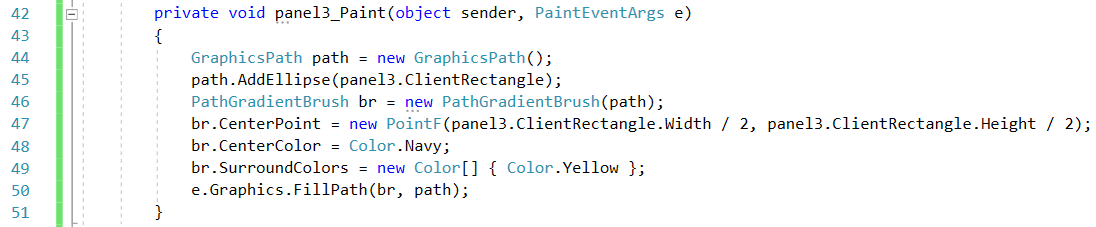
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.6.5 การระบายสีด้วย Path Gradient Brush**

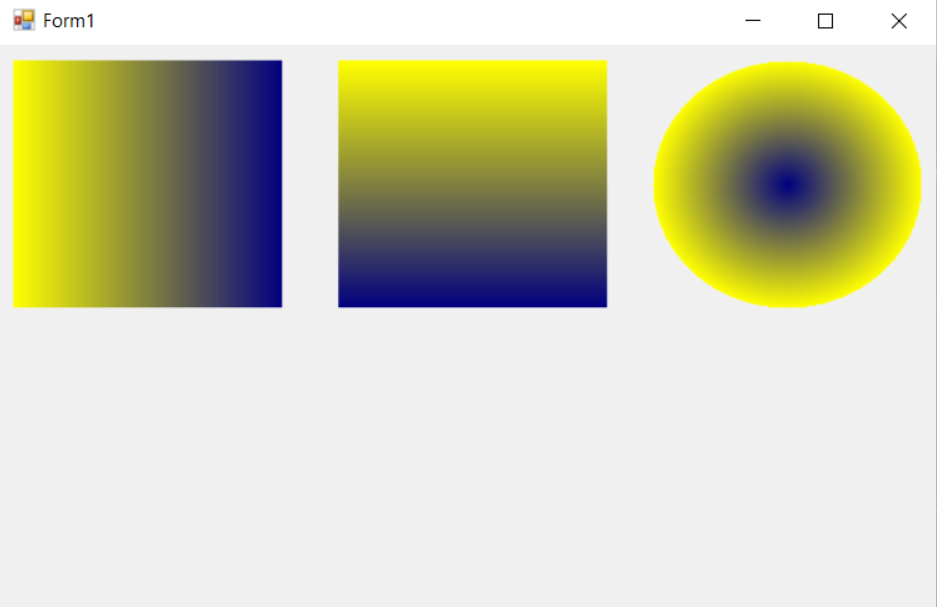
1. การระบายสีด้วย Path Gradient Brush

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-26.png)

**รูปที่ 26** การระบายสีด้วย Path Gradient Brush

1. รันโปรแกรมและบันทึกผล

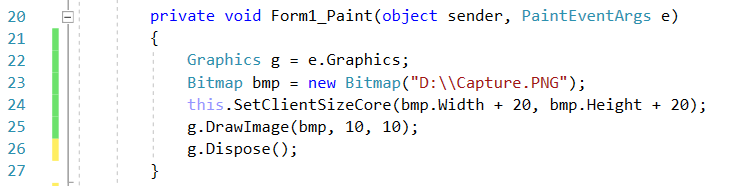
บันทึกผลการทดลอง



**3.7 การใช้งานภาพ**

**3.7.1 การโหลดภาพชนิดต่างๆ มาแสดงบนฟอร์ม**

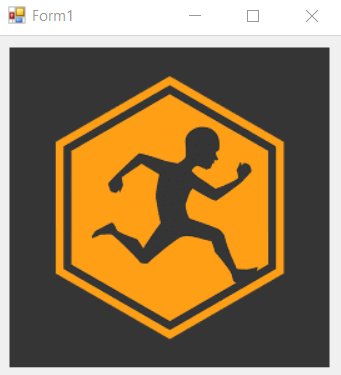
1. การโหลดภาพชนิดต่างๆ มาแสดงบนฟอร์ม

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-27.png)

**รูปที่ 27** การโหลดภาพชนิดต่างๆ มาแสดงบนฟอร์ม

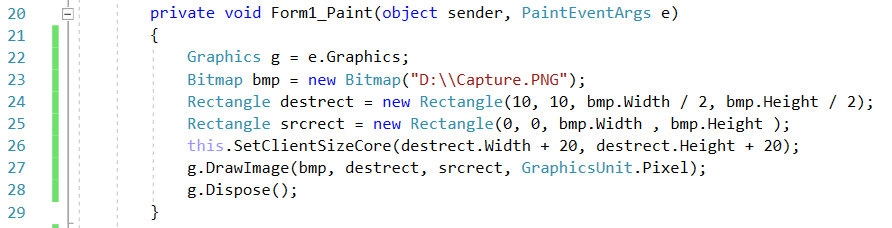
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.7.2 การ Zoom out**

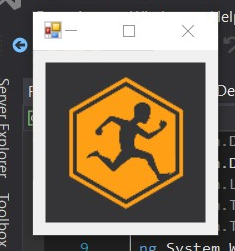
1. การ Zoom out

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-28.png)

**รูปที่ 28** การ Zoom out

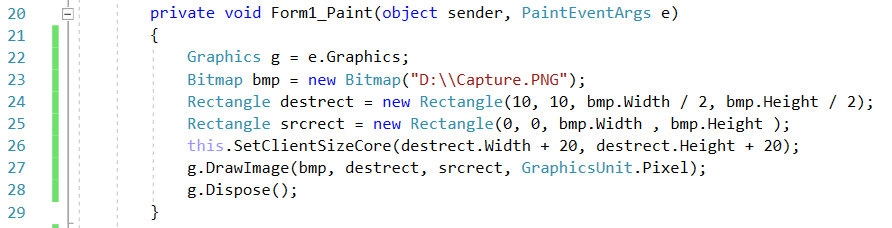
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.7.3 การ Zoom in**

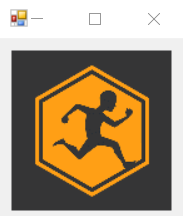
1. การ Zoom in

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-28.png)

**รูปที่ 29** การ Zoom in

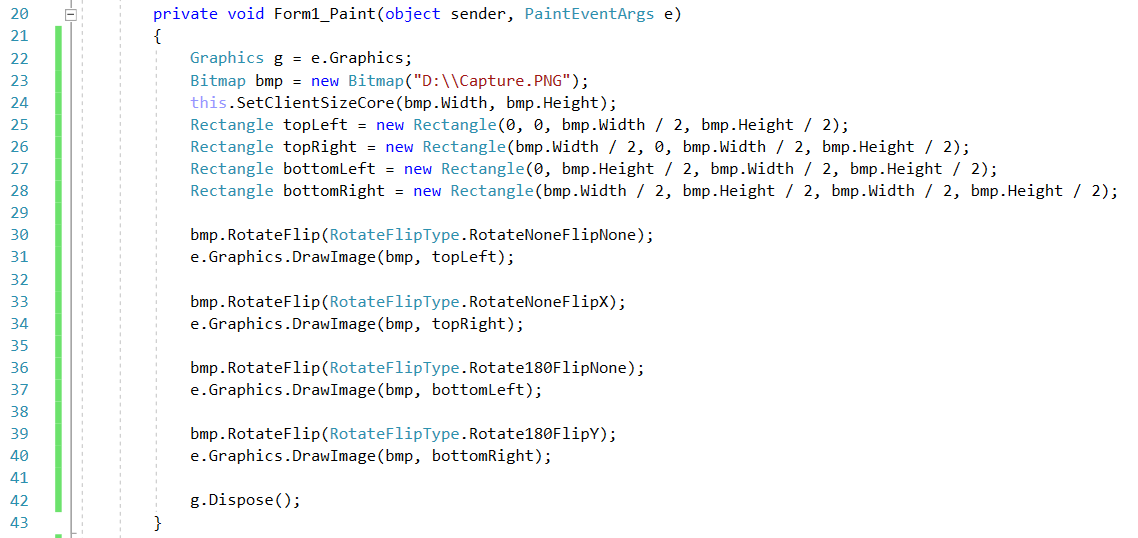
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.7.4 การพลิกและหมุนภาพ**

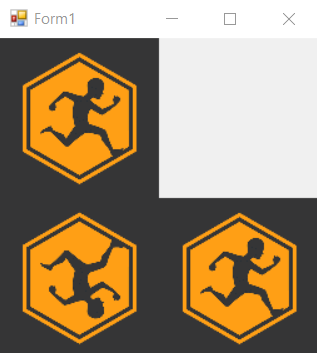
1. การพลิกและหมุนภาพ

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-30.png)

**รูปที่ 30** การพลิกและหมุนภาพ

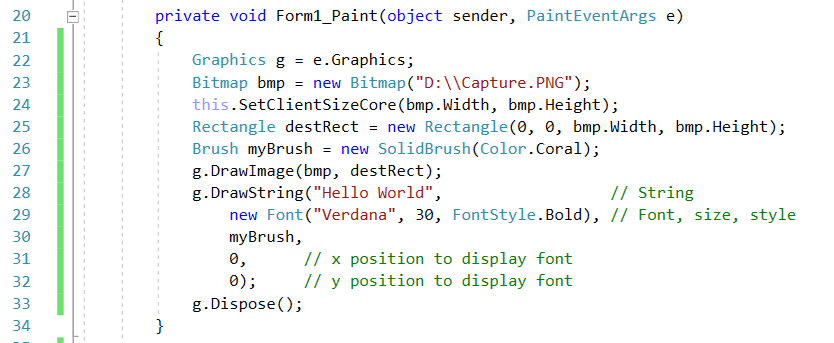
1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**3.7.5 การเขียนข้อความลงในภาพ**

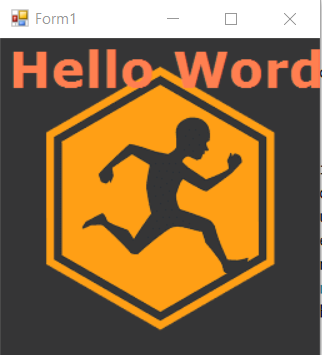
1. การเขียนข้อความลงในภาพ

[](https://github.com/aomnaherb/Week-09/blob/master/images/fig-31.png)

**รูปที่ 31** การเขียนข้อความลงในภาพ

1. รันโปรแกรมและบันทึกผล

บันทึกผลการทดลอง



**การบ้าน**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Drawing.Drawing2D;

namespace paint

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

Graphics g = e.Graphics;

Pen mypen = new Pen (Color.Black, 80);

g.DrawRectangle(mypen, 100, 100, 300, 200);

mypen.Width = 10;

Pen body = new Pen(Color.Yellow, 80);

g.DrawRectangle(body, 100, 300, 300, 200);

body.Width = 10;

Rectangle nose = new Rectangle(220,200,70,50);

g.FillEllipse(Brushes.Brown, nose);

nose.Width = 80;

Pen eyes1 = new Pen (Color.Red,50);

g.DrawEllipse(eyes1,80,80,90,80 );

Pen eyes = new Pen(Color.Red, 50);

g.DrawEllipse(eyes, 330, 80, 90, 80);

Font f = new Font("Verdana", 20, FontStyle.Italic);

Brush Text = new SolidBrush(Color.Green);

g.DrawString("Mechazawa", f,Text,180,400);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

}

}

****