Draw Pyramid

```
/*
Professional Windows GUI Programming Using C#
by Jay Glynn, Csaba Torok, Richard Conway, Wahid Choudhury,
Zach Greenvoss, Shripad Kulkarni, Neil Whitlow

Publisher: Peer Information
ISBN: 1861007663
*/
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

using System.Drawing.Drawing2D;

namespace Pyramid
{
///

/// Summary description for Pyramid.
///
public class Pyramid1 : System.Windows.Forms.Form
{
private System.Windows.Forms.Button button1;
///

/// Required designer variable.
///
private System.ComponentModel.Container components = null;

int rot = 0;
Point center = new Point(125, 100);
public Pyramid1()
```

```
{
//
// Required for Windows Form Designer support
//
InitializeComponent();

this.Text = “Pyramid by Transfomation”;
//
// TODO: Add any constructor code after InitializeComponent
call
//
}

///

/// Clean up any resources being used.
///
protected override void Dispose( bool disposing )
{
if( disposing )
{
if (components != null)
{
components.Dispose();
}
}
base.Dispose( disposing );
}

#region Windows Form Designer generated code
///

/// Required method for Designer support – do not modify
/// the contents of this method with the code editor.
///
private void InitializeComponent()
{
this.button1 = new System.Windows.Forms.Button();
this.SuspendLayout();
```

```
//
// button1
//
this.button1.Location = new System.Drawing.Point(8, 8);
this.button1.Name = “button1”;
this.button1.Size = new System.Drawing.Size(48, 24);
this.button1.TabIndex = 0;
this.button1.Text = “Rotate”;
this.button1.Click += new
System.EventHandler(this.button1_Click);
//
// Pyramid
//
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(292, 266);
this.Controls.AddRange(new System.Windows.Forms.Control[] {
this.button1});
this.Name = “Pyramid”;
this.Text = “Pyramid”;
this.ResumeLayout(false);

}
#endregion

///

/// The main entry point for the application.
///
[STAThread]
static void Main()
{
Application.Run(new Pyramid1());
}
protected override void OnPaint
(System.Windows.Forms.PaintEventArgs e)
{
Graphics g = e.Graphics;
g.Clear(this.BackColor);
```

```
Pyramid(g);
//PyramidPathRotate(g);
g.Dispose();
}
protected void Pyramid(Graphics g)
{
Pen p = new Pen(Color.Blue);
int ten = 10;
Rectangle rc = new Rectangle(50, 90, 150, ten); // the base rectangle
g.DrawRectangle(p, rc);
for(int i = 1;i <= 7; i++) { rc.Offset(0,-ten); rc.Inflate(-ten, 0); g.DrawRectangle(p, rc); } p.Dispose(); } private void button1_Click(object sender, System.EventArgs e) { // rot++; //      rot      is      a      class      member      // PyramidPathRotate(CreateGraphics()); // Refresh(); } protected void PyramidPathRotate(Graphics g) { GraphicsPath gP = new GraphicsPath(); // create an empty path Pen p = new Pen(Color.Blue); int ten = 10; Rectangle rc = new Rectangle(50, 90, 150, ten); // the base rectangle gP.AddRectangle(rc); for(int i = 1;i <= 7; i++) { rc.Offset(0,-ten); rc.Inflate(-ten,0); gP.AddRectangle(rc); } Matrix m = new Matrix(); m.RotateAt(45*rot, center, MatrixOrder.Append); gP.Transform(m); g.DrawPath(p, gP); // draw the rotated path g.FillEllipse(Brushes.Red, center.X, center.Y, 3, 3); // center point p.Dispose(); } } } [/csharp]
```