Part 1: Looking at the Microsoft .NET frame work and CIL using a simple C# program

```
namespace CIL
{
class Program
{
static void Main(string[] args)
{
int a = 10, b = 10, c;
c = a + b;
Console.WriteLine(c.ToString());
Console.ReadKey();
}
}
}
```

Find the ildasm (intermediate language disassembler) and start it by double clicking on the Icon in Windows Explorer. Use this program to open the CIL.exe (executable) file you created. You can launch the Windows explorer by holding down the Windows Key á and E.

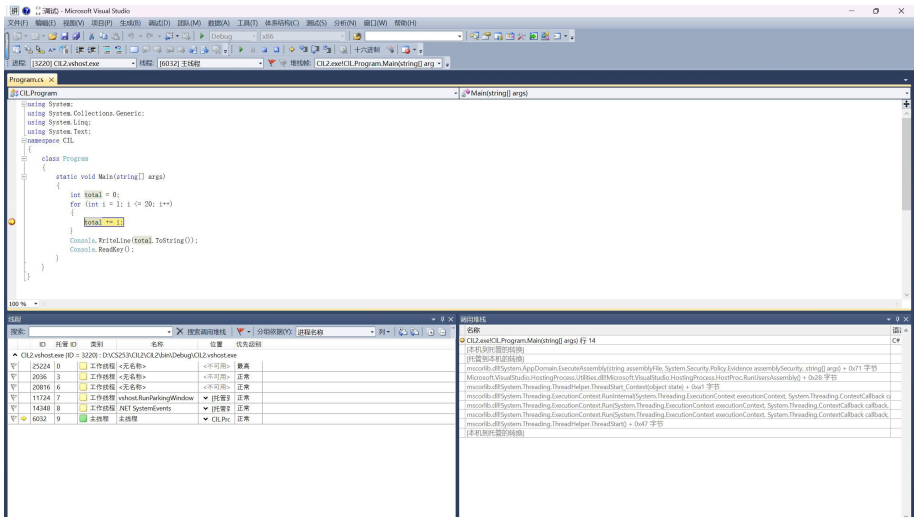
```
.method private hidebysig static void  Main(string[] args) cil managed
{
  .entrypoint
  // 代码大小       31 (0x1f)
  .maxstack  2
  .locals init (int32 V_0,
           int32 V_1,
           int32 V_2)
  IL_0000:  nop
  IL_0001:  ldc.i4.s   10
  IL_0003:  stloc.0
  IL_0004:  ldc.i4.s   10
  IL_0006:  stloc.1
  IL_0007:  ldloc.0
  IL_0008:  ldloc.1
  IL_0009:  add
  IL_000a:  stloc.2
  IL_000b:  ldloca.s   V_2
  IL_000d:  call       instance string [mscorlib]System.Int32::ToString()
  IL_0012:  call       void [mscorlib]System.Console::WriteLine(string)
  IL_0017:  nop
  IL_0018:  call       valuetype [mscorlib]System.ConsoleKeyInfo [mscorlib]System.Console::ReadKey(
  IL_001d:  pop
  IL_001e:  ret
} // end of method Program::Main
```

Figure 9 A Windows PE executable run on a DOS emulator, CIL.EXE.

Part 2: Using the debugging tools available in the IDE

Using a debugger is an essential skill when using an IDE. Use your existing CIL Windows Console Application and modify the program so that it adds the integers from 1 to 20.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace CIL
{
class Program
{
static void Main(string[] args)
{
int total = 0;
for (int i = 1; i <= 20; i++)
{
total += i;
}
Console.WriteLine(total.ToString());
Console.ReadKey();
}
}
}
```



part 3: Recreating a classic computer game, Pong as a C# Windows Form Application

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;
namespace Pong
{
    public partial class Form1 : Form
    {
        [DllImport("user32.dll")]
        static extern ushort GetAsyncKeyState(int vKey);
        public static bool IsKeyPushedDown(Keys keyData)  // For fast
keyboard input
        {
            return 0 != (GetAsyncKeyState((int)keyData) & 0x8000);
        }
        Point Ball;
        Point Ball_Direction;
        int Paddle_Left_Y = 128;
        int Paddle_Right_Y = 128;
        int Left_score = 0;
        int Right_score = 0;
        bool Game_paused = true;
        public Form1()
        {
            InitializeComponent();
            Ball = new Point(25, 13);   // Set starting position of ball
            Ball_Direction = new Point(3, 3);
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }
```

```
        private void timer1_Tick(object sender, EventArgs e)
        {
            if (IsKeyPushedDown(Keys.W)) Paddle_Left_Y -= 10;
            if (IsKeyPushedDown(Keys.X)) Paddle_Left_Y += 10;
            if (IsKeyPushedDown(Keys.P)) Paddle_Right_Y -= 10;
            if (IsKeyPushedDown(Keys.M)) Paddle_Right_Y += 10;
            if (IsKeyPushedDown(Keys.Space)) Game_paused = false;
            if (Paddle_Left_Y < 5) Paddle_Left_Y = 0;  // Clamp paddle
positions
            if (Paddle_Left_Y > 251) Paddle_Left_Y = 251;
            if (Paddle_Right_Y < 5) Paddle_Right_Y = 0;
            if (Paddle_Right_Y > 251) Paddle_Right_Y = 251;
            if (IsKeyPushedDown(Keys.Escape)) Application.Exit();
            // Ball/Paddle collision detection on left paddle
            if ((Ball.X < 30) && (Math.Abs(Ball.Y - Paddle_Left_Y) < 50) &&
(Ball_Direction.X < 0))
            {
                Ball_Direction.X = -Ball_Direction.X;
            }
```

```
            if ((Ball.X > 482) && (Math.Abs(Ball.Y - Paddle_Right_Y) < 50)
&& (Ball_Direction.X > 0))
            {
                Ball_Direction.X = -Ball_Direction.X;
            }
            if (!Game_paused)
            {
                Ball.X += Ball_Direction.X;
                if ((Ball.X > 511) && (Ball_Direction.X > 0))   // Ball hit
back wall travelling left
                {
                    Ball.X = 30;                                // Restart ball
in front of paddle
                    Ball.Y = Paddle_Left_Y;
                    Game_paused = true;
                    Left_score++;// Reverse horizontal motion
                }
                if ((Ball.X < 1) && (Ball_Direction.X < 0)) // Ball hit back
wall travelling right
                {
                    // Ball_Direction.X = -Ball_Direction.X; // Remove or
“comment out” this line
                    Ball.X = 480;                               // Restart ball
in front of paddle
```

```
                    Ball.Y = Paddle_Right_Y;
                    Game_paused = true;
                    Right_score++;// Reverse horizontal motion
                }
                Ball.Y += Ball_Direction.Y;
                if ((Ball.Y > 255) && (Ball_Direction.Y > 0)) // Ball hit
bottom wall travelling down
                {
                    Ball_Direction.Y = -Ball_Direction.Y;      // Reverse
horizontal motion
                }
                if ((Ball.Y < 1) && (Ball_Direction.Y < 0))   // Ball hit
top wall travelling down
                {
                    Ball_Direction.Y = -Ball_Direction.Y;   // Reverse
horizontal motion
                }
            }
            Bitmap image = new Bitmap(pictureBox1.Width,
pictureBox1.Height);
            Graphics g = Graphics.FromImage(image);
            g.FillEllipse(new SolidBrush(Color.Red), Ball.X - 10, Ball.Y -
10, 20, 20);
            g.FillRectangle(new SolidBrush(Color.Blue), 10, Paddle_Left_Y
- 50, 10, 100);
            g.FillRectangle(new SolidBrush(Color.Blue), 490,
Paddle_Right_Y - 50, 10, 100);
            g.Dispose();
            label1.Text = Left_score.ToString("0.#");
            label2.Text = Right_score.ToString("0.#");
            if ((Left_score > 2) || (Right_score > 2))  // Identify a winner
            {
                timer1.Stop();  // Stop game loop
                String text = "Right";
                if (Left_score > Right_score) text = "Left";
                DialogResult reply = MessageBox.Show(text + " player
wins\rDo you wish to play again?",
"Winner", MessageBoxButtons.YesNo);
                if (reply == DialogResult.Yes)
                {
                    Left_score = 0;     // Reset game values
                    Right_score = 0;
                    Paddle_Left_Y = 128;
                    Paddle_Right_Y = 128;
```

```
                    Game_paused = true;
                    Ball.X = 25; Ball.Y = 13;
                    Ball_Direction.X = 3; Ball_Direction.Y = 3;
                    timer1.Start();  // Restart game loop
                }
                else
                {
                    Application.Exit(); // Quit game
                }
            }
            pictureBox1.Image = image;

        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

    }
}
```

Pong:

```
namespace Pong
{
    partial class Form1
    {
        /// <summary>
        /// 必需的设计器变量。
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// 清理所有正在使用的资源。
        /// </summary>
        /// <param name="disposing">如果应释放托管资源，为 true；否则为 false。
</param>
```

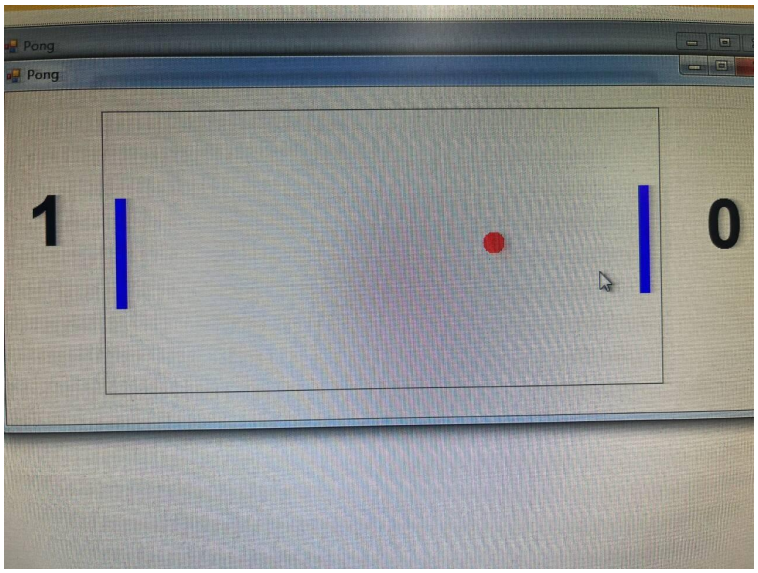
```
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
```

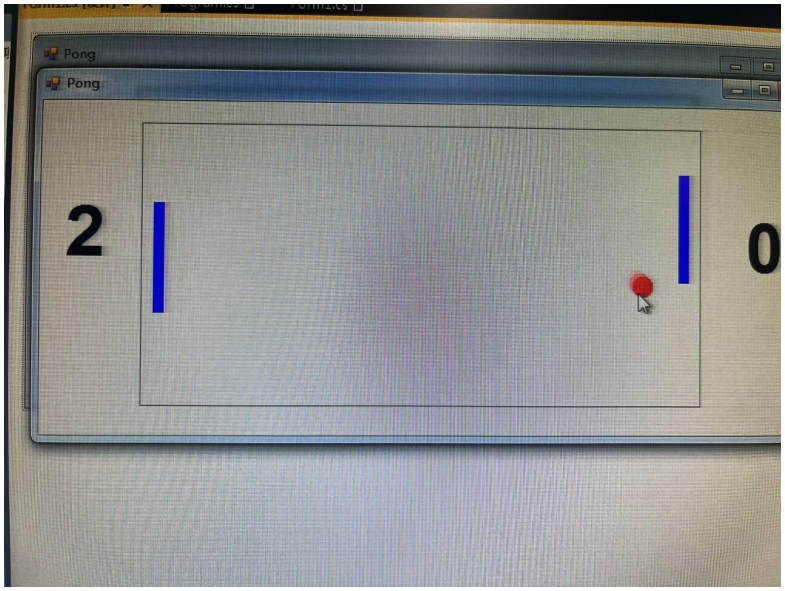
```
        #region Windows 窗体设计器生成的代码
```

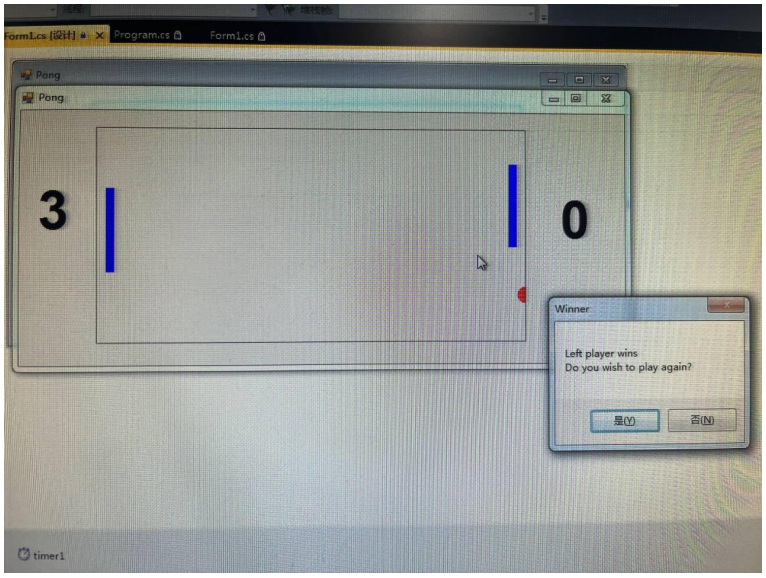
```
        /// <summary>
        /// 设计器支持所需的方法 - 不要
        /// 使用代码编辑器修改此方法的内容。
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            this.timer1 = new System.Windows.Forms.Timer(this.components);
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)
).BeginInit();
            this.SuspendLayout();
            //
            // timer1
            //
            this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font = new System.Drawing.Font("宋体", 48F);
            this.label1.Location = new System.Drawing.Point(12, 106);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(60, 64);
            this.label1.TabIndex = 0;
            this.label1.Text = "0";
            this.label1.Click += new
System.EventHandler(this.label1_Click);
            //
            // label2
            //
```

```
            this.label2.AutoSize = true;
            this.label2.Font = new System.Drawing.Font("宋体", 48F);
            this.label2.Location = new System.Drawing.Point(652, 106);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(60, 64);
            this.label2.TabIndex = 1;
            this.label2.Text = "0";
            //
            // pictureBox1
            //
            this.pictureBox1.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle;
            this.pictureBox1.Location = new System.Drawing.Point(90, 20);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(512, 256);
            this.pictureBox1.TabIndex = 2;
            this.pictureBox1.TabStop = false;
            this.pictureBox1.Click += new
System.EventHandler(this.pictureBox1_Click);
            //
            // Form1
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(724, 302);
            this.Controls.Add(this.pictureBox1);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.label1);
            this.Name = "Form1";
            this.Text = "Form1";
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)
).EndInit();
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Timer timer1;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.PictureBox pictureBox1;
    }
```

```
}
```


Pong
Pong
1
0


Pong
Pong
2
0


Program.cs
Form1.cs
Pong
Pong
3
0
Winner
Left player wins
Do you wish to play again?
timer1