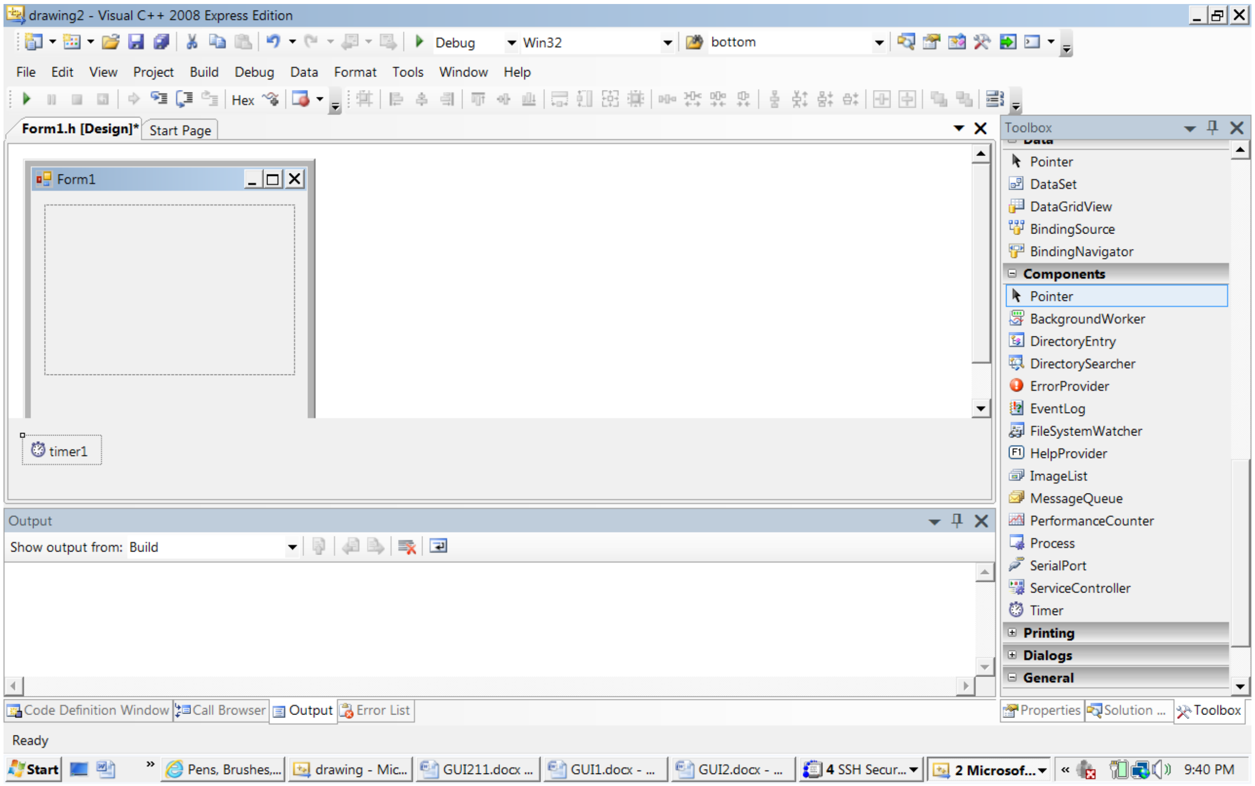
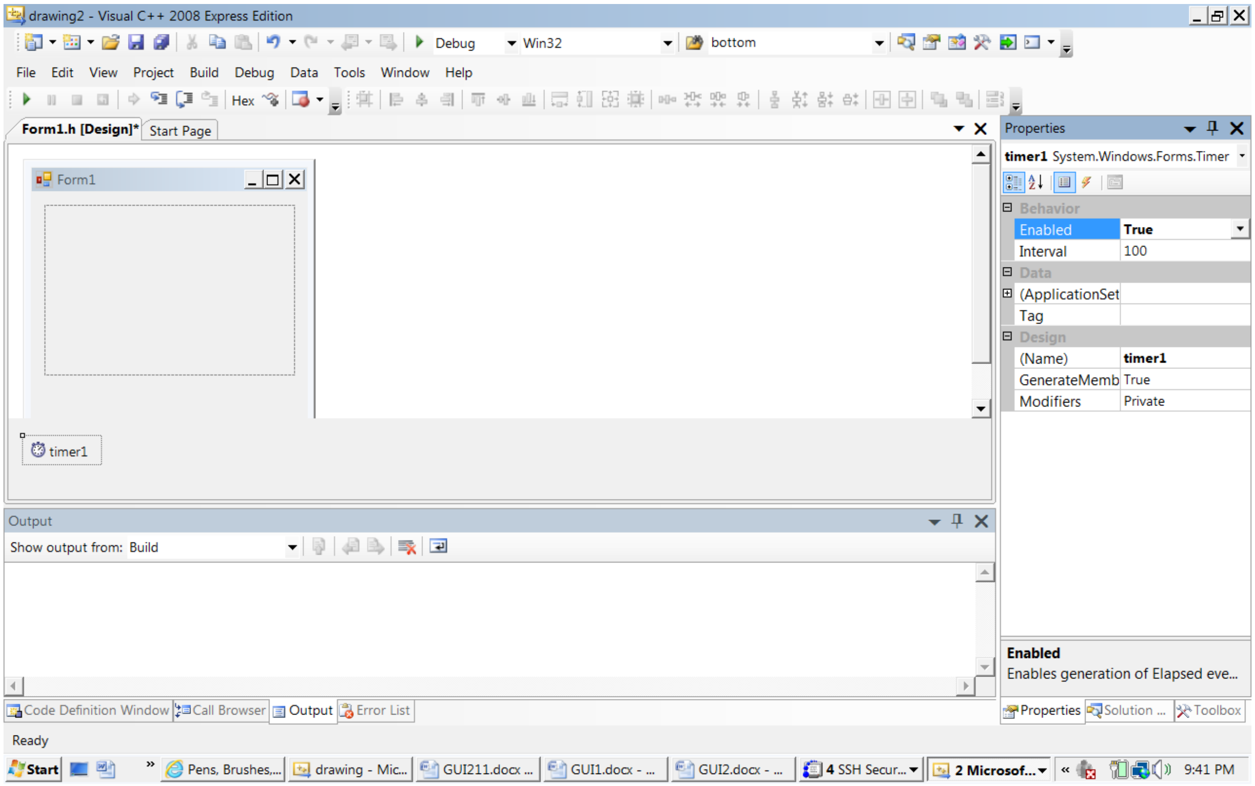
Graphic User Interface (Practice Using Objects)

Part 2

1. We will use the skills/knowledge that we obtained from the first GUI Graphics project to do a short animation. Open a CLR/Windows Form Application as in Step 1 of the GUI1.htm document.
2. Drag a **panel** (in the **Toolbox** under **Container**) onto the form. Also drag a **Timer** (in the **Toolbox** under **Components**) The Timer will appear under the Form and will not show during runtime.



3. Select the timer and make sure that the **Enabled** Property (in the Property window) is set to **true.**

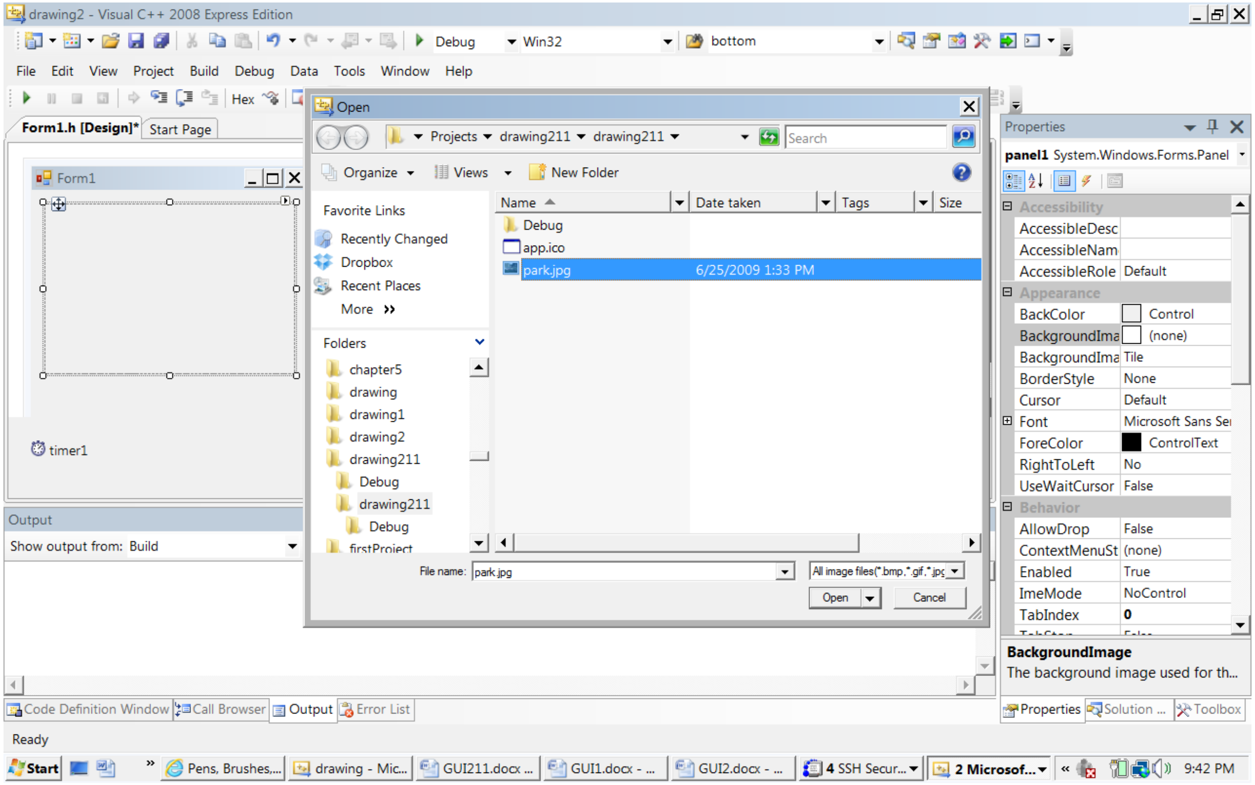


Properties: timer1 System.Windows.Forms.Timer

Behavior	
Enabled	True
Interval	100
Data	
(ApplicationSet	
Tag	
Design	
(Name)	timer1
GenerateMemb	True
Modifiers	Private

Enabled
Enables generation of Elapsed eve...

4. Select the panel, and in the Properties Window, either change the Background Color of the panel or place a picture into the panel using the BackgroundImage property. Here is one that you can use, or choose any .jpg file of your choice.



Name	Date taken	Tags	Size
Debug			
app.ico			
park.jpg	6/25/2009 1:33 PM		

File name: park.jpg

All image files(*.bmp,*.gif,*.jpg

panel1 System.Windows.Forms.Panel

Accessibility	
AccessibleDesc	
AccessibleNam	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundIma	(none)
BackgroundIma	Tile
BorderStyle	None
Cursor	Default
Font	Microsoft Sans Se
ForeColor	ControlText
RightToLeft	No
UseWaitCursor	False
Behavior	
AllowDrop	False
ContextMenuSt	(none)
Enabled	True
ImeMode	NoControl
TabIndex	0

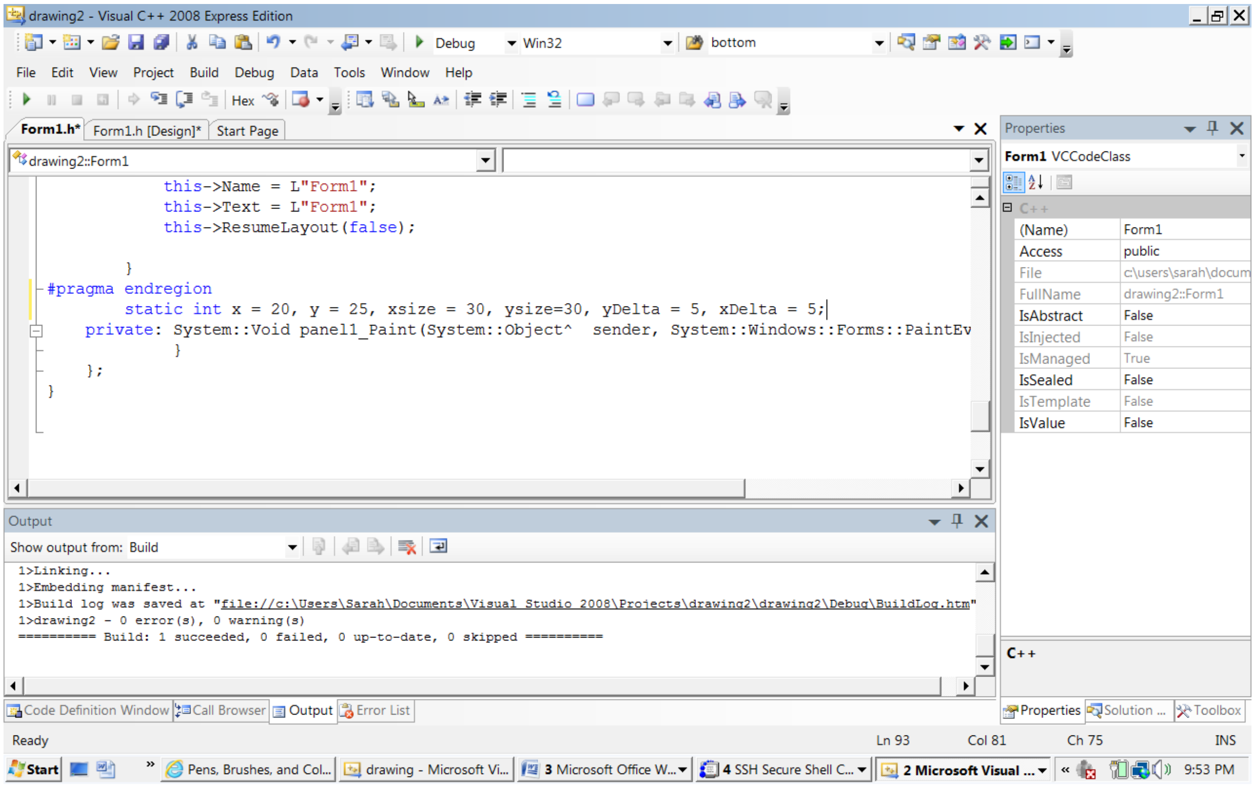
BackgroundImage
The background image used for th...

5. Use CTL-F5 to run your code.
6. We are going to have a ball bounce around the panel (or park, if you used the picture above). To do this:
   - ➢ we will write the code to draw a filled-in circle (that's just what we did in the previous GUI Lab!) in the topmost left corner of the panel
   - ➢ Then we will write code to refresh the panel with a new picture, where the ball is in a slightly different place (by changing the coordinates of the rectangle by adding to the x and y coordinates).
   - ➢ If the ball's rectangle coordinates are moving off the panel, we will begin subtracting from x or y, instead of adding

7. Double click on the panel to get to the event handler for the panel. Above the event handler (and right under the `#pragma endregion`) line, type in:

```
static int x = 0, y = 0, size = 20, yDiff = 5, xDiff = 5;
```

We did not spend time on `static` in this class, but these are integers that are outside any of the functions, but can be used by all of the functions in this class.



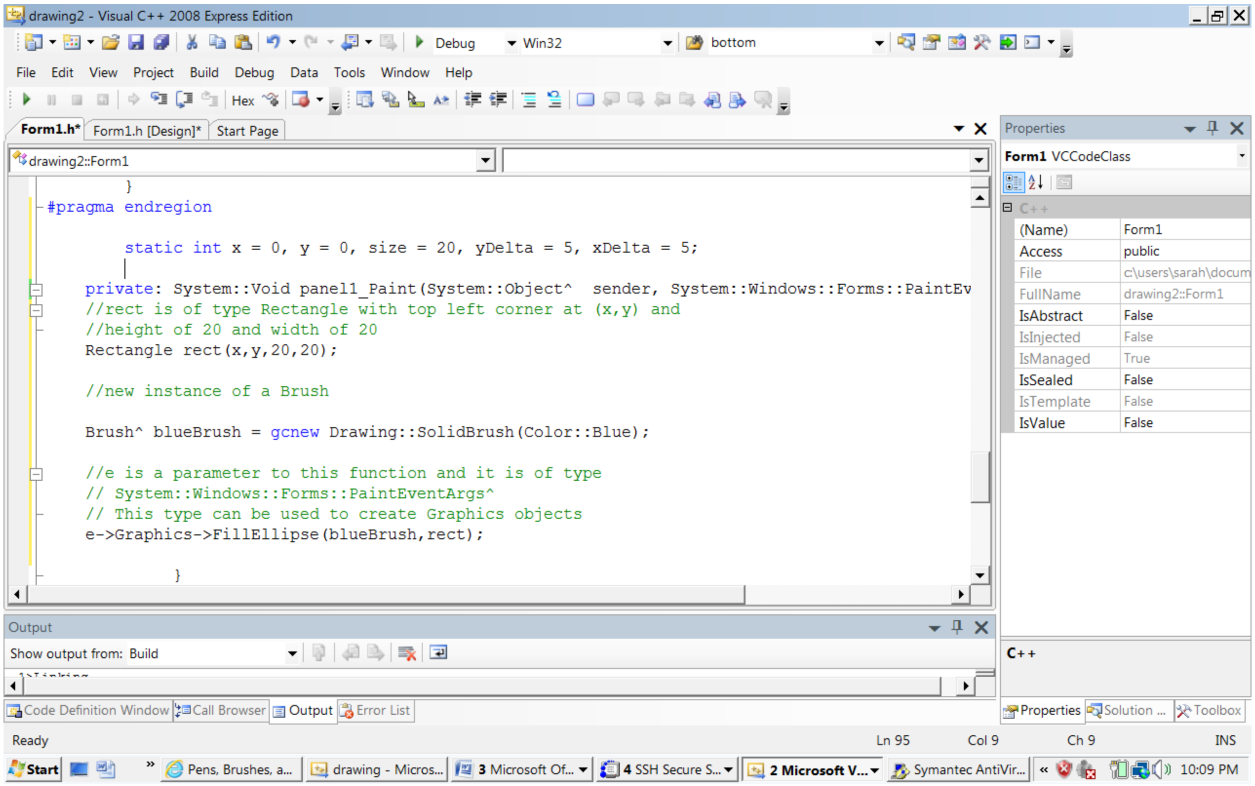
```
                this->Name = L"Form1";
                this->Text = L"Form1";
                this->ResumeLayout(false);

            }
#pragma endregion
            static int x = 20, y = 25, xsize = 30, ysize=30, yDelta = 5, xDelta = 5;
    private: System::Void panel1_Paint(System::Object^  sender, System::Windows::Forms::PaintEv
                 }
    };
}
```

```
1>Linking...
1>Embedding manifest...
1>Build log was saved at "file://c:\Users\Sarah\Documents\Visual Studio 2008\Projects\drawing2\drawing2\Debug\BuildLog.htm"
1>drawing2 - 0 error(s), 0 warning(s)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

(Name)	Form1
Access	public
File	c:\users\sarah\docum
FullName	drawing2::Form1
IsAbstract	False
IsInjected	False
IsManaged	True
IsSealed	False
IsTemplate	False
IsValue	False

8. Inside the Paint event handler for the panel type the following:

```
 //rect is of type Rectangle with top left corner at (x,y) and
//height of 20 and width of 20

Rectangle rect(x,y,20,20);

//new instance of a Brush

Brush^ blueBrush = gcnew Drawing::SolidBrush(Color::Blue);

//e is a parameter to this function and it is of type
// System::Windows::Forms::PaintEventArgs^
// This type can be used to create Graphics objects
e->Graphics->FillEllipse(blueBrush,rect);
```

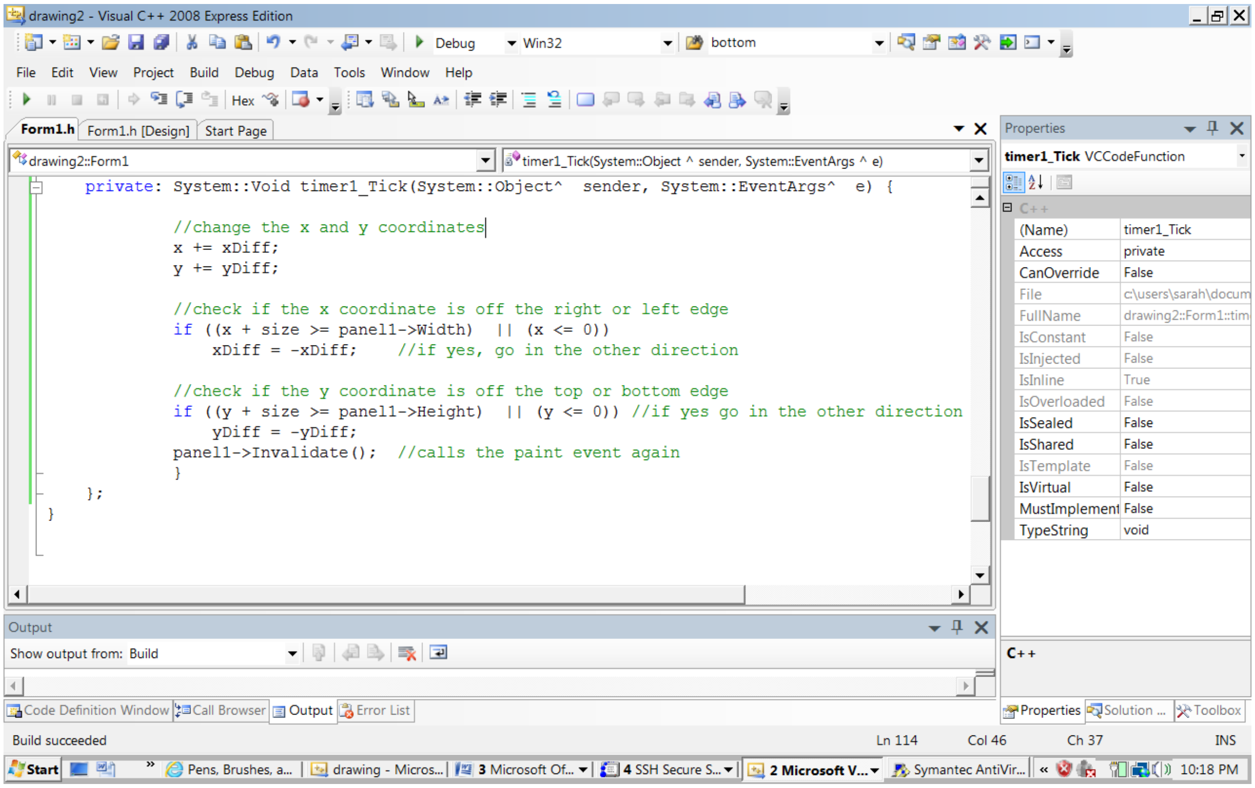


```
        }
#pragma endregion

        static int x = 0, y = 0, size = 20, yDelta = 5, xDelta = 5;

    private: System::Void panel1_Paint(System::Object^  sender, System::Windows::Forms::PaintEv
    //rect is of type Rectangle with top left corner at (x,y) and
    //height of 20 and width of 20
    Rectangle rect(x,y,20,20);

    //new instance of a Brush

    Brush^ blueBrush = gcnew Drawing::SolidBrush(Color::Blue);

    //e is a parameter to this function and it is of type
    // System::Windows::Forms::PaintEventArgs^
    // This type can be used to create Graphics objects
    e->Graphics->FillEllipse(blueBrush,rect);

            }
```

(Name)	Form1
Access	public
File	c:\users\sarah\docum
FullName	drawing2::Form1
IsAbstract	False
IsInjected	False
IsManaged	True
IsSealed	False
IsTemplate	False
IsValue	False

9. Press CTL-F5 to run your program. You should see a blue ball in the top left corner of your panel.
10. Double click on the timer to open the code for the event. Type the following into the body of the function:

```
//change the x and y coordinates
x += xDiff;
y += yDiff;

//check if the x coordinate is off the right or left edge
if ((x + size >= panel1->Width)  || (x <= 0))
      xDiff = -xDiff;     //if yes, go in the other direction

//check if the y coordinate is off the top or bottom edge
if ((y + size >= panel1->Height)  || (y <= 0))
           yDiff = -yDiff; //if yes go in the other direction

 //causes calls the paint event again

// so the ball will be painted in the new x and y spot

panel1->Invalidate();
```



drawing2 - Visual C++ 2008 Express Edition

```
    private: System::Void timer1_Tick(System::Object^  sender, System::EventArgs^  e) {

             //change the x and y coordinates
             x += xDiff;
             y += yDiff;

             //check if the x coordinate is off the right or left edge
             if ((x + size >= panel1->Width)  || (x <= 0))
                 xDiff = -xDiff;    //if yes, go in the other direction

             //check if the y coordinate is off the top or bottom edge
             if ((y + size >= panel1->Height)  || (y <= 0)) //if yes go in the other direction
                 yDiff = -yDiff;
             panel1->Invalidate();  //calls the paint event again
             }
    };
}
```

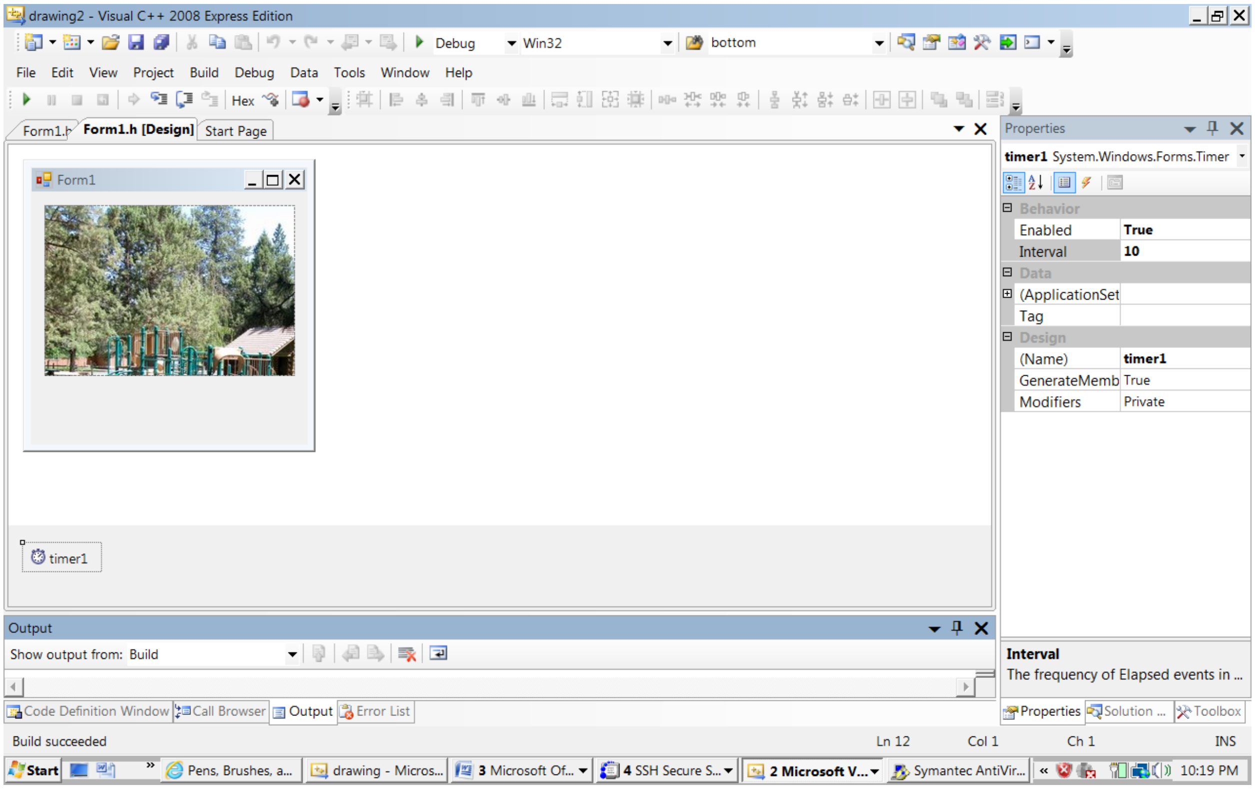
timer1_Tick VCCodeFunction

C++	
(Name)	timer1_Tick
Access	private
CanOverride	False
File	c:\users\sarah\docum
FullName	drawing2::Form1::tim
IsConstant	False
IsInjected	False
IsInline	True
IsOverloaded	False
IsSealed	False
IsShared	False
IsTemplate	False
IsVirtual	False
MustImplement	False
TypeString	void

C++

Build succeeded   Ln 114   Col 46   Ch 37   INS

11. If the ball is moving too slowly, change the Interval propery of the timer, by selecting it, and going to the property window. The Interval is the number of milliseconds between ticks.



drawing2 - Visual C++ 2008 Express Edition

timer1 System.Windows.Forms.Timer

Behavior	
Enabled	True
Interval	10
Data	
(ApplicationSet	
Tag	
Design	
(Name)	timer1
GenerateMemb	True
Modifiers	Private

Interval
The frequency of Elapsed events in ...

Build succeeded   Ln 12   Col 1   Ch 1   INS

12. For your own interest: What if there are obstacles in the way? Place shapes in the panel, and see if you can get the ball to go around them or bounce off them.