HNDIT1012 Visual Application Programming

Week 3

Events in C#

There are several categories of events in C#. We are going to focus on some events in the following categories:

1. Mouse Events
2. Keyboard Events
3. Windows Form Events
4. Timer Events

Some Mouse Events

Event	Description
Click	This event occurs when the mouse button is released, typically before the MouseUp event
MouseClick	This event occurs when the user clicks the control with the mouse.
DoubleClick	This event occurs when the control is double-clicked.
MouseDown	This event occurs when the mouse pointer is over the control and the user presses a mouse button.
MouseEnter	This event occurs when the mouse pointer enters the border or client area of the control, depending on the type of control
MouseMove	This event occurs when the mouse pointer moves while it is over a control.
MouseUp	This event occurs when the mouse pointer is over the control and the user releases a mouse button.
MouseWheel	This event occurs when the user rotates the mouse wheel while the control has focus.

MouseEventArgs Class

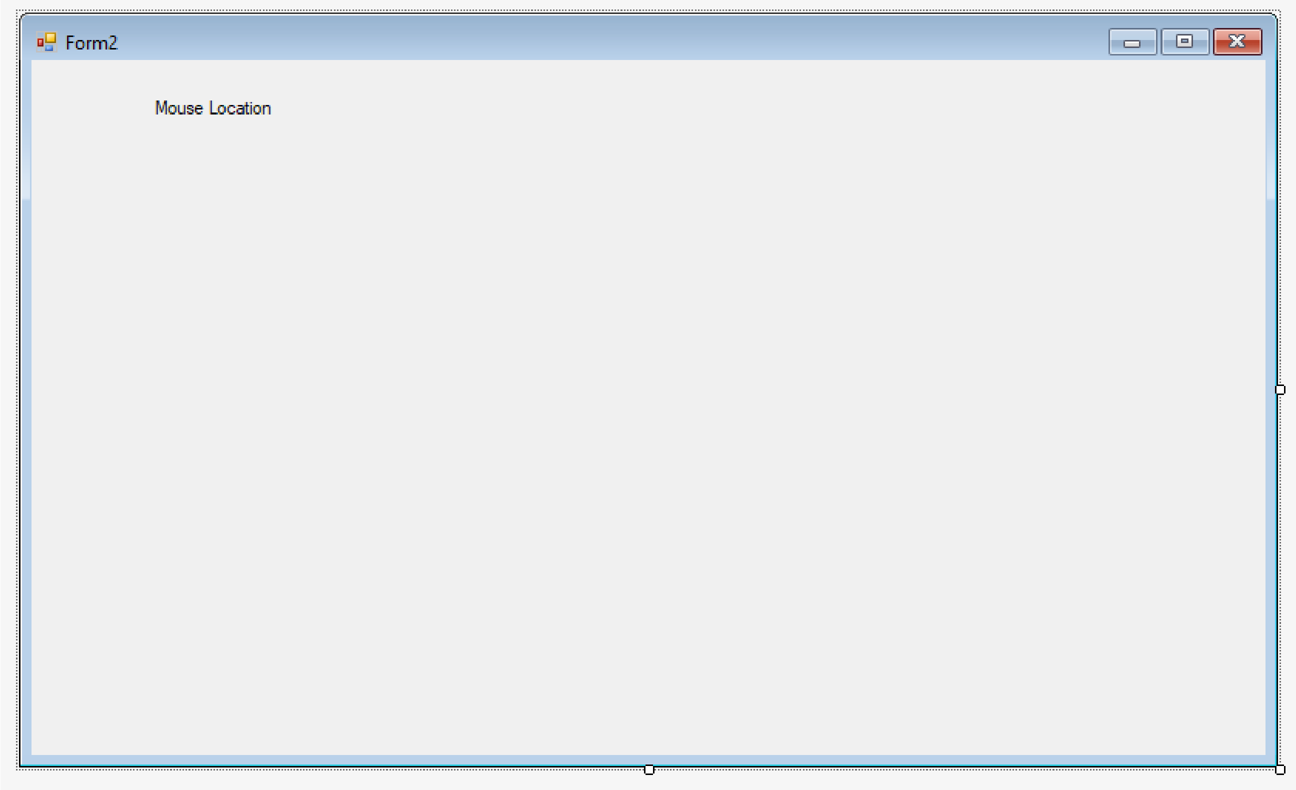
- Provides data for the MouseUp, MouseDown, and MouseMove events.
- A MouseEventArgs specifies which mouse button is pressed, how many times the mouse button was pressed and released, the coordinates of the mouse, and the amount the mouse wheel moved.

Some Properties of MouseEventArgs

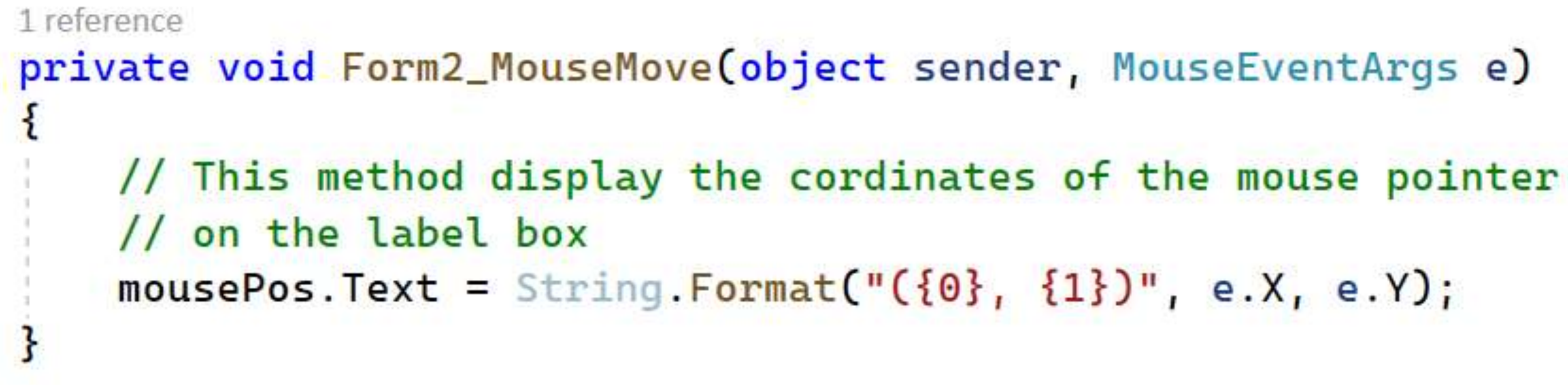
Button	Gets which mouse button was pressed.
Clicks	Gets the number of times the mouse button was pressed and released.
Location	Gets the location of the mouse during the generating mouse event.
X	Gets the x-coordinate of the mouse during the generating mouse event.
Y	Gets the y-coordinate of the mouse during the generating mouse event.

Example

- Create a windows form application and place a label on the form as shown below.



- Open the property window and select the label to view the properties.
- Change the name of the label to “mousePos”.
- Select the mouse move event of the form and type the code as shown in the next slide.



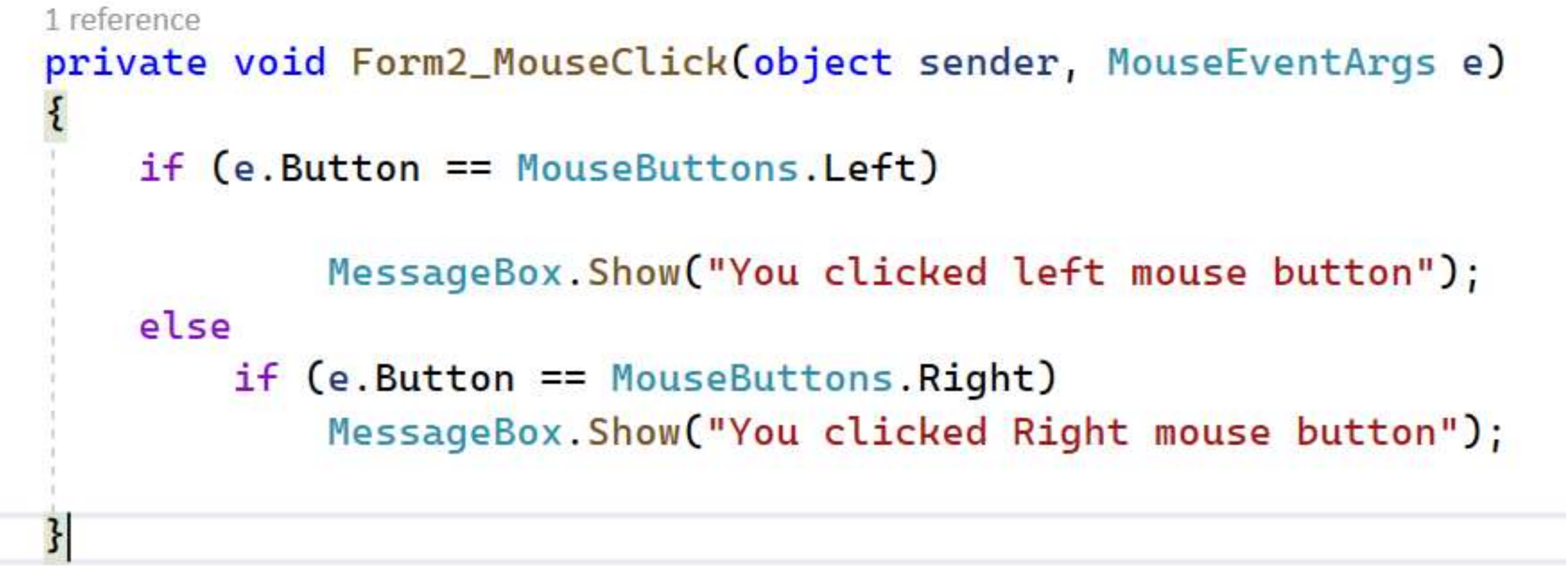
```
1 reference
private void Form2_MouseMove(object sender, MouseEventArgs e)
{
    // This method display the cordinates of the mouse pointer
    // on the label box
    mousePos.Text = String.Format("({0}, {1})", e.X, e.Y);
}
```

Run the application and view the label while moving the mouse over the form.

This method display the current coordinates of the mouse pointer on the label box.

The MouseEventArgs object e contains all the information about the event such as x and y coordinates, button pressed etc..

- Add the mouseclick event for the form as shown below



```
1 reference
private void Form2_MouseClick(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)

            MessageBox.Show("You clicked left mouse button");
    else
        if (e.Button == MouseButtons.Right)
            MessageBox.Show("You clicked Right mouse button");

}
```

Run the project and click on the form by left or right mouse button to view the message box.
MessageBox is used to display any message through a model Dialog.

Keyboard Events

Event	Description
KeyDown	This event is raised when a user presses a physical key. The KeyDown event occurs once.
KeyPress	This event is raised when the key or keys pressed result in a character. The KeyPress event, which can occur multiple times when a user holds down the same key.
KeyUp	This event is raised when a user releases a physical key. The KeyUp event occurs once when a user releases a key.

KeyEventArgs Class

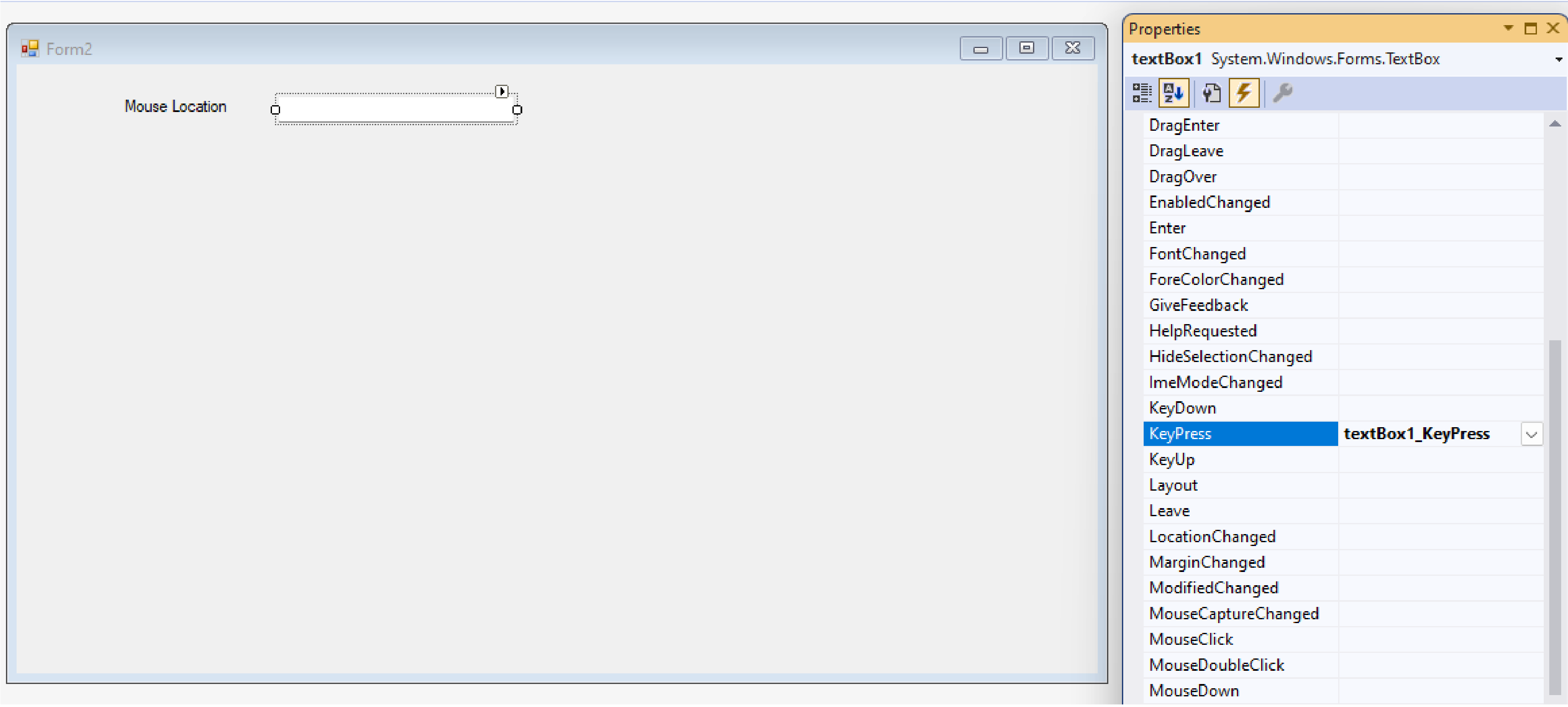
- Provides data for the KeyDown or KeyUp event.
- A KeyEventArgs, which specifies the key the user pressed and whether any modifier keys (CTRL, ALT, and SHIFT) were pressed at the same time, is passed with each KeyDown or KeyUp event.

KeyPressEventArgs Class

- Provides data for the KeyPress event.
- A KeyPressEventArgs specifies the character that is composed when the user presses a key. For example, when the user presses SHIFT + K, the KeyChar property returns an uppercase K.

Example

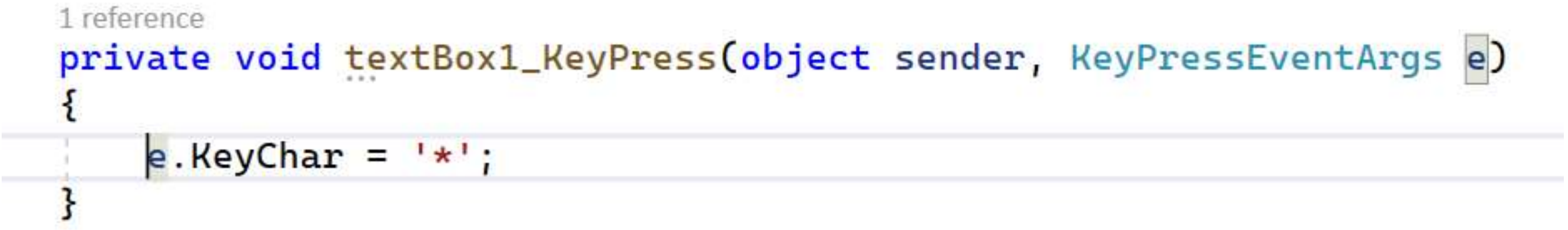
- Insert a text box to the form as shown below



textBox1 System.Windows.Forms.TextBox

KeyPress textBox1_KeyPress

- Select the keyPressed method of the textbox1.
- Type the code as shown below



```
1 reference
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    e.KeyChar = '*';
}
```

- Whatever the key user pressed inside the textbox, the character will be replaced by “*”
- Run the solution and check the result by typing some text in the text box.

Some Windows Form Events

Event	Description
Form Load	Occurs whenever the user load the form
Form Close	Occurs when the form is closed.
Form Shown	Occurs whenever the form is first displayed.

Timer Event

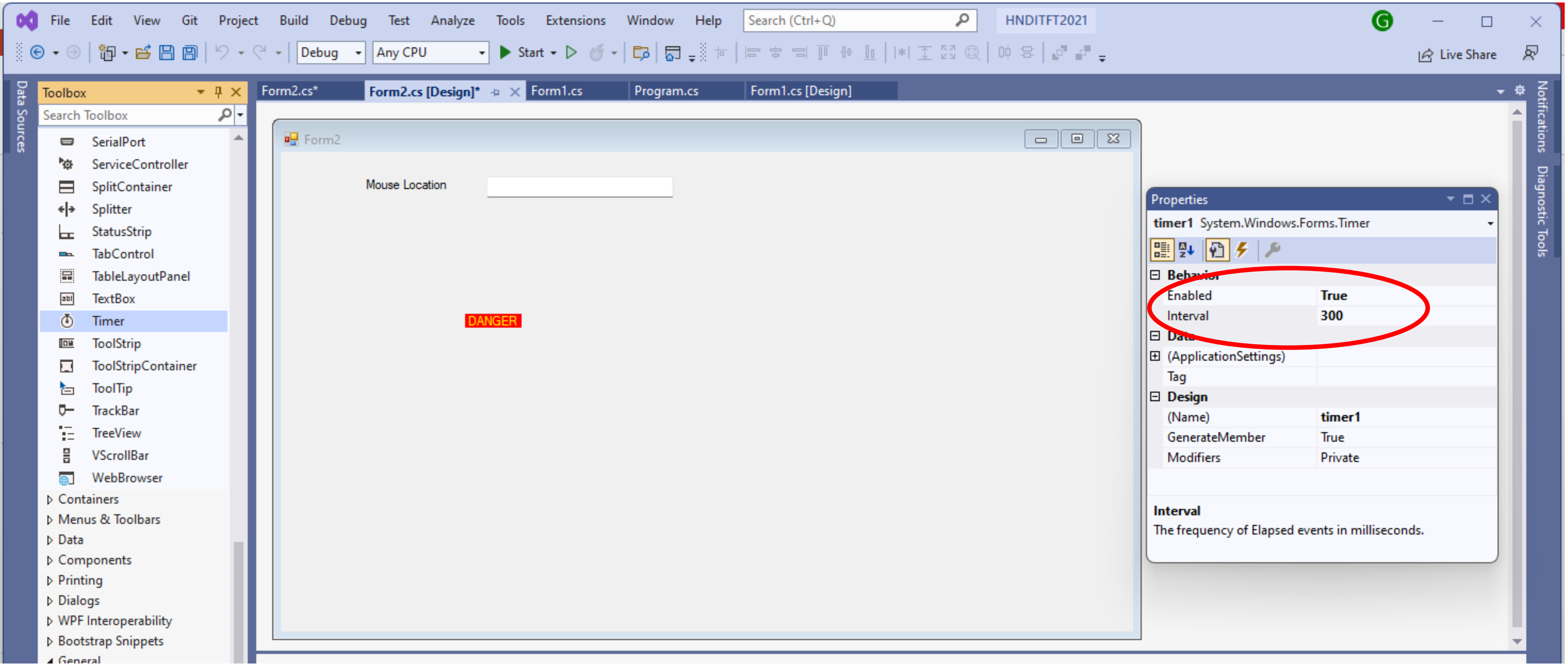
Event	Description
Tick	Occurs when the specified timer interval has elapsed and the timer is enabled.
Elapsed	Occurs when the interval elapses.

Timer.Interval Property
The time, in milliseconds, between Elapsed events. The value must be greater than zero, and less than or equal to Int32.MaxValue. The default is 100 milliseconds.

Timer.Enabled Property
Gets or sets a value indicating whether the Timer should raise the Elapsed event. True if the Timer should raise the Elapsed event; otherwise, false. The default is false.

Example

- Insert another label to the form and change the properties as shown below:
  - Name : lblDanger
  - Text : DANGER
  - Background color : Red
  - Foreground color : Yellow
- Insert a timer and set the timer properties as shown in the next slide.



Properties

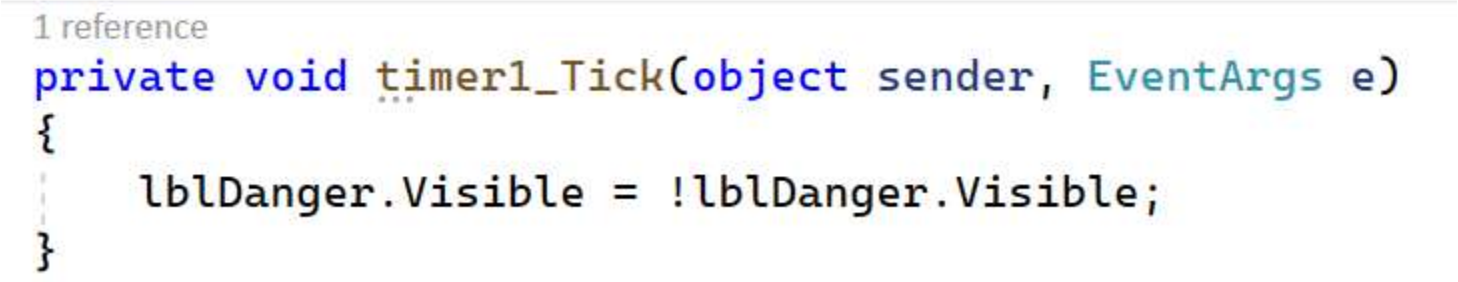
timer1 System.Windows.Forms.Timer

⊟ Behavior	
Enabled	**True**
Interval	**300**
⊟ Data	
⊞ (ApplicationSettings)	
Tag	
⊟ Design	
(Name)	**timer1**
GenerateMember	True
Modifiers	Private

Interval

The frequency of Elapsed events in milliseconds.

Double click on timer1 to open the tick event and type the code as shown below:



```
1 reference
private void timer1_Tick(object sender, EventArgs e)
{
    lblDanger.Visible = !lblDanger.Visible;
}
```

- Run the programe to view the blinking DANGER label.
- Visible property will be complemented for every 300 milliseconds. (visible property will switch the values true and false for every 300 milliseconds)

Thank You