KEYBOARD EVENTS

- Several Visual C# Express controls can recognize keyboard events, notably the form and the text box. Yet, only the control that has focus can receive a keyboard event. (the control with focus is the active control.) When trying to detect a keyboard event for a certain control, we need to make sure that control has focus. We can give a control focus by clicking on it with the mouse. But, another way to assign focus to a control is with the Focus method. The format for such a statement is:

  controlName.Focus(); This command in C# will give controlName focus and make it the active control.

- It has the same effect as clicking on the control. The control can then recognize any associated keyboard events. We use the Focus method with keyboard events to insure proper execution of each event.

- To detect keyboard events on the form, you need to set the form **KeyPreview** property to **True**. This bypasses any keystrokes used by the controls to generate events.

KEYDOWN EVENT

- The **KeyDown** event has the ability to detect the pressing of any key on the computer keyboard.

- The KeyDown event has two arguments: **sender** and **e**.

- **Sender** refers to the object that invoked the event while **e** contains additional information about the event.

- The property **e.KeyCode** can be used to determine which key was pressed. Some Examples:

  - Keys.Enter
  - Keys.Up
  - Keys.Space

- Using the KeyDown event can be challenging. For example, the KeyDown event cannot distinguish between an upper and lower case letter.
- Remember, for a keyboard even to be detected, the corresponding control MUST have **focus**.

KEYPRESS EVENTS

- The **KeyPress** event is similar to the KeyDown event, with one distinction. Many characters in the world of computers have what are called Unicode values. (Arrow Keys don't work with KeyPress Events)

- **Unicode** values are simply numbers (ranging from 0 to 255) that represent all letters (upper and lower case), all numbers, all punctuation, and many special keys like Esc, Space, and Enter.

- The KeyPress event can detect the pressing of any key that has a corresponding Unicode code

- Unicode values are related to **ASCII** (askey) values that you may have seen before in other languages.

- The information is in the value of **e.KeyChar**.

- **e.KeyChar** is a char type variable, returning a single character, corresponding to the pressed key.

- The pressed key can be a readable character (letter, number, punctuation) or a non-readable character (Esc, Enter).

- For the non-readable characters, known as control keys, we can use the corresponding Unicode value.

- Two values we will use are:

Definition	Unicode Value
Backspace	8
Enter	13

- A character (**char**) type variable is enclosed in a pair of **single quotes** NOT double quotes like strings.

- If using a Unicode value, you must **cast** the KeyChar to an **int** value.

```
if (e.KeyChar == 'b')        //user pressed the lower case b
if ((int)e.KeyChar == 98)    //user pressed the lower case b

if ((int)e.KeyChar == 13)    //user pressed the ENTER Key

//values between 0 and 9
if (e.KeyChar >= '0' && e.KeyChar <= '9')
```

- Not every key has a Unicode value, for example the **Arrow** keys. The KeyDown event must be used for Arrow Keys.

KEY TRAPPING

- **Key Trapping** is the process of detecting and ignoring unwanted key strokes.

- By comparing the input **e.KeyChar** with acceptable values, we can decide if we want to accept that value as input.

- By using the **e.Handled** property, we can ignore a pressed key we decide is **NOT** acceptable.

- If an unacceptable key is detected, we set **e.Handled** to **true**.

- If a pressed key is acceptable, we set the e.Handled property to **false**. This tells Visual C# that this method has not been handled and the KeyPress should be allowed (by default, e.Handled is false allowing all keystrokes.

- The method below will only accept a typed value from 0 to 9. Any non-numerical values will be ignored and NOT appear in the text control

```
Private void txtAnswer_KeyPress(object sender, ....

    if (e.KeyChar < '0' || e.KeyChar > '9')
    {
        e.Handled = true;
        lblMessage.Text = "Not a number";
    }
    else
    {
        e.Handled = false;
        lblMessage.Text = "You entered a number";
    }
```

COMMON UNICODE VALUES

a	97	g	103	m	109	s	115	y	121
A	65	G	71	M	77	S	83	Y	89
b	98	h	104	n	110	t	116	z	122
B	66	H	72	N	78	T	84	Z	90
c	99	i	105	o	111	u	117		
C	67	I	73	O	79	U	85		
d	100	j	106	p	112	v	118		
D	68	J	74	P	80	V	86		
e	101	k	107	q	113	w	119		
E	69	K	75	Q	81	W	87		
f	102	l	108	r	114	x	120		
F	70	L	76	R	82	X	88		
0	48	6	54	F1	112	F7	118	Space	32
1	49	7	55	F2	113	F8	119	BackSpace	8
2	50	8	56	F3	114	F9	120	Enter	13
3	51	9	57	F4	115	F10	121	Shift	16
4	52			F5	116	F11	122	Ctrl	17
5	53			F6	117	F12	123	Alt	18