

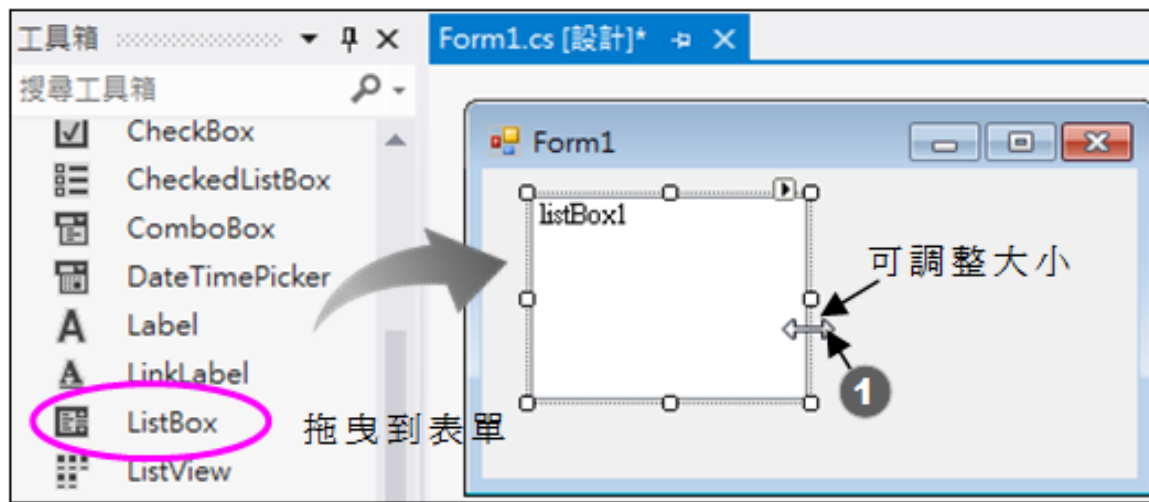


Chapter 8

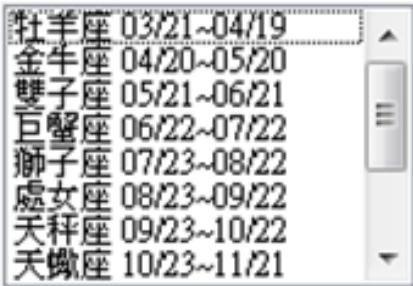

Listed Control items

8-1 ListBox Control Item

- Offer functions for choosing:
 - ① single or multiple choice
 - ② show in multiline
 - ③ insert into ListBox
 - ④ automatic sorting by alphabet



ListBox Properties

Property	Description
Text	Save selected item during runtime
Sorted	True: sort by alphabets' order False: no sorting (default)
MultiColumn	<p>Show in multiline or not, default is False</p> <p>False</p>  <p>True</p> 
Items	<p>Get or set all items of ListBox.</p> <p>Ex: get the first item: 1stTest.Items[0].ToString();</p>

Property	Description
Items.Count	Number of items in ListBox
SelectionMode	<p>One: only one can be selected (default)</p> <p>None: cannot be selected</p> <p>MultiSimple: multiple choices without Shift or Ctrl</p> <p>MultiExtended: multiple choices, Shift or Ctrl is required</p>
SelectedIndex	Get or set the index of selected item, start from 0
SelectedIndices	Get or set the collection of selected items' indexes if SelectionMode is set to MultiSimple or MultiExtended
SelectedItem	Get or set the selected item
SelectedItems	Get or set the collection of selected items if SelectionMode is set to MultiSimple or MultiExtended

Add Item in ListBox

The image shows a Visual Studio IDE with two windows open:

- 字串集合編輯器 (String Collection Editor):** This dialog box is used to edit the items in a string collection. It contains a list of zodiac signs and their date ranges. At the bottom, there are '確定' (OK) and '取消' (Cancel) buttons.
- 屬性 (Properties) window:** This window shows the properties of the selected control, 'listBox1'. The 'Items' property is highlighted, showing it is a collection. A pink oval highlights the 'Items' property and the 'Add' button (a small square with a plus sign) next to it.

Annotations with numbers indicate the steps to add an item:

- 1:** Click the 'Add' button (small square with a plus sign) next to the 'Items' property in the Properties window.
- 2:** Click the 'OK' button in the String Collection Editor dialog.
- 3:** Click the 'Add' button (small square with a plus sign) next to the 'Items' property in the Properties window.

Text overlay: 輸入完畢按 鍵 新增一個項目

Add Item in ListBox in Runtime

1. Add() Method

add a string item to the end of ListBox

Grammar

```
controllItemName.Items.Add(stringItem);
```

Ex: append “iPhone” and “iPad mini” to the end of 1stApple ListBox, usage:

```
1stApple.Items.Add("iPhone");  
1stApple.Items.Add("iPad mini");
```

2. AddRange() Method

add all elements of the string array to the ListBox

Grammar

```
controllItemName.Items.AddRange(stringArrayName);
```

Ex: add 4 elements of string array ItemName to 1stApple ListBox, usage:

```
string[] ItemName=new string[]{ "iPad" , "iPhone", "iMac" , "iTV" };  
lstApple.Items.AddRange(ItemName);
```

3. Insert() method

insert string item into designated index, the item at the original position and the following items are moved 1 position backward

Grammar

```
controllItemName.Items.Insert(index, stringItem);
```

Ex: there are 3 items called “iPad”, “iPhone” and “iTV” in the ListBox. Insert “iMac” into the position with index 2, the item “iTV” at the original position is moved backward. Usage:

```
lstApple.Items.Insert(2, "iMac",);
```

Result: "iPad" 、 "iPone" 、 "iMace" 、 "iTV" 。

Remove Item from ListBox in Runtime

1. Clear() method remove all items from ListBox

Grammar

```
controllItemName.Items.Clear();
```

Ex: remove all items from 1stApple, usage:

```
1stApple.Items.Clear();
```

2. Remove() method

remove designated string item from ListBox, the following items move 1 position forward

Grammar

```
controlItemName.Items.Remove(stringItem);
```

Ex1: remove “iPad” item from 1stApple ListBox, usage:

```
1stApple.Items.Remove("iPad");
```

Ex2: remove selected item from 1stApple ListBox, usage:

```
1stApple.Items.Remove(1stApple.SelectedItem);
```

3. RemoveAt() method

remove the item with designated index from ListBox

Grammar

```
controllItemName.Items.RemoveAt(index);
```

Ex1: remove 3rd item from 1stApple ListBox, usage:

```
1stApple.Items.RemoveAt(2);
```

Ex2: remove selected-index item from 1stApple ListBox, usage:

```
1stApple.Items.RemoveAt(1stApple.SelectedIndex);
```

Other ListBox Methods

1. SetSelected() Method

set the item of ListBox to selected(true) or unselected(false)

Grammar

```
controllItemName.SetSelected(index, true/false);
```

Ex: set the first item of 1stApple ListBox to selected, usage:

```
1stApple.SetSelected(0, true);
```

2. ClearSelected() Method

cancel all selected items and make them unselected

Grammar

```
controllItemName.ClearSelected();
```

Ex: make all items in 1stApple ListBox unselected, usage:

```
1stApple.ClearSelected();
```

3. GetSelected() Method

**See if the designated index of item is selected or not.
Return true for selected; return false for unselected**

Grammar

```
returnValue = controllItemName.GetSelected(index);
```

Ex: merge all selected items in 1stApple ListBox into sel string variable, usage:

```
string sel = "";  
for (int i=0; i<1stApple.Items.Count; i++) {  
    if (1stApple.GetSelected(i)) {  
        sel += 1stApple.Items[i] + " ~ ";  
    }  
}
```

4. Contains() Method

get whether the designated string is in the ListBox or not. Return true for existence; return false for nonexistence

Grammar

```
returnValue = controlItemName.items.Contains(string);
```

Ex: get whether “iTV” exists in 1stApple controlItemName or not, usage:

```
bool exist = comboBox1.Items.Contains("iTV");
```



ListBox Events

1. SelectedIndexChanged

- Default event
- Triggered when SelectedIndex changes value

Example(apple):

Design a program to show the apps of Apple. Requirements:

1. There are 4 options “Mac”, “iPod”, “iPhone” and “iPad” in the classification when the program starts. Mac is the default option.
2. Put all types of the designated classification into “Type List” when the option of classification is selected.
3. Every classification has corresponding types:
 - ① There are MacBook, Mac mini, iMac and MacPro in Mac classification.
 - ② There are iPod shuffle, iPod nano, iPod touch and iPod classic in iPod classification.
 - ③ There are iPhone 4, iPhone 4S and iPhone 5 in iPhone classification.
 - ④ There are iPad 2 and iPad mini in iPad classification.

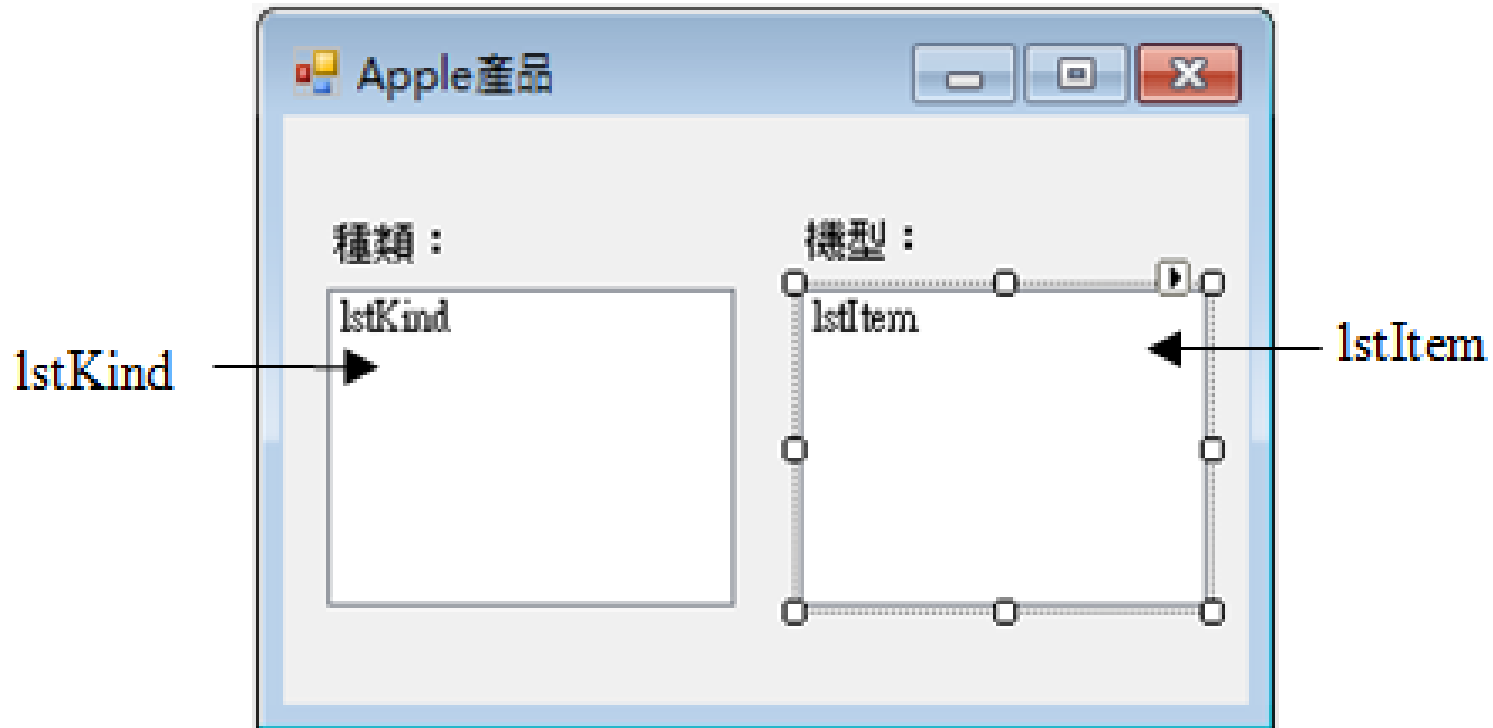
Result:



A screenshot of a web form titled "Apple產品" (Apple Products). The form has two dropdown menus. The first menu, labeled "種類:" (Type), has a list of options: Mac, iPod (highlighted with a blue bar and a mouse cursor), iPhone, and iPad. The second menu, labeled "機型:" (Model), has a list of options: iPod shuffle, iPod nano, iPod touch, and iPod classic.

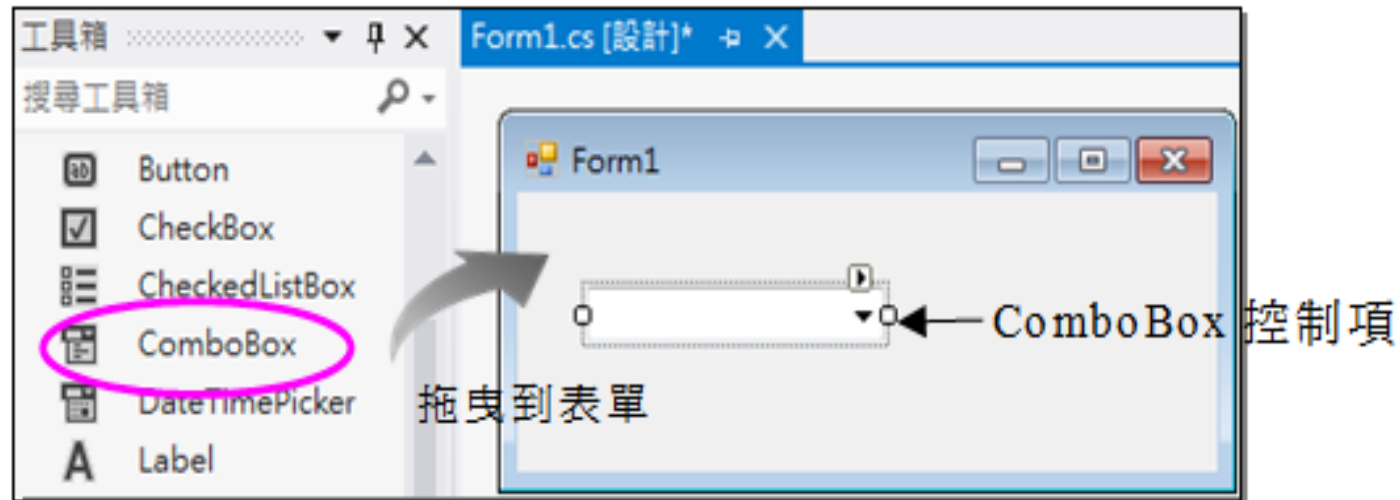
種類 :	機型 :
Mac	iPod shuffle
iPod	iPod nano
iPhone	iPod touch
iPad	iPod classic

Design User Interface




8-2 ComboBox Control Item

- Provide many string items for choosing
- Only allow one item selected
- Input new item during runtime is allowed



ComboBox Properties

Property	Description
Text	Save the selected item or inputted string data during runtime.
MaxDropDownItems	The maximum number of visible items
DropDownStyle	<p>Style of ComboBox</p> <ol style="list-style-type: none">1. DropDown: show items when the button is pressed and customized input is allowed (default)2. DropDownList: show items when the button is pressed and customized input is not allowed3. Simple: always show items and customized input is allowed <p>DropDown : DropDownList : Simple :</p>  <p>The image shows three examples of Windows ComboBox controls. The first, labeled 'DropDown', shows a standard dropdown menu with a blue arrow button on the right and a list of items (玉山, 大霸尖山, 奇萊山, 雪山) that appears when the button is pressed. The second, labeled 'DropDownList', shows a similar control but with a greyed-out text area and a blue arrow button, indicating that only one item can be selected and no custom input is allowed. The third, labeled 'Simple', shows a control where the list of items is always visible within the text area, and custom input is also allowed.</p>



ComboBox Events

1. SelectedIndexChanged

- Default event
- Triggered when SelectedIndex changes value

2. TextChanged

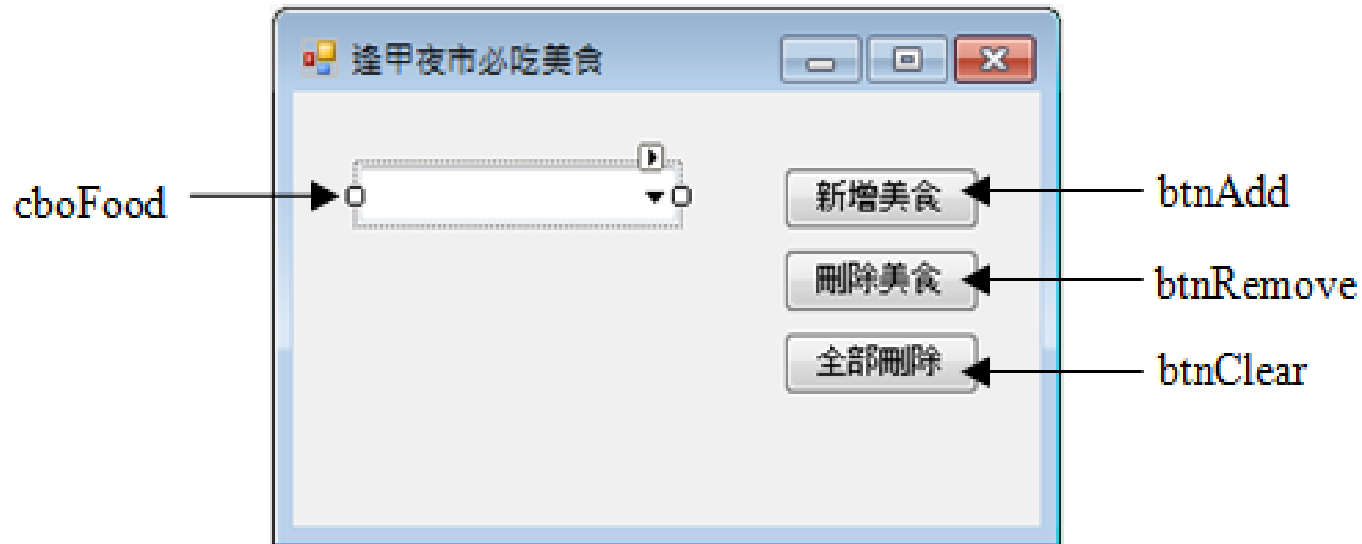
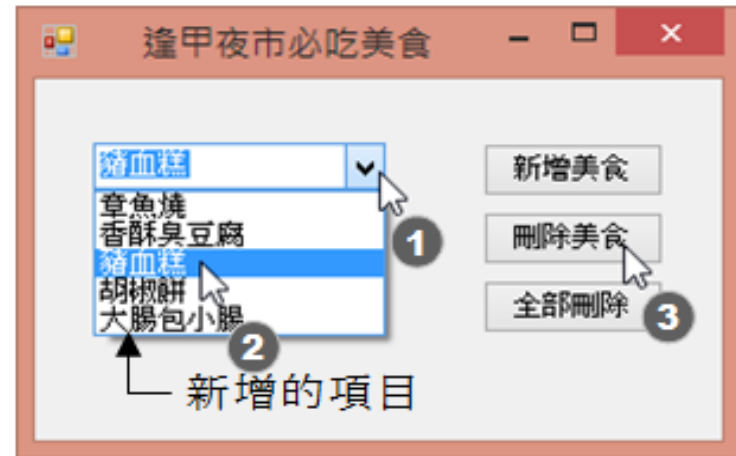
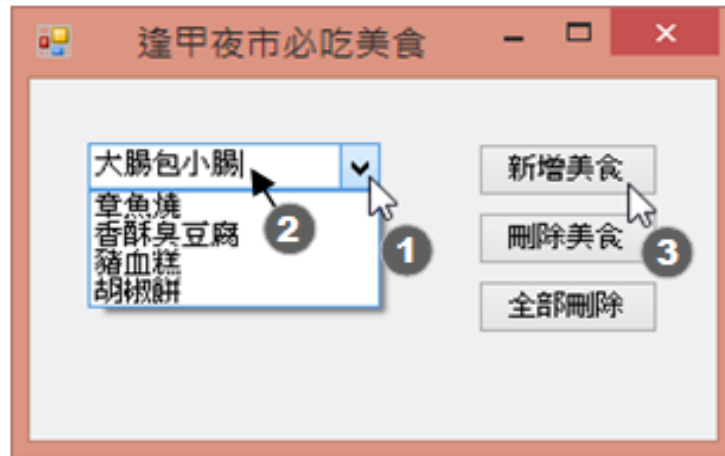
- Triggered when selected item is changed or input value changes Text property
- Trigger TextChanged event \Rightarrow SelectedIndexChanged event in order when the user selects an item

Example(nightMarket):

Design a program for must-eat foods in Fengjia Night Market. Requirements:

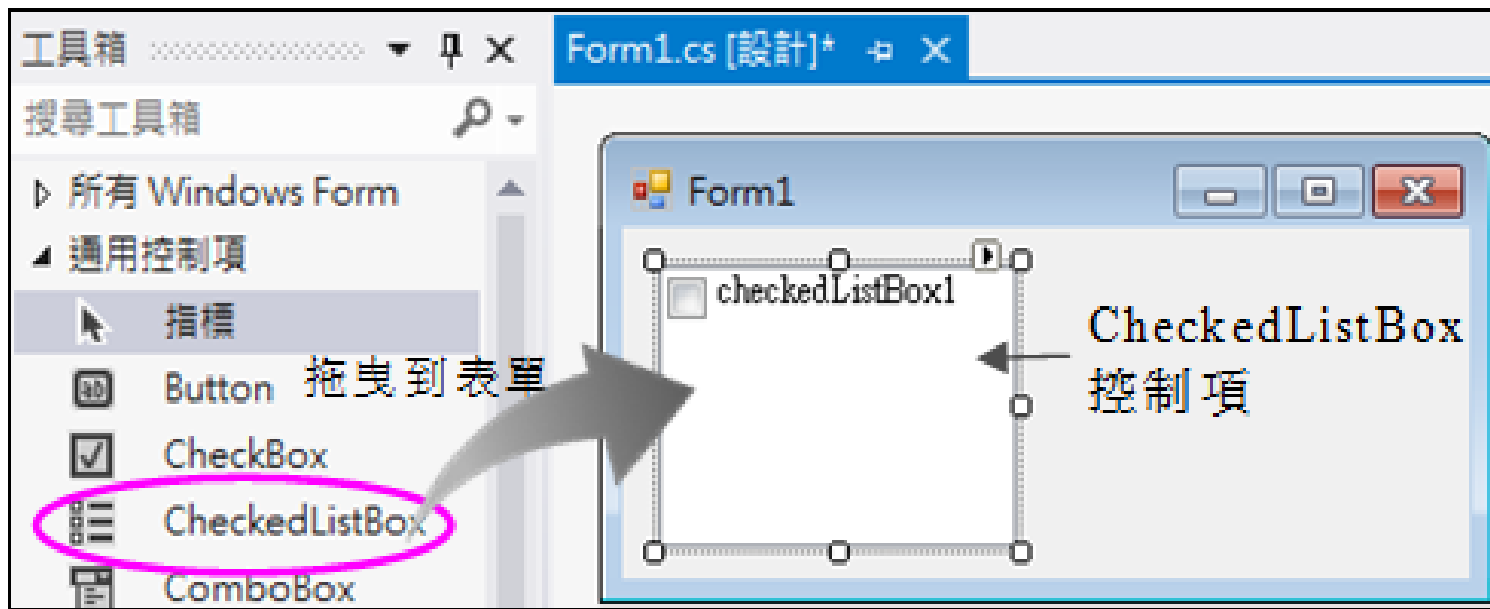
1. There are 4 items called “章魚燒”, “香酥臭豆腐”, “豬米血糕” and “胡椒餅” in ComboBox when the program starts.
2. If inputted food name is not in the menu, press “新增美食” to add to the menu.
3. Press “刪除美食” to delete selected item.
4. Press “全部刪除” to delete all items.

Result:



8-3 CheckedListBox Control Item

- Similar to ListBox
- Additional check box for selection
- Can choose items, but is more suitable for multiple selections



CheckedListBox Properties

Property	Description
CheckOnClick	True: set checked when the item is clicked False: set checked when the item is double-clicked
CheckedItems	Get collection of current selected items during runtime
CheckedIndices	Get index collection of current selected items during runtime

CheckedListBox Methods

1. add() method

add new item and assign check status in the meantime

Grammar

```
controllItemName.Items.Add(stringItem, true|false);
```

Ex: add an item “台中” into c1stCity CheckedListBox and set checked. Add an item “新北” into c1stCity and set unchecked.

```
clstCity.Items.Add("台中", true);  
clstCity.Items.Add("新北", false);
```

2. SetItemChecked() method

set the item with the designated index to be checked or unchecked

Grammar

```
controllItemName.SetItemChecked(index, true|false);
```

Ex: set the 1st item of c1stCity CheckedListBox to be checked. Set the 3rd item of c1stCity to be unchecked, usage:

```
clstCity.SetItemChecked(0,true);  
clstCity.SetItemChecked(2,false);
```

3. GetItemChecked() method

get whether the item with the designated index is checked or not. Return true if checked, return false if unchecked.

Grammar

```
controllItemName.GetItemChecked(index);
```

Ex: set zipCode variable to be “400” when the 1st item of c1stCity is checked, usage:

```
if(c1stCity.GetItemChecked(0))  
{  
    string zipCode = "400";  
}
```



SelectedIndexChanged Event

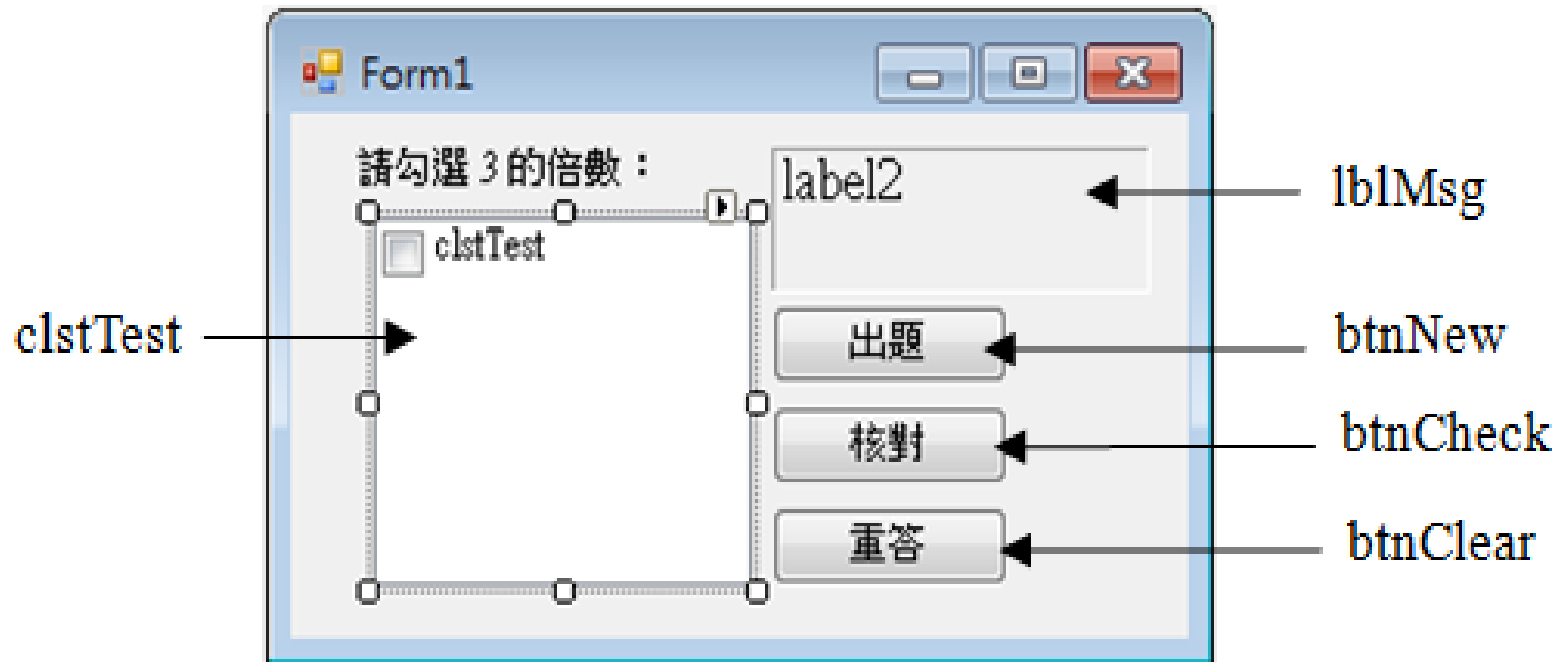
- **Default event of CheckedListBox**
- **Triggered when the user selects an item and changes SelectedIndex property**

Example(multiple):

Design a program to get multiples of 3, requirements:

1. The button “出題” is available when the program starts. The function of button is to generate 6 random between 1 and 99 and put these number into CheckedListBox. All numbers are unchecked.
2. The user can select which numbers are multiples of 3.
3. Press the button “核對” to show how many right answers they are.
4. Press the button “重答” to set all checked options to be unchecked.

Design User Interface



Practice8.1: WinChkLstBx

- Use `CheckedListBox` to create a program that allows users to bet on Big Lotto. When the form is loaded, six numbers that are not repeated from 1 to 49 are generated and stored in the array.
- Users can pick their six lotto numbers by checking the list box.
- If a user doesn't checked the six numbers in the checklist box before pressing the "Talker" button, a dialogue box will pop out a message "Please select 6 numbers!";
- If a user do select numbers that are the same as the program-generated lotto numbers, a message "Congratulations on winning your big prize..." will appear on the label. On the contrary, nothing shown on the label refers to "Tough luck! Please keep it up..."; When you press the "Clear" button, the check box 1-49 option will be set to unchecked.

Result:

Form1

Big Lotto-Please select six numbers

<input type="checkbox"/> 1	<input type="checkbox"/> 6	<input type="checkbox"/> 11	<input type="checkbox"/> 16	<input type="checkbox"/> 21	<input type="checkbox"/>
<input type="checkbox"/> 2	<input type="checkbox"/> 7	<input type="checkbox"/> 12	<input type="checkbox"/> 17	<input type="checkbox"/> 22	<input type="checkbox"/>
<input type="checkbox"/> 3	<input type="checkbox"/> 8	<input type="checkbox"/> 13	<input type="checkbox"/> 18	<input type="checkbox"/> 23	<input type="checkbox"/>
<input type="checkbox"/> 4	<input type="checkbox"/> 9	<input type="checkbox"/> 14	<input type="checkbox"/> 19	<input type="checkbox"/> 24	<input type="checkbox"/>
<input type="checkbox"/> 5	<input type="checkbox"/> 10	<input type="checkbox"/> 15	<input type="checkbox"/> 20	<input type="checkbox"/> 25	<input type="checkbox"/>

< >

not drawing the winning numbers of Lottery in current period

check the numbers Clear Close

output

Form1

Big Lotto-Please select six numbers

<input type="checkbox"/> 1	<input type="checkbox"/> 6	<input checked="" type="checkbox"/> 11	<input type="checkbox"/> 16	<input type="checkbox"/> 21	<input type="checkbox"/> 2
<input type="checkbox"/> 2	<input type="checkbox"/> 7	<input type="checkbox"/> 12	<input type="checkbox"/> 17	<input type="checkbox"/> 22	<input type="checkbox"/> 2
<input type="checkbox"/> 3	<input checked="" type="checkbox"/> 8	<input type="checkbox"/> 13	<input type="checkbox"/> 18	<input checked="" type="checkbox"/> 23	<input type="checkbox"/> 2
<input type="checkbox"/> 4	<input type="checkbox"/> 9	<input checked="" type="checkbox"/> 14	<input type="checkbox"/> 19	<input checked="" type="checkbox"/> 24	<input type="checkbox"/> 2
<input type="checkbox"/> 5	<input type="checkbox"/> 10	<input checked="" type="checkbox"/> 15	<input type="checkbox"/> 20	<input type="checkbox"/> 25	<input type="checkbox"/> 3

< >

The current Lotto numbers are as follows:
1, 19, 22, 23, 32, 47,
Tough luck! Please keep it up...

check the numbers Clear Close

Drawn no prize

Form1

Big Lotto-Please select six numbers

<input type="checkbox"/> 13	<input checked="" type="checkbox"/> 19	<input type="checkbox"/> 25	<input checked="" type="checkbox"/> 31	<input type="checkbox"/> 37	<input type="checkbox"/> 43
<input type="checkbox"/> 14	<input type="checkbox"/> 20	<input type="checkbox"/> 26	<input type="checkbox"/> 32	<input type="checkbox"/> 38	<input type="checkbox"/> 44
<input type="checkbox"/> 15	<input type="checkbox"/> 21	<input type="checkbox"/> 27	<input type="checkbox"/> 33	<input type="checkbox"/> 39	<input type="checkbox"/> 45
<input type="checkbox"/> 16	<input type="checkbox"/> 22	<input type="checkbox"/> 28	<input checked="" type="checkbox"/> 34	<input type="checkbox"/> 40	<input type="checkbox"/> 46
<input type="checkbox"/> 17	<input type="checkbox"/> 23	<input type="checkbox"/> 29	<input checked="" type="checkbox"/> 35	<input type="checkbox"/> 41	<input type="checkbox"/> 47
<input checked="" type="checkbox"/> 18	<input type="checkbox"/> 24	<input checked="" type="checkbox"/> 30	<input type="checkbox"/> 36	<input type="checkbox"/> 42	<input type="checkbox"/> 48

< >

The current Lotto numbers are as follows:
18, 19, 30, 31, 34, 35,
Congratulations on winning your big prize...

check the numbers Clear Close

Win the Lottery



The End

Take a Break