

# 教學意見調查

---

- 系主任重視的是分數（和科院其他系比較）。
- 但數字對老師改進教學幫助不大。所以請同學們盡量撥空在問卷最後寫下你的意見。
  1. 小考很佔時間（幾乎一節課）。但不考又怕同學累積很多進度，到期中考前才念書。如果不必小考，助教也可省下改考卷的時間。你贊成廢除小考嗎？
  2. 學期前半要教基本的C++語法，再加上**Object-Oriented Programming**。剩下教**MFC**的時間很少，這樣真的足以應付大二所需嗎？是否應獨立出一門課「**Windows Programming**」？應開在幾年級？

# 長干行

## ～李白

妾髮初覆額，折花門前劇；  
郎騎竹馬來，遶床弄青梅。  
同居長干里，兩小無嫌猜。  
十四為君婦，羞顏未嘗開；  
低頭向暗壁，千喚不一回。  
十五始展眉，願同塵與灰；  
常存抱柱信，豈上望夫臺？

十六君遠行，瞿塘濼瀨堆；  
五月不可觸，猿鳴天上哀。  
門前遲行跡，一一生綠苔；  
苔深不能掃，落葉秋風早。  
八月蝴蝶來，雙飛西園草；  
感此傷妾心，坐愁紅顏老。  
早晚下三巴，預將書報家；  
相迎不道遠，直至長風沙。

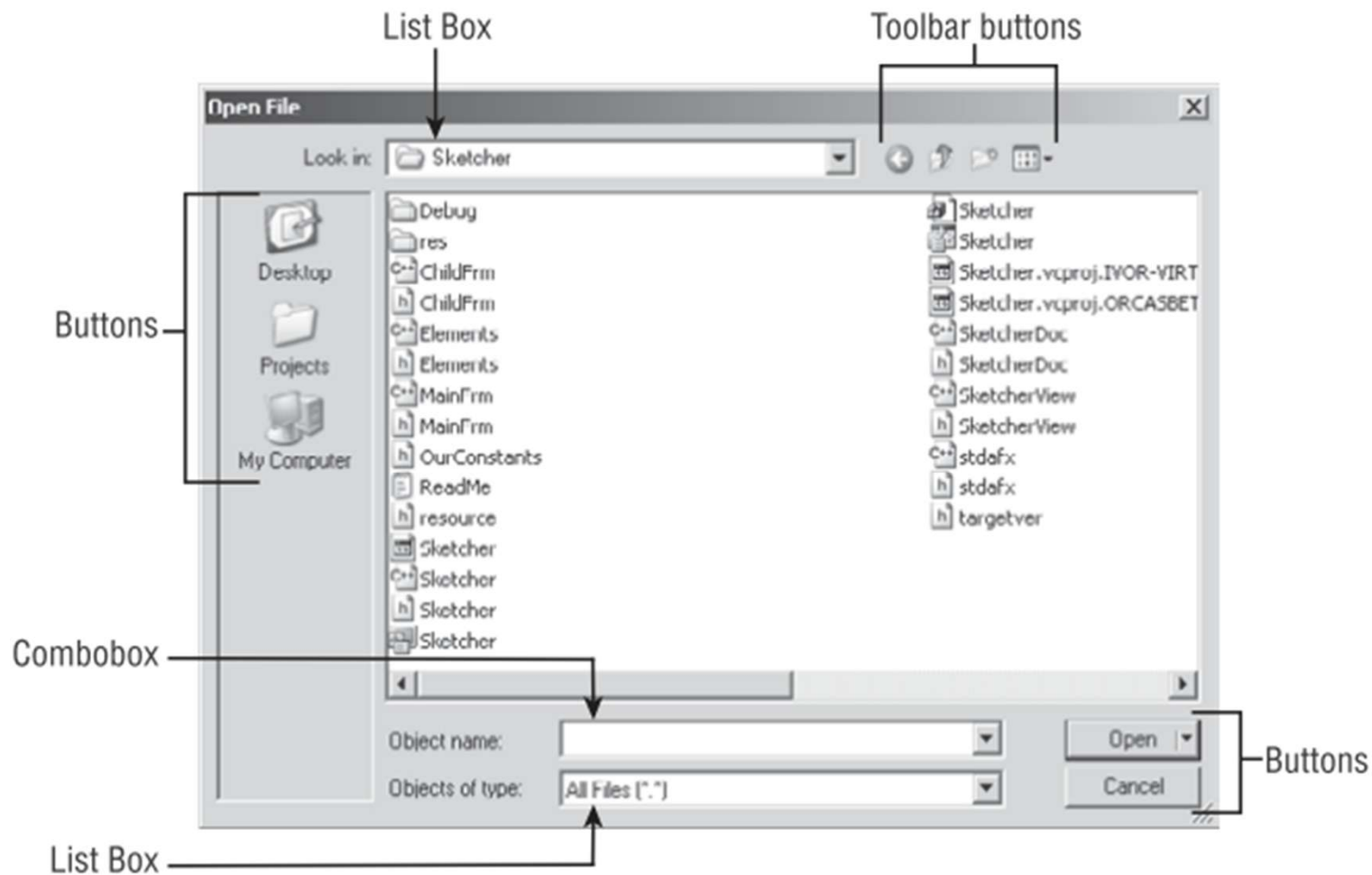
# Chapter 16



## Working with Dialogs and Controls

# Controls in a Dialog Box

File > Open > File



# Common Controls (P.748)

Static controls provide static information, such as titles or instructions, or simply provide decoration in a dialog in the form of an icon or a filled rectangle.

Radio buttons are usually grouped so that if one is checked all the others are unchecked.

Check boxes are individually checked and more than one can be checked at one time.

Buttons can have labels as here and they can also display icons.

You have already seen scroll bars attached to the client area of the Sketcher window. Scroll bars can also be free standing.

A list box presents a predefined list of items from which you can choose. The scroll bar need not be present for a short list. A list can also have multiple columns and be scrolled horizontally. A version of the list box is available that can display icons as well as text.

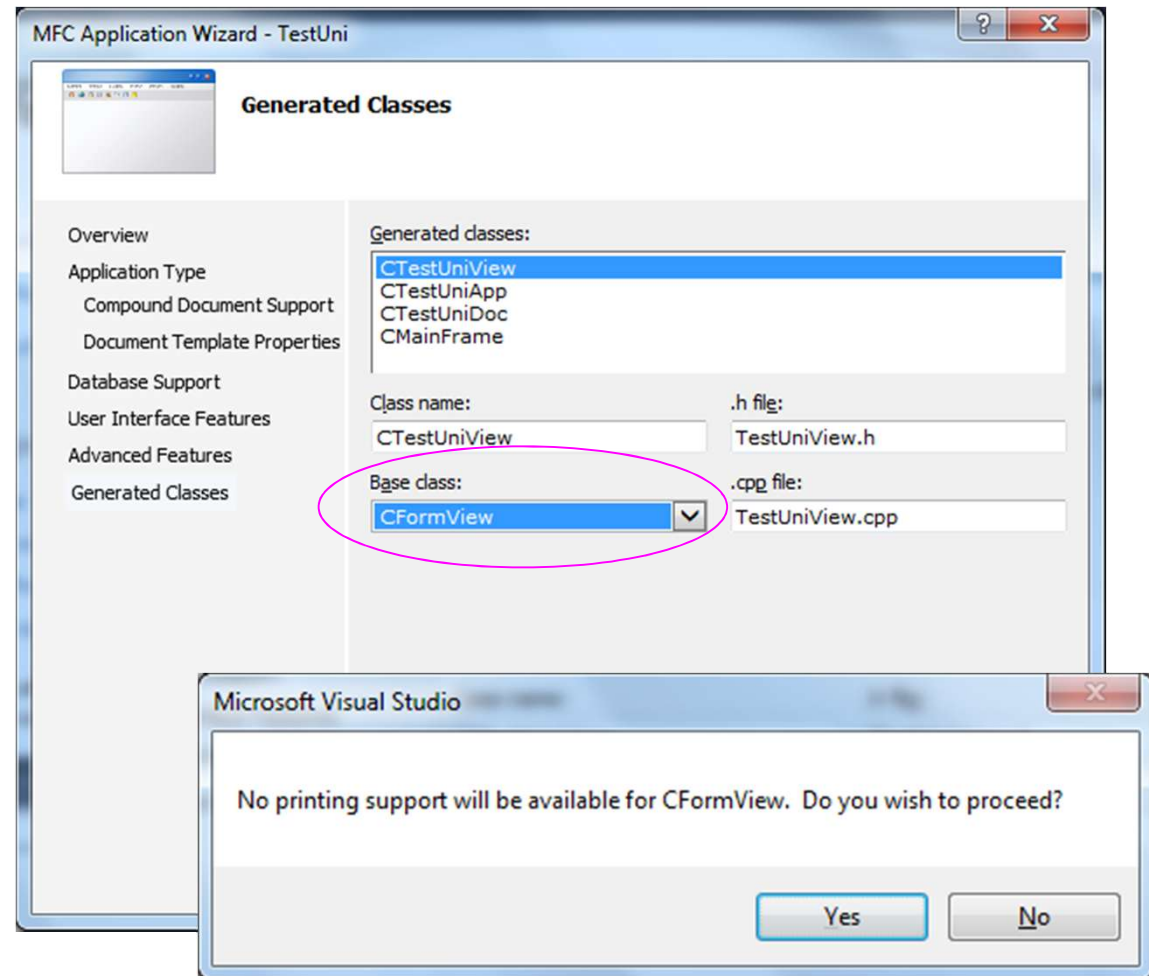
This text box is the simplest form of edit control that allows you to enter and/or edit a line of text. More sophisticated edit controls can display multiple lines of text and support scrolling of the text.

Comboboxes combine the capability of a dropdown list from which you can select with the option of entering data yourself. The Save As... dialog uses a combobox to enable you to enter the file name.

- ❑ Static Controls
- ❑ Button Controls
- ❑ Scrollbars
- ❑ List Boxes
- ❑ Edit Controls
- ❑ Comboboxes

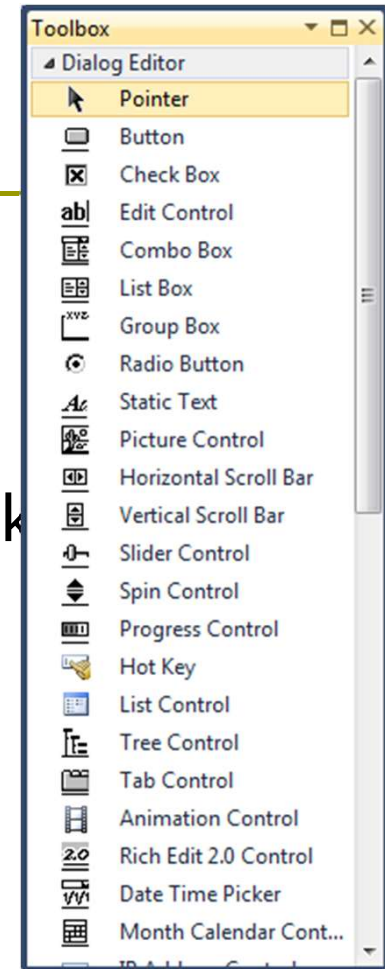
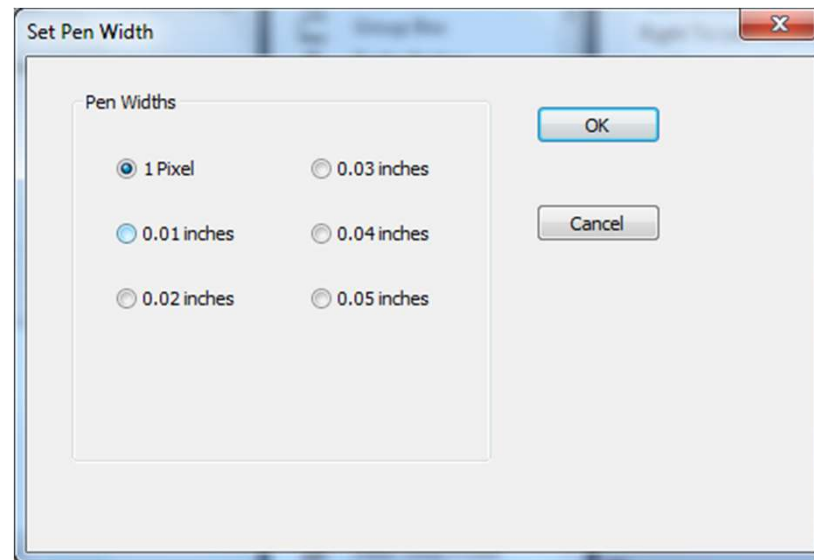
# Create a Project with Controls

- Generate your project based on the **CFormView** class, which provide easy support to controls



# Adding New Controls

- Adding new controls is easy.
  - You can drag the control from the toolbox.
  - You can click the control and then click in the dialog.
  - You can double-click the control.



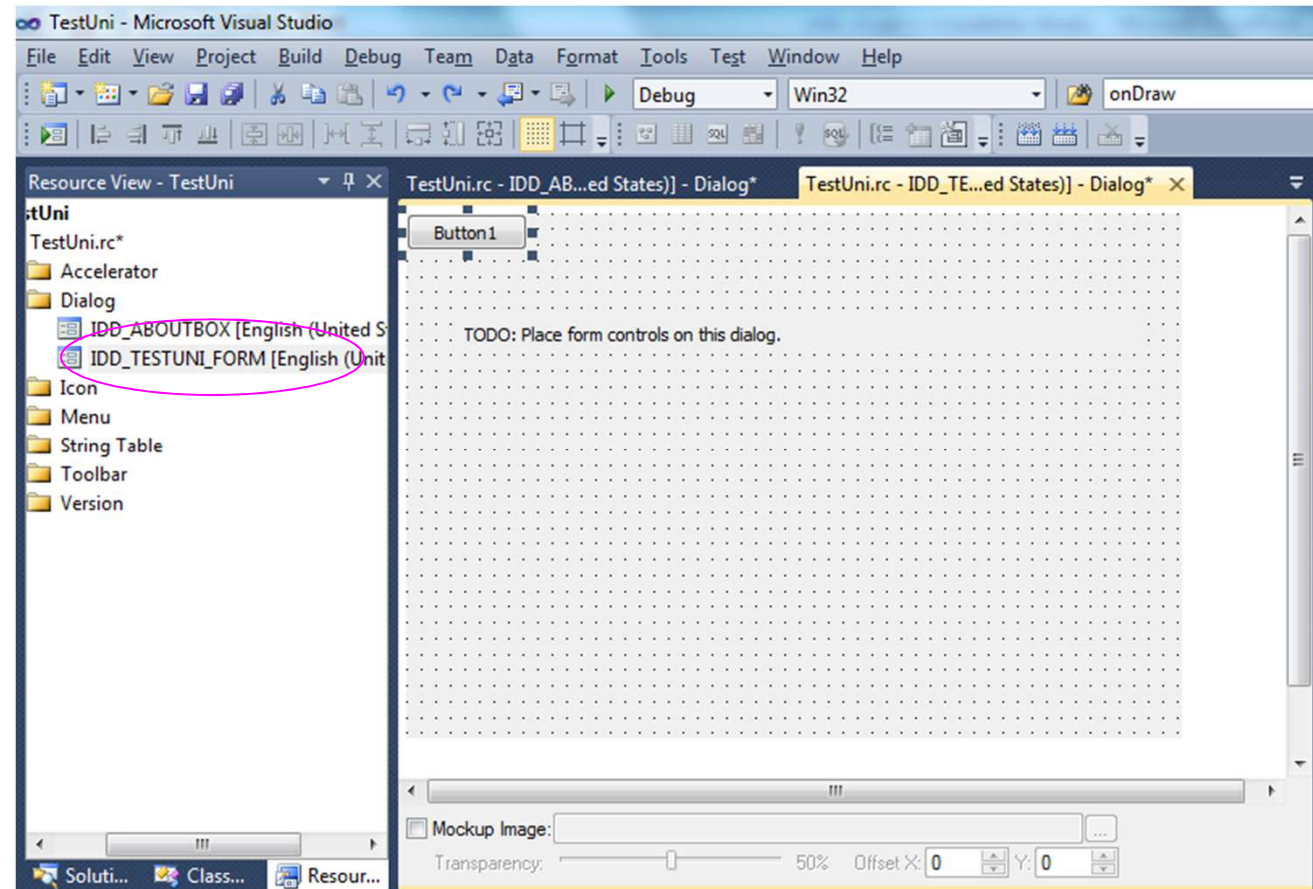
# Change the Button Caption





# Create a Button on the Form

- ❑ Resource View
- ❑ Click the Dialog folder to expand.
- ❑ Click the form to modify its design.



# OnBnClickedButton1 ()

---

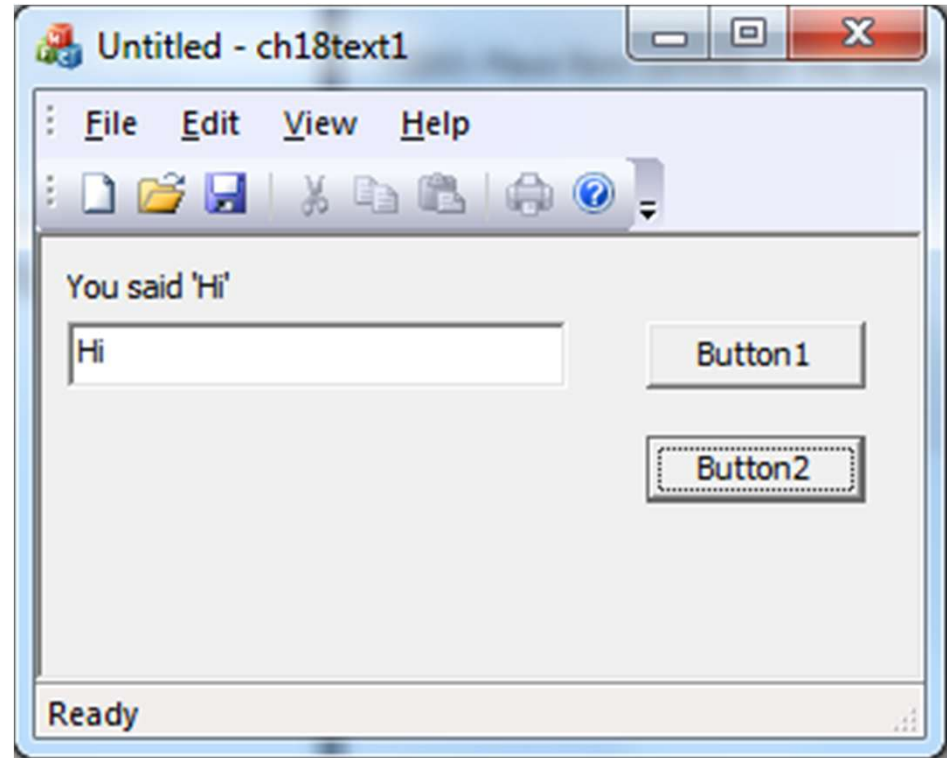
```
void Cch18puzzle1View::OnBnClickedButton1 ()
{
    static bool shown = false;           // P.212

    CWnd* pBtn = GetDlgItem(IDC_BUTTON1);
    if (shown)
        pBtn->SetWindowText( _T("&Bye Bye") );
    else
        pBtn->SetWindowText( _T("H&ello") );
    shown = !shown;
}
```

**Try this!**

# Using an Edit Box Control (P.780)

- ❑ Mouse clicks are convenient for users to make choices, however, when the programs need more detailed information, you'll need to get text input from the user.



# Set the Contents of an Edit Box

---

- ❑ Create an MFC application based on the `CFormView` class.
- ❑ Create an edit box control for input.
- ❑ Double-click the button control to create a message handler

```
void Cch18text1View::OnBnClickedButton1()  
{  
    GetDlgItem(IDC_EDIT1)->SetWindowTextA("Hello");  
}
```

# Get the Contents of an Edit Box

---

```
void Cch18text1View::OnBnClickedButton2 ()
{
    CString str, msg;

    GetDlgItem(IDC_EDIT1) -> GetWindowText (str);
    int i = _ttoi(str);
    msg.Format( _T("The value is %d"), i);

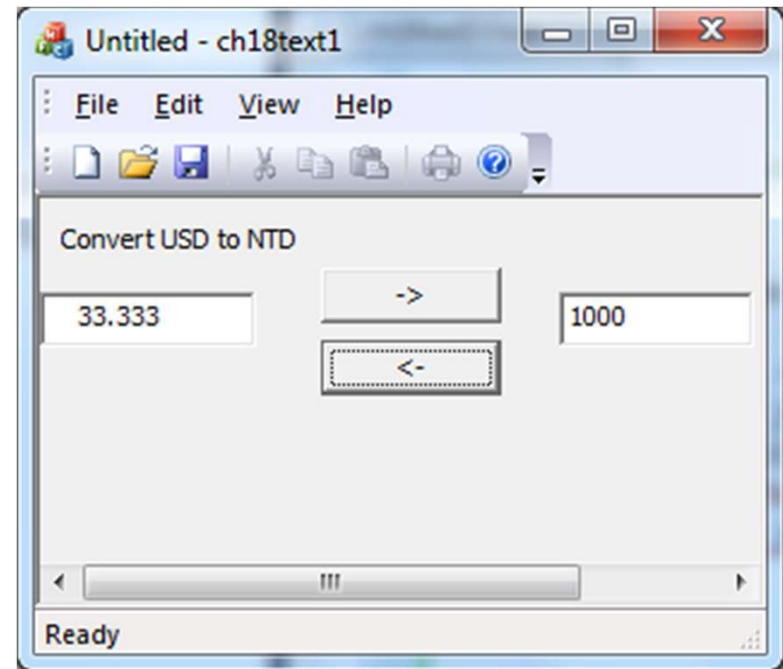
    GetDlgItem(IDC_STATIC) -> SetWindowText (msg);
}
```



Similar to `sprintf(str, "%d", i);`

# Exercise: Currency Converter

- Design an application to convert US dollars to NT dollars (assume  $1 \text{ USD} = 30 \text{ NT dollars}$ ), and vice versa.
- Note that the input may not always be integers.
  - Hint: Use `_ttof()`.



# Initial Contents of an EditBox

---

- Put the following code in `CView::OnInitialUpdate()`

```
CWnd* pEditBox =  
    static_cast<CWnd*>(GetDlgItem(IDC_EDIT1));  
pEditBox->SetWindowText(  
    _T("Please type a string here") );
```

# Properties of an Edit Box Control

---

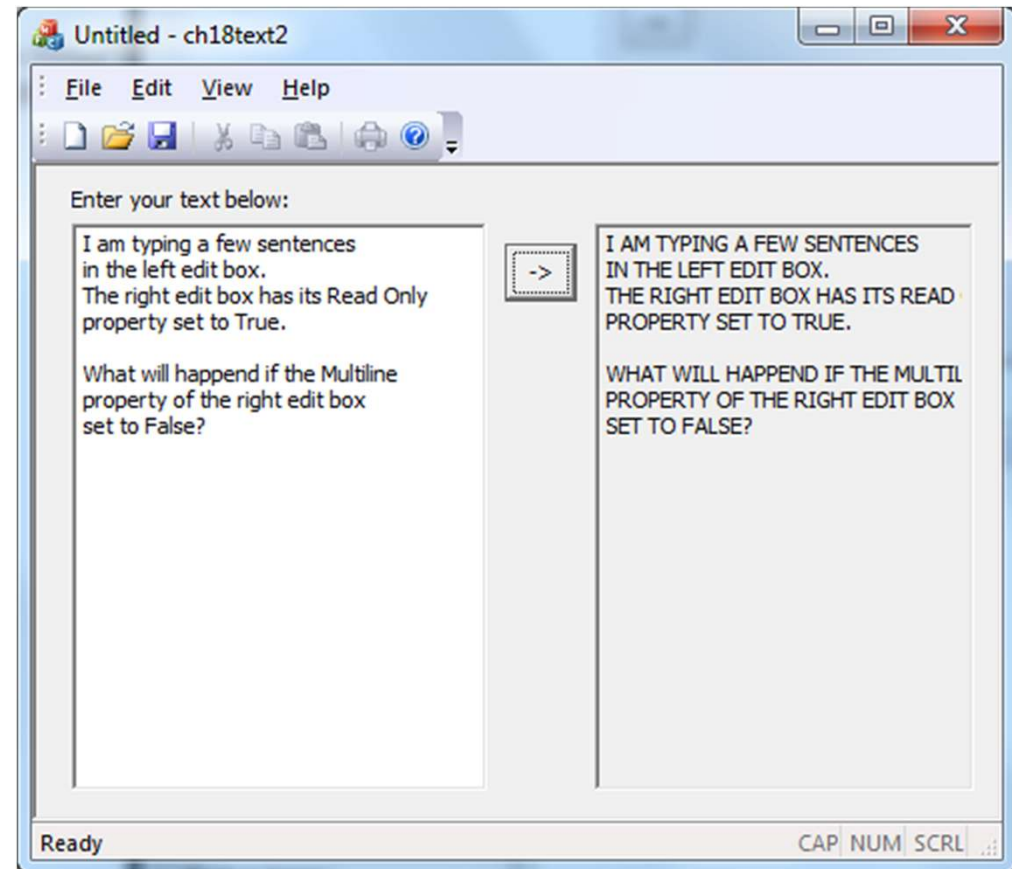
- Multiline
  - The text you enter can span more than one line.
- Align text
  - Left/Center/Right
- Want return
  - Insert a RETURN character into the text string.
  - If False, pressing enter will select the default control (generally the OK button).
- Auto HScroll
- Auto VScroll

You may double-click a property to toggle its value.



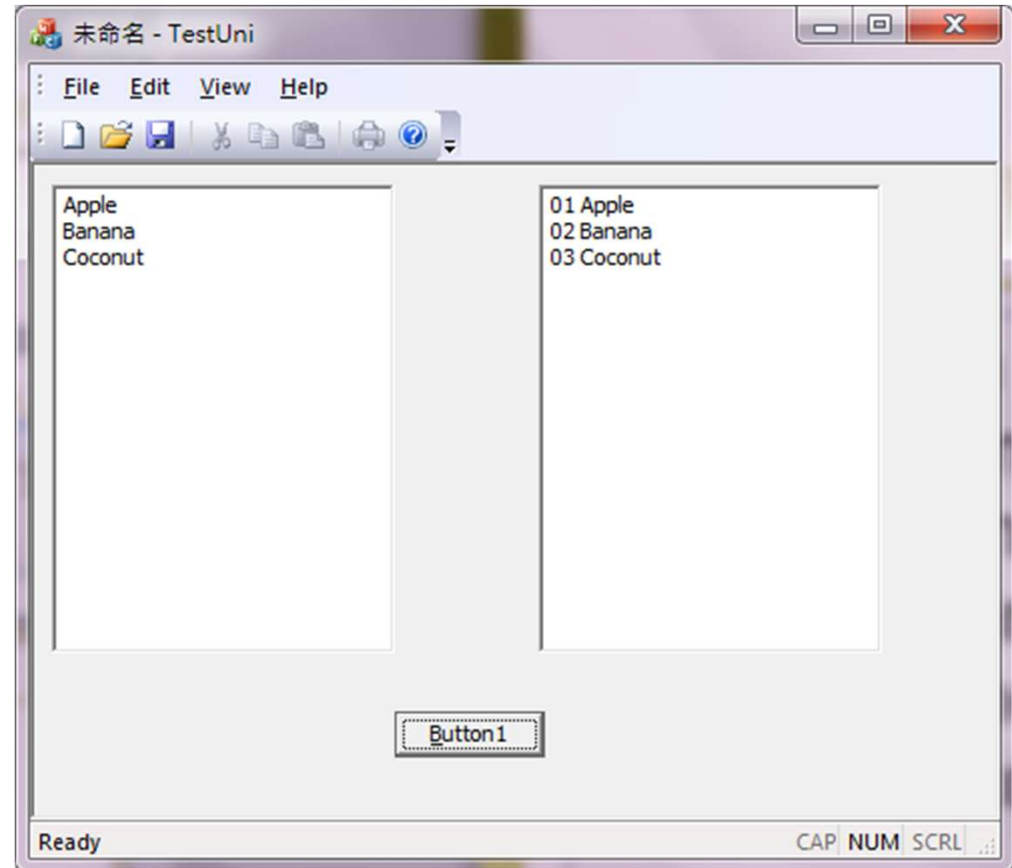
# Exercise: Multiline Edit Box

- Design an application which will convert all the English alphabets (a-z) from lowercase to uppercase.
  - Hint: Use CString member function MakeUpper() and MakeLower().



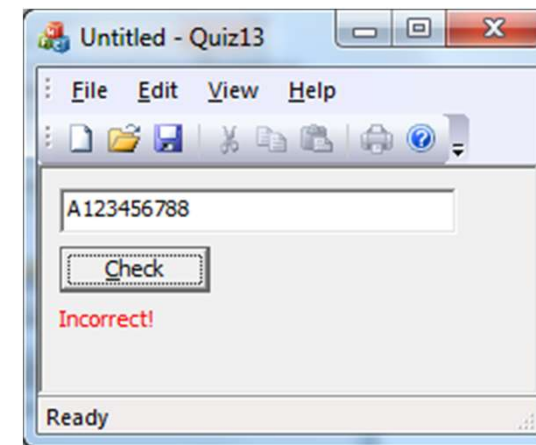
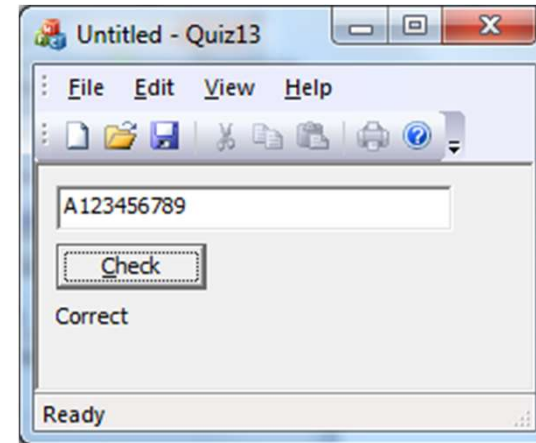
# Exercise: Line Number

- Design an application which will add line numbers in front of each line in an Edit Box control.
  - A "new line" actually consists of `CHR(13)+CHR(10)`.



# Enable a Button

- We want to design an application which requires the user to input his/her ID number for validation.
- When the application starts, it shows a message in the edit box to prompt the user typing an ID number.
- Initially, the button was disabled. When the user types a 10-character string, the button will be enabled.
- When the button was clicked, following [this rule](#) to validate the ID number.
  - If it is valid, change the static text control to show "Correct".
  - If it is invalid, change the static text control to show "Incorrect!"



# CView::OnInitialUpdate()

---

## □ Initialize an Edit Box control

- `CWnd* pEditBox = static_cast<CWnd*>(GetDlgItem(IDC_EDIT1));`
- `pEditBox->SetWindowText( _T("Please type your ID here") );`

## □ Disable a Button

- `GetDlgItem(IDC_BUTTON1)->EnableWindow(FALSE);`

# Check the Input String Length

---

- Create a handler for the message EN\_CHANGE.

```
CView:: OnEnChangeEdit1 ()
{
    int len =
        GetDlgItem(IDC_EDIT1) ->GetWindowTextLength();

    if (len == 10)
        GetDlgItem(IDC_BUTTON1) ->EnableWindow(TRUE);
    else
        GetDlgItem(IDC_BUTTON1) ->EnableWindow(FALSE);
}
```

## Change the Color of Edit Box /Static Text

---

- Create a handler for the message WM\_CTLCOLOR.
- Add the following code in CView::OnCtlColor()

```
if( pWnd->GetDlgCtrlID() == IDC_STATIC &&  
m_WrongID)  
    pDC->SetTextColor( RGB(255,0,0) );
```

# HW: ID Number Validation

---

- ❑ Complete the message handler `OnBnClickedButton1()`.
- ❑ If the user inputs lowercase letter like "a123456789", convert it to uppercase by the `MakeUpper()` member function of `CString`.
- ❑ If the first character is not an English alphabet, or the other nine characters are not digits, the string is certainly not a valid ID number.
- ❑ If the second character is not 1 or 2, it is also invalid.

# Principle of Learning

---

- Tell me and I forgot.
  - Show me and I remember.
  - Involve me and I understand.
- 不聞不若聞之，  
聞之不若見之，  
見之不若知之，  
知之不若行之，  
學至乎行而止矣。  
——荀子
  - 古人學問無遺力，  
少壯工夫老始成。  
紙上得來終覺淺，  
絕知此事要躬行。  
——陸游



# Common Controls (P.748)

Static controls provide static information, such as titles or instructions, or simply provide decoration in a dialog in the form of an icon or a filled rectangle.

Radio buttons are usually grouped so that if one is checked all the others are unchecked.

Check boxes are individually checked and more than one can be checked at one time.

Buttons can have labels as here and they can also display icons.

You have already seen scroll bars attached to the client area of the Sketcher window. Scroll bars can also be free standing.

A list box presents a predefined list of items from which you can choose. The scroll bar need not be present for a short list. A list can also have multiple columns and be scrolled horizontally. A version of the list box is available that can display icons as well as text.

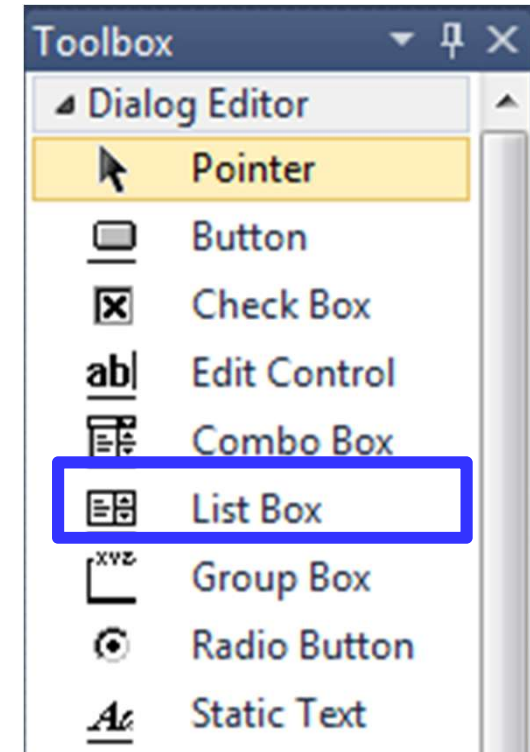
This text box is the simplest form of edit control that allows you to enter and/or edit a line of text. More sophisticated edit controls can display multiple lines of text and support scrolling of the text.

Comboboxes combine the capability of a dropdown list from which you can select with the option of entering data yourself. The Save As... dialog uses a combobox to enable you to enter the file name.

- ❑ Static Controls
- ❑ Button Controls
- ❑ Edit Controls
- ❑ List Boxes
- ❑ Combo boxes
- ❑ Check Boxes
- ❑ Radio Buttons
- ❑ Scrollbars

# Using a List Box

- ❑ Create a List Box Control on your form.
- ❑ The default value of the `Sort` property is `True`.
  - Set it to `False` if you want to keep the sequence as items are appended to the list.
    - ❑ e.g. "Jan", "Feb", "Mar"



# Initialize a List Box

---

- ❑ Put the initialization in the overrides `OnInitialUpdate()` instead of the constructor of `CView`.

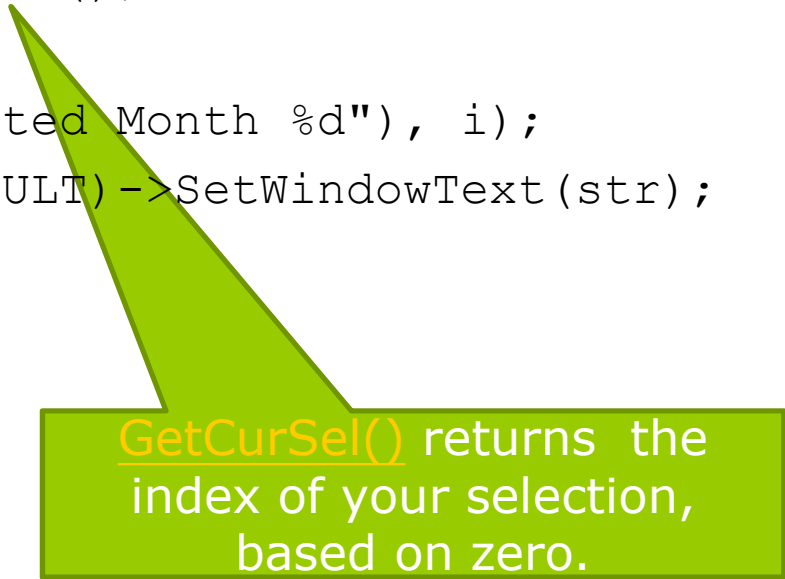
```
void CQuiz13View::OnInitialUpdate()
{
    CFormView::OnInitialUpdate();
    GetParentFrame()->RecalcLayout();
    ResizeParentToFit();

    CListBox* pListBox = static_cast<CListBox*>(GetDlgItem(IDC_LIST1));
    CString str;
    for (int i=0; i<3; i++)
    {
        str.Format(_T("Month %d"), i);
        pListBox->AddString(str);
    }
    pListBox->SetCurSel(0);
}
```

# Get the Index of the Selected Item

---

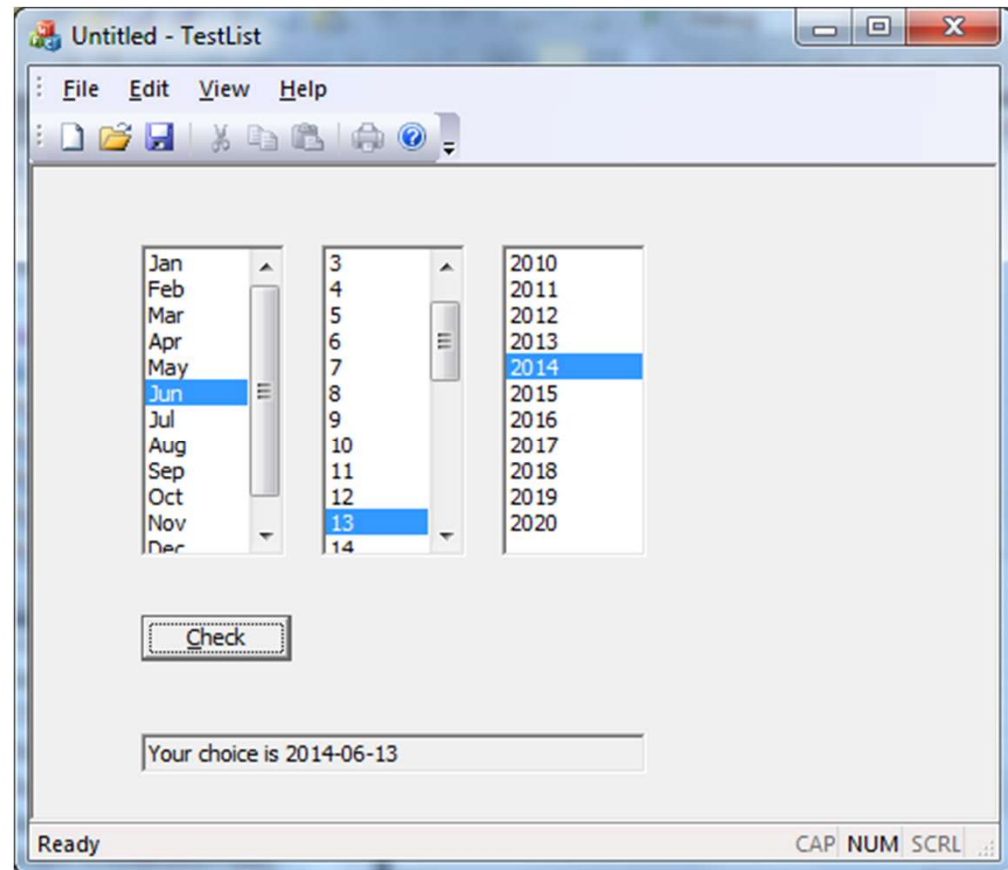
```
void Cch18listView::OnBnClickedButton1 ()
{
    CListBox* pListBox =
        static_cast<CListBox*>(GetDlgItem(IDC_LIST1));
    int i = pListBox->GetCurSel();
    CString str;
    str.Format( _T("You selected Month %d"), i);
    GetDlgItem(IDC_STATIC_RESULT)->SetWindowText(str);
}
```



GetCurSel() returns the index of your selection, based on zero.

# Exercise: List Box Controls

- Use three list boxes to represent Month, Day, and Year, respectively.
- When the user click the button, show the date chosen by the lists.



# CListBox Member Functions

---

- GetText
  - Gets a string from a list box.
- GetCount
  - Returns the number of strings in a list box.
- GetTextLen
  - Returns the length in bytes of a list-box item.
- DeleteString
  - Deletes a string from a list box.

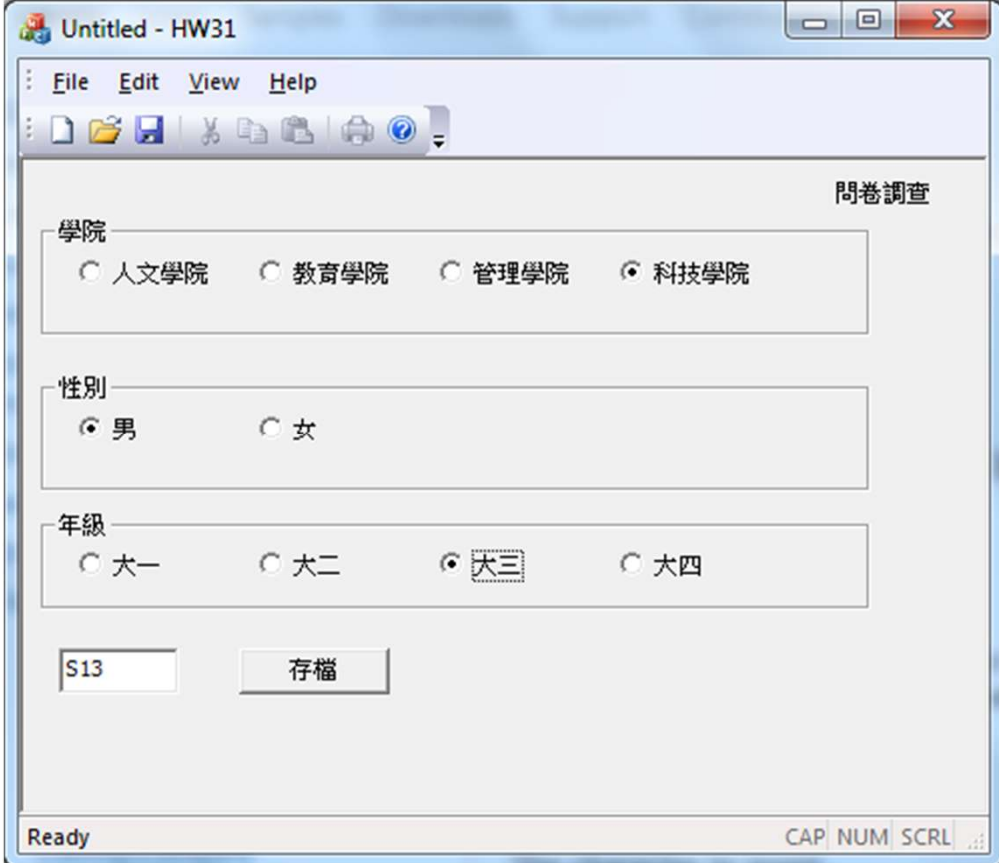
# Exercise: Dynamically Change a List

---

- Modify the previous exercise, so that
  - When the user selects a month, your program will automatically adjust the list for days.
    - January – 31 days
    - April – 30 days
    - June – 30 days
    - etc.
  - When the user clicks the button, your program will calculate what day the selected day is, and show a message like “2013-06-04 is Tuesday”.

# Using Radio Buttons (P.750)

- Sometime you want to enclose a few radio buttons in a group box, so that only one member of a group can be checked at any given time.



The screenshot shows a Windows application window titled "Untitled - HW31" with a menu bar (File, Edit, View, Help) and a toolbar. The main content area is titled "問卷調查" (Survey) and contains three groups of radio buttons, each enclosed in a rectangular box:

- 學院 (College):** Four radio buttons labeled "人文學院", "教育學院", "管理學院", and "科技學院". The "科技學院" button is selected.
- 性別 (Gender):** Two radio buttons labeled "男" (Male) and "女" (Female). The "男" button is selected.
- 年級 (Grade):** Four radio buttons labeled "大一", "大二", "大三", and "大四". The "大三" button is selected.

Below the radio button groups, there is a text input field containing "S13" and a button labeled "存檔" (Save). The status bar at the bottom of the window displays "Ready" and "CAP NUM SCRL".



## I have 3 groups of radio buttons, but

---

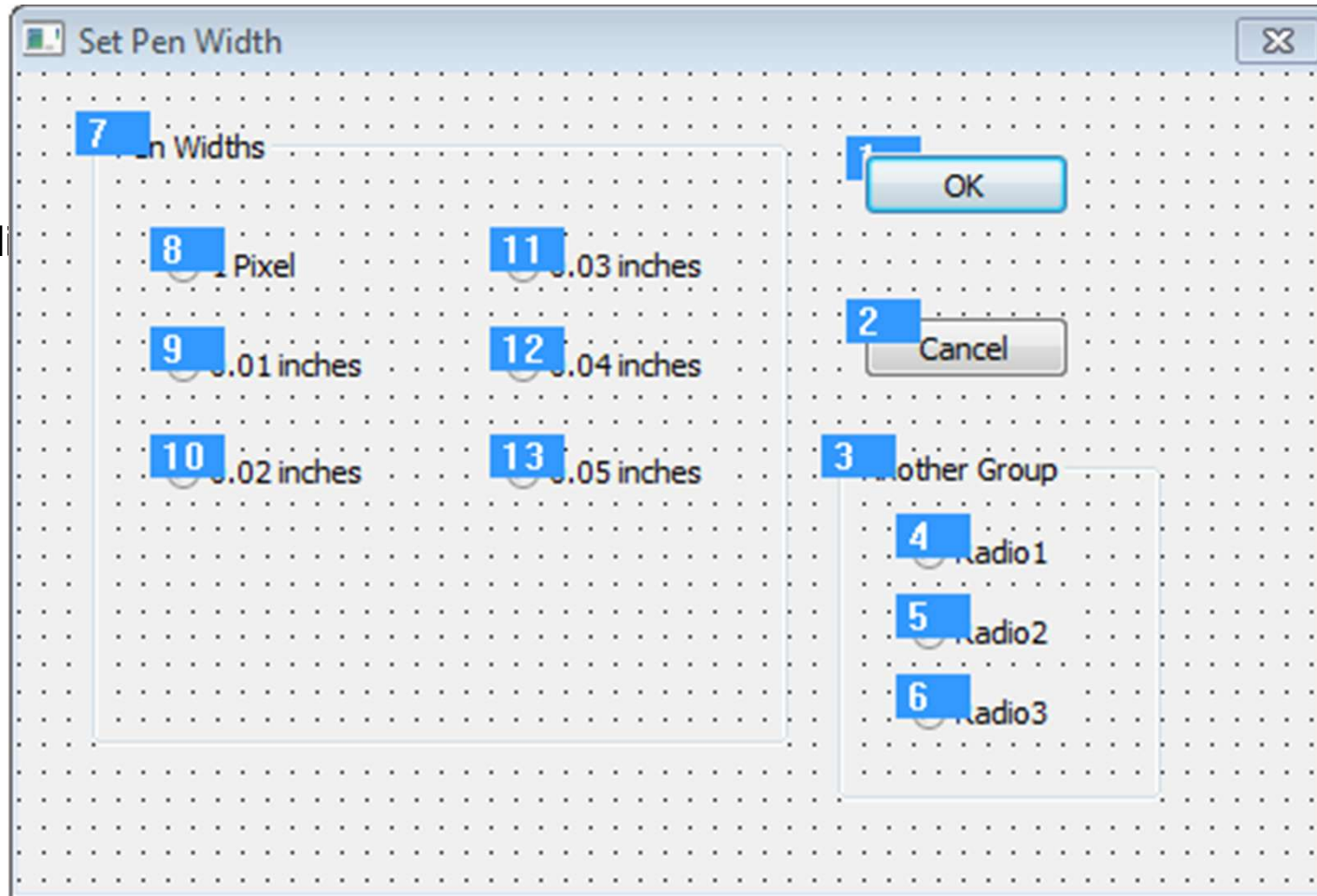
- 框起來只是美觀，實際運作上所有 radio button 還是屬於同一個 group.
- 你得把每個 group 的第一個（由 tab order 決定）radio button 的 Group property 設為 True.
  - Ctrl-D 可以看 tab order. 再按一次則取消.
  - Ctrl-T 可以測試這個 form 是否介面設計好了。

# Radio Buttons in a Group Box

---

- ❑ Only one member of the group can be checked at any given time.
- ❑ However, enclosing radio buttons within two group boxes does not make them to join two separate groups.
  - The tab order of the dialog, and the `Group` property completely dictates which radio buttons belong to which groups.
  - Each radio button belongs to the group of the previous main radio button in the tab order. The main radio button has the `Group` property set to `True`.
  - You can set the tab order by going to the dialog editor and pressing `Ctrl+D`. You then click on each control from 1 to N in the order you want the tabbing to go.

# Dialog Editor (Ctrl-D)



main radi

o button

# OnBnClickedRadio1()

---

- When a radio button is checked, do the following :

```
void CView::OnBnClickedRadio1()  
{  
    m_str.SetAt(0, 'H');  
    GetDlgItem(IDC_EDIT1)->SetWindowText(  
        CString(m_str) );  
}
```

- Your **CView class** may declare a private **data member** `m_str` with initial value `_T("0000")`.

# Check/Uncheck Radio Buttons in Your Program

---

- You may have a RESET button to uncheck all radio buttons

```
void CHW31View::OnBnClickedButtonClear()
{
    // Reset all radio buttons
    CheckDlgButton(IDC_RADIO1, 0);
    CheckDlgButton(IDC_RADIO2, 0);
    CheckDlgButton(IDC_RADIO3, 0);
    CheckDlgButton(IDC_RADIO4, 0);
    CheckDlgButton(IDC_RADIO5, 0);
    CheckDlgButton(IDC_RADIO6, 0);
    CheckDlgButton(IDC_RADIO7, 0);
    CheckDlgButton(IDC_RADIO8, 0);
    CheckDlgButton(IDC_RADIO9, 0);
    CheckDlgButton(IDC_RADIO10, 0);
}
```

1 to check the radio button; 0 to uncheck.

# Append Data to a File

In CView.h  
#include <fstream>  
using std::ofstream;  
using std::ios;

```
void CView::OnBnClickedSave()
{
    ofstream Log("log.txt", ios::app);
    Log << CT2A(m_str) << '\n';
    Log.close();
```

Convert Unicode to ASCII

```
m_str = _T("0000");
GetDlgItem(IDC_EDIT1)->SetWindowText(_T("  "));

OnBnClickedButtonClear(); // Reset radio buttons
}
```

- You may find the file "log.txt" in the same directory as your .cpp files.



# Displaying a BITMAP Image

# A Quick Demonstration

---

- Create an SDI application.
- In the Resource View, right-click to Add Resource, and create a new Bitmap image.
  - The default name should be IDB\_BITMAP1.
  - Draw something in the bitmap.



# Sample Code

---

- ❑ In the OnDraw() function in the View class, add the following code:
  - `CBitmap Bitmap;`
  - `Bitmap.LoadBitmapW(IDB_BITMAP1);`
  - `CDC memDC;`
  - `memDC.CreateCompatibleDC( pDC );`
  - `memDC.SelectObject( &Bitmap );`
  - `pDC->BitBlt( 0, 0, 48, 48, &memDC, 0, 0, SRCCOPY );`
- ❑ If Intelli-sense warns you that it does not recognize `IDB_BITMAP1` (yet), just ignore the warning.
- ❑ Build and execute the program.

# The meaning of each line of code

---

- ❑ A CBitmap object must be declared.
- ❑ The CBitmap object must be initialized by calling its `LoadBitmap()` function.
- ❑ A CDC object must be declared.
- ❑ The CDC object must be initialized by calling its `CreateCompatibleDC()` member function.
  - This DC is referred to as a “memory device context”.
- ❑ Select the CBitmap object into the CDC object, by calling the CDC’s `SelectObject()` function.
- ❑ Copy the memberDC to an on-screen windows, by calling the screen CDC’s `BitBlt()` function.



# Can I Load an External Bitmap File?

---

```
CString szFilename ("Y:\\Waste\\micky.bmp");
HBITMAP hBmp = (HBITMAP)::LoadImageW(NULL, szFilename,
IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE|LR_CREATEDIBSECTION);

CBitmap Bitmap;
// Bitmap.LoadBitmapW(IDB_BITMAP2);
Bitmap.Attach(hBmp);

CDC memDC;
memDC.CreateCompatibleDC( pDC );
CBitmap *pOldbmp = memDC.SelectObject( &Bitmap );
BITMAP bi;    // Get the BMP Height and Width
Bitmap.GetBitmap(&bi);
pDC->BitBlt( 0, 0, bi.bmWidth, bi.bmHeight, &memDC, 0, 0,
SRCCOPY );
memDC.SelectObject( pOldbmp );
```

# Add a Button to Trigger the Action

---

- ❑ Create an SDI (single document interface) application.
- ❑ Select `CFormView` as the base class.
- ❑ In the Resource View, expand the Dialog resource and double-click the form.
- ❑ Drag a button control to the form.
- ❑ Double-click the button to add a handler.
  - Insert the code for displaying a BITMAP file.

# Terminate an MFC Application

---

- You may have an Exit button so that your MFC application can terminate itself.

```
void ExitMFCApp()  
{  
    ASSERT(AfxGetMainWnd() != NULL);  
    AfxGetMainWnd()->SendMessage(WM_CLOSE);  
}
```

```
void CView::OnBnClickedButtonEnd()  
{  
    ExitMFCApp();  
}
```

# HW21

---

- ❑ Design an application with 3 buttons. Each button will display a different bitmap file when it is pushed.
- ❑ Bonus: Design your button so that it will show/hide the corresponding bitmap as a toggle switch.

