# COMSC-200 Lecture Topic 13
# Microsoft Foundation Classes

### Reference
codeproject.com
codeproject.com
microsoft.com

### Using Visual Studio 2010
New Project->Visual C++->MFC->MFC Application
  type a name; click OK
Application->Application Type
  CHOOSE: Dialog Based radio button
  CHOOSE: Use MFC In A Static Library radio button
  Click Finished
set Release mode
  `F7` build before 1st `Ctrl-F7` compile!
use the Resource and Solutions tabs

### Backuping
save only the .SLN file...
...and the application's folder
lose *any* "release" or "debug" folder

### Visual Components
placing "controls" in a window
View->Toolbox
in Toolbox: Edit Control *(CEdit)*,
  Static Text *(CStatic)*, Button *(CButton)*
control IDs
  change IDs for static text...
Format->Align, Arrange Buttons, Space Evenly,
  Make Same Size (last clicked is base)
Format->Tab Order
& and Static Text

### Writing "Handlers"
Double-click button to edit it's handler
do not directly call a handler

### To Delete A Control
Edit->Find & Replace->Find In Files, Find All
  and delete all references.
  Then rebuild

### Casting
```
CEdit* pEdit1 = (CEdit*)GetDlgItem(IDC_EDIT1);
```
msdn.microsoft.com/library

### Getting Text
```
CString edit1;
pEdit1->GetWindowText(edit1);
```

### Writing To Edit Boxes
```
CEdit* result = (CEdit*)GetDlgItem(IDC_EDIT4);
result->SetWindowText(CString(...));
```
  *...any quoted literal like this:* `CString(_T("Hello"))`
  *...or any C const char**
  *...or any C++ string with .c_str()*
property: multi-line
  if true, use `\r\n` line-separator
`_T` is a macro to make strings "character set neutral"

### The `CString` Class
the MFC's string class

### Converting CString to `const char*`
Step 1: convert CString to C++ string
Step 2: convert C++ string to const char*

```
string s = CStringA(cs);
const char* c = s.c_str();
```

### Building Strings
```
#include <sstream>
using std::ostringstream;
ostringstream sout; // a buffer
sout << ...
```
`sout.str()` returns a `string`
`sout.str().c_str()` returns a `const char*`
put this in SetWindowText: `CString(sout.str().c_str())`

### Editable Static Text
change ID property to match name (e.g., IDC_STATIC to IDC_STATIC1)
```
CStatic* pStatic1 = (CStatic*)GetDlgItem(IDC_STATIC1);
```

### Checkboxes and Radios
objects of class **CButton**
CButton::SetCheck(true), CButton::GetCheck()
Radios: 1st radio in group: Group=true
  tab order determines 2nd, etc.

### The MFC Class Hierarchy
classes derived from **CObject**

### Adding Code Modules
put PROTOTYPES in H, function defs in CPP
*"unexpected end of file*
  *while looking for*
  *precompiled header directive"*
`#include "stdafx.h"` in the .cpp file

### Building Strings Using `printf` Features
```
CString cs;
cs.Format(_T("i=%d and x=%f"), i, x); // or %f for floating point
```
  variations: %10d, %.2f, %10.2f, %s
  ref: www.cplusplus.com
put this in SetWindowText: `...->SetWindowText(cs)`

### Reading Combo Boxes
```
CComboBox* object = (CComboBox*)GetDlgItem(IDC_COMBO1);
int n = object->GetCurSel();
```

### Icon Editing
replace the res\{app}.ico image file

### Picture Controls
use BMP format (in Windows Paint: save-as BMP)
no dynamic resizing! (in Windows Paint: resize)
1. place BMPs in /res folder
2. in resource view: "add resource", import bitmap
   each BMP assigned an "IDB_BITMAPx" ID
3. in Dialog, place Picture Control
   only upper-left matters -- not size
4. in OnInitDialog, set Picture Control "style"
   and load default picture
5. in handler(s), load picture(s)

msdn.microsoft.com/library and codeproject.com

`GetLength()`
`operator==`
`MakeUpper()` and `MakeLower()`
`CString cs = ...;`
converting to int: `int i = _ttoi(cs);`
converting to double: `double d = _ttof(cs);`
converting to std::string: `string s = CStringA(cs);`
std::string to const char*: `s.c_str()`

**Private Member Functions**
to be called from various handlers in dlg.cpp
add to class definition in dlg.h

**File Dialogs**

```
CFileDialog dlg(true, NULL, NULL,
  OFN_HIDEREADONLY|
  OFN_OVERWRITEPROMPT,
  NULL, this);
if (dlg.DoModal() == IDOK)
```

the `CFile` class: msdn.microsoft.com/library

---

Here's another legitimate place to get a Visual Studio 2010: http://imaginecup.com/. It's a Microsoft-sponsored competition. If you register, you'll get access to some programs that could be used for program development.

---

**GetWindowText vs GetWindowTextA vs GetWindowTextW**
The version of an MFC member function without a trailing "A" is a *macro* that the compiler interprets as either the GetWindowTextA or the GetWindowTextW function. Projects with the (default) unicode option interpret this as GetWindowTextW, while ASCII projects use GetWindowTextA instead. GetWindowText itself does not show up in the popup menus, therefore, because it's not a *function*. So just use GetWindowText and other function macros, without the trailing A (or W).

---

```
BOOL CTestDlg::OnInitDialog()
{
  ...
  // ref. http://msdn.microsoft.com/en-us/library/yft127ws(v=vs.80).aspx
  CTabCtrl* pTab = (CTabCtrl*)GetDlgItem(IDC_TAB1);
  pTab->InsertItem(0, _T("One"));
  pTab->InsertItem(1, _T("Two"));
  pTab->InsertItem(2, _T("Three"));
  pTab->InsertItem(3, _T("Four"));
  pTab->SetCurSel(0); // set initial control visibility to match this

  // to size the pull-down area, click on the down arrow first...
  // ref. http://www.functionx.com/visualc/controls/combobox.htm
  CComboBox* pCombo = (CComboBox*)GetDlgItem(IDC_COMBO1);
  pCombo->InsertString(0, _T("One"));
  pCombo->InsertString(1, _T("Two"));
  pCombo->InsertString(2, _T("Three"));
  pCombo->InsertString(3, _T("Four"));
  pCombo->SetCurSel(0); // or -1 for unselected
  ...

  // Radio Buttons and Checkboxes (class CButton)
  // ref. http://www.functionx.com/visualc/controls/radiobutton.htm
  pRadio->SetCheck(BST_CHECKED);
  pCheck->SetCheck(BST_CHECKED);
  ...

  // Picture control setup
  // place BMPs in /res, import as bitmap resources (IDB_BITMAP1, etc)
  CStatic* pPicture = (CStatic*)GetDlgItem(IDC_STATIC3);
  pPicture->ModifyStyle(0xF, SS_BITMAP, SWP_NOSIZE);
}

void CTestDlg::OnSelchangeTab1(NMHDR* pNMHDR, LRESULT* pResult)
{
  // get pointers to all controls in all tabs
  CEdit* pEdit1 = (CEdit*)GetDlgItem(IDC_EDIT1);
  //...

  // hide all controls in all tabs
  pEdit1->ShowWindow(SW_HIDE);
  //...

  // show controls for selected tab
  CTabCtrl* pTab = (CTabCtrl*)GetDlgItem(IDC_TAB1);
```

```
switch (pTab->GetCurFocus())
{
  case 0:
    pEdit1->ShowWindow(SW_SHOW);
    //...
    break;

  case 1:
    //...
    break;

  case 2:
    //...
    break;

  case 3:
    //...
    break; // YES, YOU NEED THE LAST BREAK
}

...
CComboBox* pCombo = (CComboBox*)GetDlgItem(IDC_COMBO1);
int n = pCombo->GetCurSel(); // -1 for unselected, 0 for first item...
...
CButton* pRadio = (CButton*)GetDlgItem(IDC_RADIO1);
int r = pRadio->GetCheck(); // BST_UNCHECKED for unselected, BST_CHECKED for selected
...
CButton* pCheck = (CButton*)GetDlgItem(IDC_CHECK1);
int c = pCheck->GetCheck(); // BST_UNCHECKED for unselected, BST_CHECKED for selected

CStatic* pPicture = (CStatic*)GetDlgItem(IDC_STATIC3);
HBITMAP hb = (HBITMAP)::LoadImage(AfxGetInstanceHandle(),
  MAKEINTRESOURCE(IDB_BITMAP1), IMAGE_BITMAP,0,0,LR_CREATEDIBSECTION);
pPicture->SetBitmap(hb);
```

MFC ref: MSDN