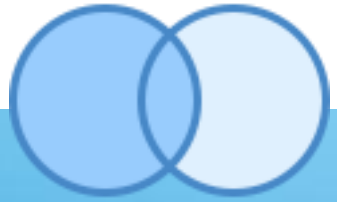


Windows basic concepts

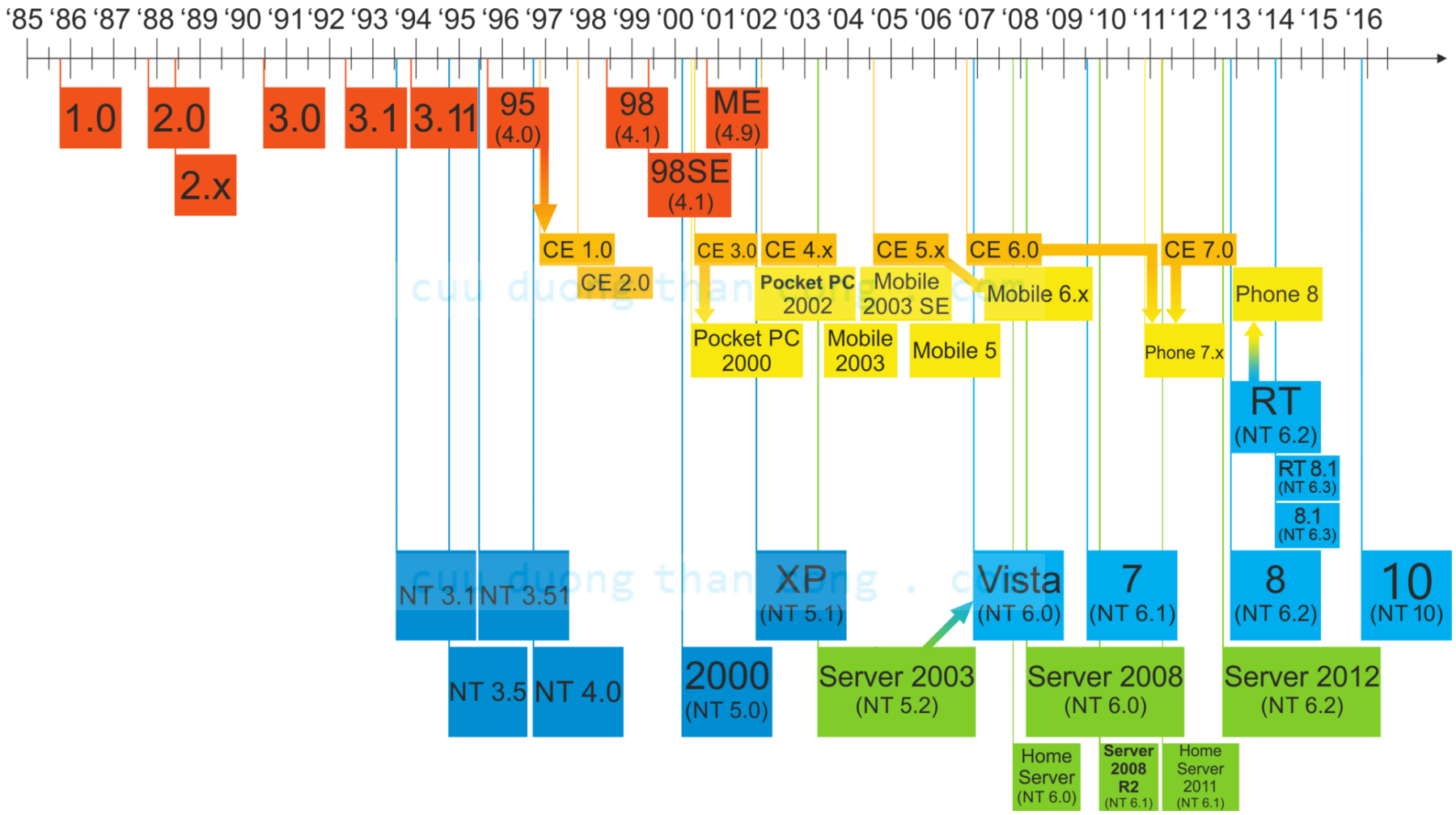
Windows programming

In this talk

1. Windows history & characteristics
2. Window's anatomy
3. Event-driven programming using messages

cuu duong than cong . com

Windows history



Popularity

Desktop OS	Net Applications	StatCounter
Windows XP	6.94%	4.10%
Windows Vista	0.53%	0.78%
Windows 7	49.04%	38.59%
Windows 8	1.23%	2.03%
Windows 8.1	6.40%	7.90%
Windows 10	26.80%	30.87%
All listed versions	**90.94%**	**84.27%**

Wikipedia, https://en.wikipedia.org/wiki/Microsoft_Windows (06.2017)

Basic charateristics

- **Event-driven**: wait for input from system
- **Multitask** / multithread
- 32 bit & 64 bit
- **G**raphical **U**ser **I**nterface

Programming libraries

- **Windows API ("Win32 API" and "Win64 API")**
- OWL - Object Windows Library
- **MFC - Microsoft Foundation Classes**
- ATL - Active Template Library
- WTL - Windows Template Library
- **BCL - .NET Framework Base Class Library**
- …

Covered in this course

- Windows **A**pplication **P**rogramming **I**nterface (11 weeks)
  - Win32 API (C++)
- C# - Universal app with WPF (3-4 weeks)

Microsoft .Net framework

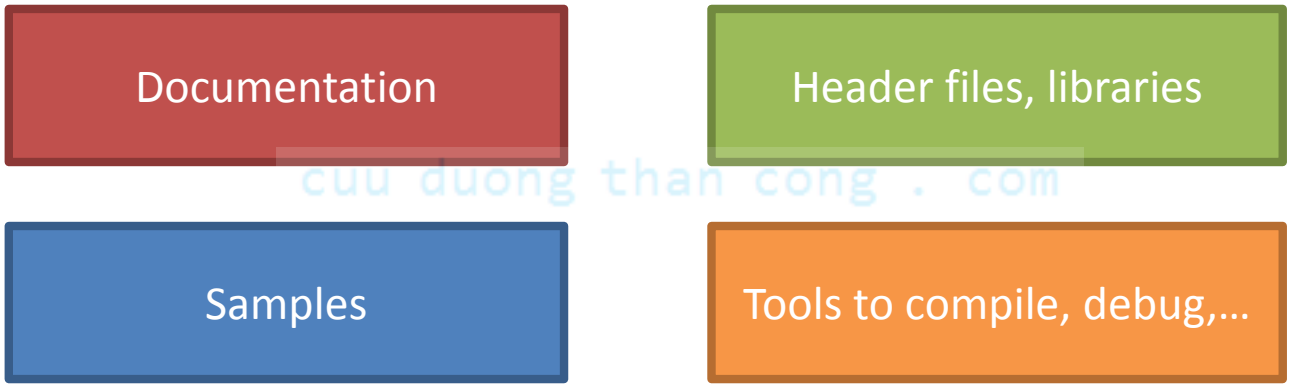
Version number	CLR version	Release date	Support ended	Development tool	Included in		Replaces
					Windows	Windows Server	
1.0	1.0	2002-02-13	2009-07-14[22]	Visual Studio .NET[23]	XP[a]	N/A	N/A
1.1	1.1	2003-04-24		Visual Studio .NET 2003[23]	N/A	2003	1.0[24]
2.0	2.0	2005-11-07	2011-07-12[22]	Visual Studio 2005[25]	N/A	2003, 2003 R2,[26] 2008 SP2, 2008 R2 SP1	N/A
3.0	2.0	2006-11-06	2011-07-12[22]	Expression Blend[27][b]	Vista	2008 SP2, 2008 R2 SP1	2.0
3.5	2.0	2007-11-19	2011-07-12[22] (except 3.5 SP1)	Visual Studio 2008[28]	7, 8[c], 8.1[c], 10[c]	2008 R2 SP1	2.0, 3.0
4.0	4	2010-04-12	2016-01-12[29]	Visual Studio 2010[30]	N/A	N/A	N/A
4.5	4	2012-08-15	2016-01-12[29]	Visual Studio 2012[31]	8	2012	4.0
4.5.1	4	2013-10-17	2016-01-12[29]	Visual Studio 2013[32]	8.1	2012 R2	4.0, 4.5
4.5.2	4	2014-05-05	N/A	N/A	N/A	N/A	4.0–4.5.1
4.6	4	2015-07-20	N/A	Visual Studio 2015[33]	10	N/A	4.0–4.5.2
4.6.1	4	2015-11-30[34]	N/A	Visual Studio 2015 Update 1	10 v1511	N/A	4.0–4.6
4.6.2	4	2016-08-02[35]	N/A		10 v1607	2016	4.0–4.6.1
4.7	4	2017-04-05[36]	N/A	Visual Studio 2017	10 v1703	N/A	4.0–4.6.2

Windows API

- Main components
  - Base Services: file systems, devices, processes & threads, Windows registry…
  - Graphics Device Interface
  - User Interface
  - Windows Shell
  - Network Services
  - Multimedia related APIs: DirectX
  - Programs interaction APIs: DDE, OLE, COM…

Windows SDK

- **S**oftware **D**evelopment **K**it



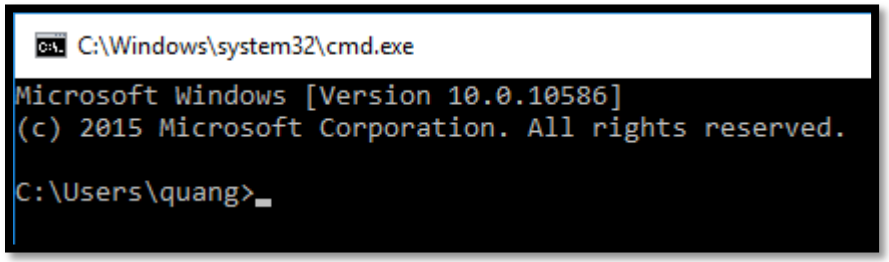
- Windows (SDK) for Windows 10
  - Already in Visual Studio

Version of VS in this course

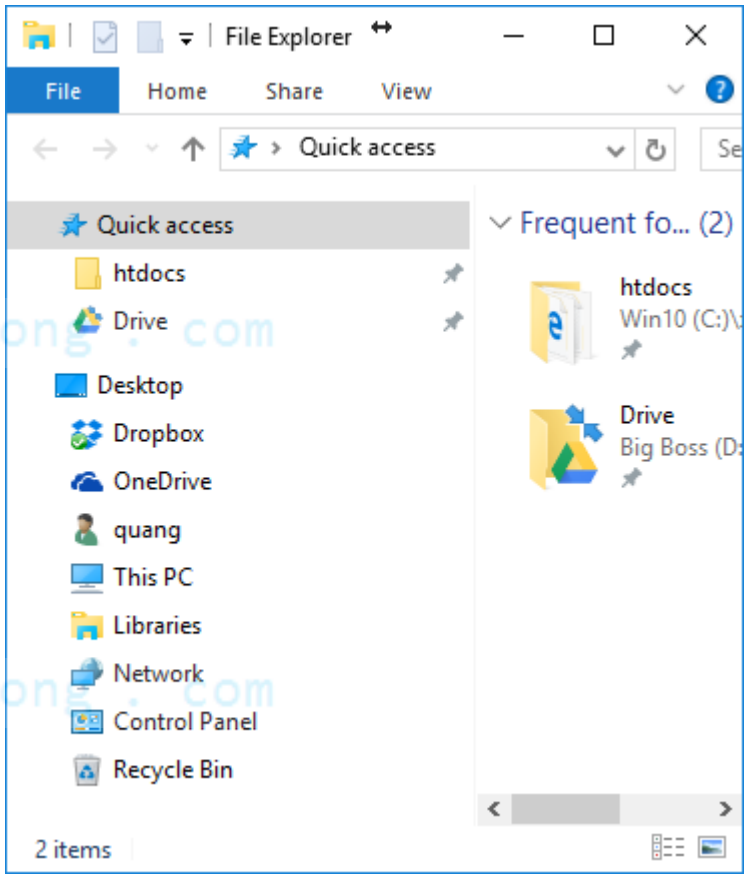
- Visual Studio **2017 Community** Edition
  - **Free**

Other versions are okay, as long as you specify in **readme.txt** or **readme.docx**

Types of application



Console UI

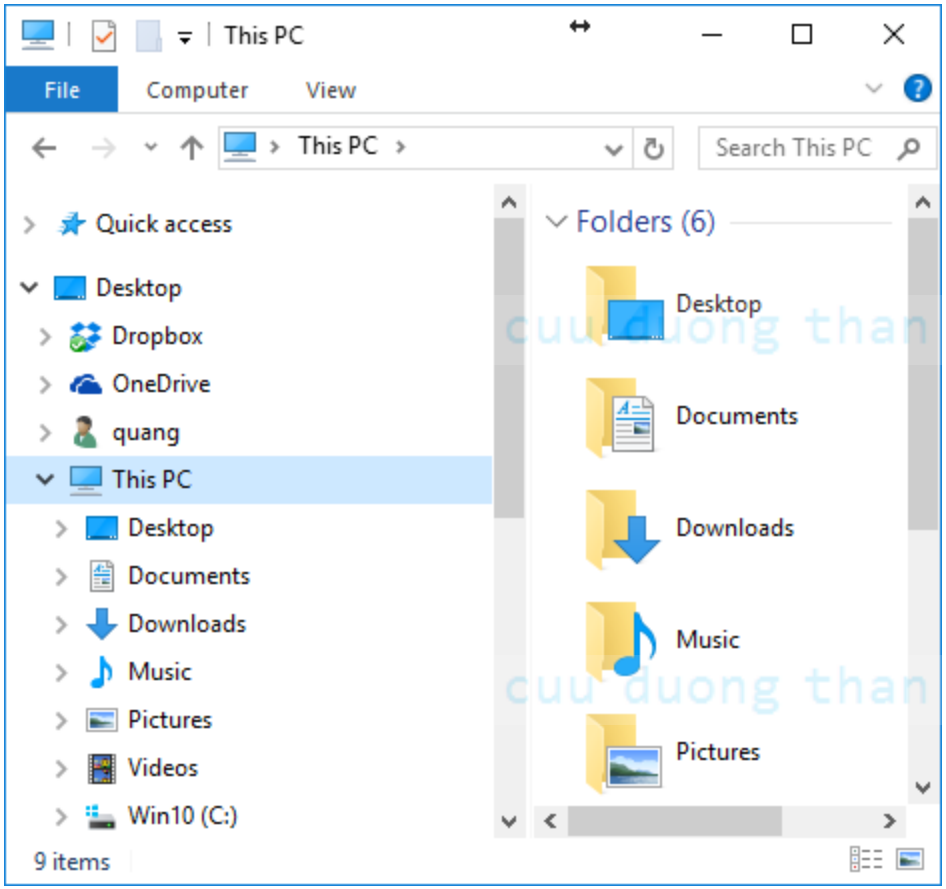


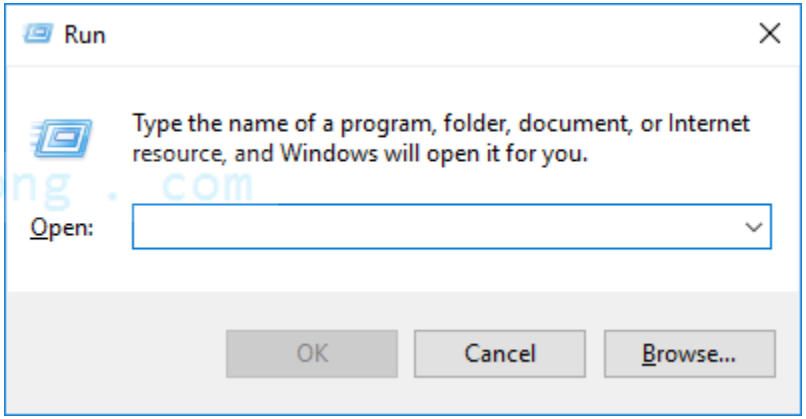
- Web-service
- Web-based
- And many more

Focus in this course: **Graphical UI**

Window's anatomy

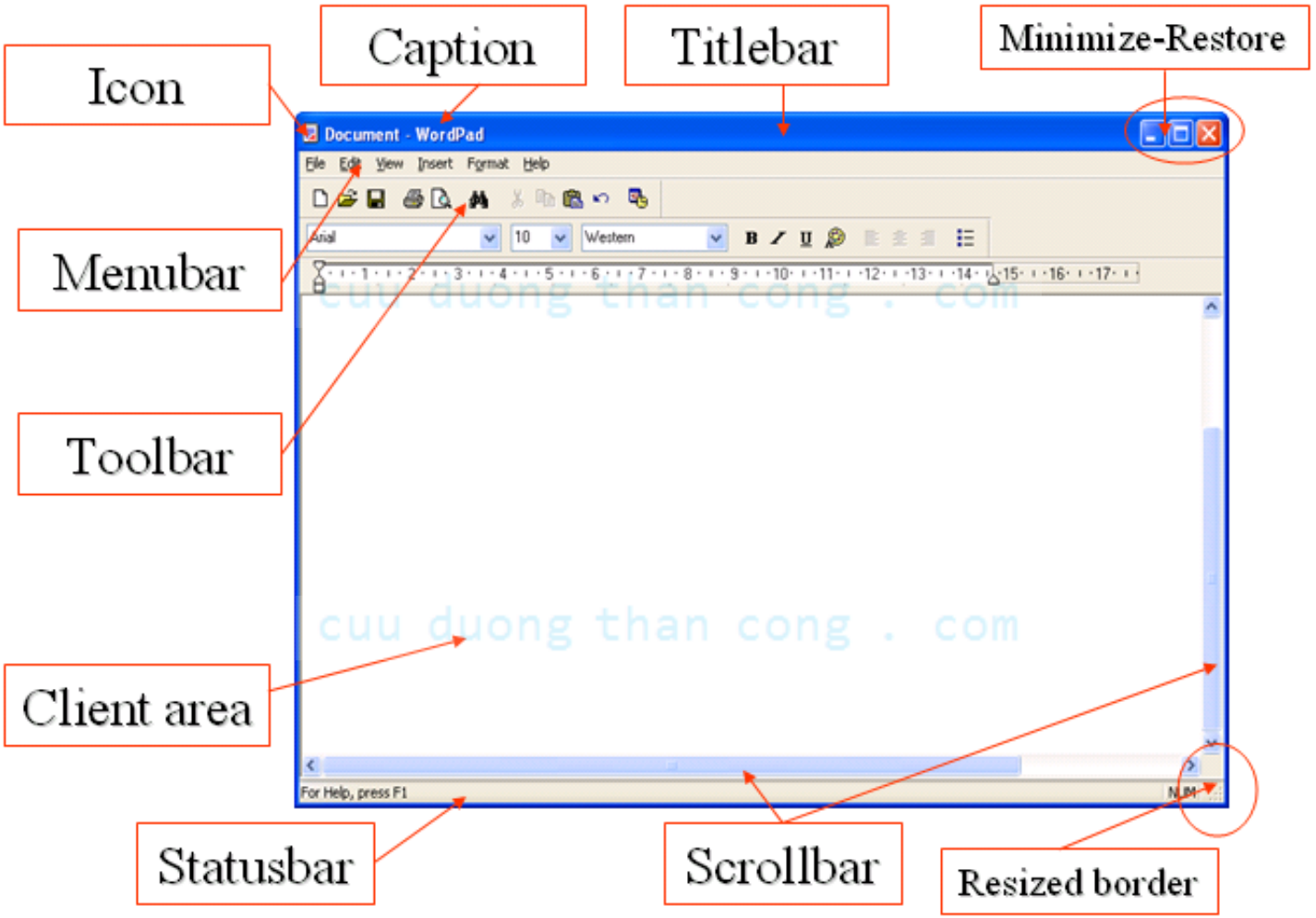
What is a window?



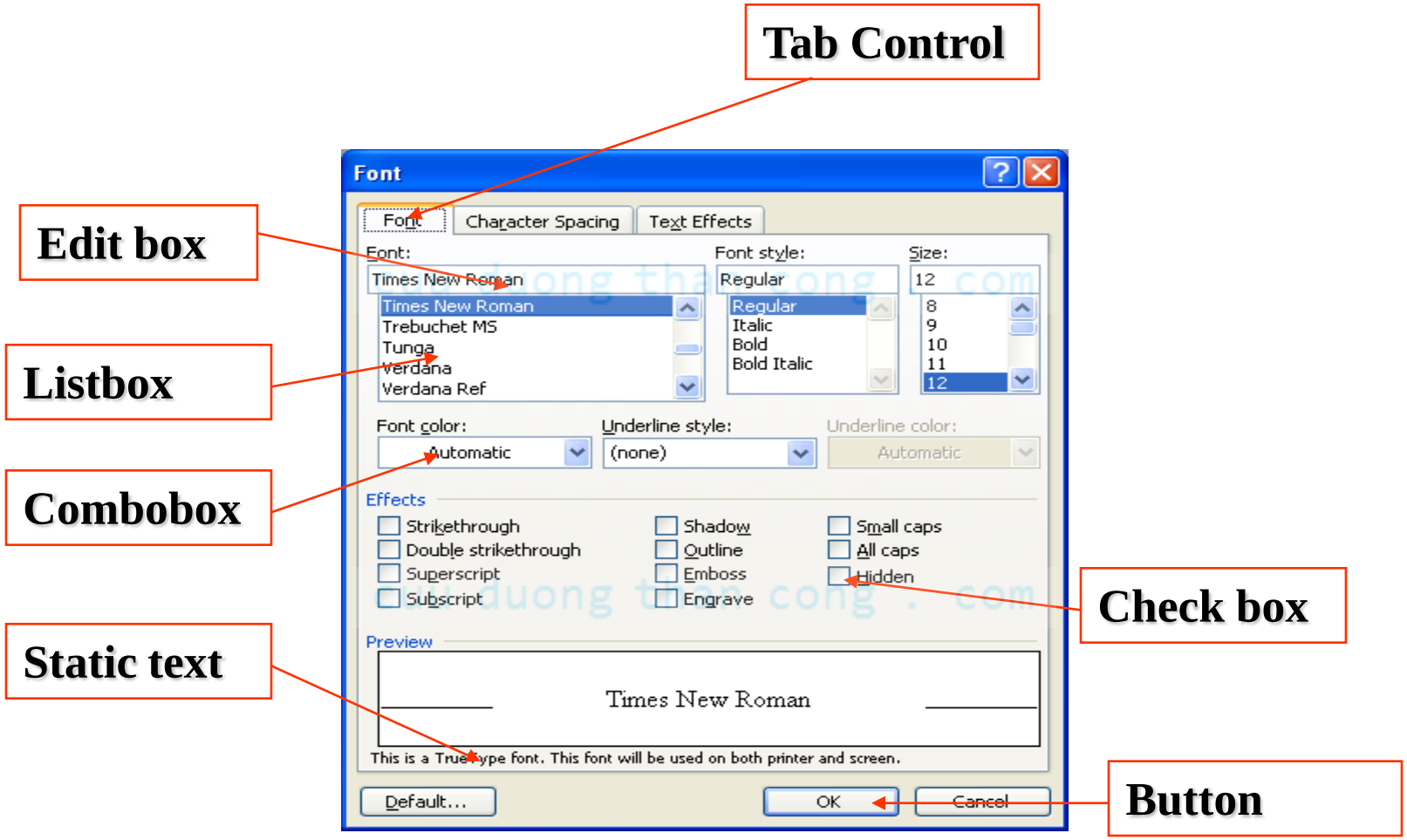


Quiz: How many windows are there?

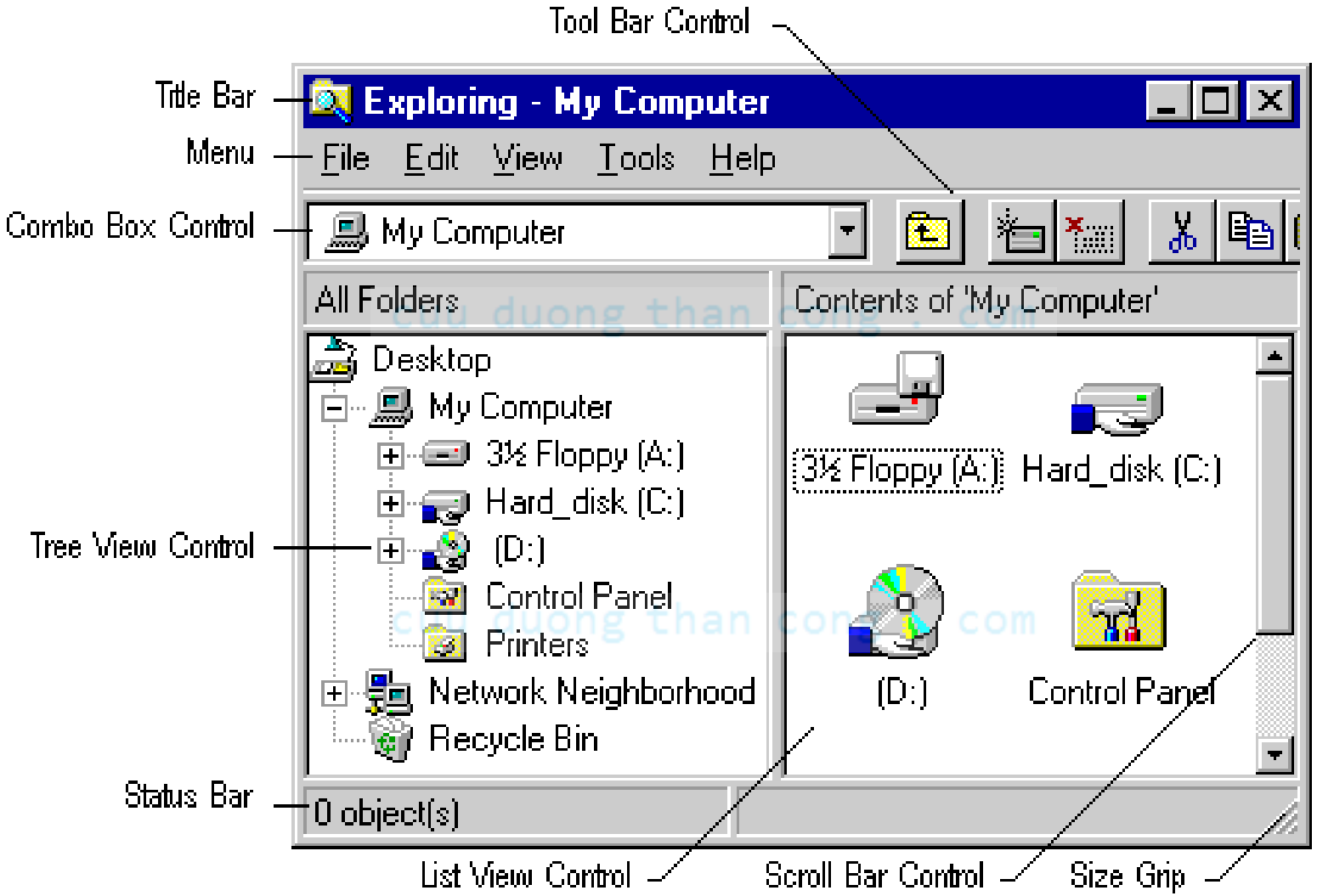
Components of a Window



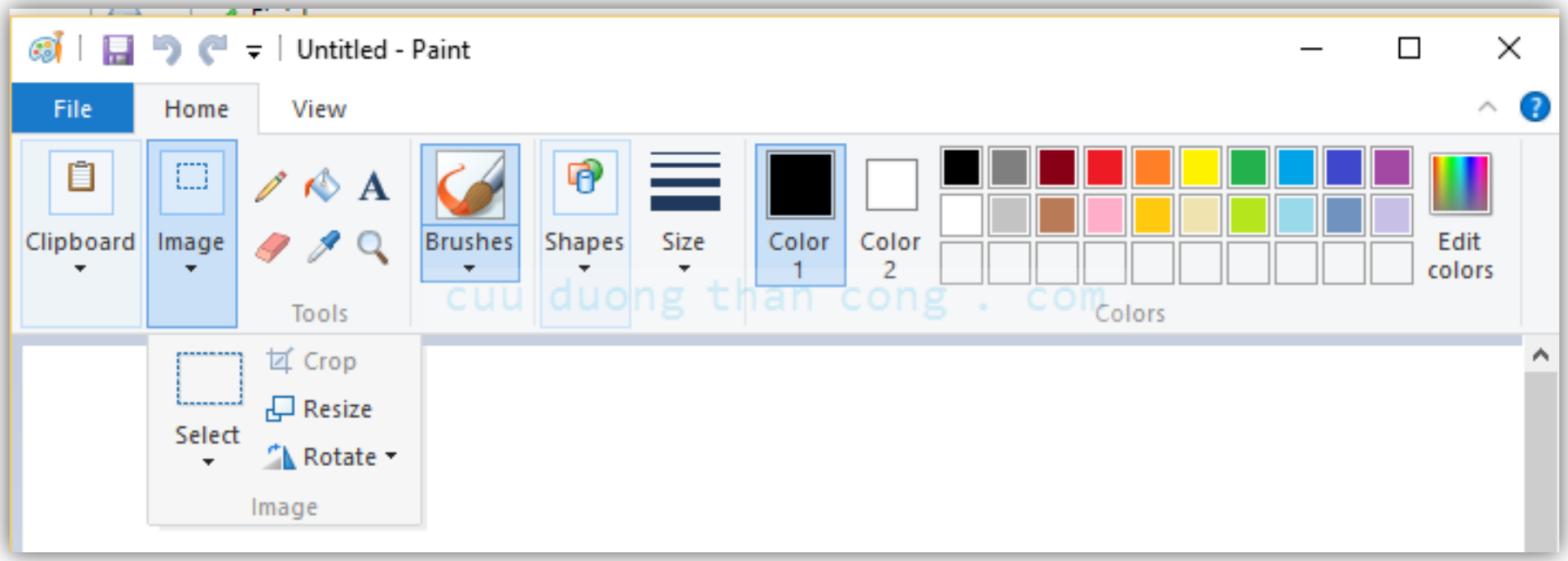
Common controls 1



Common controls 2



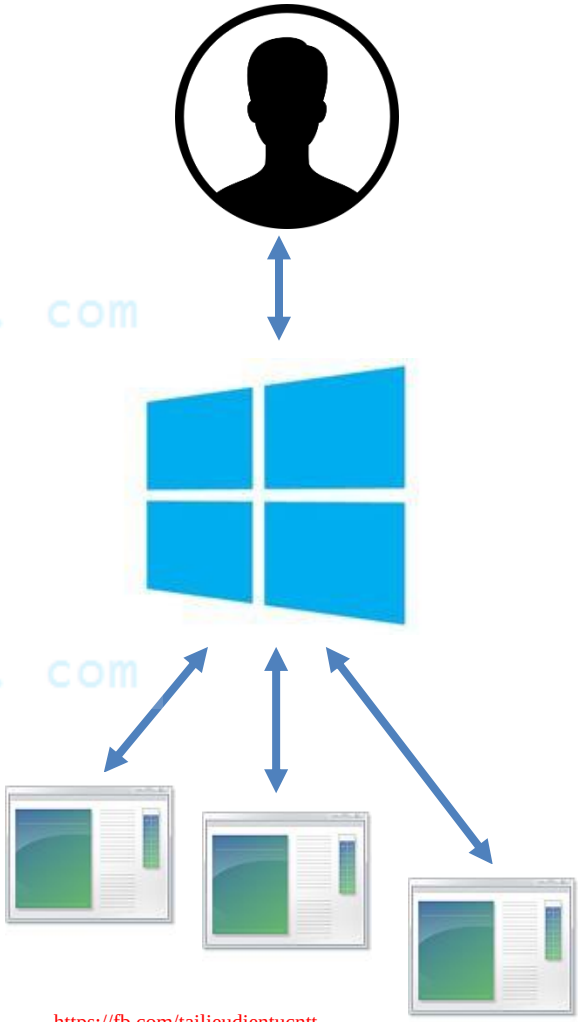
Ribbon / fluent



How application started?

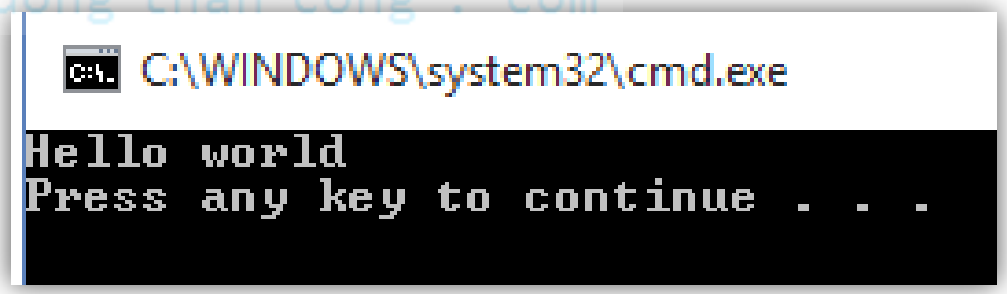
- OS create process and thread
- Load binary code into memory (DLL, if needed)
- Data is allocated & mapping into virtual memory
- Application start the thread

DOS vs Windows app



Console application

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello world";
    return 0;
}
```

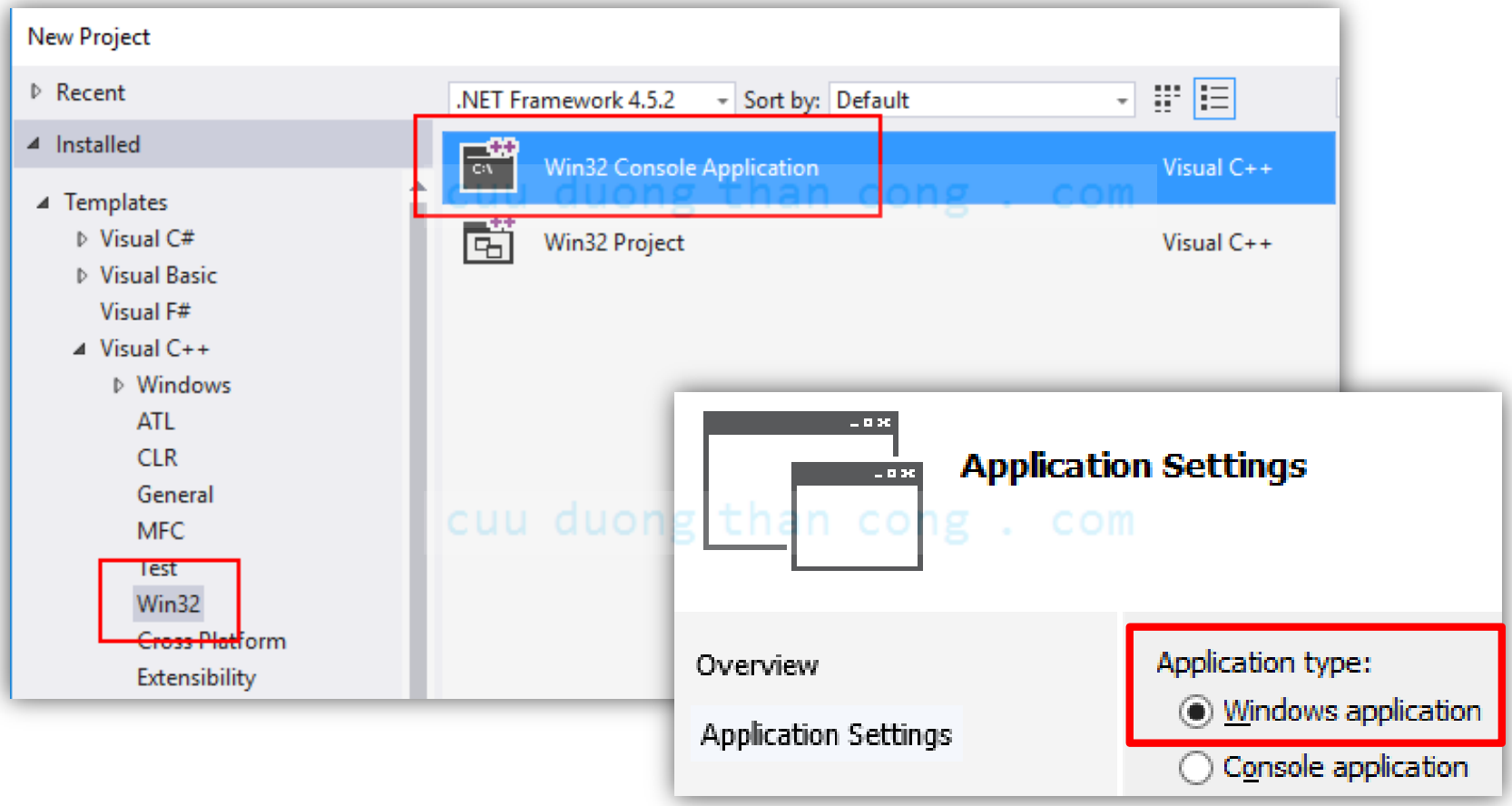
Simple Windows program

```
#include <windows.h>

int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
    _In_opt_ HINSTANCE hPrevInstance,
    _In_ LPWSTR    lpCmdLine,
    _In_ int       nCmdShow)
{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

    MessageBox(0, L"Hello world", L"Info", 0);

    return 0;
}
```

10 Questions?

Create code from template

- File > New Project



Full program - main

```
int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
                      _In_opt_ HINSTANCE hPrevInstance,
                      _In_ LPWSTR    lpCmdLine,
                      _In_ int       nCmdShow)
{
    UNREFERENCED_PARAMETER(hPrevInstance);
    UNREFERENCED_PARAMETER(lpCmdLine);

    LoadStringW(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadStringW(hInstance, IDC_WIN32PROJECT1, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    if (!InitInstance (hInstance, nCmdShow)) {
        return FALSE;
    }

    HACCEL hAccelTable = LoadAccelerators(hInstance, MAKEINTRESOURCE(IDC_WIN32PROJECT1));
    MSG msg;

    // Main message loop:
    while (GetMessage(&msg, nullptr, 0, 0)) {
        if (!TranslateAccelerator(msg.hwnd, hAccelTable, &msg)) {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }
    return (int) msg.wParam;
}
```

WM_QUIT

To sum up

- Register window class
- Init an instance
- Message loop

Register class

```
ATOM MyRegisterClass(HINSTANCE hInstance)
{
    WNDCLASSEXW wcex;
    wcex.cbSize = sizeof(WNDCLASSEX);

    wcex.style          = CS_HREDRAW | CS_VREDRAW;
    wcex.lpfnWndProc    = WndProc;
    wcex.cbClsExtra     = 0;
    wcex.cbWndExtra     = 0;
    wcex.hInstance      = hInstance;
    wcex.hIcon          = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_WIN32PROJECT1));
    wcex.hCursor        = LoadCursor(nullptr, IDC_ARROW);
    wcex.hbrBackground  = (HBRUSH)(COLOR_WINDOW+1);
    wcex.lpszMenuName   = MAKEINTRESOURCEW(IDC_WIN32PROJECT1);
    wcex.lpszClassName  = szWindowClass;
    wcex.hIconSm        = LoadIcon(wcex.hInstance, MAKEINTRESOURCE(IDI_SMALL));

    return RegisterClassExW(&wcex);
}
```

Initialize main window

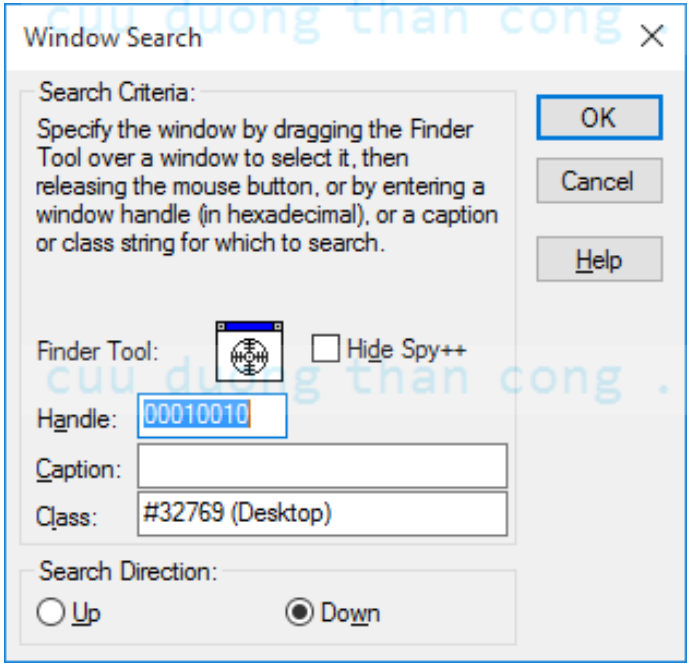
```
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)
{
   hInst = hInstance; // Store instance handle in our global variable

   HWND hWnd = CreateWindowW(szWindowClass, szTitle, WS_OVERLAPPEDWINDOW,
      CW_USEDEFAULT, 0, CW_USEDEFAULT, 0, nullptr, nullptr, hInstance, nullptr);

   if (!hWnd)
   {
      return FALSE;
   }

   ShowWindow(hWnd, nCmdShow);
   UpdateWindow(hWnd);

   return TRUE;
}
```

Some important terms

- **Handle**
  - 32 bits unsigned integer created by OS for an object (window, file, memory, menu,…)
- **ID** (Identifier)
  - Unsigned integer to identify between objects in a program
- Instance
  - Unsigned integer of current instance
- Callback function: function called by another function

Activity – Spy++

- .*Visual Studio installation folder*\Common7\Tools\spyxx_amd64.exe
- Identify a window handle, caption & classname



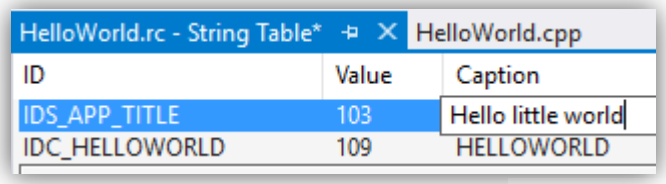
What can we do with handle?

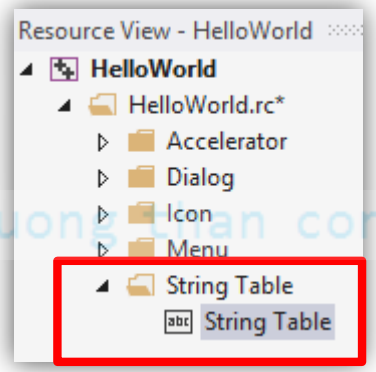
- Get handle using Spy++
- Add button to desktop!
- Change label of OK button of Run dialog to Hello

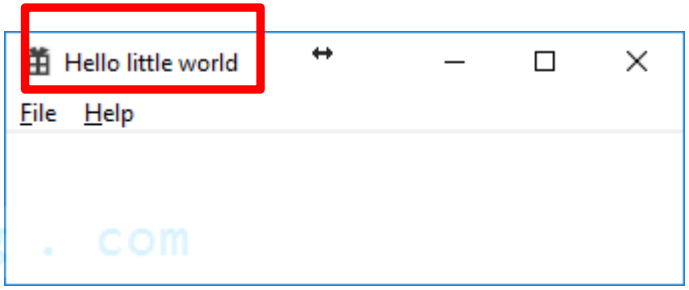
- Force redraw
  - InvalidateRect (hWnd, NULL, TRUE);
    UpdateWindow (hWnd);

Little customization

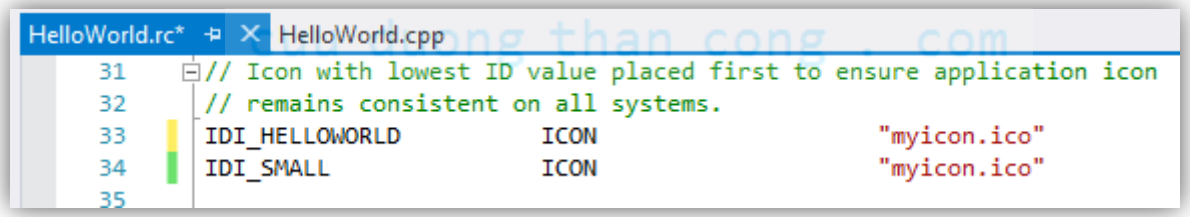
- Change title of program to "*Hello little world*"







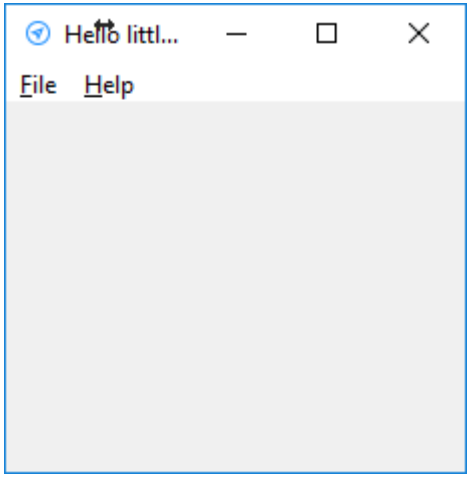
- Change icon of program



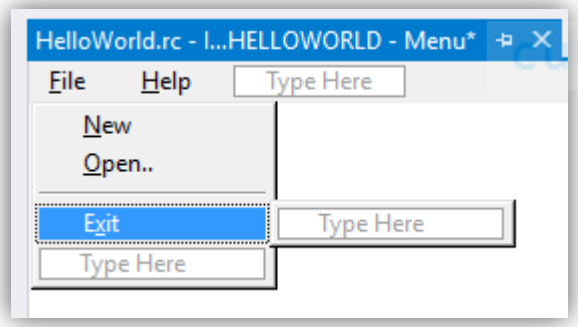
More customization

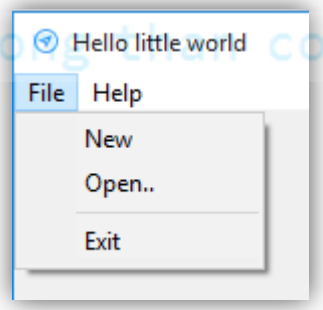
- Change color of background to better color

```
wcex.hbrBackground  = (HBRUSH)(COLOR_BTNFACE + 1);
```



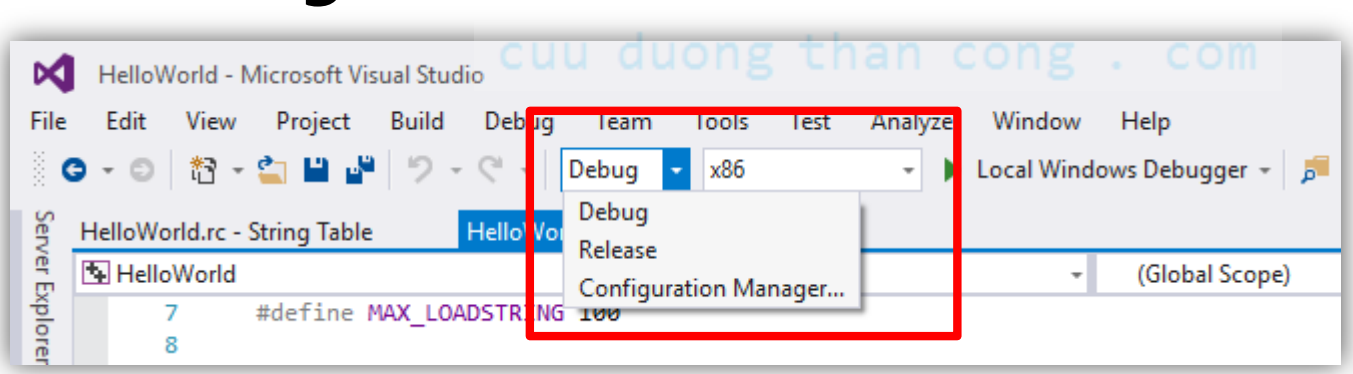
- Change menu to this

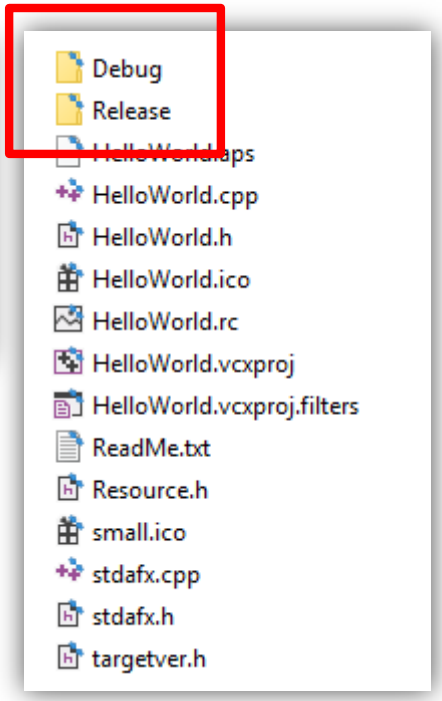




Deliverables

- Find the source code path
  - Open folder in File explorer
- **Debug** & **Release** mode

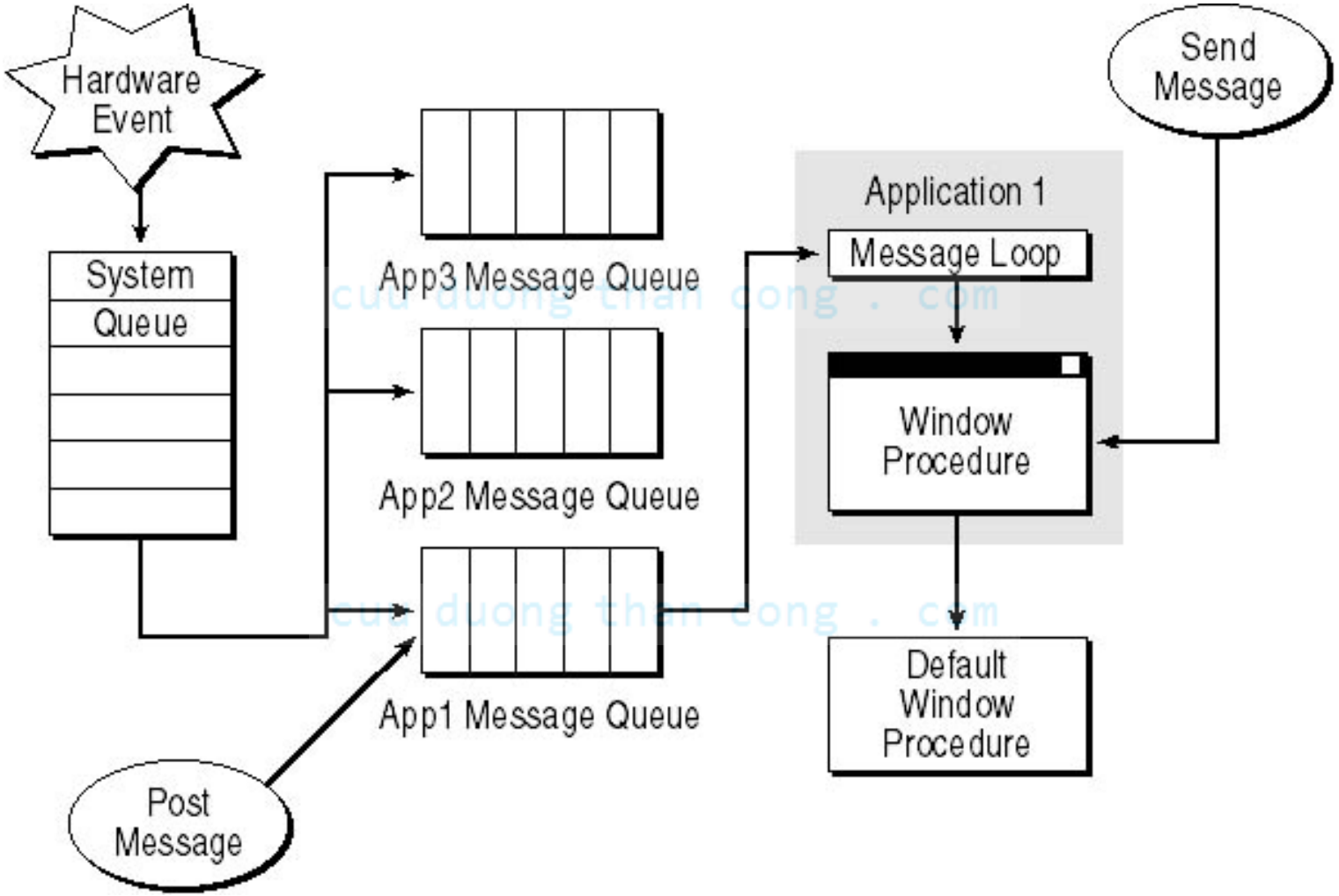




- Remember to **clean** solution before submission!
- Hidden .vs folder!

Event-driven & Messages

Event-driven programming



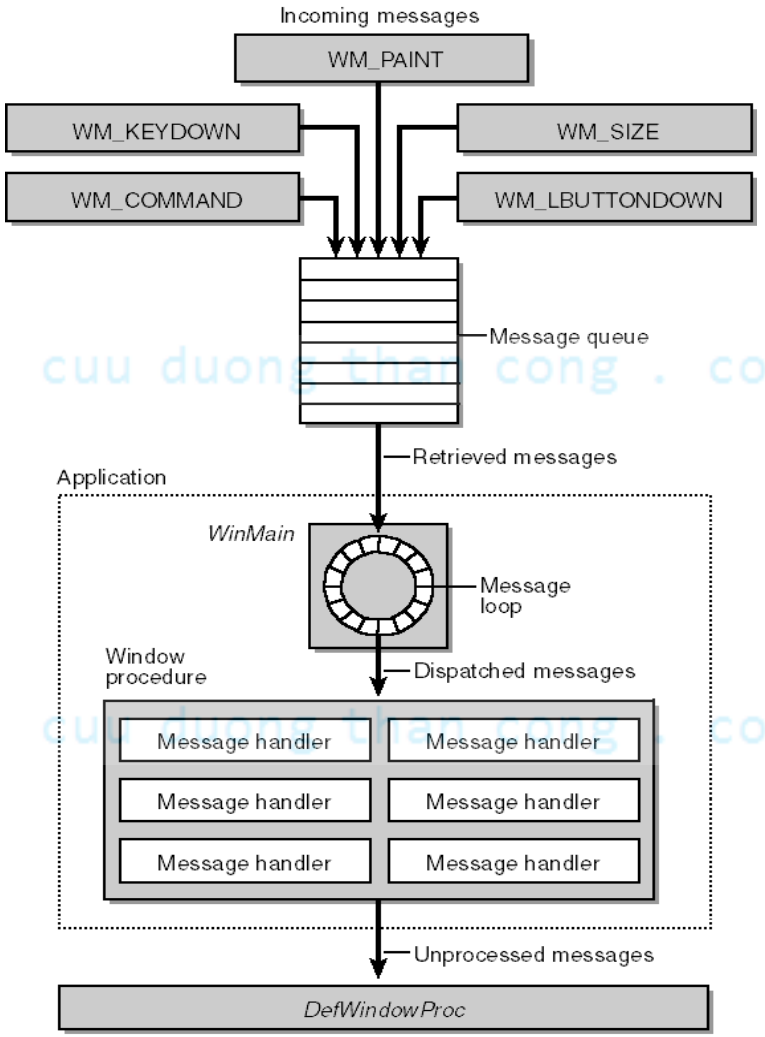
Quiz

- List 10 messages
  - Kick start: A key is pressed

Some messages

Message	Sent when
WM_CHAR	A key is pressed
WM_COMMAND	User click on button, menu
WM_CREATE	A window is created
WM_DESTROY	A window is destroyed
WM_LBUTTONDOWN	Left button is pressed (not released)
WM_LBUTTONUP	Left button is released after being pressed
WM_MOUSEMOVE	Mouse movement
WM_PAINT	A windows needs to be repainted
WM_QUIT	Application is going to close
WM_SIZE	A window is about to change its size

Event processing model



Window messages

- Generated by system & applications

- Activity: Reading comprehension
  - Reading-MessageQueue.docx

Message components

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
```

Window handle

Message identifier

Window procedure

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
    case WM_COMMAND:
        {
            int wmId = LOWORD(wParam);
            // Parse the menu selections:
            switch (wmId)
            {
            case IDM_ABOUT:
                DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
                break;
            case IDM_EXIT:
                DestroyWindow(hWnd);
                break;
            default:
                return DefWindowProc(hWnd, message, wParam, lParam);
            }
        }
        break;
    case WM_PAINT:
        {
            PAINTSTRUCT ps;
```

Flashback - Function pointer

- Pointer to another function to call

God mode: http://www.cprogramming.com/tutorial/function-pointers.html

Two ways for message

1. Postmessage( )
   - Send message into queue
2. Sendmessage( )
   - Send message immediately, skip queue and acquire instant response

Quiz

What should I change if I want to

1. Change icon of the program?
2. Change background color of the main window?
3. Change main window size to 300, 400?

   How can we make it appear in the center?

   How can we make our program appear fullscreen?
4. Save working progress on exit if dirty?
5. Get configuration of the program?

Question

- What if we have multiple windows, how many WndProc does we have to write?

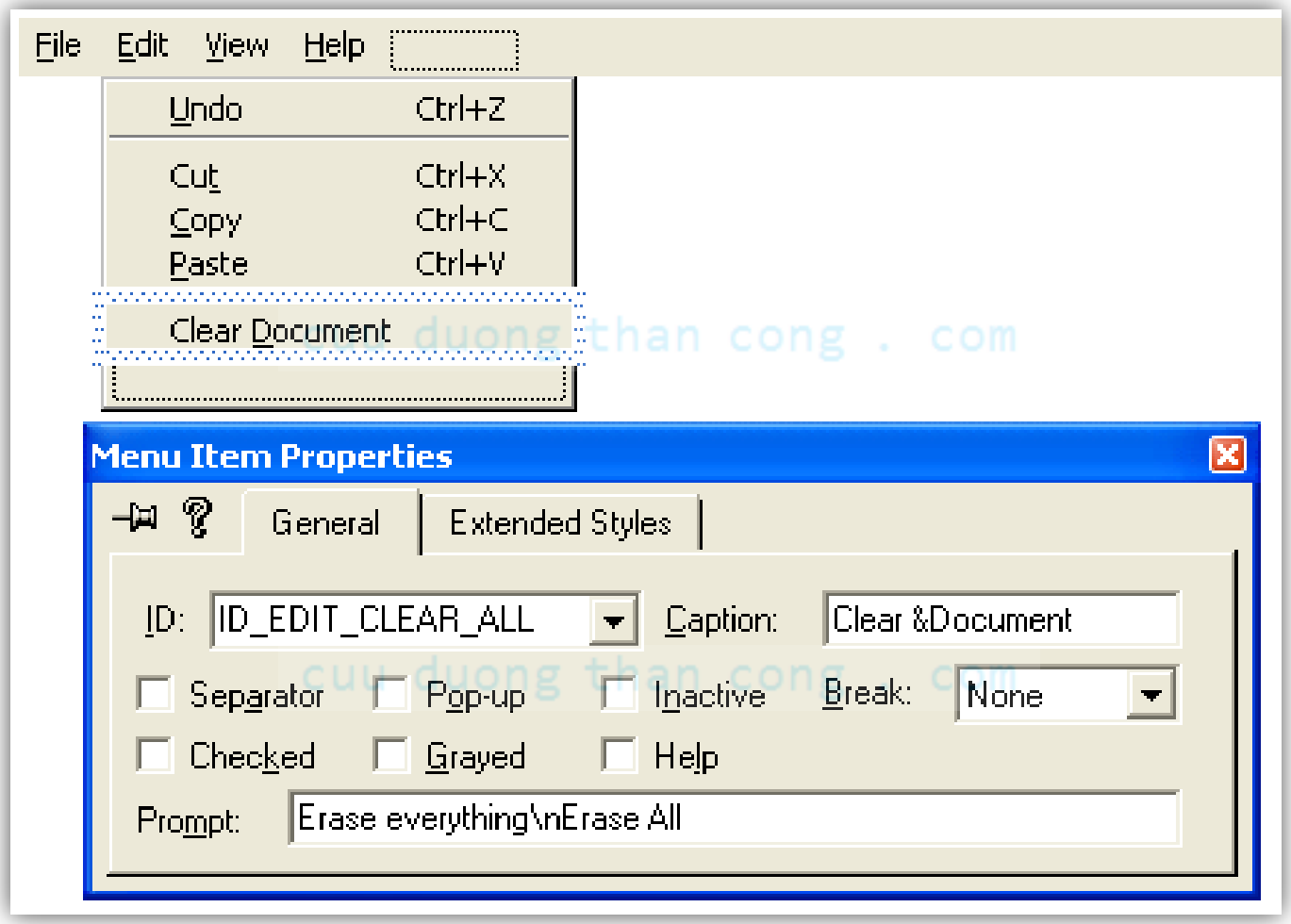
Resources

- Defined in .rc file
- Added to executable file when compiled
- Types of resources
  - **Accelerator table**: hot-key
  - **Bitmap**: bitmap
  - **Caret**, **Cursor**
  - **Dialog box**
  - **Font, Icon**
  - **Menu**
  - **String-table entry**
  - **Version information**

Accelerator table

ID	Key
ID_EDIT_COPY	Ctrl + C
ID_FILE_NEW	Ctrl + N
ID_FILE_OPEN	Ctrl + O
ID_FILE_PRINT	Ctrl + P
ID_FILE_SAVE	Ctrl + S
ID_EDIT_PASTE	Ctrl + V
ID_EDIT_UNDO	Alt + VK_BACK
ID_EDIT_CUT	Shift + VK_DELETE
ID_NEXT_PANE	VK_F6
ID_PREV_PANE	Shift + VK_F6
ID_EDIT_COPY	Ctrl + VK_INSERT
ID_EDIT_PASTE	Shift + VK_INSERT
ID_EDIT_CUT	Ctrl + X
ID_EDIT_UNDO	Ctrl + Z

Menu


File Edit View Help
Undo Ctrl+Z
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Clear Document
Menu Item Properties
General
Extended Styles
ID: ID_EDIT_CLEAR_ALL
Caption: Clear &Document
Separator
Pop-up
Inactive
Break: None
Checked
Grayed
Help
Prompt: Erase everything\nErase All

Quiz – Hungarian notation

- HWND, HINSTANCE, BOOL, LRESULT
- WCHAR, TCHAR, CHAR
- LPCWSTR, LPTSTR
- WPARAM, LPARAM
- cbSize
- szWindowClass, szTitle

- L“” & _T(“”) difference?

Graphical user interface

Types

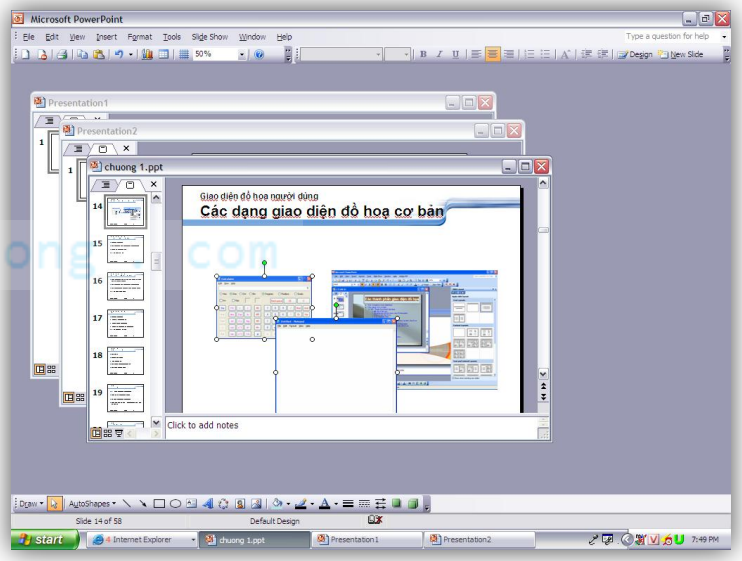
- SDI - Single Document Interface
- MDI - Multiple Document Interface
- Dialog

Single Document Interface - SDI

- One working window
- Resizeable
- No child window
- Ex: Notepad, Paint,…

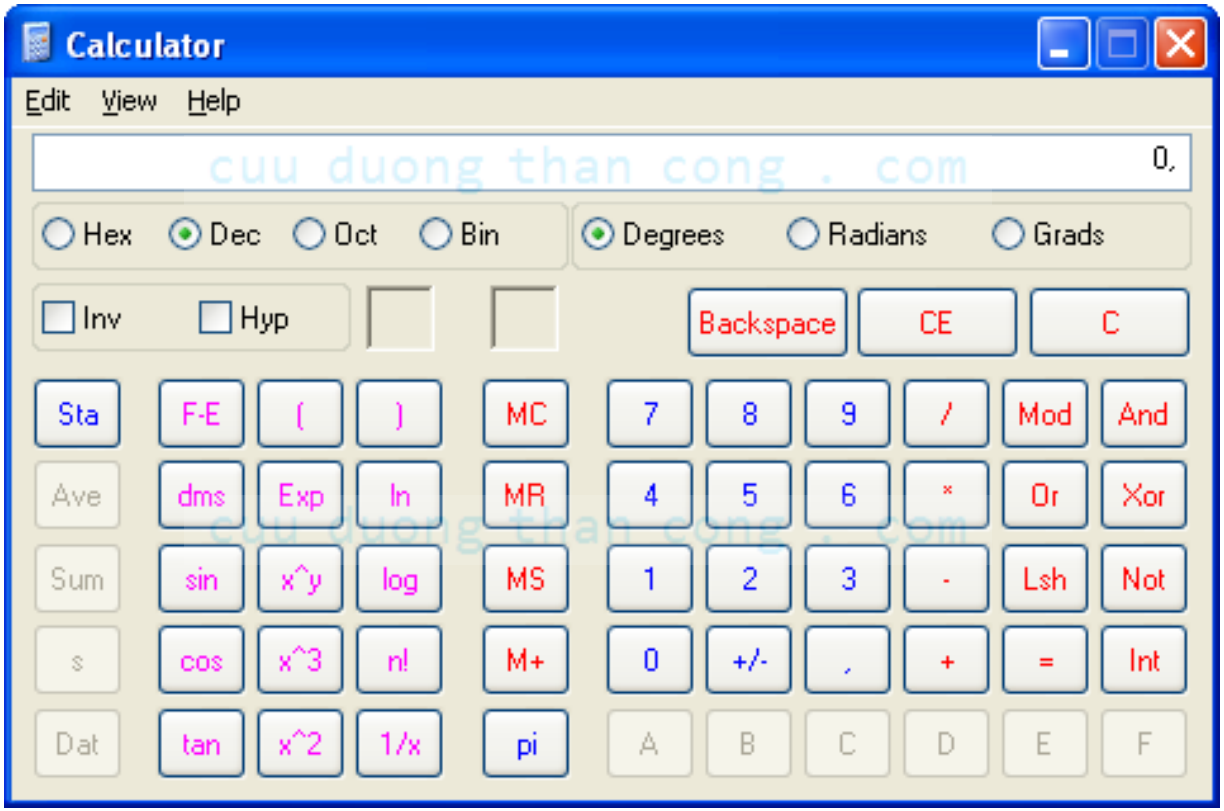
Multiple Document Interface - MDI

- One main window(Frame window) and many child windows
- Resizeable
- Maximize/Minimize/Close child windows
- Ex: Word, Excel, VC++,…



Dialog

- Fixed size



Other OS out there

Linux

- Android
- iOS
- BlackBerryOS
- Tizen…

Refereces

- Windows API Index
  - https://msdn.microsoft.com/en-us/library/windows/desktop/ff818516(v=vs.85).aspx
- Windows data types
  - https://msdn.microsoft.com/en-us/library/windows/desktop/aa383751(v=vs.85).aspx