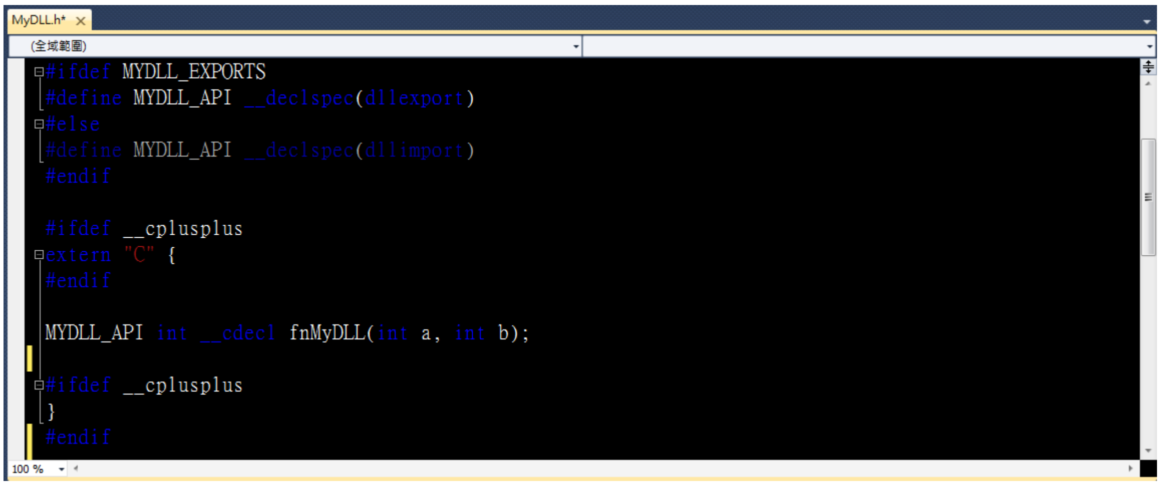
國立臺北科技大學 自動化所 – 人機介面

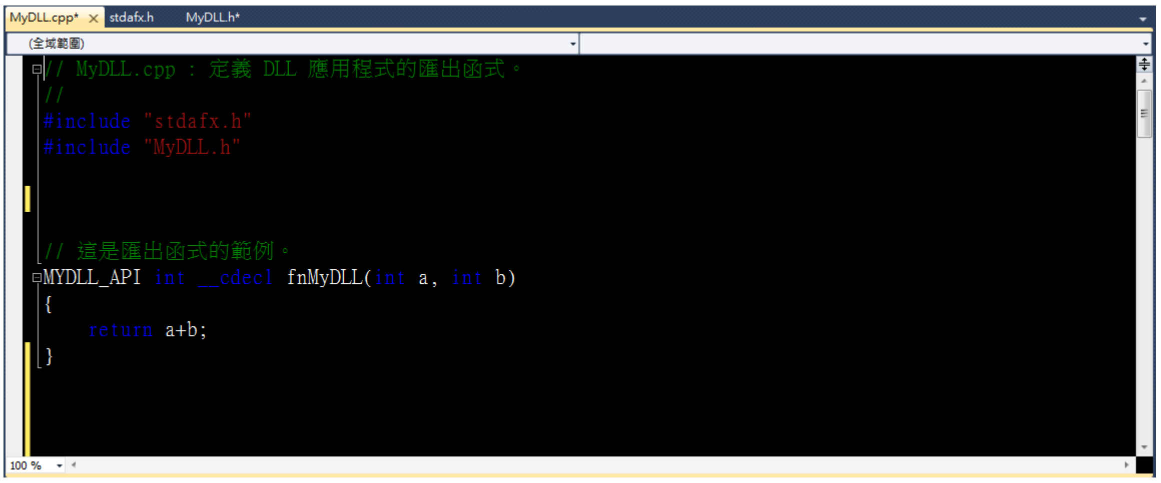
API Design (Flat C API (.dll) for C#)*

The following steps describe how to design a flat C API (.dll), and then this API will be called using C# GUI. These steps are for Microsoft Visual Studio 2010, although the steps are similar for other versions of Visual Studio.

1. Following the instructions of Practice 2 to create your DLL project as shown below.

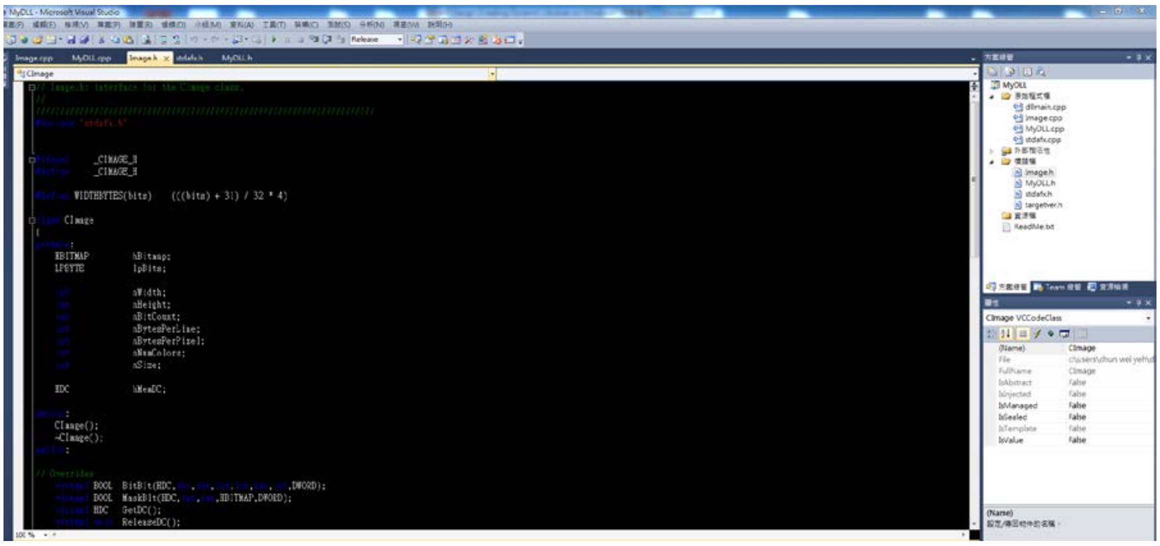


```
#ifdef MYDLL_EXPORTS
#define MYDLL_API __declspec(dllexport)
#else
#define MYDLL_API __declspec(dllimport)
#endif

#ifdef __cplusplus
extern "C" {
#endif

MYDLL_API int __cdecl fnMyDLL(int a, int b);

#ifdef __cplusplus
}
#endif
```



```
// MyDLL.cpp : 定義 DLL 應用程式的匯出函式。
//
#include "stdafx.h"
#include "MyDLL.h"



// 這是匯出函式的範例。
MYDLL_API int __cdecl fnMyDLL(int a, int b)
{
    return a+b;
}
```

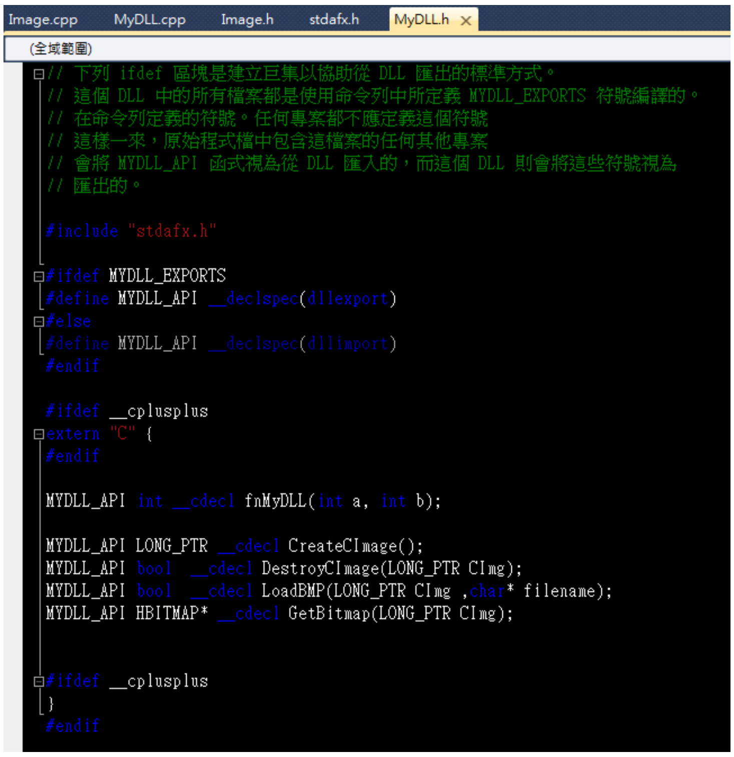
2. Add Image.h and Image.cpp files downloaded from NTUT i School to this project.



```
// Image.h: interface for the CImage class.
//
//////////////////////////////////////////////////////////////////////
#include "stdafx.h"


#ifndef     _CIMAGE_H
#define     _CIMAGE_H

#define WIDTHBYTES(bits)    (((bits) + 31) / 32 * 4)

class CImage
{
private:
    HBITMAP         hBitmap;
    LPBYTE          lpBits;

    int             nWidth;
    int             nHeight;
    int             nBitCount;
    int             nBytesPerLine;
    int             nBytesPerPixel;
    int             nNumColors;
    int             nSize;

    HDC             hMemDC;

public:
    CImage();
    ~CImage();
public:

// Overrides
    virtual BOOL  BitBlt(HDC,int,int,int,int,int,int,DWORD);
    virtual BOOL  MaskBlt(HDC,int,int,HBITMAP,DWORD);
    virtual HDC   GetDC();
    virtual void  ReleaseDC();
```

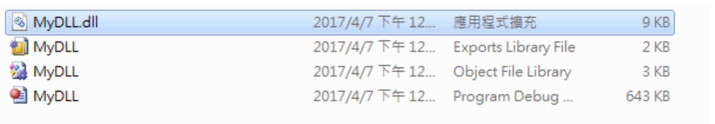
(Name)	CImage
File	c:\users\chun wei yeh\...
FullName	CImage
IsAbstract	False
IsInjected	False
IsManaged	False
IsSealed	False
IsTemplate	False
IsValue	False

3. Design the flat C API for the Class – Cimage



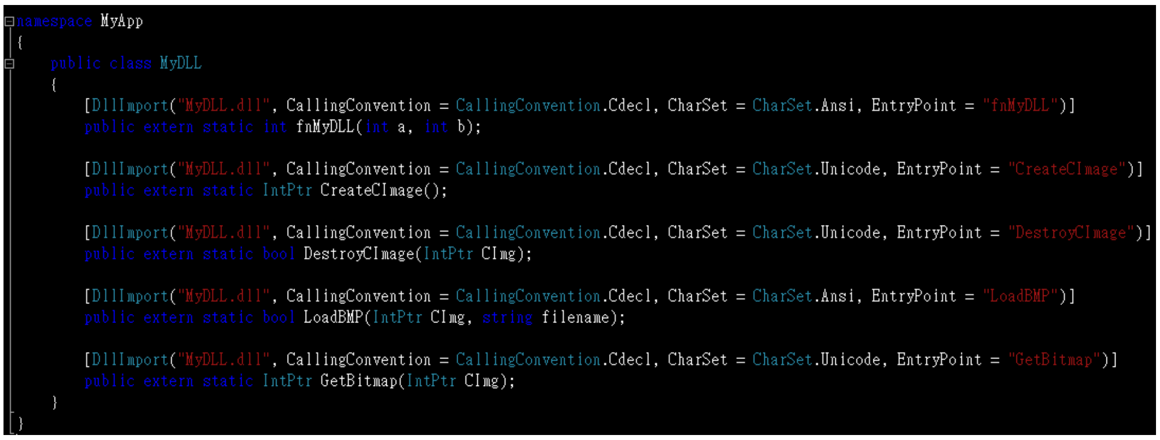
```
// 下列 ifdef 區塊是建立巨集以協助從 DLL 匯出的標準方式。
// 這個 DLL 中的所有檔案都是使用命令列中所定義 MYDLL_EXPORTS 符號編譯的。
// 在命令列定義的符號。任何專案都不應定義這個符號
// 這樣一來，原始程式檔中包含這檔案的任何其他專案
// 會將 MYDLL_API 函式視為從 DLL 匯入的，而這個 DLL 則會將這些符號視為
// 匯出的。

#include "stdafx.h"

#ifdef MYDLL_EXPORTS
#define MYDLL_API __declspec(dllexport)
#else
#define MYDLL_API __declspec(dllimport)
#endif

#ifdef __cplusplus
extern "C" {
#endif

MYDLL_API int __cdecl fnMyDLL(int a, int b);

MYDLL_API LONG_PTR __cdecl CreateCImage();
MYDLL_API bool  __cdecl DestroyCImage(LONG_PTR CImg);
MYDLL_API bool  __cdecl LoadBMP(LONG_PTR CImg ,char* filename);
MYDLL_API HBITMAP* __cdecl GetBitmap(LONG_PTR CImg);


#ifdef __cplusplus
}
#endif
```

4. Build MyDLL.dll file for C# use



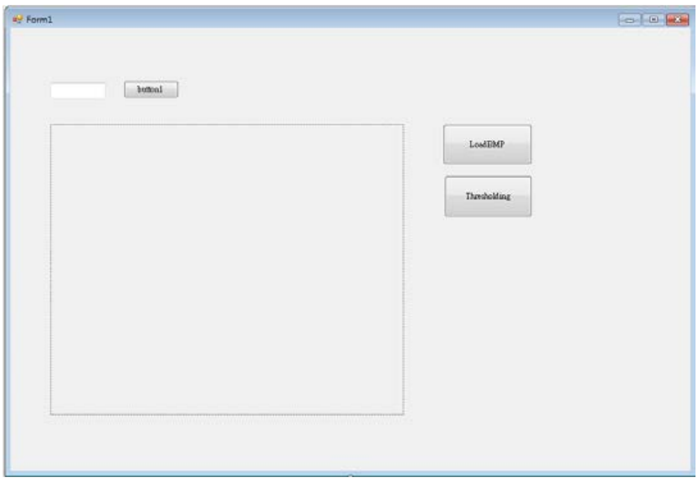
MyDLL.dll	2017/4/7 下午 12...	應用程式擴充	9 KB
MyDLL	2017/4/7 下午 12...	Exports Library File	2 KB
MyDLL	2017/4/7 下午 12...	Object File Library	3 KB
MyDLL	2017/4/7 下午 12...	Program Debug ...	643 KB

5. Modify MyDLL.cs with the new C API functions

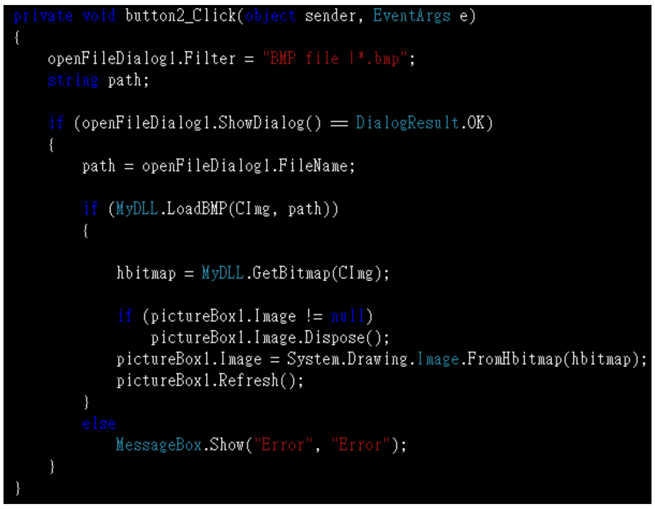


```
namespace MyApp
{
    public class MyDLL
    {
        [DllImport("MyDLL.dll", CallingConvention = CallingConvention.Cdecl, CharSet = CharSet.Ansi, EntryPoint = "fnMyDLL")]
        public extern static int fnMyDLL(int a, int b);

        [DllImport("MyDLL.dll", CallingConvention = CallingConvention.Cdecl, CharSet = CharSet.Unicode, EntryPoint = "CreateCImage")]
        public extern static IntPtr CreateCImage();

        [DllImport("MyDLL.dll", CallingConvention = CallingConvention.Cdecl, CharSet = CharSet.Unicode, EntryPoint = "DestroyCImage")]
        public extern static bool DestroyCImage(IntPtr CImg);

        [DllImport("MyDLL.dll", CallingConvention = CallingConvention.Cdecl, CharSet = CharSet.Ansi, EntryPoint = "LoadBMP")]
        public extern static bool LoadBMP(IntPtr CImg, string filename);

        [DllImport("MyDLL.dll", CallingConvention = CallingConvention.Cdecl, CharSet = CharSet.Unicode, EntryPoint = "GetBitmap")]
        public extern static IntPtr GetBitmap(IntPtr CImg);
    }
}
```

6. Design C# GUI by adding a Picturebox, an OpenFileDialog and a Button as follows.

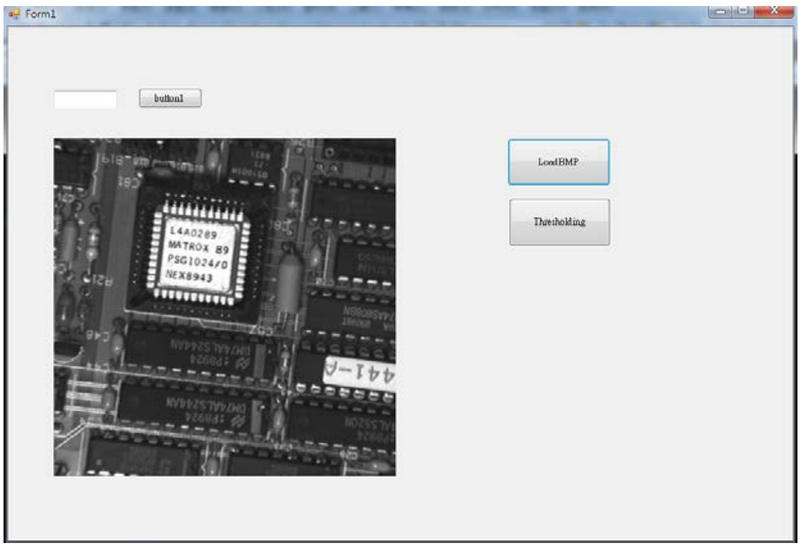


7. Add codes to LoadBMP_click



```
private void button2_Click(object sender, EventArgs e)
{
    openFileDialog1.Filter = "BMP file |*.bmp";
    string path;

    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        path = openFileDialog1.FileName;

        if (MyDLL.LoadBMP(CImg, path))
        {

            hbitmap = MyDLL.GetBitmap(CImg);

            if (pictureBox1.Image != null)
                pictureBox1.Image.Dispose();
            pictureBox1.Image = System.Drawing.Image.FromHbitmap(hbitmap);
            pictureBox1.Refresh();
        }
        else
            MessageBox.Show("Error", "Error");
    }
}
```

8. Build → Build MyApp, and copy the MyDLL.dll file built in step 4 to the folder of MyApp.exe. Now your MyApp.exe is executable. Please load a bmp file, and then show it.



9. Exercises

(a) Add C API of thresholding functions to MyDLL.dll

(b) Call this function in C#