實驗二

Visual 的^符號

於宣告時有使用此符號的物件，表示此物件的類別是屬於 Managed class。
Managed Class 是一種特別的 Class，這邊不多做敘述

一般而言，只要不是我們自己寫的類別，而是 Visual 內建的，往往都是使用 Managed Class。(如： Button, Form, ImageList, PictureBox, String)

此種 Managed Class，從宣告、new、到呼叫函數、使用變數等，都很類似 Pointer 的用法，以下舉例說明：
在 C++，我們自己定義完一個類別(MyClass)之後，除了靜態的方式產生物件，亦可以動態的使用 Pointer 產生物件。如下

```
MyClass *myclass = new MyClass();

myclass->DoSomething();
```

而在 Visual C++中，Managed Class 的使用就很類似 C++ 的 Pointer to object 用法，以下以讀檔時所需要的程式碼為例。

```
StreamReader^ inputFile

    = gcnew StreamReader("inputFile.txt");

inputFile->ReadLine();
```

其對應關係請同學自己觀察。

讀檔(StreamReader)

從前面的例子就可以知道如何讀檔，同學自己嘗試看看。
其中檔案的 Path 部分，直接使用字串輸入絕對或相對位置都可以。
相對位置是以執行檔所在目錄作為目前位置。
另外，要在前面加上

```
using namespace System::IO;
```

Visual 的 String

Visual C++ 中，除了原本 C 與 C++就有的 char[]及 string 之外，尚有另一種字串類別，那就是 String。
同樣的，String 是屬於 Managed Class。
所有 GUI 元件無論是回傳資料或是參數傳遞，與字串相關的常常都是使用此一種類別。大家請特別注意架構的問題！
很多人第一次用 Visual 時，可能會使用 C++的 string 去接 Managed class 回傳的字串。如：

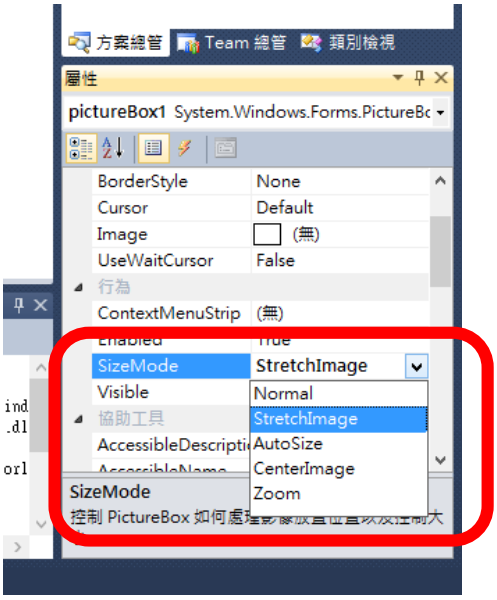
```
StreamReader^ inputFile
    = gcnew StreamReader("inputFile.txt");
string strOfCpp = inputFile->ReadLine();
```

這是因為 String 是微軟自己開發出的新類別，string 則是比較早的 C++建立起來的。理所當然的，string 的 Copy Constructor 不可能接受 String 的類別。
因此，在 Visual C++，建議大家就直接使用 String 而不要再用 string。
所以，上面的例子就必須把最後一行改成

```
String^ strOfCpp = inputFile->ReadLine();
```

PictureBox 的 SizeMode

使用 StretchImage 可以無論原照片解析度，自動的縮小放大圖片到 PictureBox 的大小。



MSDN 搜尋

comboBox 的 MSDN 搜尋範例

物件屬性的存與取

pictureBox1->Left

pictureBox1->Width

Width or this->Width