

1-1 物件導向程式設計簡介

1-1-1 前言

在物件導向程式(Object Oriented Programming, OOP)語言出現前，軟體開發者所使用的語言稱為程序導向語言，例如 DOS 下的 QB、用於 8051 單晶片上的 Keil C 語言就是屬於此類。此類程式在設計上多以**結構化程式設計**(Structured Programming)的方式來進行應用系統的開發，其主要概念就是把問題切割成各個比較小的問題，這些比較小的問題，如果還是複雜到無法以一個函式來解決時，就再切割成為更小的問題。直到可以用一個單一的函式來解決為止。每個函式處理個別的功能，主程式藉由呼叫各個函式，來完成他的工作。

怎樣的程式方能稱為結構化的程式設計呢？一般而言需符合以下的特性：

- 1.由上而下(top-down)程式設計。
- 2.模組化(module)程式設計。
- 3.程式中只能包含循序結構(Sequence Structure)、選擇結構(Selection Structure)、重覆結構(Loop/Iteration/Repetition Structure)。並且少用 GOTO 指令來破壞邏輯結構等。

採用結構化程式設計的方式將具有以下幾個優勢：

- 1.使程式易於了解與閱讀。
- 2.一個入口、一個出口減少程式邏輯錯誤。
- 3.減少程式維護成本與提高程式設計師的生產力。

當然，凡事有利必有弊，結構化程式設計也免不了有些令人遺憾的缺失，這些缺點最主要有以下幾點：

- 1.所需指令較多，使程式變大，且占用主記憶體空間較多。
- 2.程式執行時間較長。
- 3.資料與方法之間缺乏關連性等。
- 4.當程式龐大時後續維護相當困難。

這種程式設計的方式可想而知，當程式龐大時維護就會變得很困難，即使只是要修改某一個小功能，也必須追蹤整個程式的邏輯，物件導向設計的出現正是為了避免此一缺點。**一個程式語言是否為物件導向最主要評估因素有三：封裝(Encapsulation)、繼承(Inheritance)與多型(Polymorphism)，簡稱為 PIE**。本節先導入封裝的概念，另外兩個留待後面章節再行介紹。

所謂封裝，就是把程式**指令與資料包裝起來成為一個黑盒子**，並定義黑盒子對外的溝通介面。例如 MP3 隨身聽，我們不知它如何設計，也毋須知道，只要弄清楚機器上對外的每一個插孔的用途，知道何時及如何使用即可。同樣的道理，封裝好的物件，外界的使用者，無須知道物件內部如何執行他的作業。只需要知道，如何使用物件的介面，來完成自己的工作即可。所以，設計物件的開發者，日後要更新物件的程式時，只要維持物件的介面不變，便

可以確保，使用這物件的外部程式，不需要改寫，一樣可以執行。而外界程式的開發者，只要遵循不直接存取物件內部的功能及資料，僅透過介面來使用物件的規則，便可以確保程式執行無誤。

1-1-2 學習目標

- 1.明白類別與物件的分別，並能舉例說明。
- 2.了解封裝觀念與事件驅動(Event Driven)觀念。
- 3.能使用 Visual Studio 2015 設計一個簡單的 Windows Form Application。

1-1-3 相關知識

(一)、何謂類別?何謂物件?

類別(Class)就如設計藍圖，物件(Object)則是根據此藍圖所實際造出的物體。所以，同一類別的物件，基本功能是相同，但細部的某些性質可不同。例如利用同一房屋設計藍圖所建造出來的房子其基本功能與型式應是相同，但可能屋瓦的顏色卻不同，如圖 1-1-1 所示。



圖 1-1-1 類別：藍圖

物件 1：紅瓦房子

物件 2:黑瓦房子

(二)、物件結構

程式中物件的結構可參考下圖 1-1-2，在 Windows 的系統下，每開啟一個視窗就相當於執行了一個程式，如果將此程式所佔用的記憶體空間放大來看就是一個個的物件，再將每一個物件所佔用的記憶體空間放大出來，就可清楚看到分成兩部份：**資料與程式**。其中資料分成兩種欄位與屬性，程式是以副程式為單位分為事件與方法兩種，說明如下。

1. 欄位：物件中對外公開的變數。
2. 屬性：封裝在物件中的資料，需按一定機制存取。例如圖 1-1-2 的"乘除運算"視窗程式中有兩個按鈕(Button)，顯示的文字是"相乘"與"相除"，這些顯示文字是儲存在 Button 物件上的 Text 屬中，所以若想更改 Button 上顯示的文字，就必須將 Text 屬

性修改為新的字串。

3. 方法：封裝在物件中的副程式，需自行呼叫。例如 PictureBox 物件中的 Load 方法可載入指定圖片顯示於 PictureBox 物件上。
4. 事件：封裝在物件中的副程式，會因使用者的動作而自動執行。例如 Button 中的 Click 事件，會因使用者在此 Button 上按下滑鼠左鍵而自動執行。

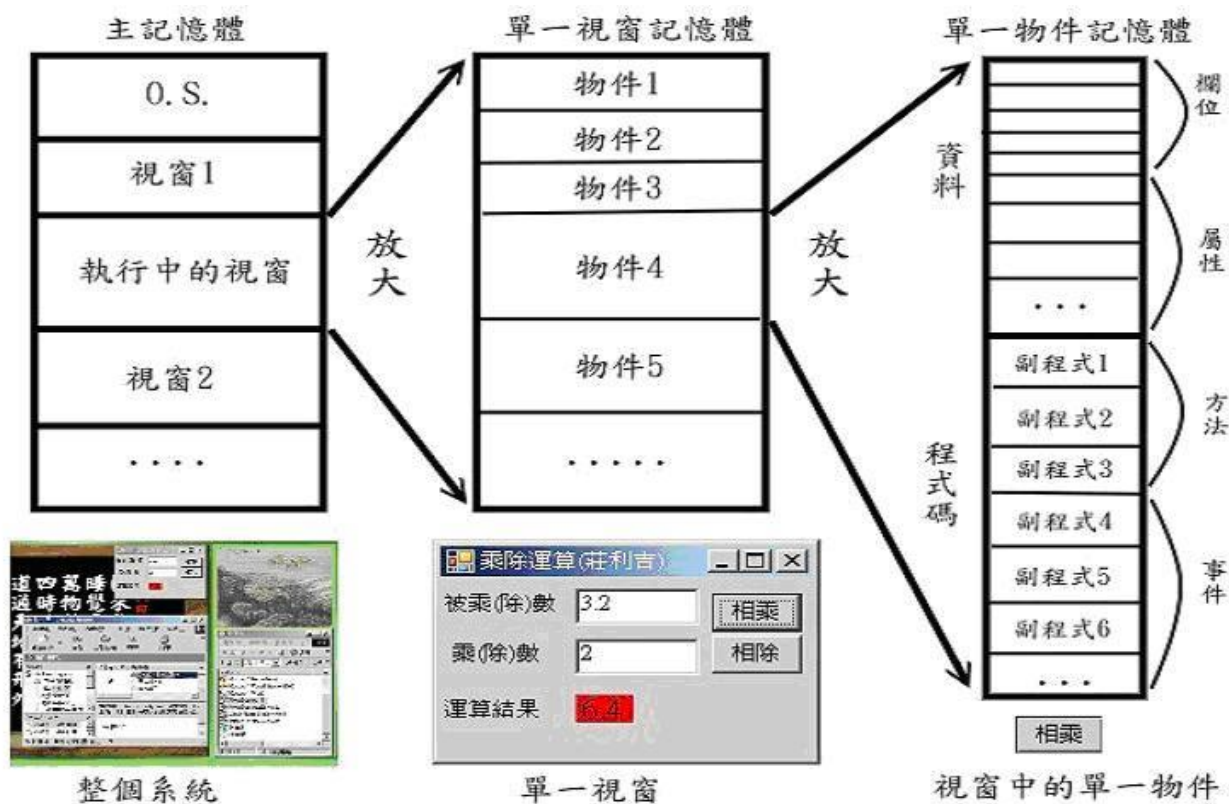


圖 1-1-2 物件結構

(三)、物件導向程式設計步驟

1. 決定所需的物件
2. 設定各物件的必要屬性
3. 撰寫各物件必要的事件程式碼
4. 執行及偵錯

1-1-4 程式範例

(一)、程式功能

設計一個標準視窗程式，上面有一個 Button 與 Label，當使用者在 Button 上按下滑鼠左鍵時，會在右邊 Label 上顯示"半徑為 3 的圓面積為 28.27431"的訊息，如下圖 1-1-3。



圖 1-1-3 程式執行畫面

(二)、設計步驟

1. 啟動 Visual Studio 2015，選**建立專案**、選 **Visual C#**、選 **Windows**，在 Visual Studio 安裝的範本處選 **Windows Form 應用程式**，如下圖 1-1-4。在名稱處輸入程式專案名稱，中英文皆可，此處修改為**顯示圓面積**。



圖 1-1-4 Windows Form 應用程式專案

2. 輸入完後按確定進入程式表單設計畫面，會顯示如圖 1-1-5 畫面。此時 Visual C# 已自動產生一個空白表單(Form)供程式設計師擺放所需物件。

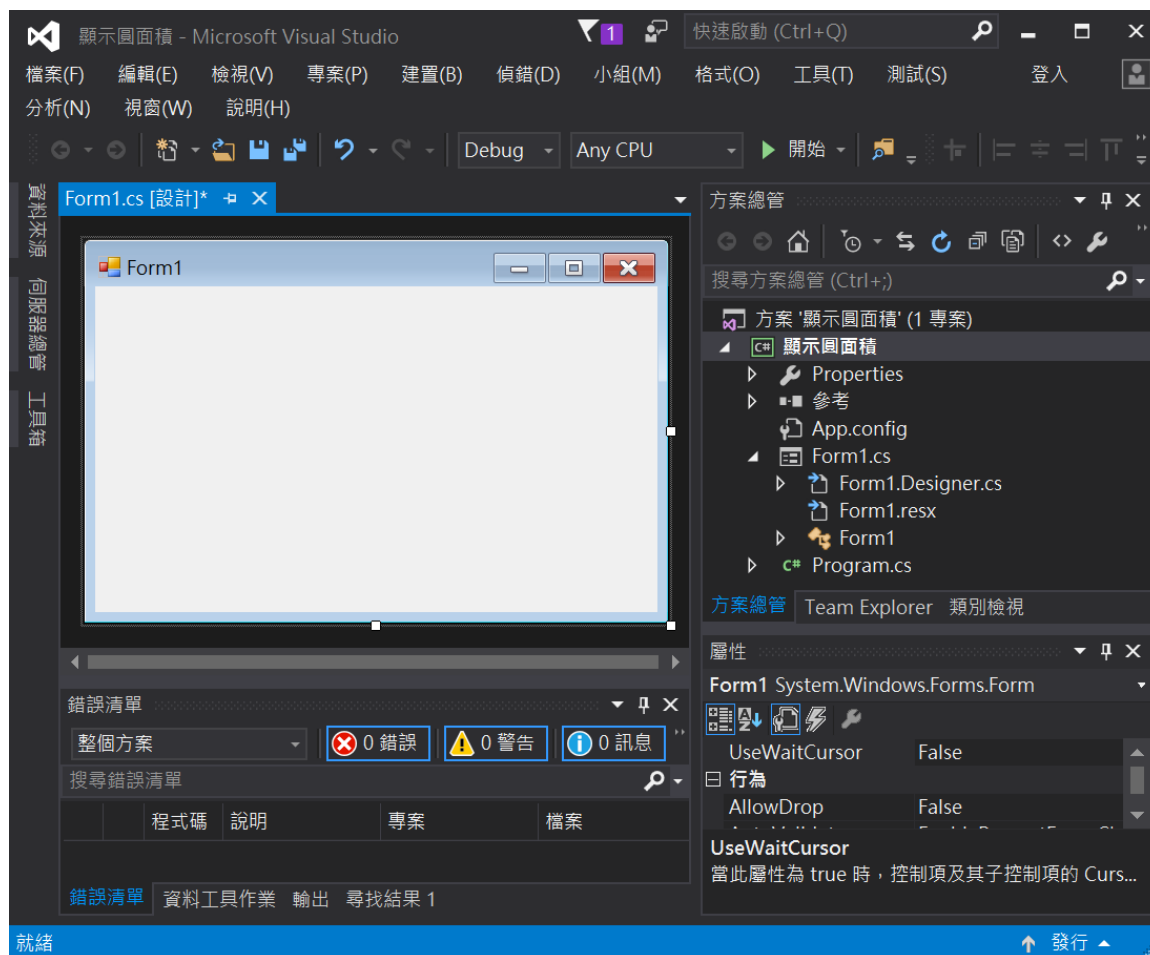


圖 1-1-5 空白表單

3.接著開始部署物件。將滑鼠移到左邊工具箱，然後在 Button 與 Label 類別圖示上 Double Click，就可在表單產生物件然後拖曳到表單上的適當位置，也可以自行從工具箱拖曳物件到表單也，結果如圖 1-1-6。

4.物件產生後接者開始修改其屬性，先點選 Button1 物件，在右下角屬性視窗處找到 Text 屬性(可先點 A→Z 排序圖示方便尋找)，然後輸入顯示圓面積，找到 Name 屬性(物件名稱)確認為 Button1。再點選 Label1 找到 Text 屬性修改為圓面積顯示處，找到 Name 屬性(物件名稱)確認為 Label1。最後點選最外圍表單，同樣找到 Text 屬性修改為顯示圓面積(莊利吉)，此時表單抬頭就會修正為輸入文字，完成後如圖 1-1-7。

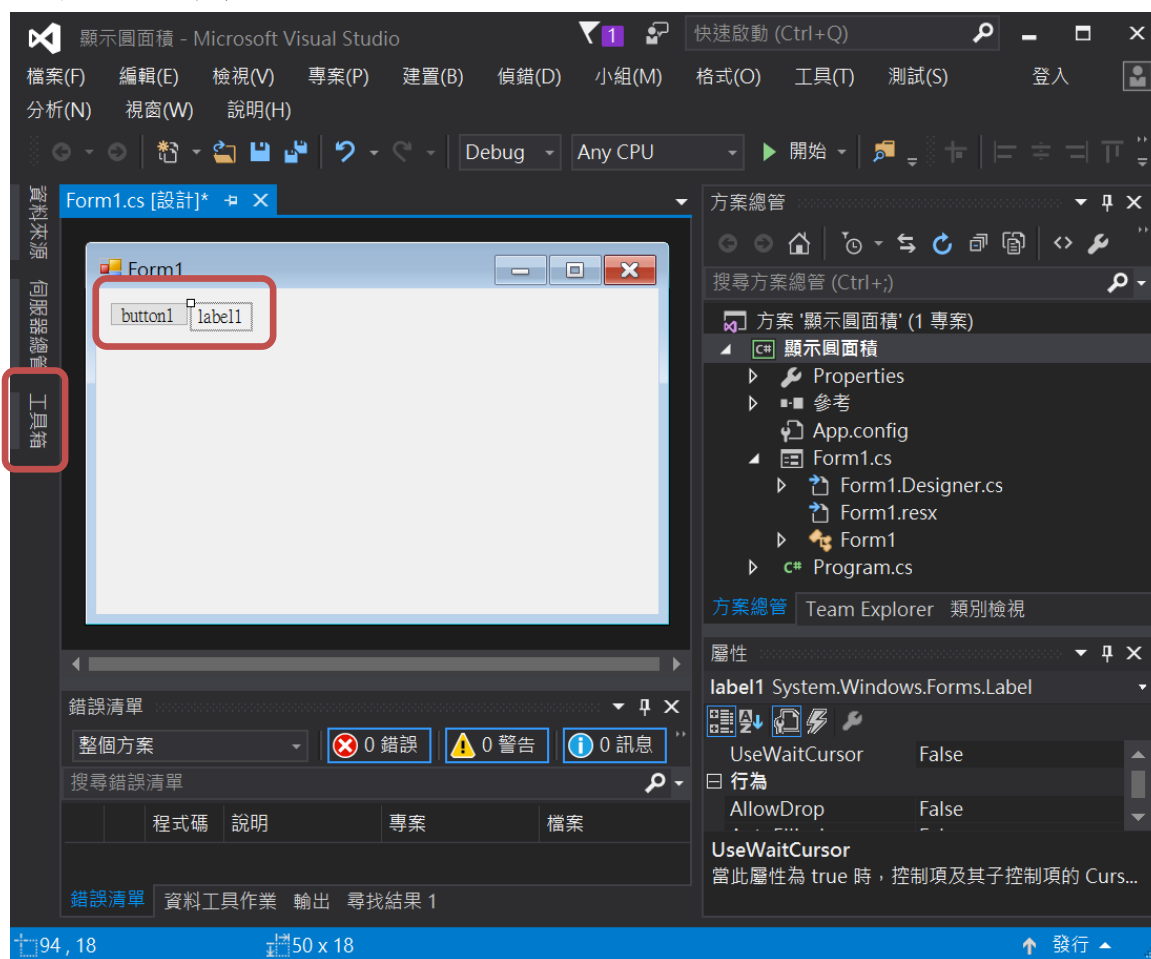


圖 1-1-6 加入所需物件

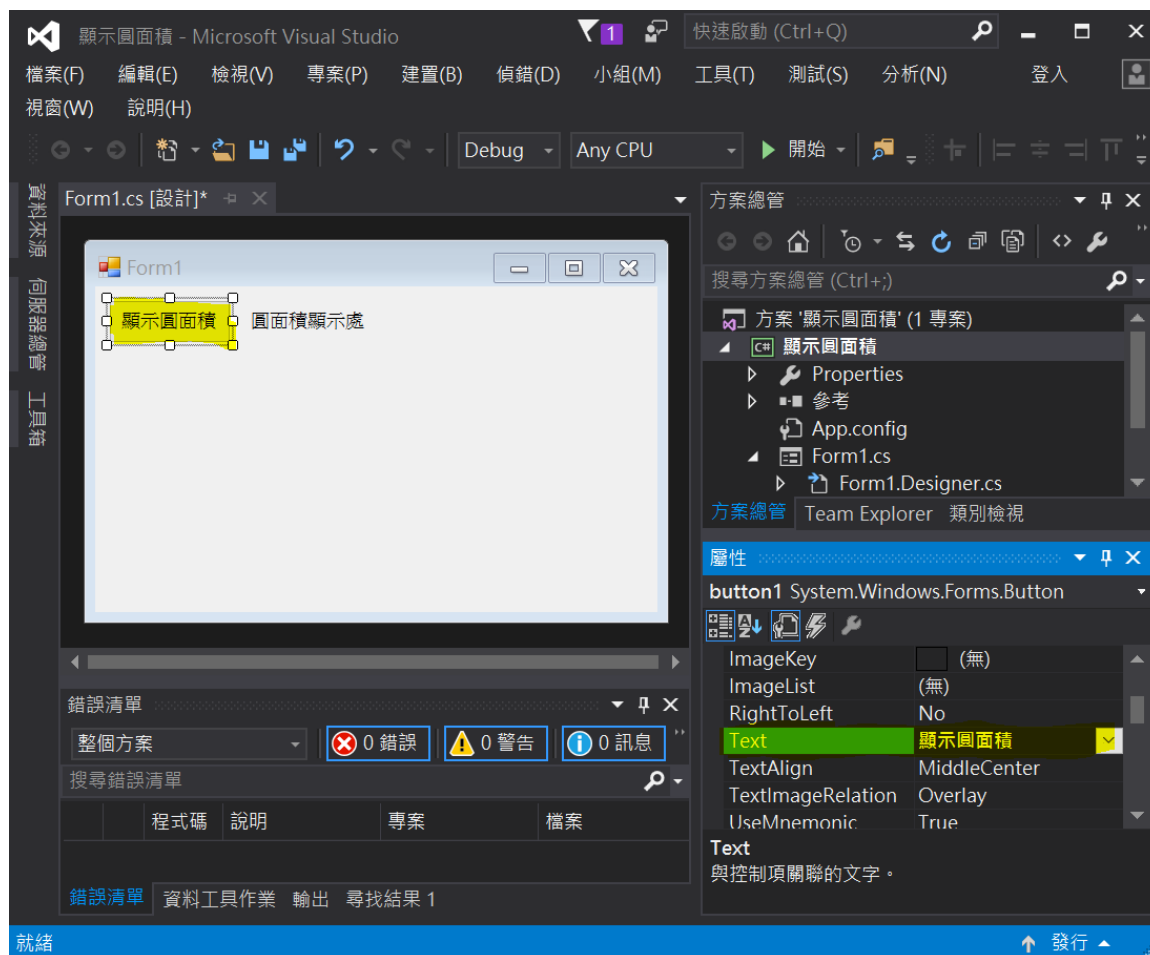



圖 1-1-7 設定物件屬性

5. 撰寫事件程式碼。在方案總管處點選 Form1.cs，然後按工具列上的檢視程式碼圖示 ，以便切換至檢視程式碼畫面。在 Form1 表單上點選物件 Button1 兩下左鍵，此時 C# 會自動產生一個空白事件副程式如下。

```
private void button1_Click(object sender, EventArgs e)
{
}

```

6. 因為要顯示圓面積在 Label1 上，所以必須將要顯示訊息放到 Label1 的 Text 屬性，最終結果如下圖 1-1-8。

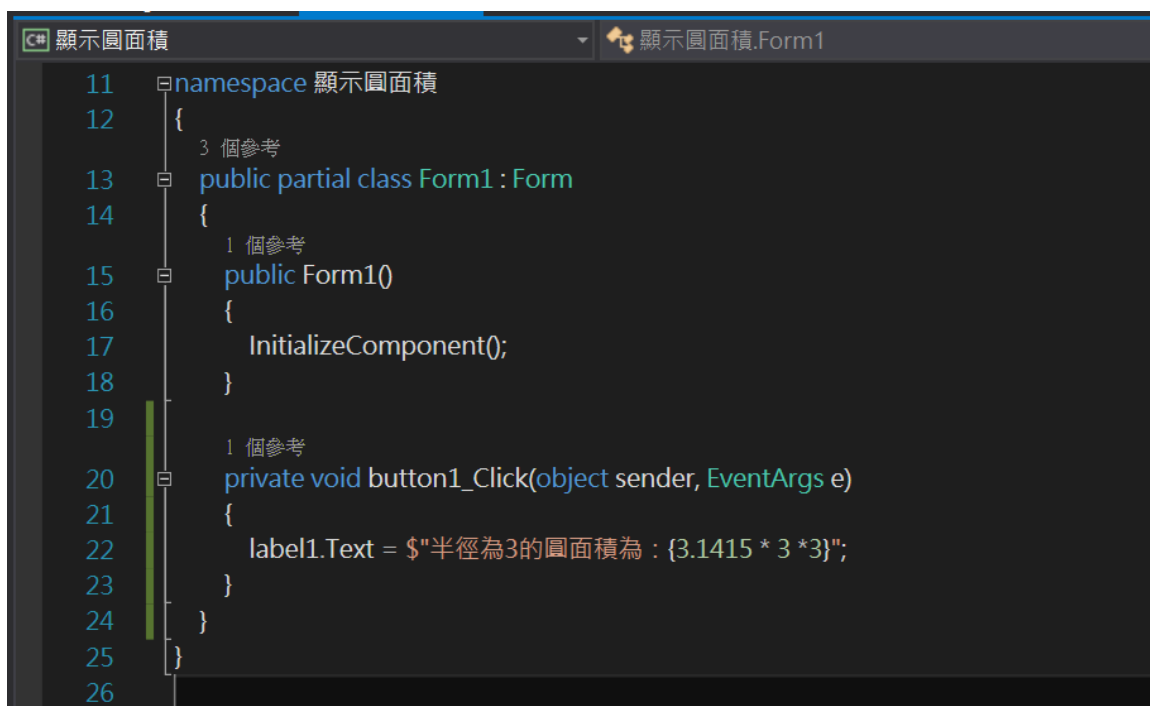


圖 1-1-8 撰寫事件程式碼

7.測試並偵錯。最後點選工具列上的開始偵錯圖示，如圖 1-1-9，就可執行程式，顯示如圖 1-1-3 的畫面。

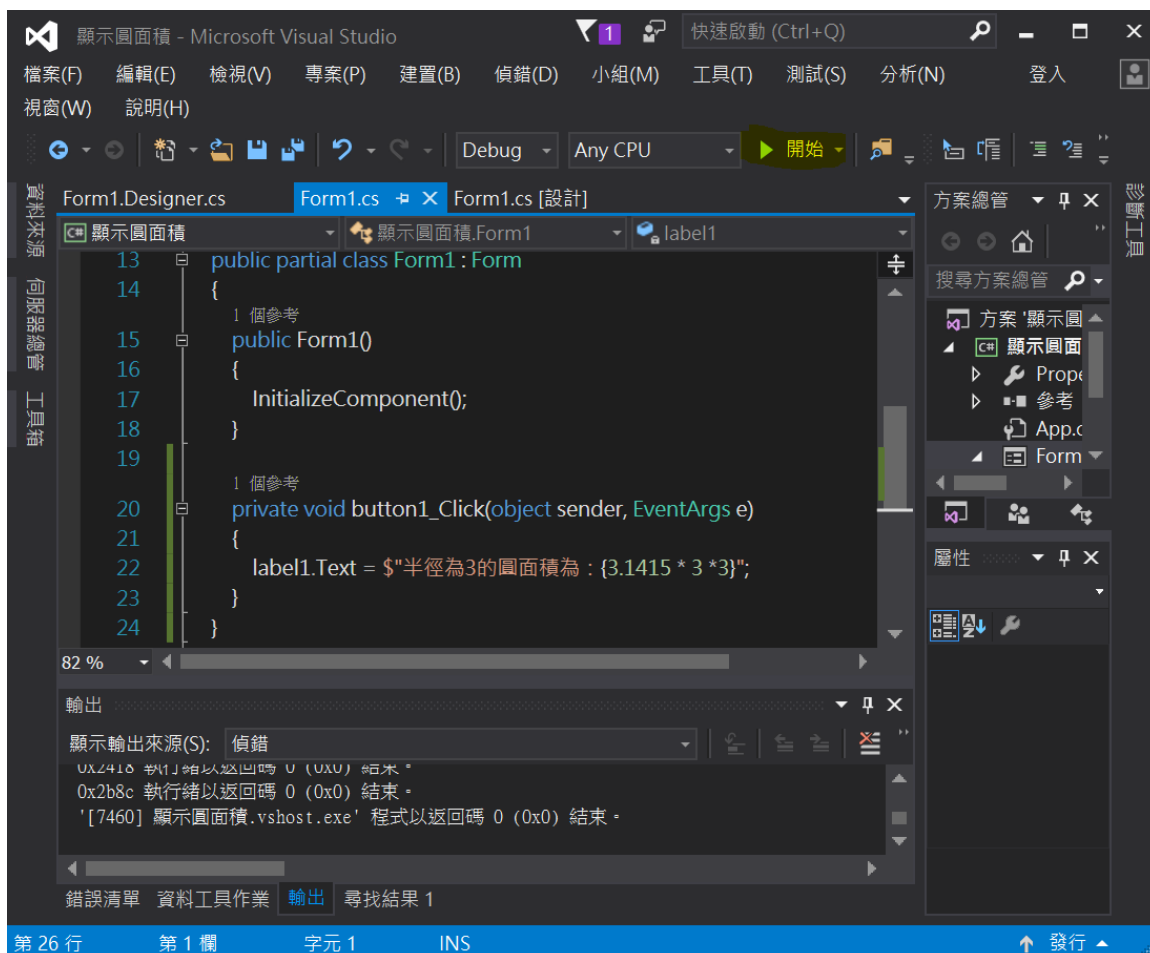


圖 1-1-9 執行程式

(三)、程式儲存與修改

1. 欲儲存程式，可按工具列上全部儲存圖示，如圖 1-1-10，然後按瀏覽選擇儲存位置，如圖 1-1-11，建議勾選為方案建立目錄，然後按儲存。

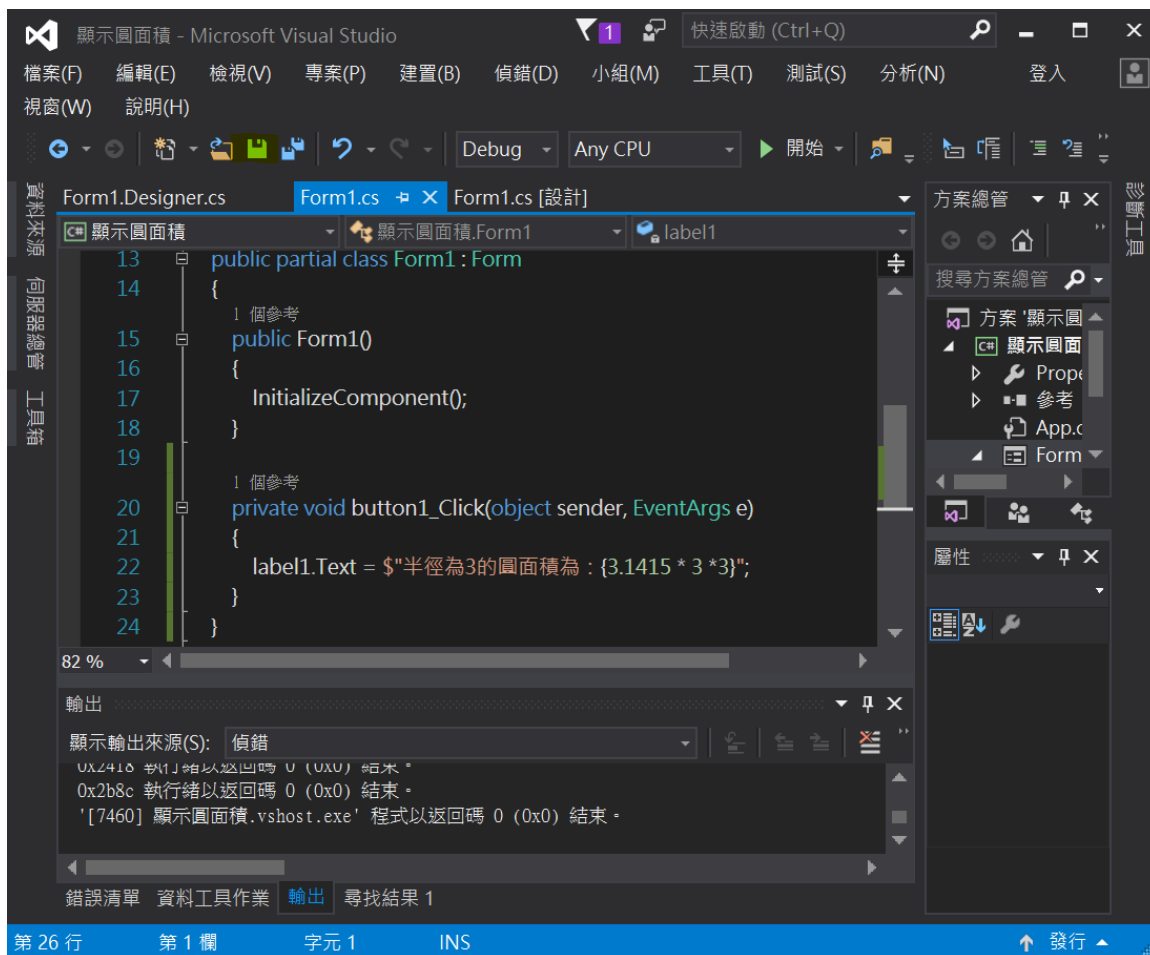


圖 1-1-10 儲存所有相關檔案

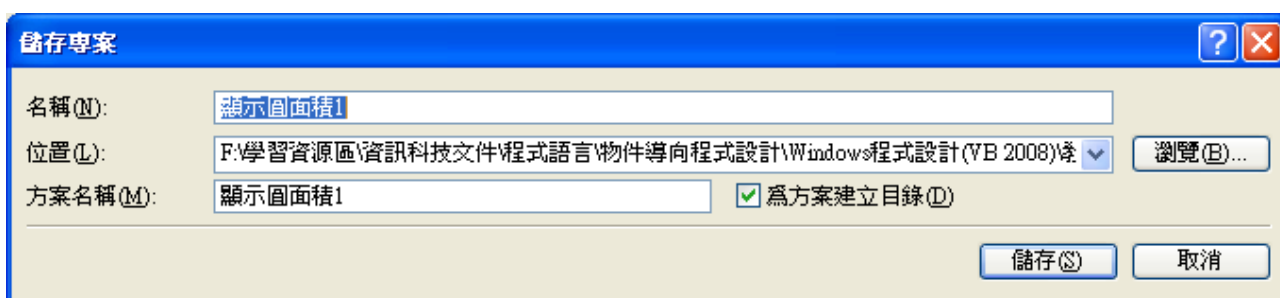


圖 1-1-11 選擇儲存位置

2. 檢視儲存的資料夾後會發現如圖 1-1-12 的結構，其中副檔名為 sln 者就是程式專案，日後如要修改程式直接在此 sln 專案上 Double Click 即可開啟程式。但真正的表單與程式是存在下一層同名的資料夾下，如圖 1-1-13。當要拷貝程式時必須從上層的資料夾開始整個子資料夾都拷貝方可。

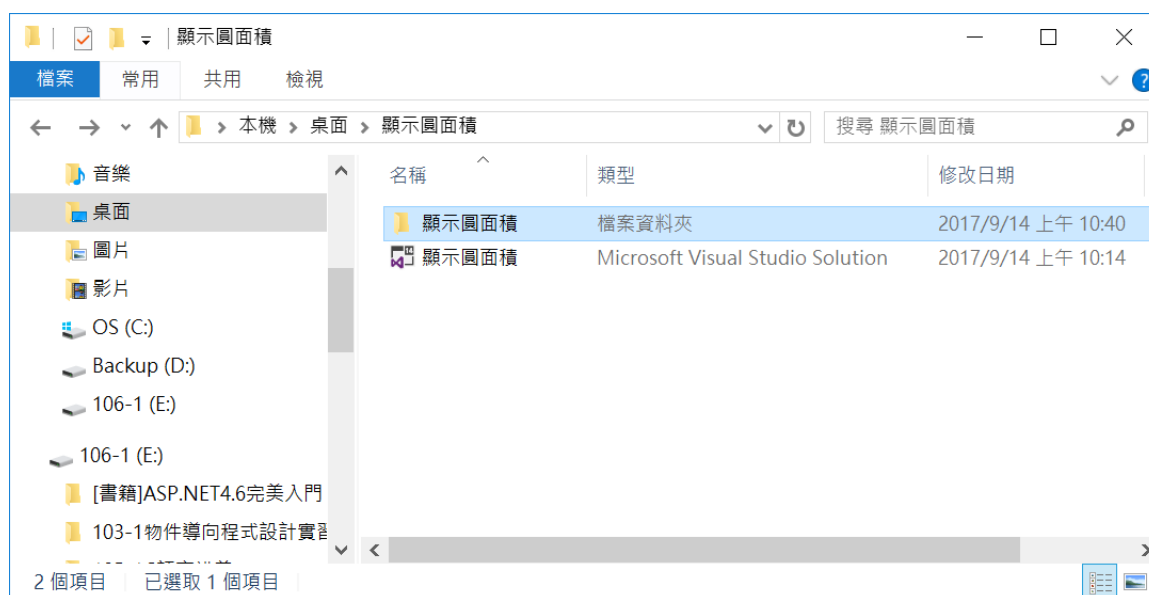


圖 1-1-12 程式專案結構

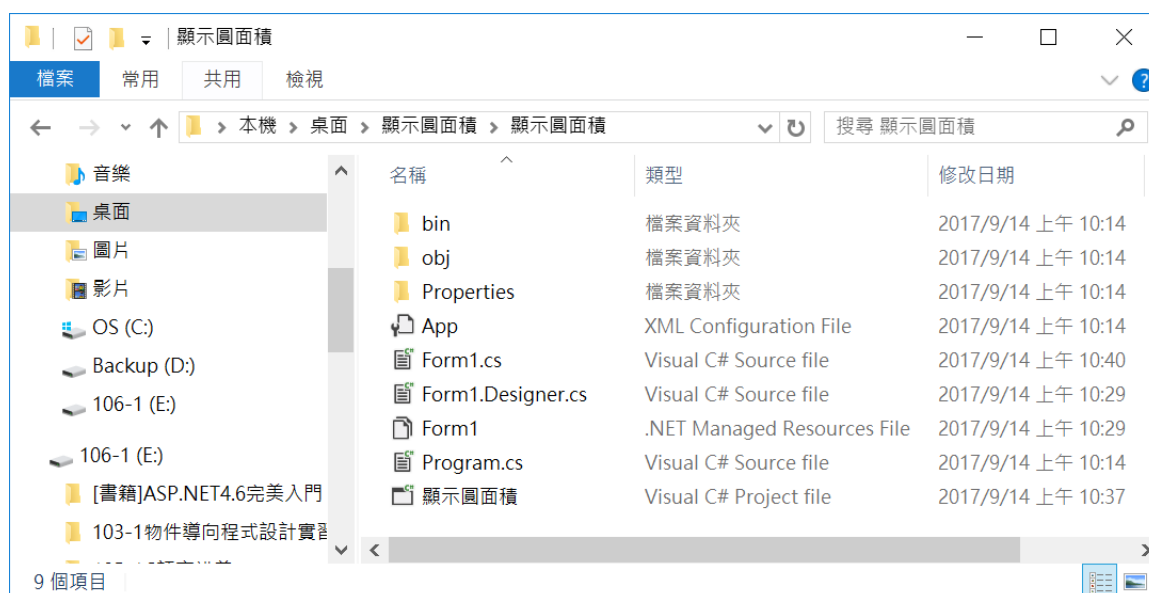


圖 1-1-13 專案所包含檔案

1-1-5 習題

仿照上面程式範例顯示邊長為 5 的正方形之周長。