附錄C

Visual Basic 6.0

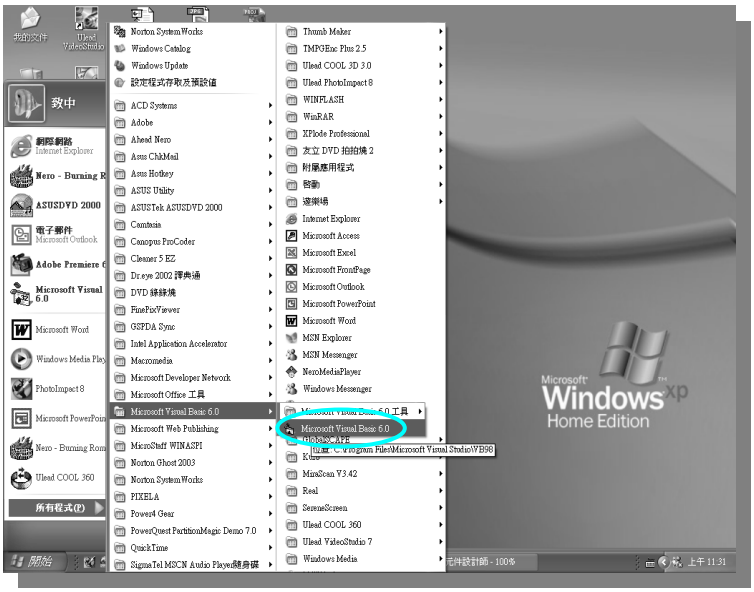
基本操作說明

C-1 進入與離開 Visual Basic 6.0
C-2 Visual Basic 工作視窗介紹
C-3 Visual Basic 基本組成要素
C-4 Visual Basic 資料型態
C-5 Visual Basic 常數與變數
C-6 Visual Basic 陣列介紹
C-7 Visual Basic 運算子介紹
C-8 Visual Basic 選擇性判斷指令介紹
C-9 Visual Basic 模組（Module）介紹
C-10 Visual Basic 6.0 完整操作流程介紹

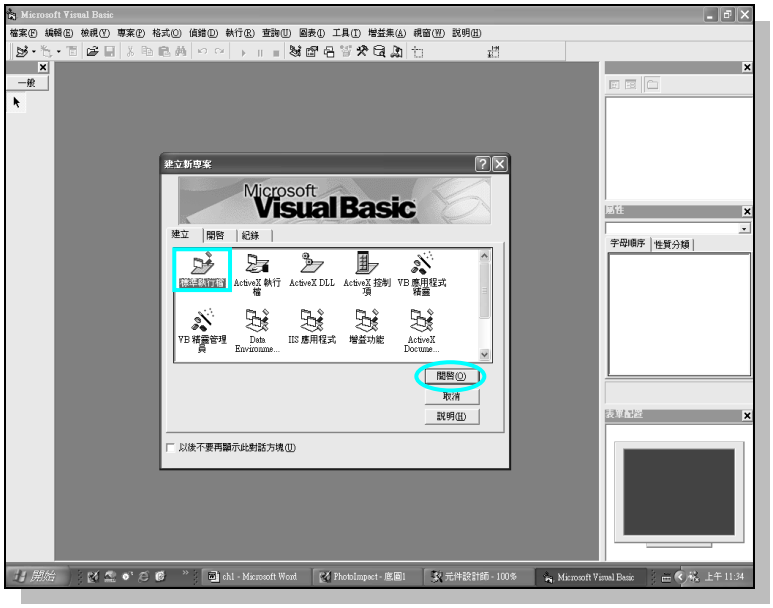
C-1 進入與離開 Visual Basic 6.0

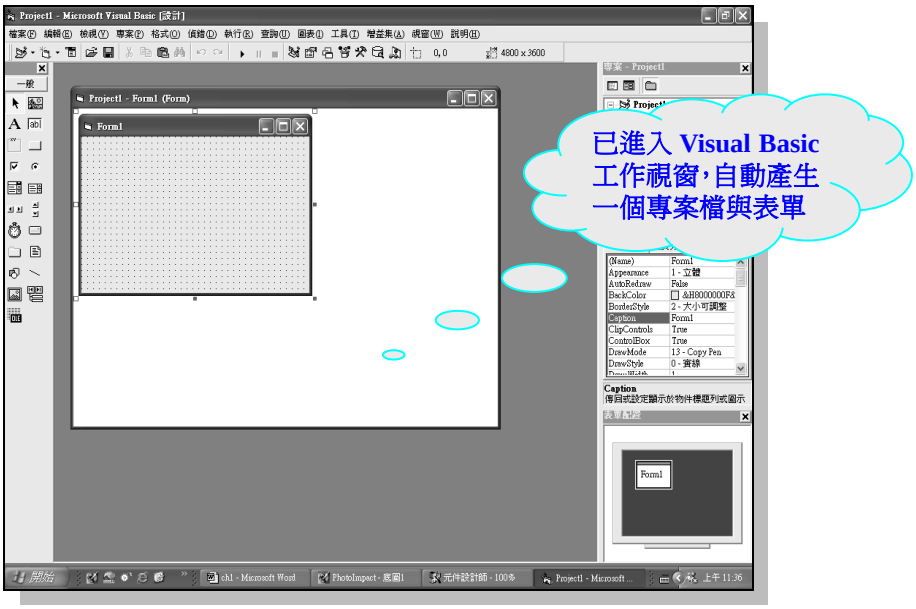
Visual Basic 是採用『**視覺化**』圖形的操作介面，使用者可以直接點選按鈕或圖示（Icon）的方式，直接來進行程式的撰寫與視窗外觀的設計。首先我們先開機實際操作如何進入與離開 Visual Basic 6.0。

1. 按『**開始 → 程式集 → Microsoft Visual Basic 6.0 → Microsoft Visual Basic 6.0**』。



2. **在『標準執行檔』上，利用滑鼠連續左點兩下，直接開啟空白專案**，或點選之後按『**開啟**』鈕皆可。

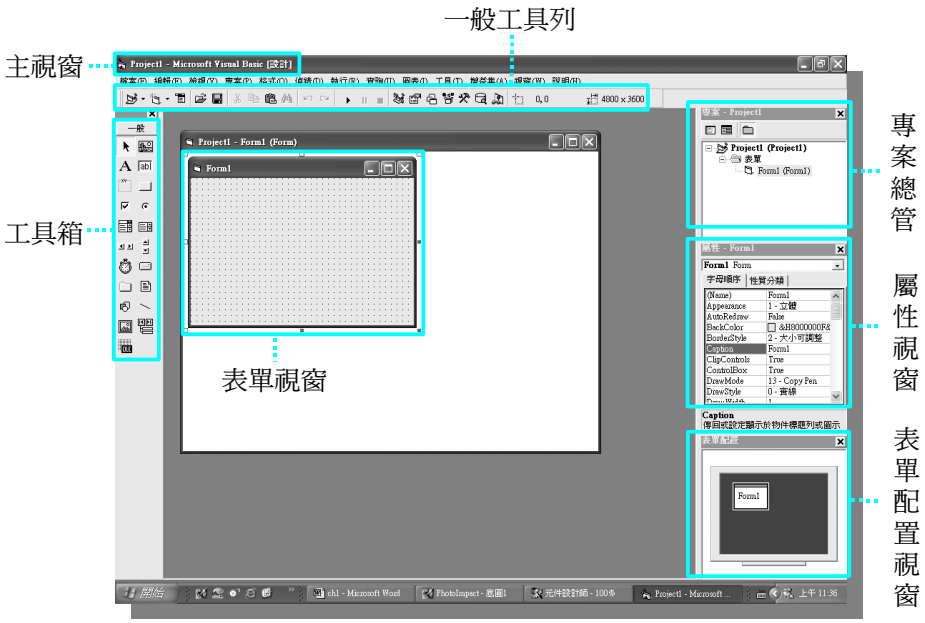




3 按『**檔案 → 結束**』可以關閉該程式。

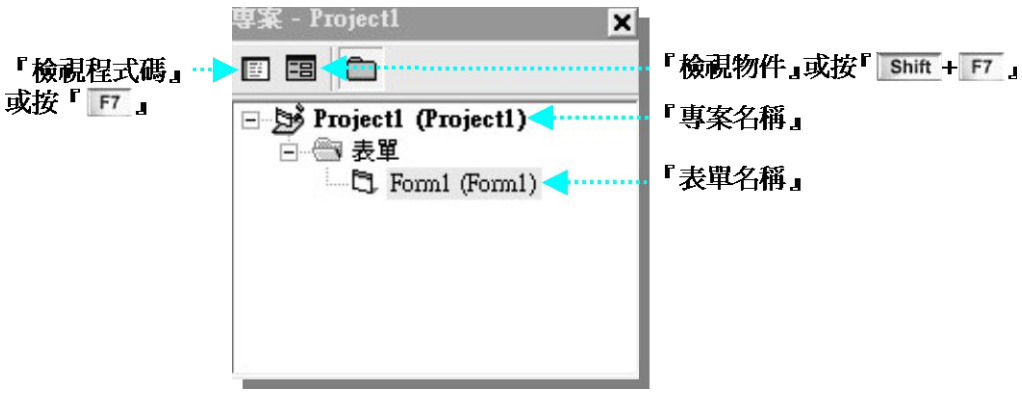
註：當關閉之前會要求儲存所有相關檔案，需全部都按『確定』鈕進行儲存，且盡量放置在桌面或 C 碟的根目錄，以方便尋找修改。

C-2 Visual Basic 工作視窗介紹



由上圖我們可以看到 Visual Basic 工作視窗包含主視窗、表單視窗、專案總管視窗、屬性視窗、表單配置視窗以及工具箱，分別詳細說明如下：

1. **主視窗**：用來顯示專案名稱以及工作模式。
2. **表單視窗**：可讓您在應用程式中建立視窗、對話方塊及控制項。
3. **專案總管視窗**：用來管理本專案所有表單與程式，並可利用『**檢視程式碼**』與『**檢視物件**』按鈕切換，以方便使用者撰寫程式，**這兩個按鈕經常使用應熟記**，若無法顯示該視窗，則按一般工具列的『**檢視→專案總管**』。



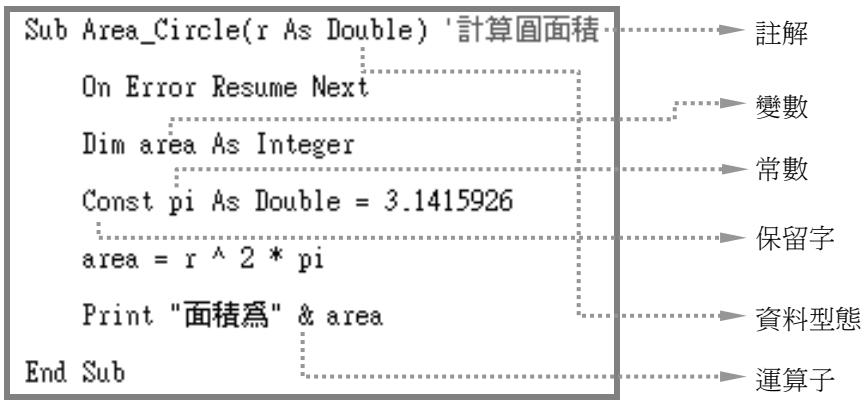
4. **屬性視窗：**用來瀏覽或設定每個控制物件自己的屬性（例如名稱、顏色、大小、樣式…），不同的物件有不同的屬性，**若要變更屬性，僅需在屬性上方點擊兩下即可**，若無法顯示該視窗，則按一般工具列的『**檢視 → 屬性視窗**』。

在許多的控制物件都有『***Index***』屬性，它可傳回或設定在集合物件中唯一可辨識物件數目。按照預設值，***Index*** 屬性被設定為在集合物件中建立物件的順序，當有多個相同物件時，物件名稱必須不同，若需使用相同名稱，則要採用陣列方式，稱為**控制項陣列**，例如：Command1(0)、Command1(1)、Command1(2)，括號內的值即為 ***Index*** 的內容。

為了加快測驗時間，以及簡化程式內容，建議先設定好第一個 Command1 各項屬性後，再利用 ***Ctrl+C*** 複製物件，然後再按 ***Ctrl+V*** 貼上物件，此時會出現如下圖的詢問視窗，務必選擇 是(Y) 按鈕，利用控制項陣列會比較方便，此時原物件名稱已變為 Command1(0)與 Command1(1)，其 ***Index*** 屬性自動設為 0 與 1，當再按 ***Ctrl+V*** 貼上第三個按鈕物件，此時將不再詢問，直接會設定為 Command1(2)名稱。

C-3 Visual Basic 基本組成要素

每一種程式語言，都有其特定的格式，在設計程式時，我們必須遵循該程式語言的使用規則。首先我們可以看到 Visual Basic 語言是由下面四個基本要素所組合而成（如下圖所示），包括註解、變數、常數與保留字等，現在分別說明如下：



- **註解（Comment）**：為程式碼的說明，可用來提高程式的可讀性，並不會影響程式的執行，且可放置於程式碼任意位置，但須以『**單引號**』為起始，緊接著所有文字皆視為註解，一般預設值以『**綠色**』呈現，例如，上圖的『**'計算圓面積**』。
- **變數（Variable）**：一種命名之儲存位置，可在程式執行階段存放可更改之資料。每一變數在其有效範圍內均有單一的名稱，在同一個有效範圍內必須是唯一的，例如上圖的『area』。
- **常數（Constant）**：常數可以是字串、數值、另一常數、任何算術運算子或邏輯運算子的組合，例如上圖的『pi』。
- **保留字（Reserved Word）**：為 Visual Basic 程式碼中具有特別意義的字元組合，例如列印資料的敘述中必須使用『***Print***』保留字，所以在撰寫程式時，不能再以『***Print***』作為常數或變數的名稱。

  保留字最大的特徵是無論您輸入的程式碼皆為大寫或小寫，當游標移到另外一行時，保留字會自動變成大小寫，且字型顏色也會跟著改變，因此，當不小心輸入錯誤的保留字程式碼，應可立即察覺，並加以修正錯誤，所以在撰寫程式時，應全程都使用小寫較佳。

C-4　Visual Basic 資料型態

Visual Basic 程式中用到的資料，可以區分為多種不同的類型，例如數值資料、字串資料、布林資料與日期資料，以下分別介紹這四種資料型態，說明如下：

- **數值類（*Value*）**：數值類資料有整數、長整數、單精準度與倍精準度等等，不同數值資料類型，所佔用空間大小不同，各種資料類型的說明如下表所示：

資料類型	英文名稱	數值表示範圍
整數	***Integer***	**－32,768～32,767**
長整數	***Long***	－2,147,483,648～2,147,483,647
單精準度	***Single***	正數：1.401298E-45～3.402823E+38 負數：－3.402823E+38～－1.401298E-45
倍精準度	***Double***	正數：4.94065645841247E-324～ 1.79769313486232E+308 負數：－1.79769313486232E+308～ －4.94065645841247E-324

- **字串類（*String*）**：是由若干個中文字、英文字母、數字或特殊符號所組成。在撰寫程式時，字串的前後均需使用『**雙引號**』符號。
- **布林類（*Boolean*）**：通常用來表示條件的成立與否，若條件成立，則資料值為 ***True***（真）；若條件不成立，則資料值為 ***False***（假）。
- **日期類（*Date*）**：適用於表示日期與時間的資料型態，資料的前後必須以符號"#"包圍，例如：**#2008/3/22#** 表示日期格式。
- **可變資料類（*Variant*）**：如果一個變數沒有明確宣告其他資料型態，那麼均會自動變成 ***Variant*** 資料型態。

下一章節將介紹如何宣告常數或變數的資料型態，通常我們會利用『**一般程序**』，以宣告常數或變數的資料型態，其目的是為了在整個專案的所有程序都可使用該宣告的常數或變數，若是變數的宣告還可以保存其內容，我們稱之為『**生存期**』。**若只是在個別的程序中宣告時，則該變數僅可存在於該程序之中，一旦脫離該程序，則該變數內容會自動清除**，因此，在使用上要特別注意。

C-5 Visual Basic 常數與變數

常 數

在運算過程中不會改變內容的資料項稱為『常數』，例如：圓周率為 3.14，其值是永遠不會改變的。每個主應用程式皆可定義自己的常數，使用者也可採用 ***Const*** 陳述式來定義附加的常數，同時您可以在程式中的任意位置使用常數以代替真正的值。在程序中使用 ***Const*** 陳述式時，通常是將 ***Const*** 陳述式放在程序最開始的地方。常數宣告的語法如下：

Const 常數名稱 ***As*** 資料型態 = 常數內容

當我們在撰寫程式，若是**經常重複**引用某一數字，例如在計算圓的面積或是圓周都要用到圓周率（3.14），若採用常數來取代可有下列三種優點：

- **讓程式較易修改**：若需將圓周率 3.14 改為 3.1415926535897932，則只需修改常數定義的地方即可，而不用修改整個程式碼。
- **減少程式錯誤的機率**：圓周率 3.14 若為 3.1415926535897932，您會很容易打錯。
- **讓程式易於閱讀**：當使用常數名稱（pi）會比使用一大串的數字（3.1415926535897932）更容易記憶。

Visual Basic 的常數可分成下列兩種：

- **內建常數**：Visual Basic 的內建常數大多以 ***vb*** 當開頭，每一內建常數都有其特定的含意，在程式碼中任何地方都可以替代實際數值使用，例如：我們可以利用 ***vbRed***（紅色）、***vbBlue***（藍色）直接指定顏色的種類，另外還有 ***vbCrLf*** **表示將游標移到下一行的第一個位置**。

- **自訂常數**：使用 ***Const*** 陳述式來定義附加的常數，例如可將 3.14 直接定義為 pi 常數。常數名稱最好使用相關語意的英文單字或縮寫，若是您的英文底子不是很好，可任取較短名稱，但要在該名稱後面加上註解，以利程式的閱讀與日後程式的修改。

變　數

在程式語言中的變數（variable）是指一種內容不固定的資料選項，這種資料選項的值可以隨著程式的執行而改變。當程式設計師在撰寫程式碼，為了資料處理的需要，通常程式設計師會在程式中定義一些變數，並利用這些變數來進行各種運算處理。

程式中使用到的變數，應該先宣告正確的資料型態，例如：當變數如果為整數，通常宣告成整數（***Integer***）或長整數（***Long***）資料型態；若有小數點的數值，則可以宣告成單精準度（***Single***）或倍精準度（***Double***）資料的型態。字串變數最好在宣告時，同時定義佔用記憶體的長度，否則會浪費記憶體資源。變數宣告的語法如下：（若同時有許多變數具有相同資料型態，可用**逗號**連接使用）

語法	***Dim*　變數名稱 1，變數名稱 2，變數名稱 3　[*As*　資料型態]**
範例 1	***Dim*　no　*As*　*Integer*　（宣告　no 變數　當成　整數）
範例 2	***Dim*　s1　*As*　*String**12　（宣告　s1 變數　當成　字串）

佔用 12 Bytes 記憶空間（固定字串變數）

語法的中括號[ ]表示可有可無，若未設定陣列資料型態，則將預設的資料型態自動視為可變資料型態（***Variant***）。在程序中使用 ***Dim*** 陳述式時，通常是將 ***Dim*** 陳述式放在程序最開始的地方，而 Visual Basic 的**變數名稱**的命名規則，必須遵照以下規則：

- 變數名稱的長度必須在 255 個字元以內。
- 變數名稱不可與 Visual Basic 的保留字重複。
- 變數名稱必須以英文字母當開頭，後面則可以加上英文字母、數字或底線。
- 變數名稱中間不可以含有句點（“.”）或者是特殊字元（!、@、#、$、%、&等等）。
- 變數名稱在同一個有效範圍（生存期）之內，不可以重複命名。

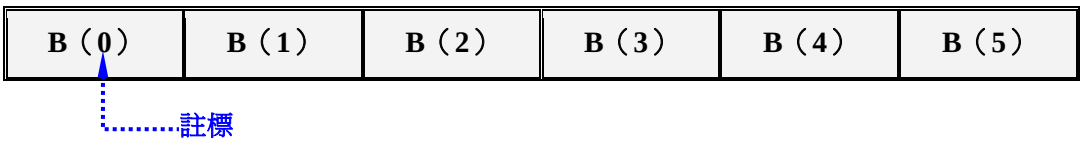
當變數在初始化時，數值變數的初始值為 0，可變字串初始值為零長度字串（""），固定長度字串則都填上 0。當您宣告變數的資料型態，應可發現每次輸入『*As*』，並按下空白鍵之後會自動列出所有資料類型，以方便輸入，每當選擇完畢之後，直接利用『空白鍵』即可完成輸入。

C-6 Visual Basic 陣列介紹

陣列是由一群資料型態相同，而且具有順序排列的陣列元素（Element）所組成，每個陣列元素，可以用來儲存一項資料。使用陣列來撰寫程式時，可以使用相同名稱去引用一連串的資料項目，再以『註標值』來識別這些資料項目，所以更能有效率的管理與使用這些資料項目。

陣列可分為一維陣列、二維陣列…，最多可到 60 維度，底下僅介紹最常使用的一維陣列，其餘均可同理類推。一維陣列宣告的語法如下：

語法	***Dim*** **變數名稱 1，變數名稱 2，變數名稱 3** **[*As* 資料型態]**
範例 1	***Dim*** **B（5）** ***As*** ***String***
範例 2	***Dim*** **B（5）** （忽略資料型態的指定，則自動視為 ***Variant*** 型態）



陣列名稱的命名規則，與變數的命名規則相同。N 代表陣列的註標，因為預設的註標值是從 0 開始算起，所以陣列元素的實際個數為 **N+1**，例如：陣列 B(5)的註標由 **0～5**，所以共有 6 個陣列元素。語法的中括號[ ]表示可有可無，若未設定陣列資料型態，則將預設的資料型態自動視為可變資料型態（***Variant***）。

請特別注意，陣列與變數的宣告十分類似，除了多增加註標大小之外，最大的差異是變數可以忽略宣告，也能正常使用，但在使用陣列之前，若尚未事先宣告，則無法正常使用，並會出現如右圖的執行錯誤，因為陣列會被誤判為副程式或是函數。

C-7　Visual Basic 運算子介紹

Visual Basic 依資料間的運算方式種類，可將運算子區分為算術、比較、連結與邏輯等四類。分別敘述如下：

運　算　子	說　明
算術運算子	用來執行數學計算的運算子
比較運算子	用來執行比較的運算子
連結運算子	用來合併字串的運算子
邏輯運算子	用來執行邏輯運算的運算子

算術運算子：用於執行數值間的運算，運算規則大致與數學上的規定相同，例如：先乘除後加減，由左至右進行運算等。底下列出 Visual Basic 語言中各種算術運算子的表示方式與功能使用說明以及優先順序：

算術運算子	功　能　說　明	運　算　式	結　果	優 先 順 序
^	**計算某數的次方值**	**2 ^ 3**	**8**	**1**
-	表示負數	-6	-6	2
*	兩數相乘	2 * 3	6	3
/	兩數相除，取商值	8 / 6	1.33	3
\	兩數四捨五入取整數後再相除，並只取商數的整數部分	8 \ 6	1	4
MOD	兩數四捨五入取整數後再相除，並取餘數部分	8 ***MOD*** 6	2	5
+	兩數相加	1 + 2	3	6
-	兩數相減	2 - 1	1	6

比較運算子：用來比較運算式兩邊的關係，若比較的結果成立，則傳回 ***True***（真）；若比較結果不成立，則傳回 ***False***（假）。一般比較運算子會配合判斷性結構語法的指令或是條件式迴圈指令。

比較運算子	功能說明	運算實例	結果
=	等於	A = B	F
>	大於	A > B	F
<	小於	A < B	T
> =	大於或等於	A > = B	F
< =	小於或等於	A < = B	T
< > 或 > <	不等於	A < > B	T

連結運算子：用來做資料與資料間的串接運算，可以運用在字串與字串的連接，字串的連接是將兩個或更多的字串連接成一個新的字串內容。

- "+"運算子：僅運用在**字串與字串**間的連結運算。
- **"&"運算子：若要連接不同類型的資料，則需使用"&"運算子。當您並無法確定是加法或是字串的連結，最好使用&運算子來做連結，如此可避免混淆，並且增加程式碼的可讀性。請特別注意，&運算子的前後務必間隔一個以上的空白字元。**

邏輯運算子：用來將運算式兩邊的內容做邏輯運算，包括 ***AND***（而且）、***OR***（或）、***XOR***（互斥或）、***NOT***（相反）等等。

A	B	A *AND* B	A *OR* B	A *XOR* B	*NOT* A
F	F	**F**	**F**	**F**	**T**
F	T	**F**	**T**	**T**	**T**
T	F	**F**	**T**	**T**	**F**
T	T	**T**	**T**	**F**	**F**

C-8　Visual Basic 選擇性判斷指令介紹

If 條件選擇的格式很多，您可以使用單行形式語法來簡化程式碼，但是，區塊形式語法則提供結構化與彈性的功能，而且也較容易閱讀、維護以及除錯，現在分別說明如下：

- 『**單行式 *If* - *Then***』敘述：為條件判斷處理的基本形式，當 ***If* - *Then*** 敘述的條件成立，只需執行『**一行程式**』敘述時，通常我們會利用這種 ***If*** 敘述來作條件判斷的處理，來簡化程式碼。

語法　***If***　條件式　***Then***　程式敘述

說明

1. 如果「條件式」成立，則執行 ***Then*** 後面的「程式敘述」，否則繼續往下執行。
2. 「程式敘述」須與 ***If* - *Then*** 敘述的「條件式」撰寫於同一行。

範例

If A > 10 ***Then*** A=A+1（如果 A>10 然後 A 自動加 1）

附註：在單行形式中，也可以放上多行的陳述式，只要在這些陳述式間加上冒號即可，如底下這行陳述式所示：（**註：程式碼不分大小寫**）

例如 ***If*** A > 10 ***Then*** A = A + 1：B = B + A：C = C + B

『區塊式 *If - Then*』敘述：在上述單行式 *If - Then* 敘述中，若 *If - Then* 敘述的條件成立，必須執行多個程式敘述時，通常我們會利用這種 *If* 敘述來做為條件判斷的處理。

語法

If 條件式 *Then*
　　程式敘述區塊（一行以上的程式敘述）
End If

說明

1. 如果「條件式」成立，則執行 *Then* 後的「程式敘述區塊」，否則直接跳至 *End If*，並繼續往下執行。
2. 區塊式 *If - Then* 敘述必須與 *End If* 成對使用，由於程式敘述區塊不知有幾行，所以需利用 *End If* 來表示判斷敘述的結束。

範例

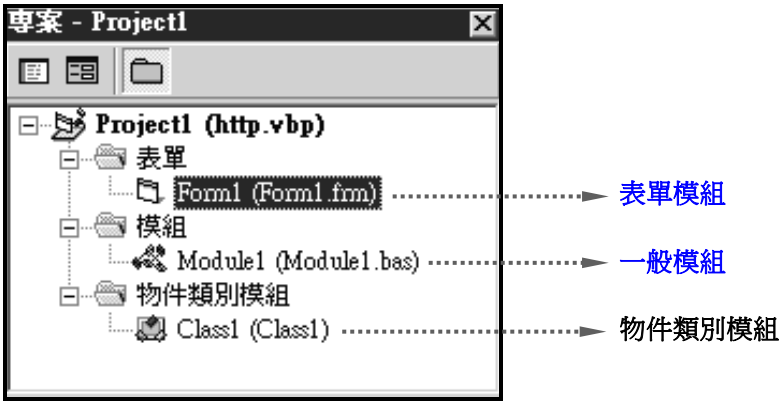
```
If  A > 10  Then
    A = A + 1
    B = B + A
    C = C + B
End If
```

程式敘述區塊

註 當您在輸入 *End If* 的程式碼時，不用區分大小寫，而且 *End* 與 *If* 之間不需留空白，因為當游標移到下一行時，會自動安排大小寫，且中間自動隔一個空白。另外使用縮排的撰寫方式，可以很清楚了解程式敘述區塊所對應的 *If - End If* 關係，強烈建議各位養成習慣。

C-9　Visual Basic 模組（Module）介紹

Visual Basic 的程式碼儲存在模組之中，它的模組共有三種類型：**表單**、**一般**和**物件類別模組**。每個一般模組、物件類別模組和表單模組，在電腦上都是以一個檔案的方式存在，您可以利用『**專案視窗**』檢查共有哪些模組被使用，如下圖所示：



表單模組（Form）

表單模組（副檔名為**.frm**）是大多數 Visual Basic 應用程式的基本元件。表單模組可以包含處理事件的程序、一般程序以及變數、常數、型態和外部程序的表單層次宣告。如果您在文字編輯器中檢視表單模組，則您還會看到表單及其控制項的說明，包括它們的屬性設定值。

一般模組（Module）

一般模組（副檔名為.bas）包含應用程式中可被其他模組共用的程序和宣告。它們可以包含變數、常數、型態、外部程序和全域程序的全域宣告（在整個應用程式範圍內都有效）。若需要新增一個『**一般模組**』，請利用『**專案 → 新增模組**』或在**專案視窗任意空白處按滑鼠右鍵『新增→模組』**的方式增加，**電腦硬體裝修乙級試題 USB 呼叫就是利用此方式撰寫**。

物件類別模組（Class）

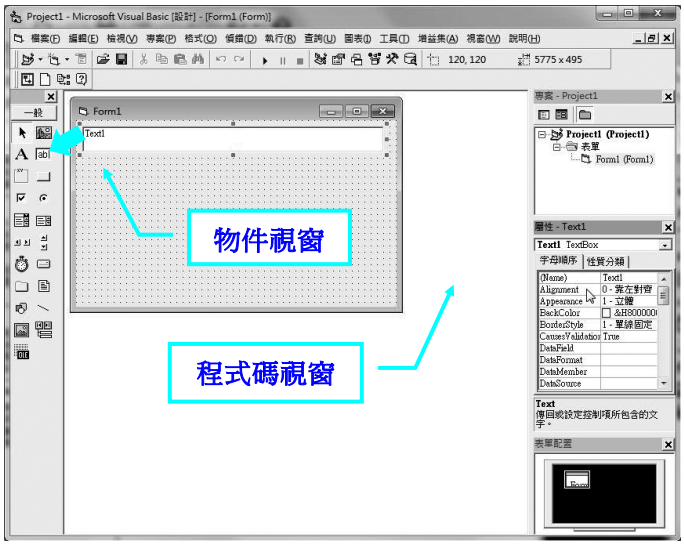
在 Visual Basic 中，物件類別模組（副檔名為**.cls**）可說是物件導向程式設計基礎。您可在物件類別模組中撰寫程式碼以建立新的物件。這些新物件可以包含自訂的屬性和方法。若需要新增一個『**物件類別模組**』，請利用『**專案 → 新增物件類別模組**』的方式增加。

C-10 Visual Basic 6.0 完整操作流程介紹

若您之前從未接觸過 Visual Basic 6.0，不妨先觀看 DVD 光碟所錄製的完整操作流程影片，底下僅針對較重要步驟進行補充說明，操作流程說明如下：

Step …1

剛開啟的專案視窗，建議**先將程式碼視窗最大化，以方便程式碼的撰寫**，但是物件視窗則千萬不可最大化。



顯示目前時間，可利用 TextBox（文字方塊）控制物件 ab|，先點選該工具之後，再到物件視窗拖曳適當的大小，因為下方還要放置 16 個圓圈圈，建議拉寬一點。

接著將文字方塊 TextBox 屬性變更如下：

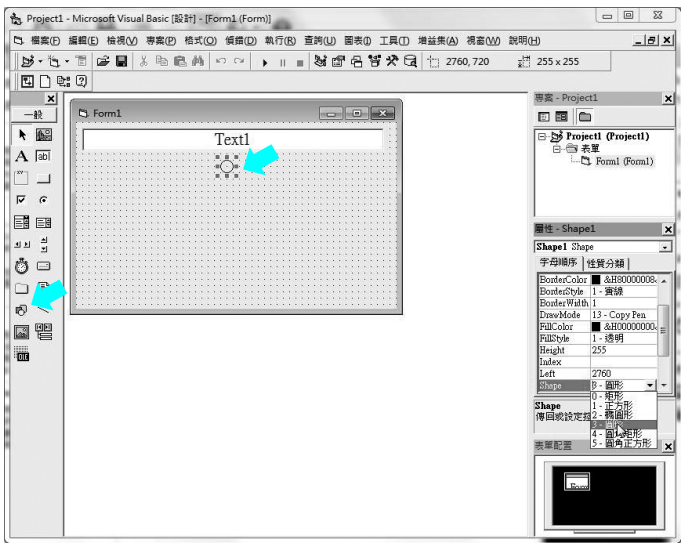
a.Alignment = 2 置中對齊。

b.利用 font 設定適當字型，以便上下置中對齊。

c.將 text 屬性內容全清除。

Step …2**

2014 年新增 16 個和 LED 同步的物件，其實這是 shape 的物件。



將 shape 屬性變更如下：

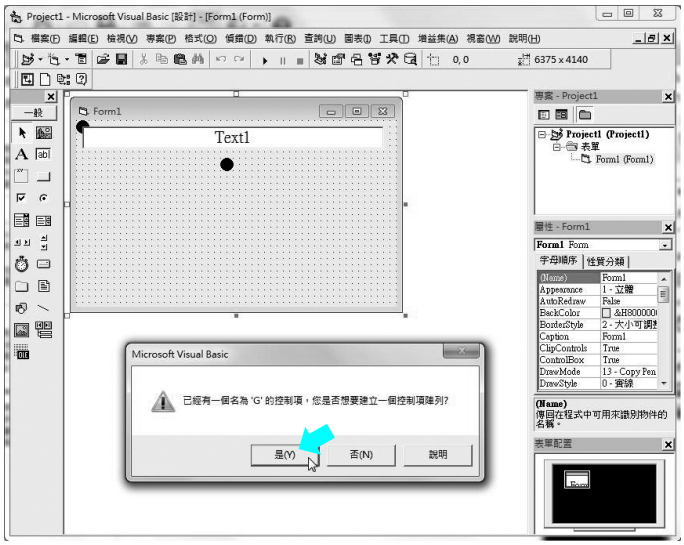
a.Shape 形狀=**3** 圓形。

b.Fillstyle=**0** 填滿。

c.將 name 改為 **G，方便撰寫**。

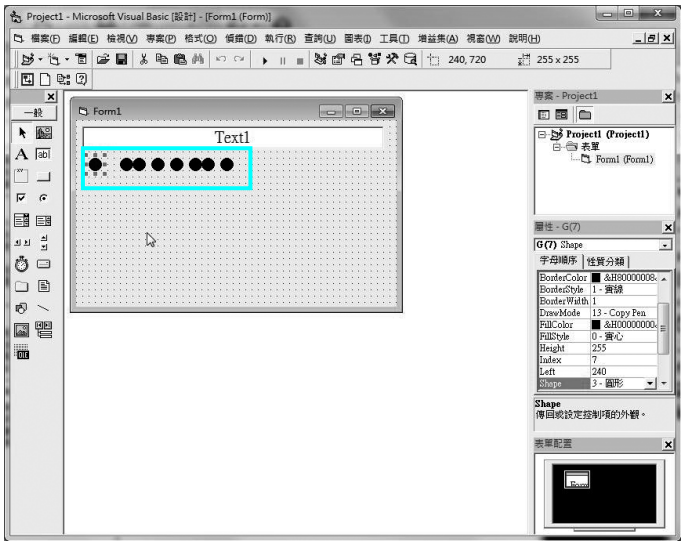
Step ...3***

將 G 物件所有屬性設定完畢之後，再點選該物件，按『Ctrl + C』做**複製**動作，緊接著按『Ctrl + V』做**貼上**動作，之後會出現下面畫面，詢問是否建立**控制項陣列**，一定要回答『**是**』，表示會複製相同名稱控制項物件陣列，此時物件名稱已自動改為 G(0)與 G(1)。



Step ...4**

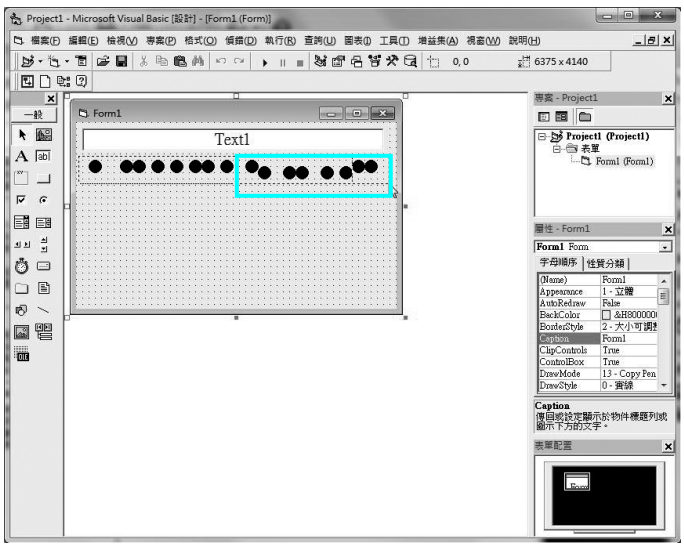
一直重複步驟 3，依序建立 G(0)~G(7)，排列位置差不多即可，但要特別留意 G(7)是在最**左**側，因為它對應到介面卡**LED16(MSB)**，而 G(0) 是在最**右**側，因為它對應到介面卡**LED9(LSB)**。



Step ...5***

一直重複步驟 2~4，依序建立 R(0)~R(7)，排列位置差不多即可，但要特別留意 R(7)是在最**左**側，因為它對應到介面卡**LED8(MSB)**，而 R(0) 是在最**右**側，因為它對應到介面卡**LED1(LSB)**。

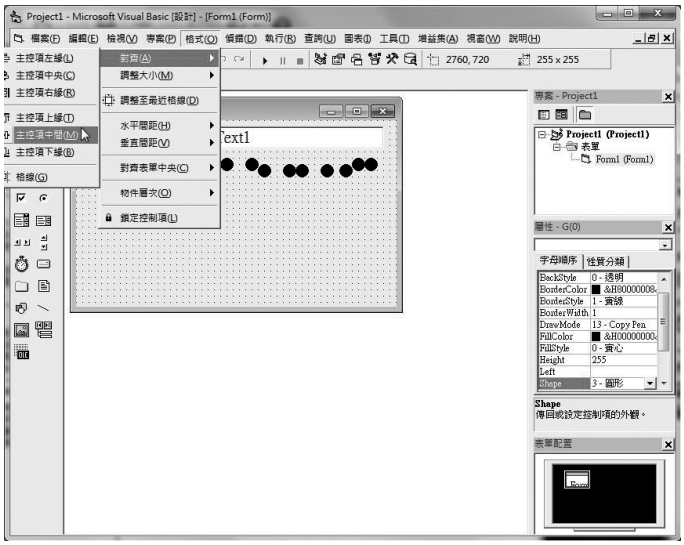
註：**所有 Index 順序不要弄錯**。



Step …6***

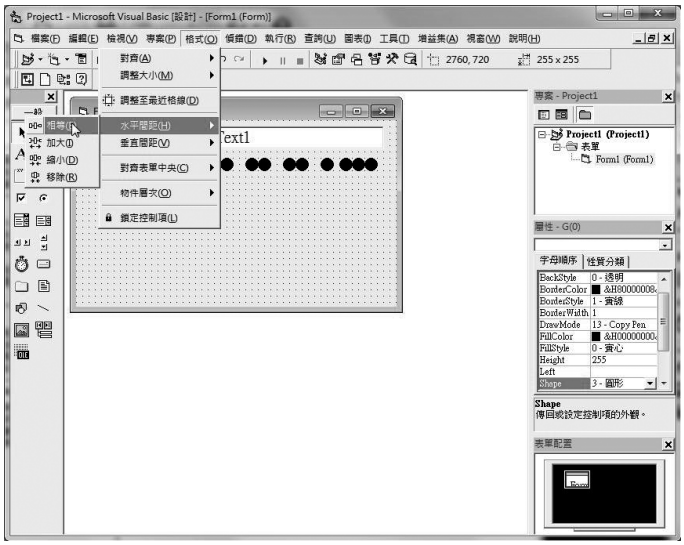
以拖曳框選方式同時選取 G(7)~G(0)與 R(7)~R(0)等 16 個物件。

利用『**格式→對齊→主控項中間**』，即可將 16 個●物件**上下對整**。



Step …7**

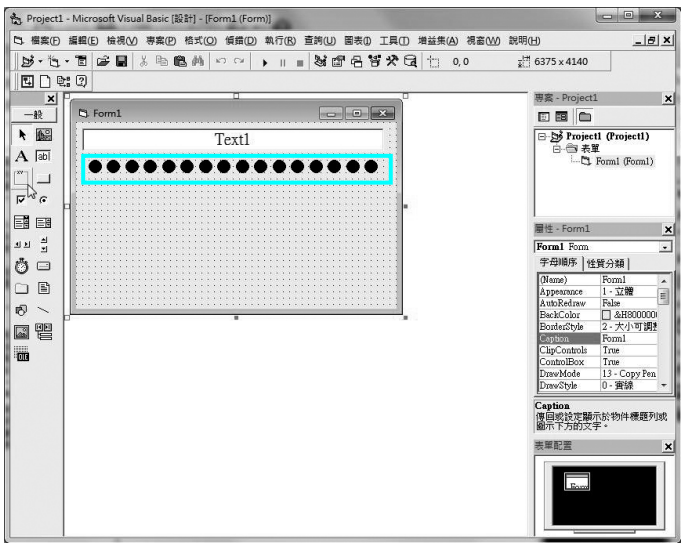
再利用『**格式→水平間距→相等**』，即可將 16 個●物件**左右間隔相等**。



Step …8**

原本 G(7)~G(0)與 R(7)~R(0)等 16 個物件排列很凌亂，一下子就變整齊。

試題當中要求未插上 USB 介面卡要以中空顯示，並非在此設定，而是利用程式碼控制即可。



Step …9***

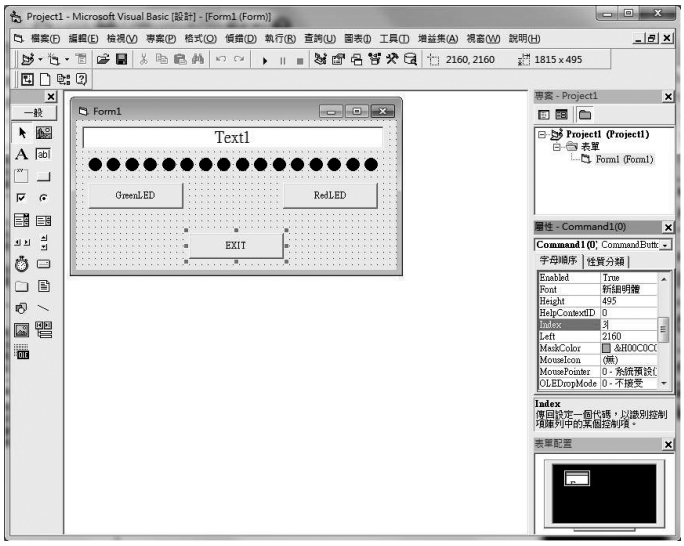
接著點選左側一般工具箱的 CommandButton（命令鈕）控制物件，再到物件視窗拖曳適當的大小，然後將屬性變更：

a.Caption 輸入 **EXIT**。

b.利用 font 設定適當字型。

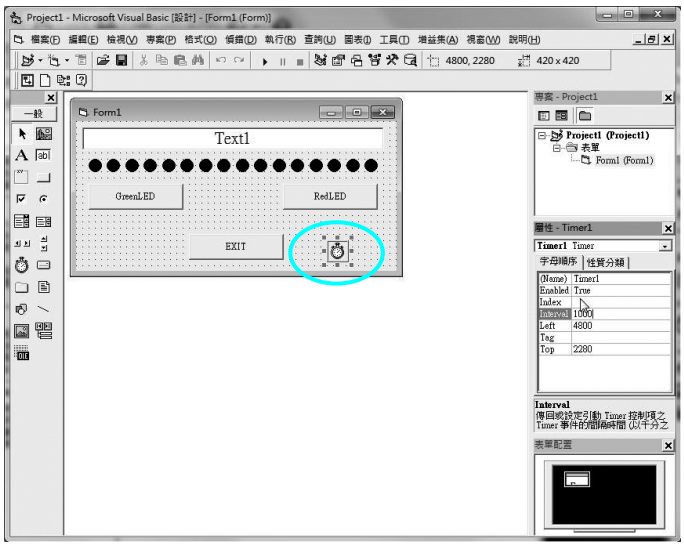
然後複製→貼上，建立兩個**控制項陣列**，分別修改 Caption 為 GreenLED 與 RedLED。

最後要將 Command1(0) 屬性 Index=3，即為 **Command1(3)**。



Step …10***

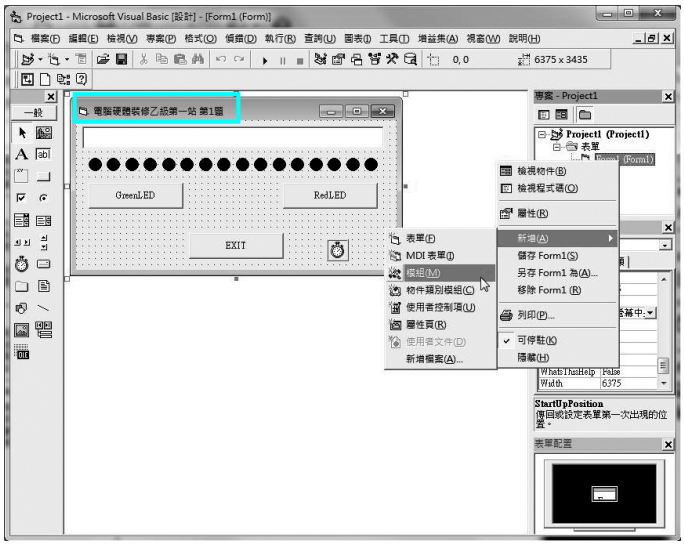
最後加入 Timer（定時器）控制物件，Timer 控制項用於背景處理中，無論放置何處，在表單都不會呈現出來，所以不用特地去調整該控制項的位置與大小。同時將 ***Interval* 屬性設定為 1000**（以**千分之一秒**為單位），因為每隔一秒 LED 燈號要變化一次，然後在物件上方點擊兩下，即可撰寫程式碼。



Step …11***

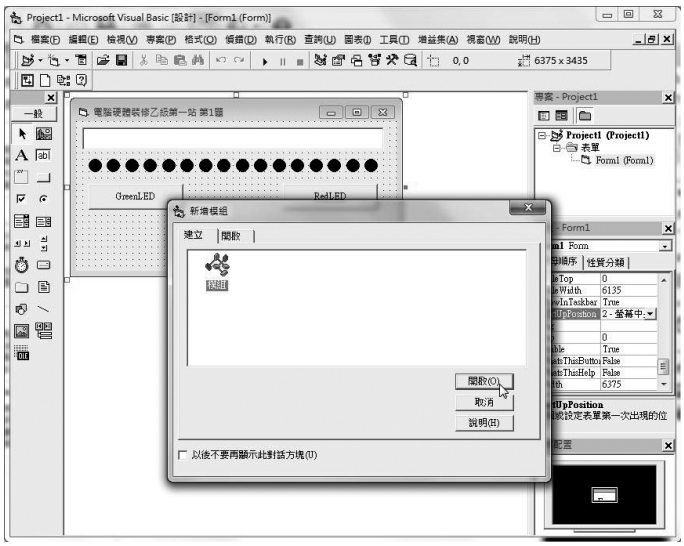
依據試題修改表單 form1 的標題 caption 屬性，內容最好完全相同，包括數字樣式。

然後在專案視窗任意空白處按滑鼠右鍵『**新增→模組**』的方式增加。



Step ...12

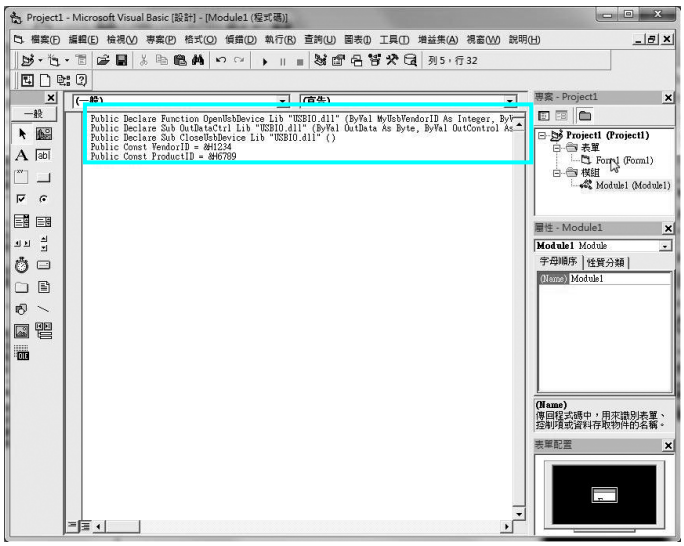
直接按『開啟』。



Step ...13***

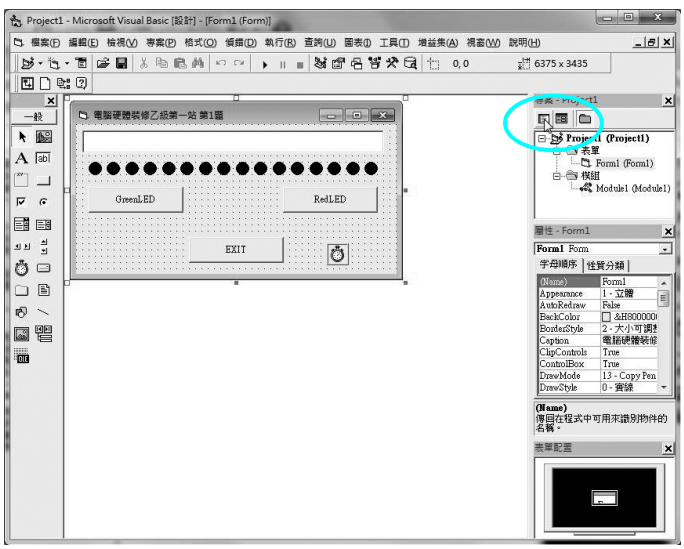
建議先撰寫模組的程式碼，並注意 USBIO.DLL 裡面函數名稱大小寫需一致。

完成後切換到 Form1 表單視窗。



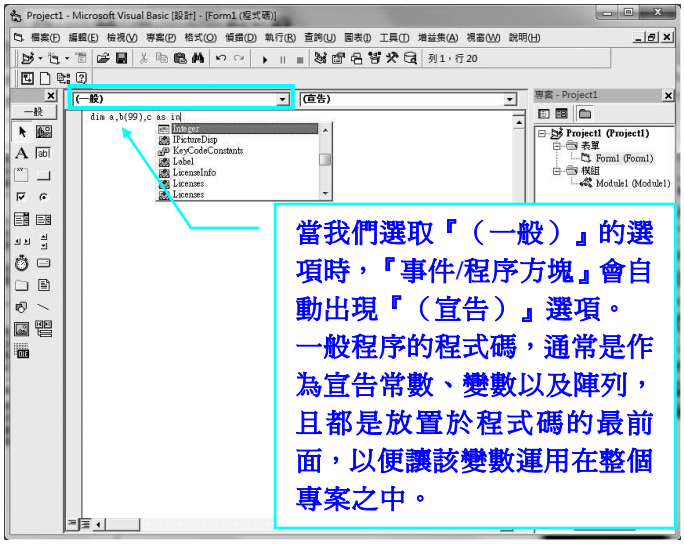
Step ...14**

利用  切換到程式碼視窗，以便開始輸入主程式碼。



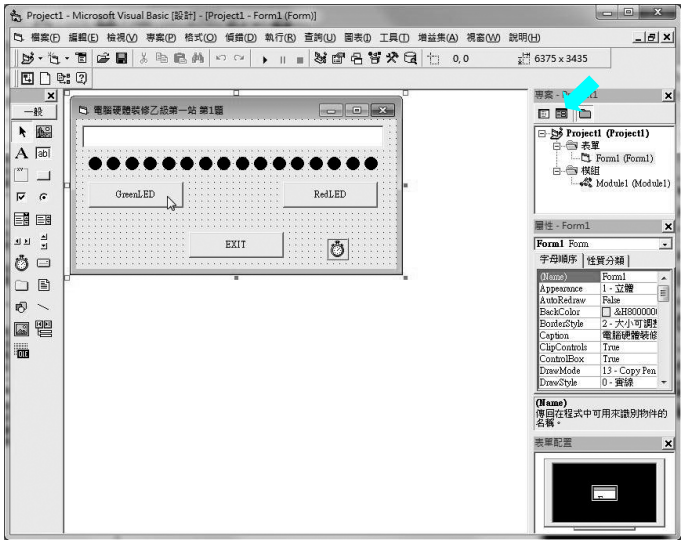
Step ...15***

點選 (一般) ，以便先輸入**程式宣告**的部分，**程式碼不分大小寫，所以全部先輸入小寫『dim a,b(99),c as』文字，當在輸入一個空白字元後，會自動出現許多選項，此時就像是電子字典，輸入第一個英文字母，則自動跳到該字母的位置，例如輸入『in』，然後按空白鍵即自動產生 integer，當按 Enter ，該行程式會自動變成大小寫**。



Step ...16*

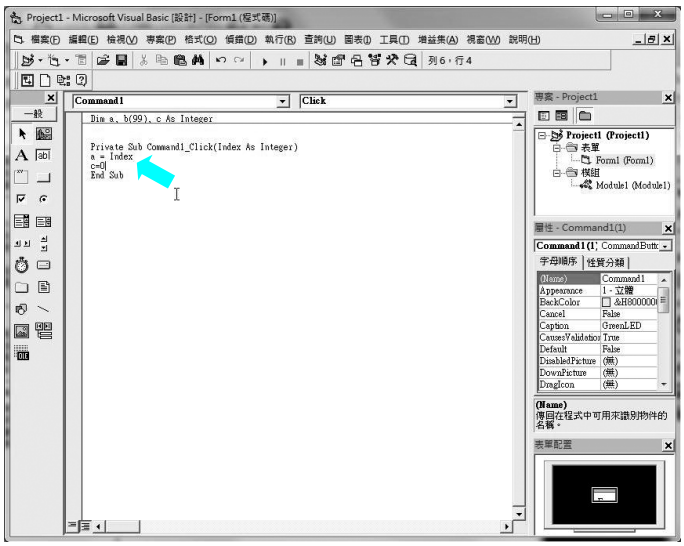
程式宣告僅一行，輸入完畢後，按 再切換回『**物件檢視**』視窗。



Step ...17***

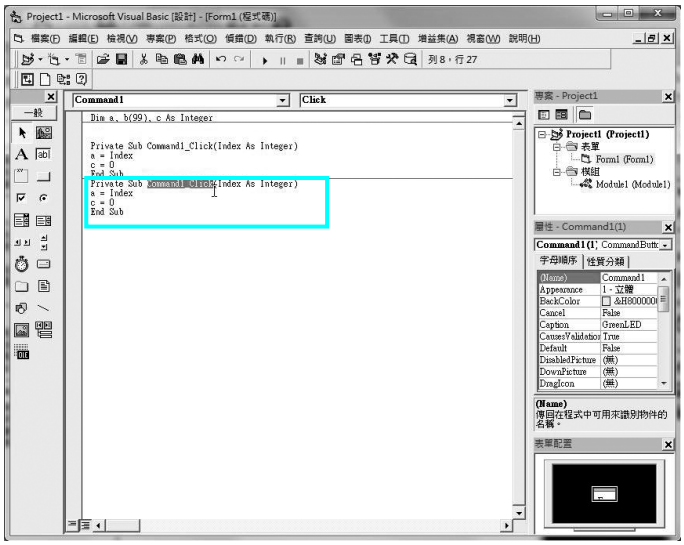
Command1(1)~ Command1(3)三個按鈕為控制項陣列，所以在其中任意三個按鈕的上方點擊兩下，即可針對按鈕物件撰寫程式碼，副程式的頭尾均不用輸入，僅輸入兩行程式，亦可合併一行『**a=index:c=0**』。

※Index 是按 Enter 才自動變成字首大寫的。



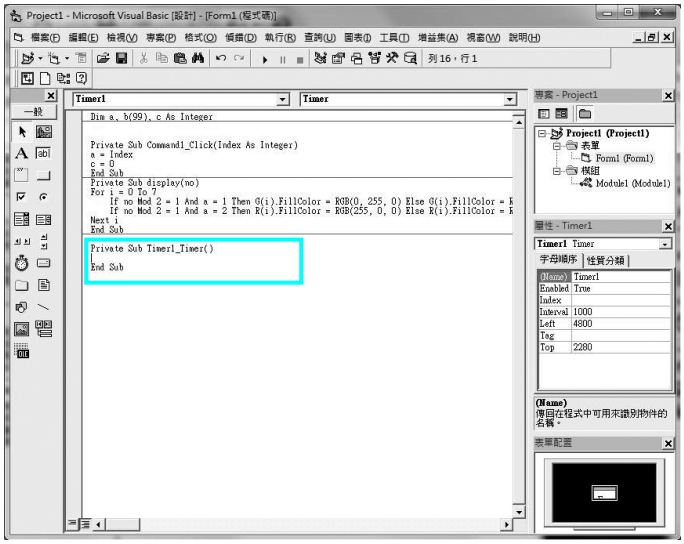
Step ...18****

因為試題要在表單新增燈號同步顯示，故自行建立 display 副程式，該名稱可自訂，透過 no 變數傳遞資料即可，**此副程式頭尾均需自行輸入**，可複製步驟 17 建立的 Command1_click 副程式全部，再修改 display 副程式名稱與內容較快速。



Step ...19*

按再切換回『**物件檢視**』視窗，在 Timer 控制物件點擊兩下，自行輸入 Timer1_Timer( ) 裡面相關的程式碼。



Step ...20***

程式碼均輸入完畢後，若馬上執行一定出錯，請看下一頁補充說明畫面。

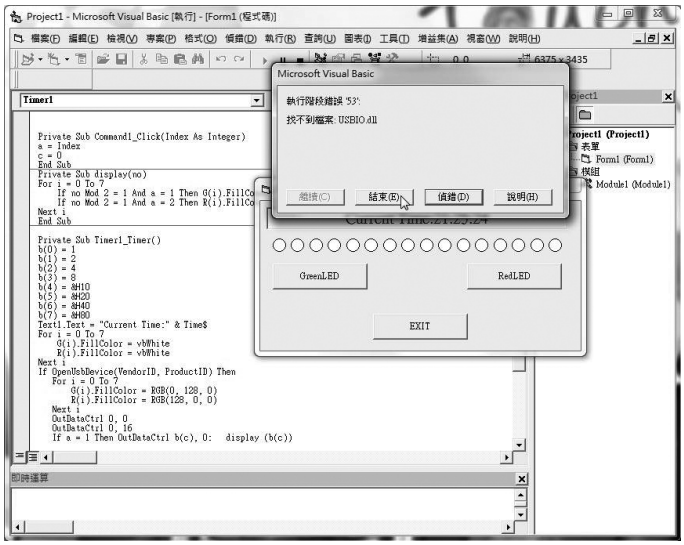
利用『**檔案→儲存專案**』先儲存整個專案所有相關檔案，記得要儲存在有 USBIO.DLL 相同的資料夾內。

Step ...21***

關閉整個程式碼，然後再開啟 Project1.vbp 專案檔，這樣程式才會正確找到 USBIO.DLL 檔。

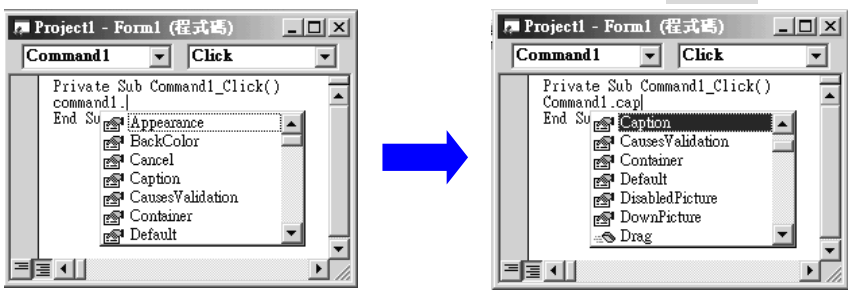
Step ...補充****

若沒做『儲存→關閉程式→開啟程式』就執行，就會出現右側錯誤畫面，原因是主程式找不到 USBIO.DLL 檔案。



當每輸入一行程式，並按下 Enter 鍵確認後，則該行程式所有內容（不含屬性值）會**自動變成大小寫（註：當我們在輸入程式時，大小寫皆可），表示該行程式的語法沒有問題，若還是全部小寫，表示程式輸入錯誤，這樣可以預防程式撰寫錯誤發生**！

當我們輸入小數點之後，會自動顯示所有該控制物件的屬性以及方法，**若無法正常顯示，則表示控制物件的名稱輸入不正確**。當再輸入"c"時，該選項會**自動移到 c 開頭的屬性或方法的位置**，所以當輸入 cap 之後，即可直接按『**空白鍵**』直接設定。



心得筆記