2006 計算機程式語言

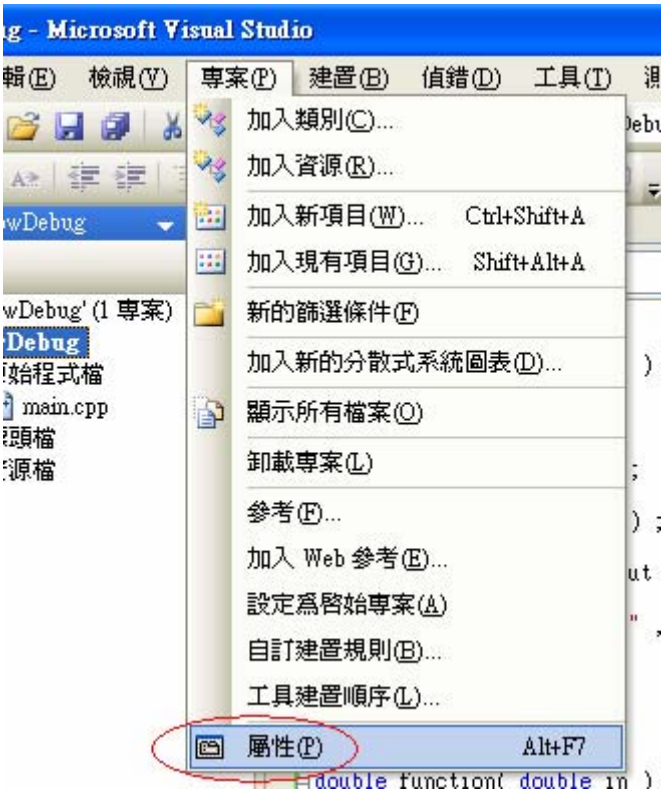
Visual Studio 2005 C++之 Debug 模式開啟與使用說明

1. **前言**：利用 Visual Studio 2005 撰寫程式時，常常需要對撰寫的內容做驗證，以確認撰寫的程式碼是否有達到預期的功效，過程中常需要檢查變數中的值是否正確，早期除錯的方法是將變數值直接在該程式碼執行到時顯示在電腦螢幕上來觀看變數的內容，但是這是非常沒有效率的方式，而 Visual Studio 2005 所提供的 IDE（**I**ntegrated **D**evelop **E**nvironment）中就提供了 Debug（除錯）模式，前幾版的 Visual Studio（VS、VS .Net、VS .Net 2003）可以直接設定中斷點並進行除錯，但很不幸的，**Visual Studio 2005 的預設值將除錯功能關閉了**，也因此就算設定好中斷點，程式執行到中斷點的位置也不會停止，本文件的目的即在教導各位如何開啟 Debug 模式以及如何使用與設定，以增進寫程式的效率。

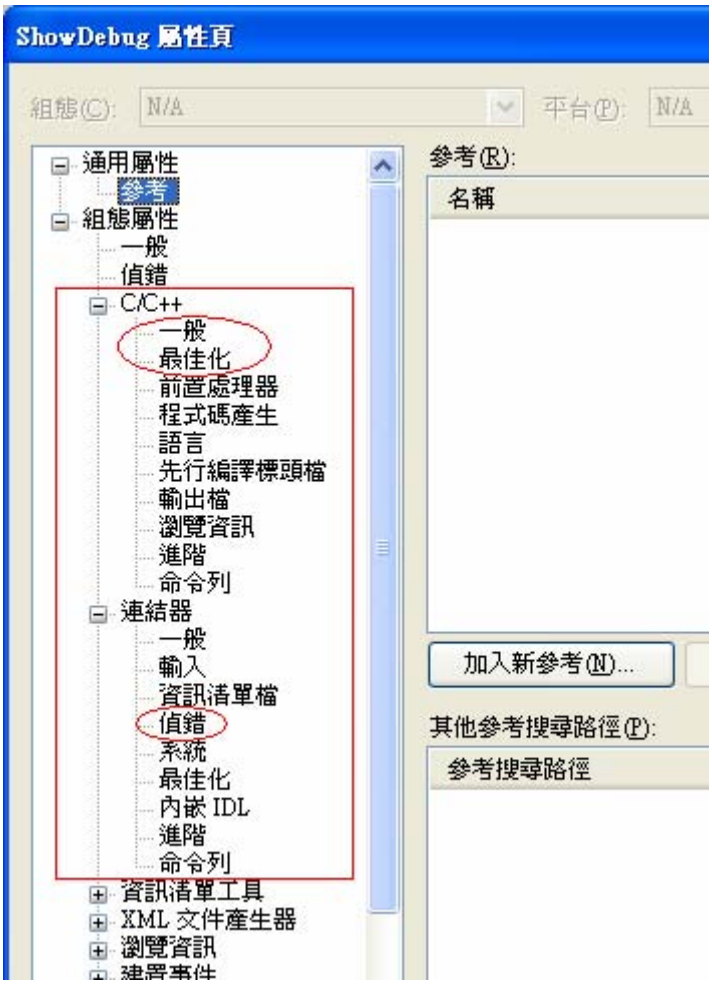
2. **開啟除錯模式**：此處將以 Step-by-step 的方式告訴各位該如何開啟除錯模式的功能。
   - **Step 1**：此段程式碼目的在利用 function 來計算圓面積，數學好的一看就知道哪裡有問題了吧，現在我們先假設我們並不曉得問題點在哪，來開啟 Debug 模式檢查。

```
#include <stdio.h>

double function( double in ) ;

int main( void )
{
    double input , output ;

    scanf( "%lf" , &input ) ;

    output = function( input ) ;

    printf( "Output: %lf\n" , output ) ;

    return 0 ;
}

//利用函式求圓面積
double function( double in )
{
    const double pi = 3.14 ;
    double return_value ;

    return_value = pi + in * in ;

    return return_value ;
}
```

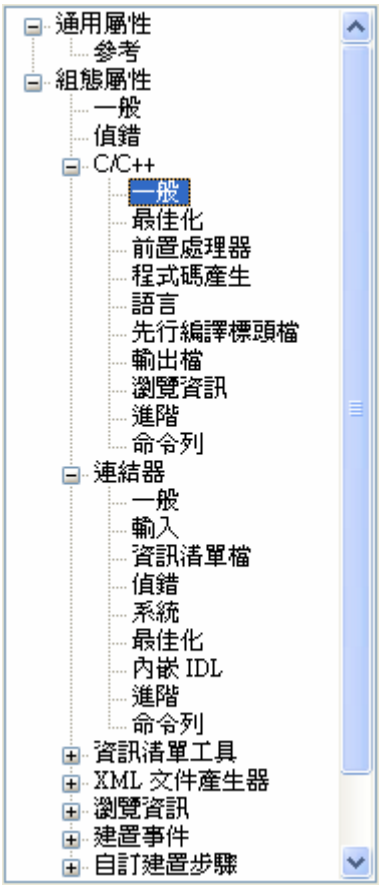
- **Step 2**：再來對要除錯的專案開啟屬性視窗，開啟的方式有兩種，一種是在方案總管視窗中對要開啟 Debug 的專案上按滑鼠右鍵點選，另一種是在選單列上選擇【專案】->【屬性】，要注意的是此種方式必須要在方案總管中先選定要開啟 Debug 的專案。

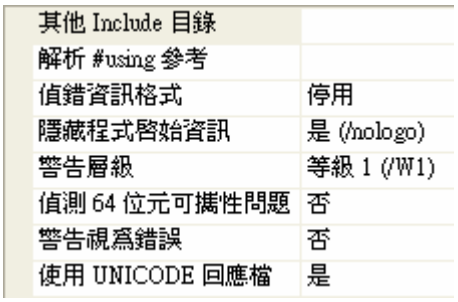


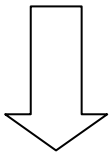
- **Step 3**：開啟屬性頁後，必須要修改三項設定，他們分別在【C/C++】下的【一般】與【最佳化】和【連結器】下的【偵錯】。

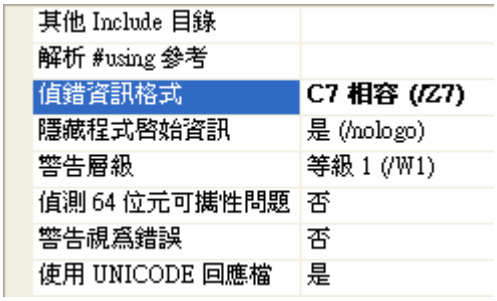


- Step 4：首先先選擇【C/C++】下的【一般】，然後將右邊【偵錯資訊格式】由『停用』設定為『C7 相容 (/Z7)』，Visual Studio 2005 會將曾經修改過的設定以粗體字表示。

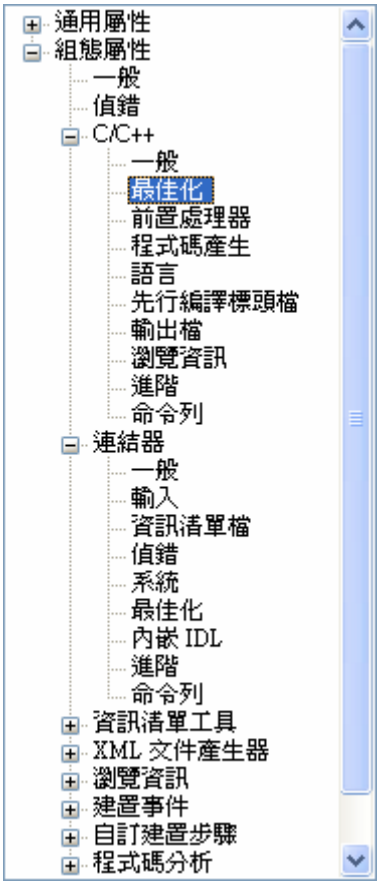




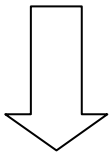




- **Step 5**：接著一樣是【C/C++】下的【最佳化】，然後將右邊【最佳化】由『最快速執行效能 (/O2)』設定為『停用 (/Od)』。

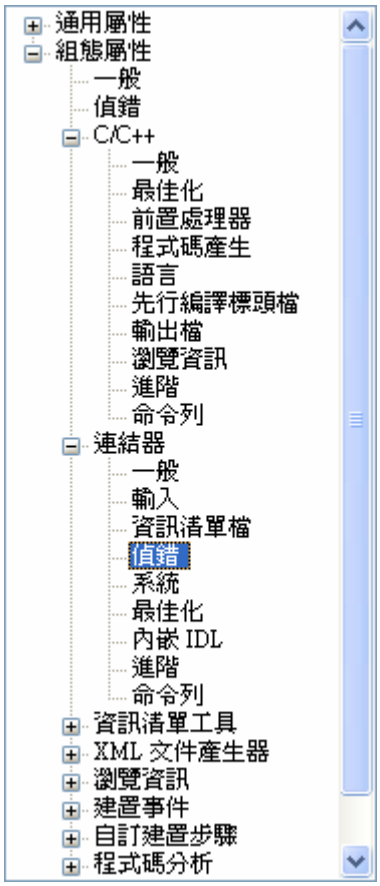


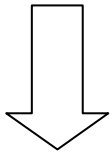
最佳化	最快速執行效能 (/O2)
內嵌函式展開	預設
啓用內建函式	否
偏好大小或速度	都不
省略框架指標	否
啓用 Fiber-Safe 最佳化	否
整個程式最佳化	否



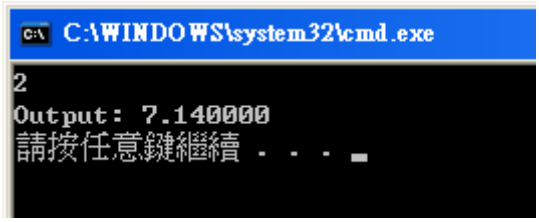
最佳化	**停用 (/Od)**
內嵌函式展開	預設
啓用內建函式	否
偏好大小或速度	都不
省略框架指標	否
啓用 Fiber-Safe 最佳化	否
整個程式最佳化	否

- **Step 6**：最後選擇到【連結器】下的【偵錯】，然後將右邊【產生偵錯資訊】由『否』設定為『是 (/DEBUG)』，到此已將必要的設定調整過，接下來就是按下右下角的 確定 了。

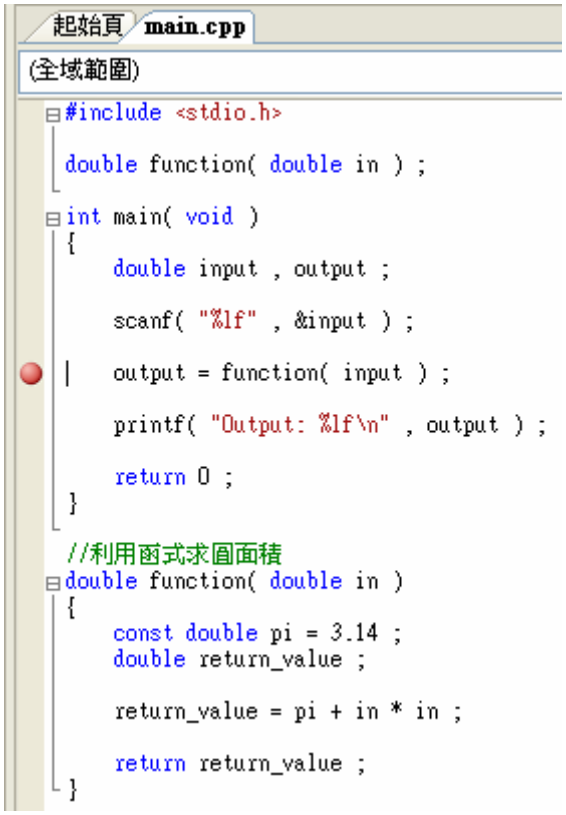


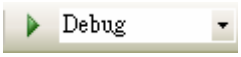


3. **使用 Debug 功能**：開啟完 Debug 的設定後就可以開始使用，下圖為本文開頭的程式所執行出來的結果，可以發現當輸入半徑為 2 時輸出結果為 7.140000，但對於求圓面積來說是錯誤的結果，現在將以 Step-by-step 的方式說明如何使用 Debug 功能。



- **Step 1**：一開始不曉得是哪一段程式碼錯誤時，就必須推測可能的錯誤點在哪裡，下圖的左側紅點部份即為中斷點（break point），所代表的意義為『程式執行到此停住』，而設定的方式也很簡單，只要在**該行程式的左側的灰色邊條上按滑鼠左鍵兩下**即可設定完成。



- **Step 2**：設定完中斷點後可以按【F5】或是工具列上的 

圖示，開始執行程式，下圖為程式執行到中斷點停下來時的畫面，中斷點上的**箭頭代表程式目前執行到的位置**。

  工具列 

為操作 Debug 的選項：

   ：繼續程式的執行。

   ：停止偵錯。

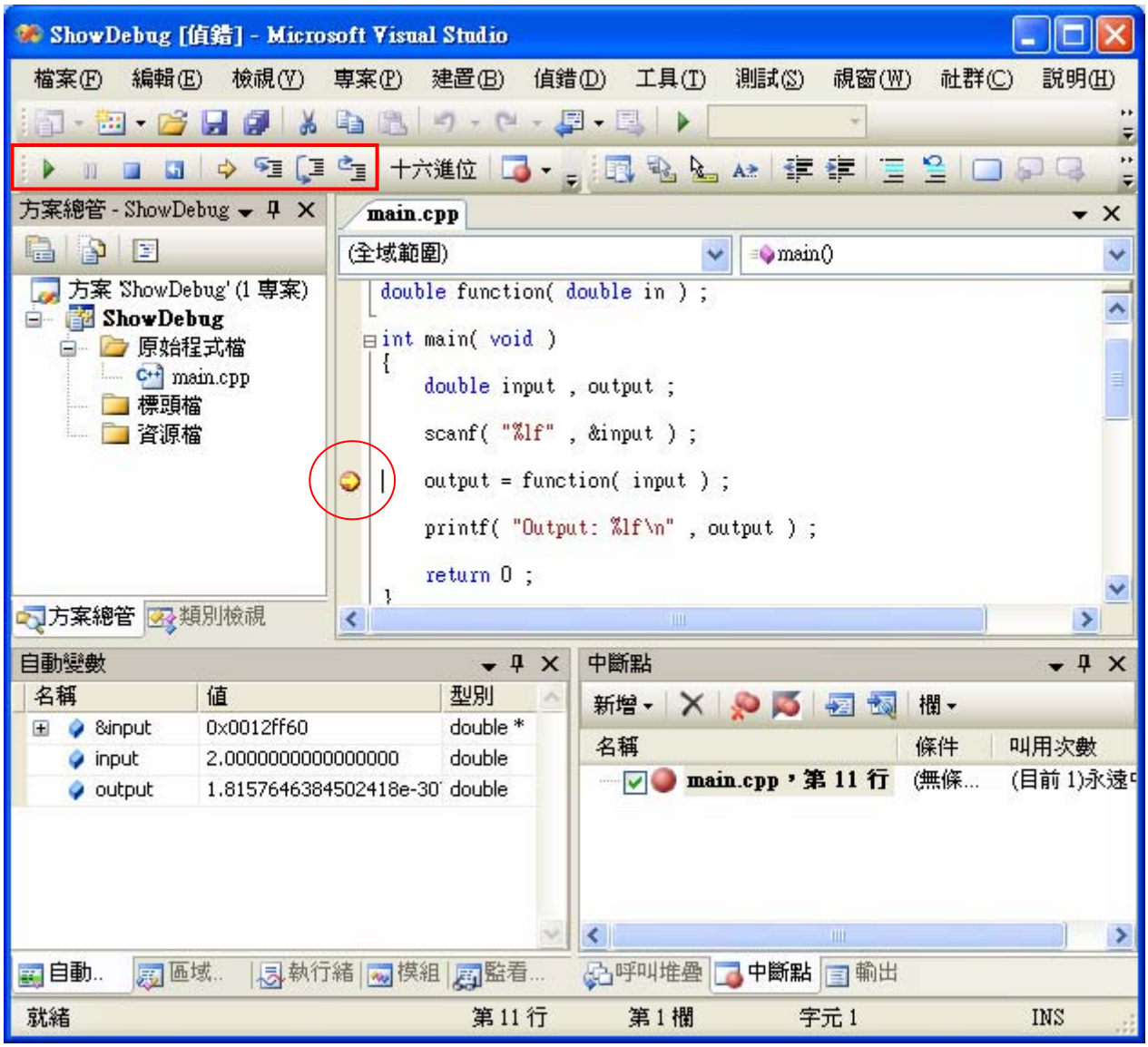
   ：重新執行程式。

   ：**逐步執行每行程式，當遇到函式時，會進入函式裡面的第一行執行。**

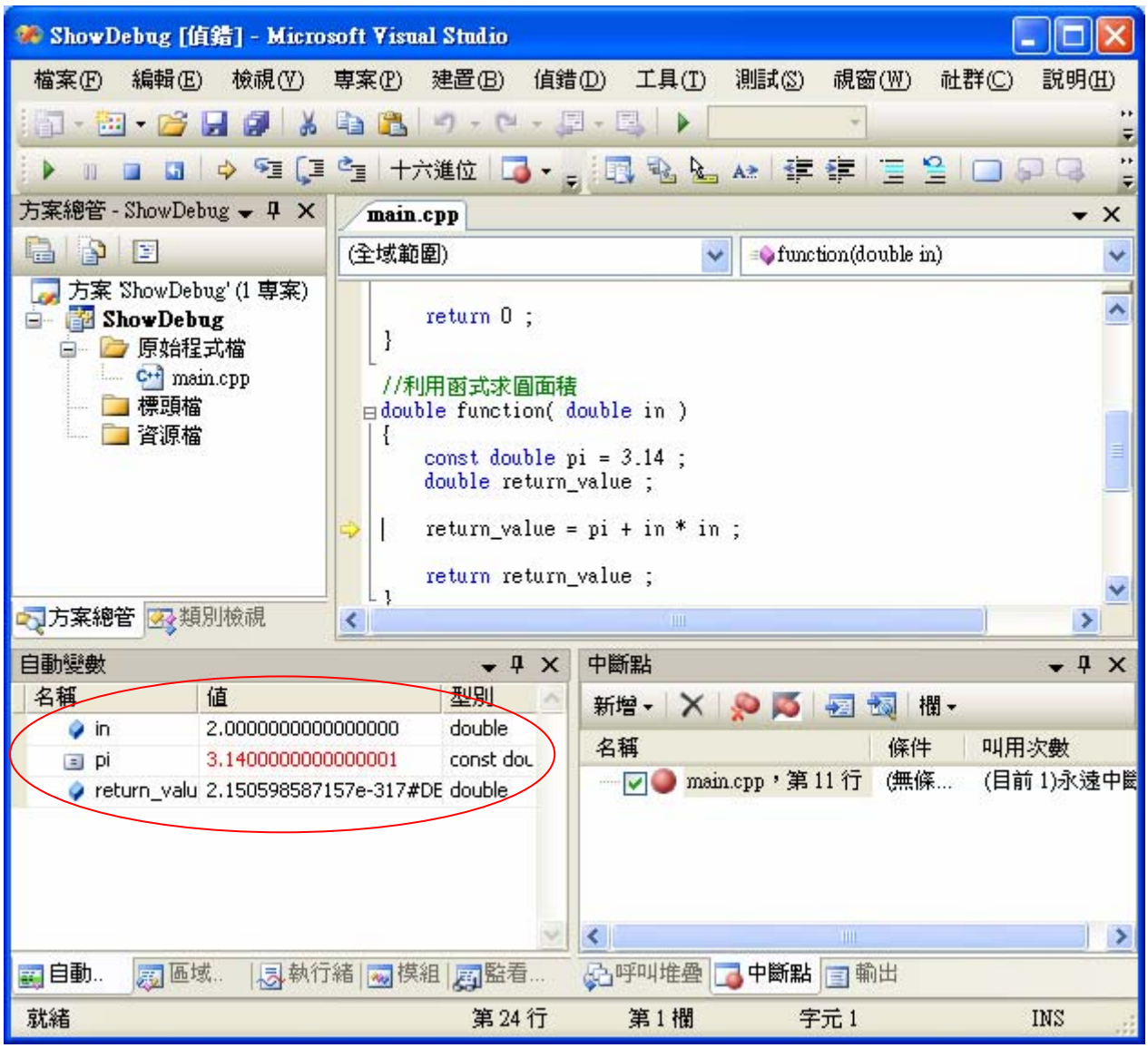
   ：**逐步執行每行程式，但遇到函式會將函式當作一行程式，直接執行完。**

   ：如果是在函式內的話，會直接將函式內剩餘的內容執行完畢後出去函式。

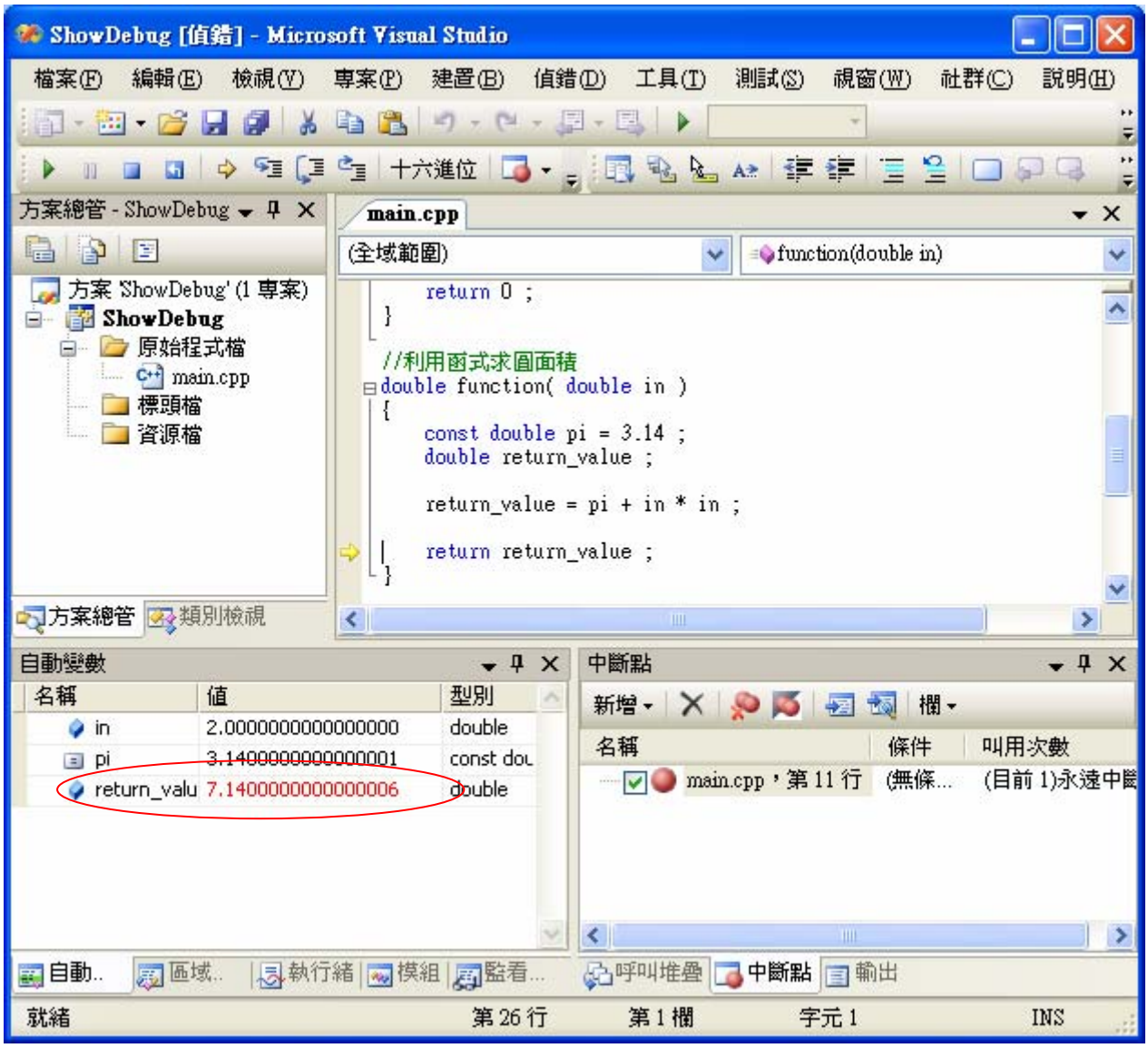
  這些為最常使用的功能。



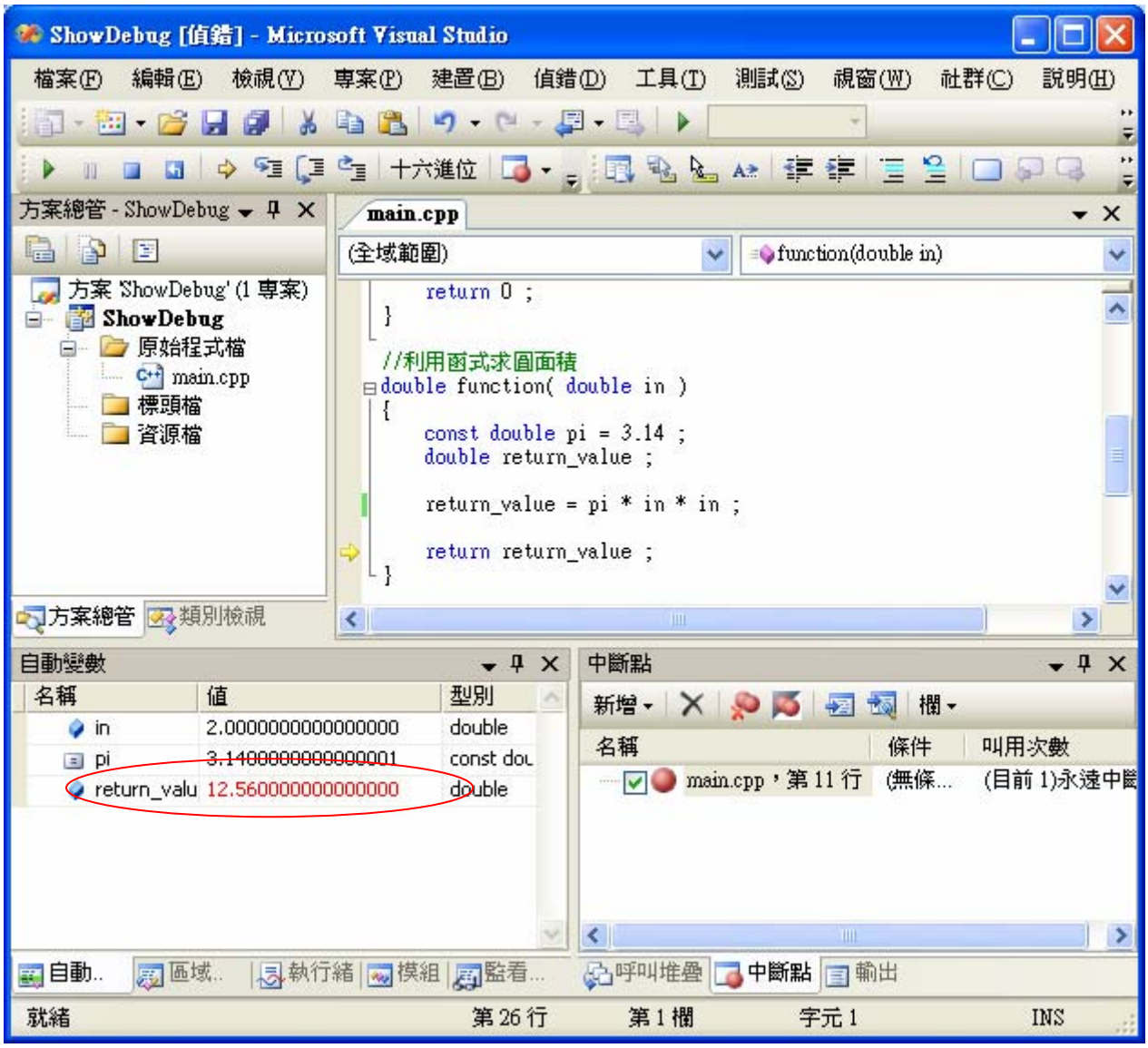
- **Step 3**：此時程式停止在函式的地方，接下來按進入函式並執行到運算式的地方，由圖中可以發現箭頭目前停在函式中運算式的位置，現在看看左下角自動變數的位置，可以看到裡面顯示出該函式的變數內的值，可以看到 *return_value* 的值很奇怪，那是因為並沒有 *return_value* 初始值，而變數 *pi* 的值呈現紅色則代表變數內的值剛被改變過。



- **Step 4**：再按一次後執行該行敘述，可以看到執行後的 *return_value* 值是錯的，到此就找到整個程式的錯誤（Bug）在哪裡。



- **Step 5**：最後按停止偵錯，然後修正計算圓面積公式後，再來看看執行結果，圖中可以看到 *return_value* 的值已經計算正確了。

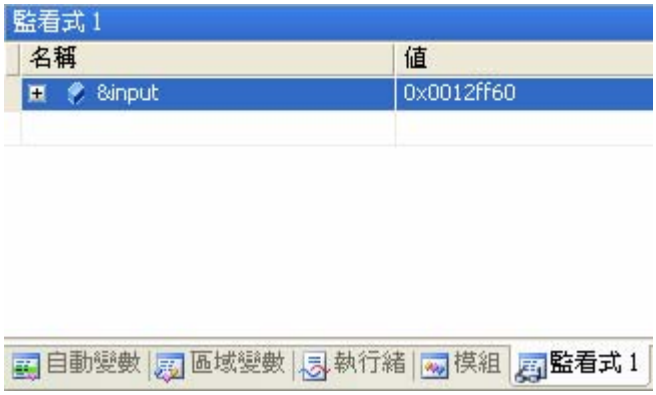


- **Step 6**：確認變數的值後，可以按  來繼續執行程式，現在程式已經有了正確的結果了。

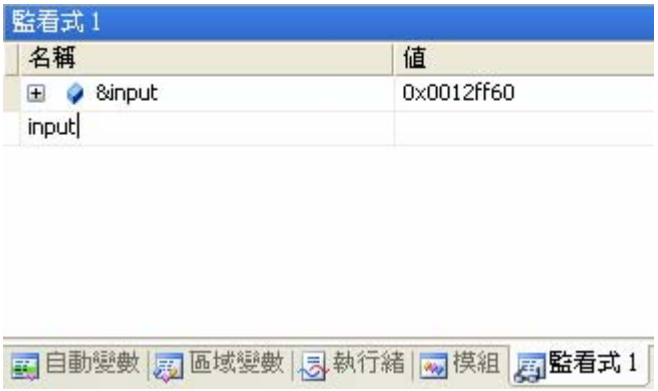
4. **其他功能說明**：前面介紹了開啟與使用基本的 Debug 功能，但是其中還有一些進階的使用方法可增進除錯的效率。

- 在自動變數的區域按右鍵選擇【加入監看式】

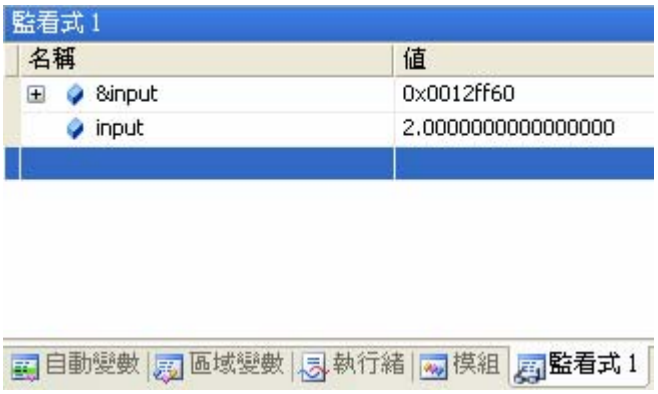
- 監看式就會出現加入的變數，由於&符號為取址運算子，所以會看到其值為 input 的所在位址。



- 或是在其下方直接輸入你要看的變數。



- 按 Enter 後就會被加入監看式中，而 input 變數內的值即為例子中的輸入值。



- 另外，在斷點處按右鍵選內容則可以針對迴圈的【叫用次數(H)】做個設定。

- 設定畫面如下圖所示，剛設定的斷點並未設定次數，所以是永遠中斷。

- 中斷次數可以設定『等於』、『倍數於』、『大於或等於』三種。

- 假如我們要設定回圈跑到第 N 圈時停下來，那就可以選擇『等於』N 次，這樣一來迴圈就會在跑第 N 次時停止，此時就可以檢視有問題的變數內的值是否跟預期的一樣。

5. Debug 的功能最主要還是在增進程式的完成度，避免掉可能的問題，但是天下並沒有十全十美的程式，今天開發一套軟體，開始發售前測試的問題都修正掉了，但是到了使用者身上時卻又發現之前未抓到的錯誤，這是常有的事，所以我們只能儘可能的抓出問題，並提高程式的完成度，這也就是為何要仰賴這些功能的主要原因。

6. 在 Visual Studio 2005 內還有許多能夠幫助除錯的功能在，希望各位同學能夠去慢慢發掘與學習。