

# C# 示波器控制應用與相關基礎語法

Adviser: Yung-Chun Lee

Advisee: Yuan-Chia Hsu

Date: 2026/07/29

# Outline

## I 課程內容

- 課程內容前置
- C#課程內容參考資源

## II C# 基礎語法

- 儀器通訊建立
- SCPI寫入與讀取
- 數據儲存

## III. 實作作業



# Outline

## I 課程內容

- 課程內容前置
- C#課程內容參考資源

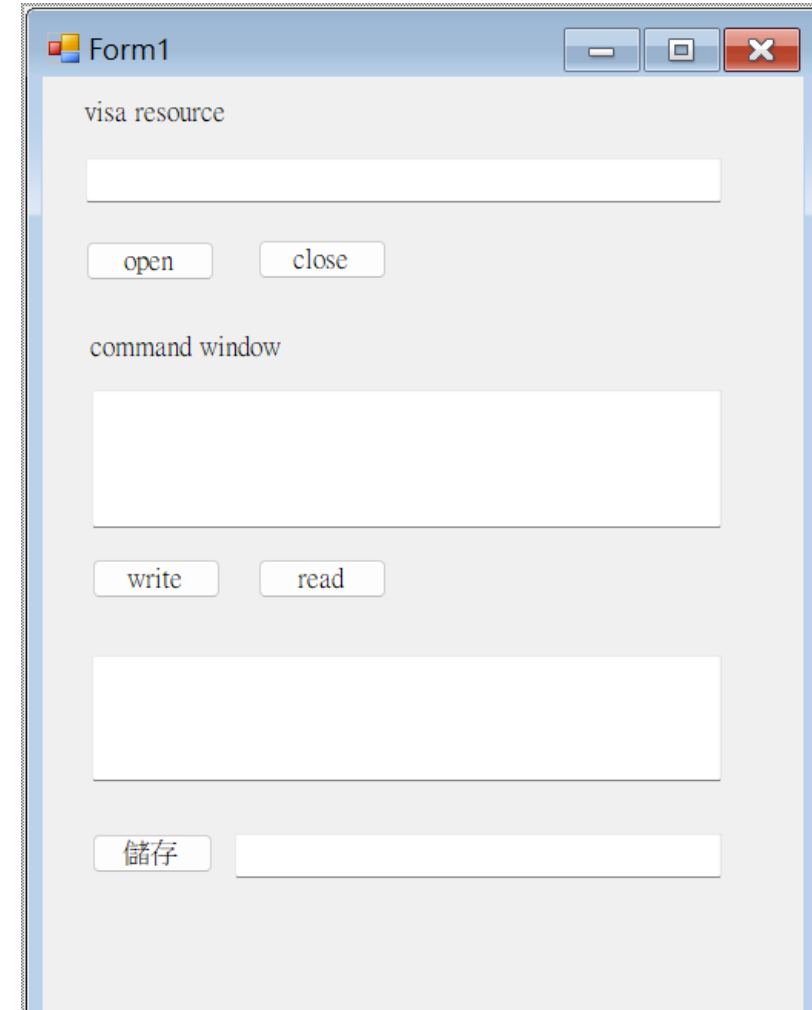
## II C# 基礎語法

- 儀器通訊建立
- SCPI寫入與讀取
- 數據儲存

## III. 實作作業

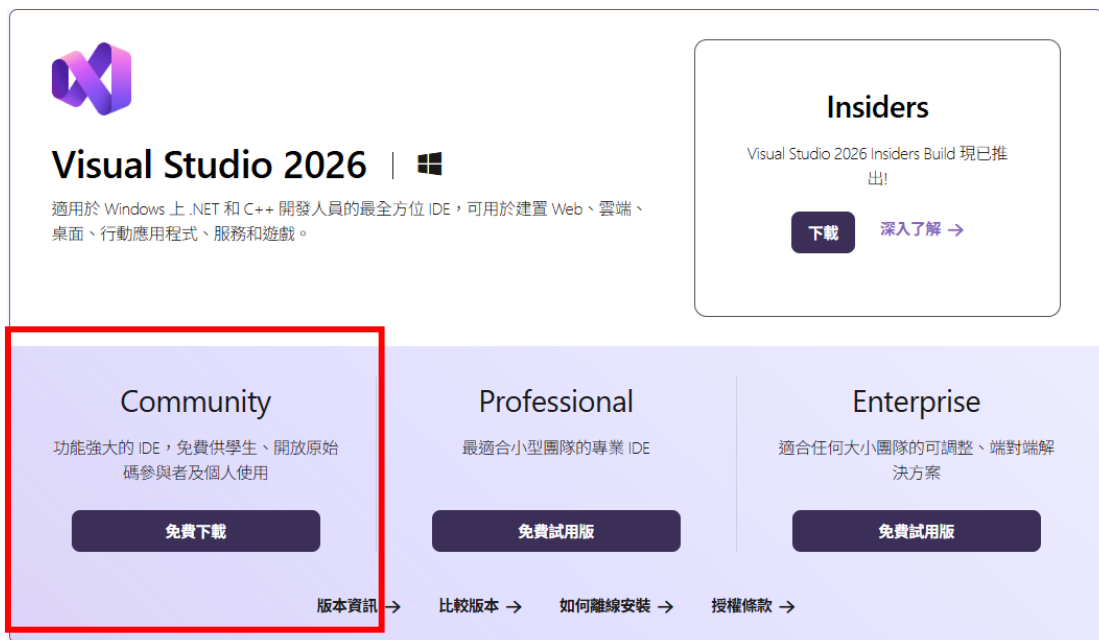
# I. 課程內容

- C# (示波器控制與應用)
- 相關基礎語法介紹



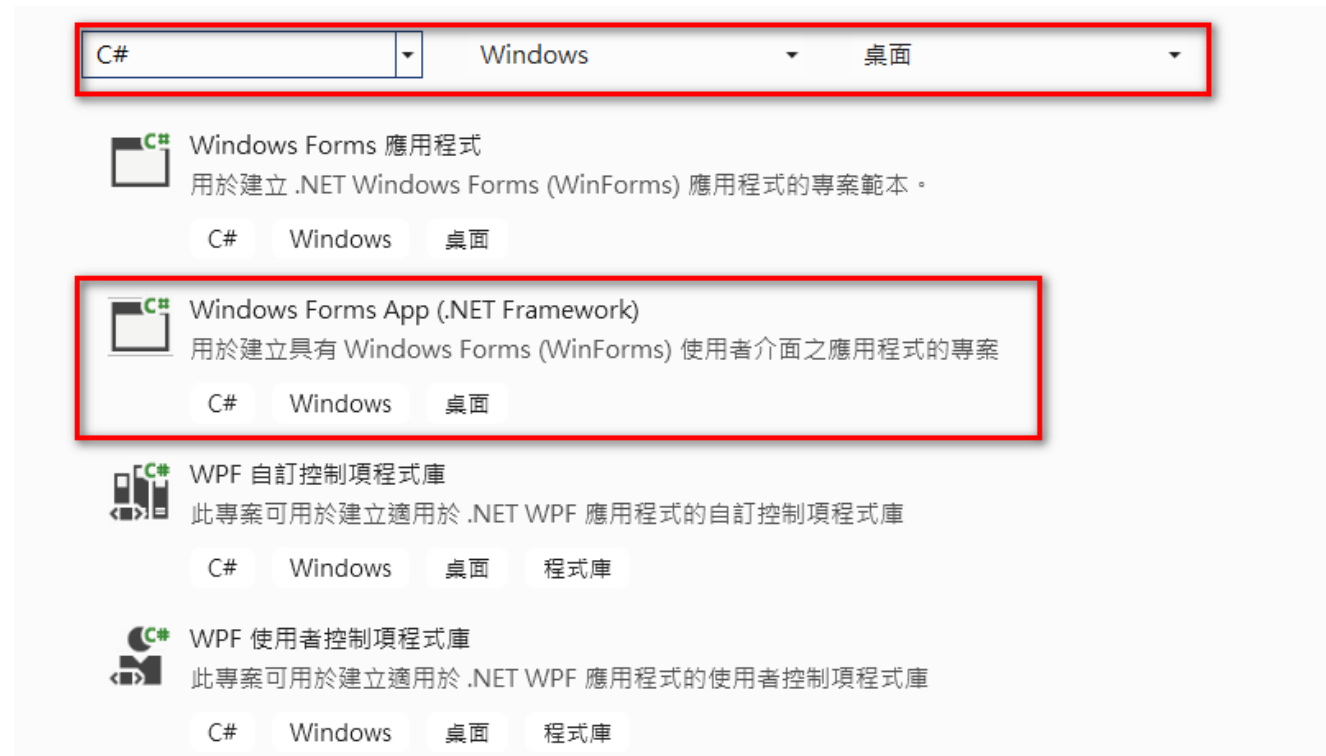
# I. 課程內容前置

- 上網下載 Visual Studio 2026 Community 版，完成後點開 Setup.exe 請選擇 .NET 桌面開發再選擇安裝



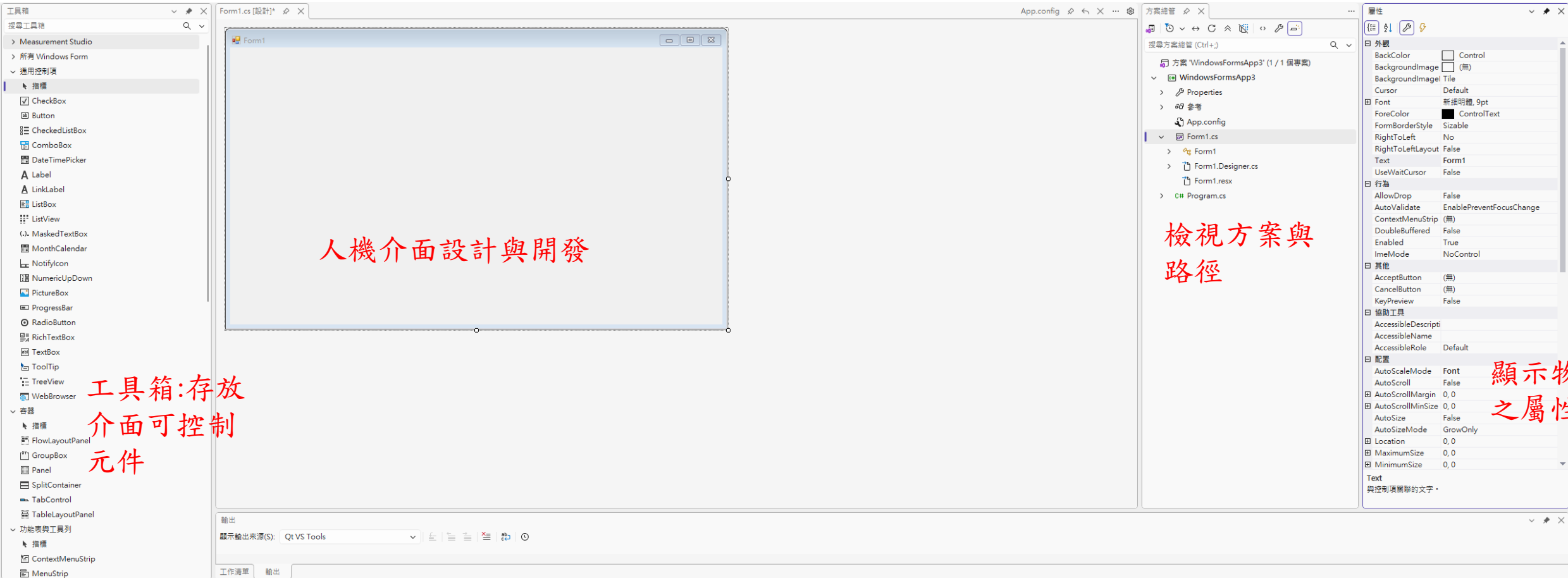
# I. C#課程內容前置

- 下拉選單選擇使用C#進行開發，平台於Windows，應用於桌面，找到Windows Forms App (.NET framework) 並選擇下一步，訂定專案名稱、位置，最後選擇建立。視窗



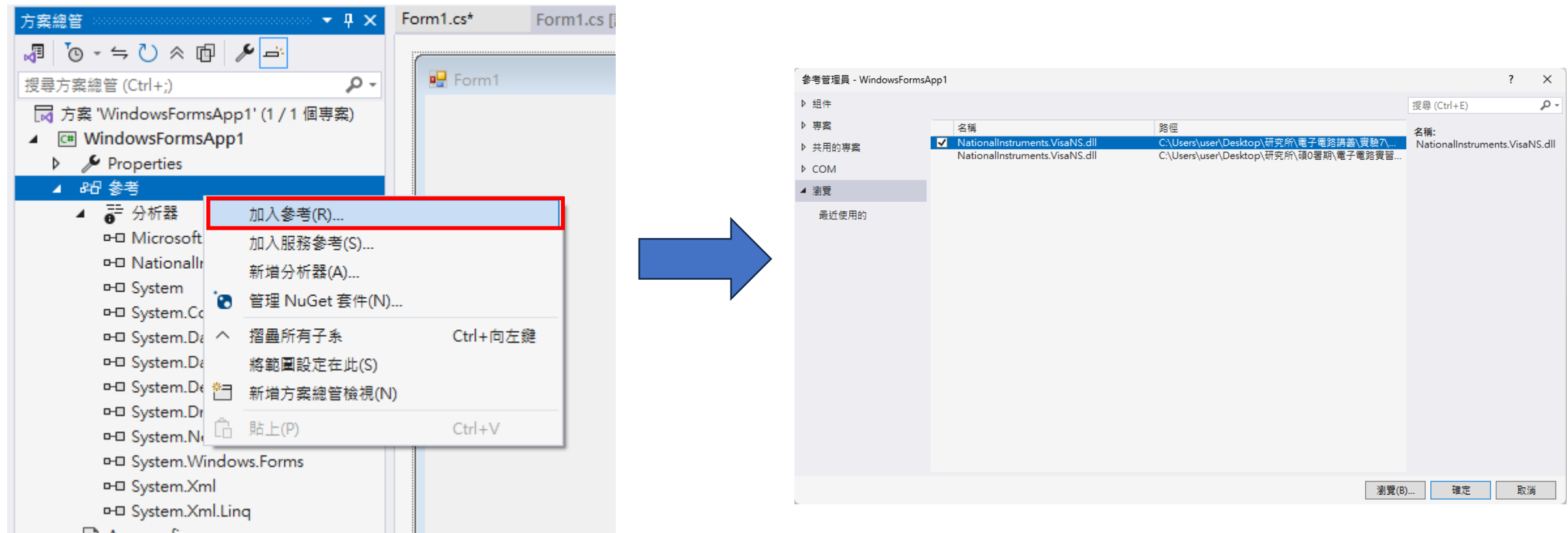
# I. C#課程內容前置

- C#專案視窗



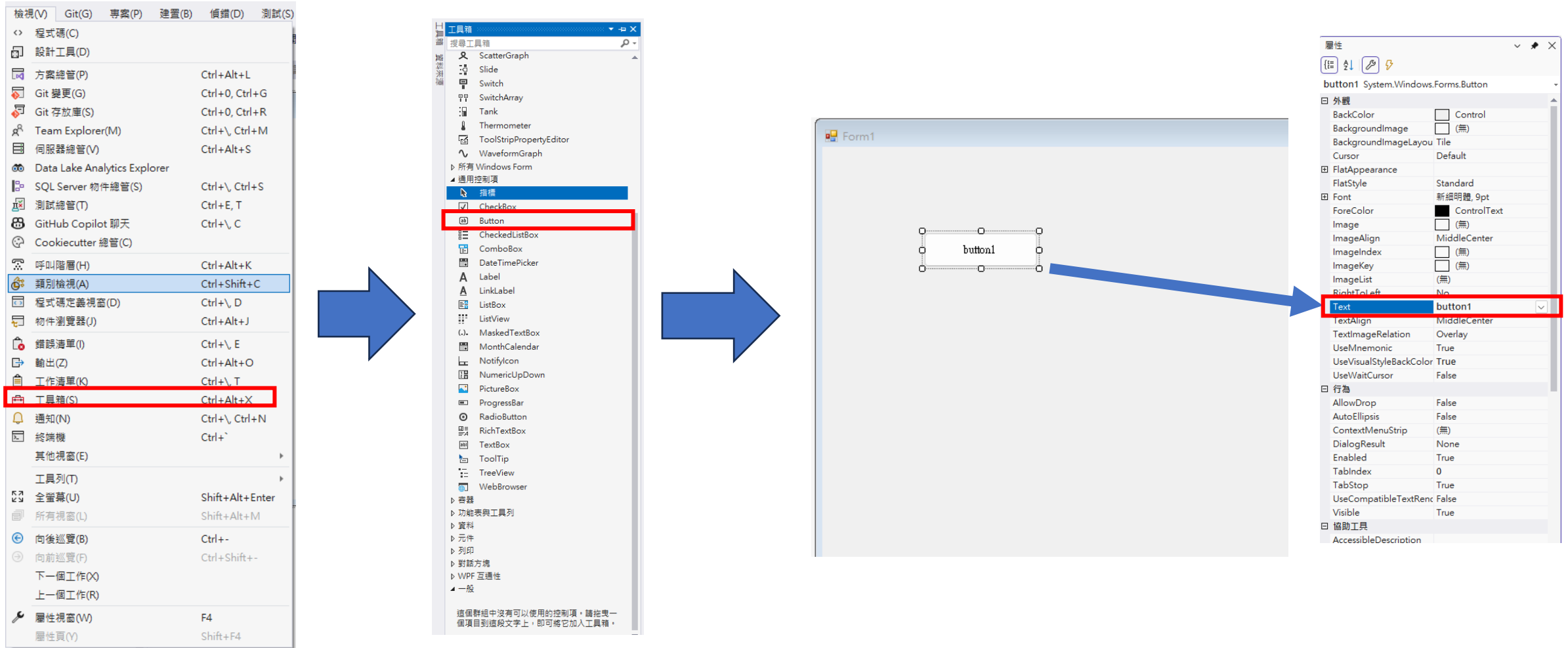
# I. C#課程內容前置

- 加入編程環境



# I. C#課程內容前置

- 建立相關物件之流程



# I. C#課程內容參考資源

- <https://www.youtube.com/watch?v=qKEFE5hfu-Q&list=PLGLZVd2c1WxdpDpjxqyeSjjYSwfeEbTkU&index=5>
- [https://www.youtube.com/watch?v=zzJIWC\\_HFU4&list=PLGLZVd2c1WxdpDpjxqyeSjjYSwfeEbTkU&index=6](https://www.youtube.com/watch?v=zzJIWC_HFU4&list=PLGLZVd2c1WxdpDpjxqyeSjjYSwfeEbTkU&index=6)
- <https://youtu.be/wCqze14g2ak?si=xfXF0Lez9aZma8OX>

# Outline

## I 課程內容

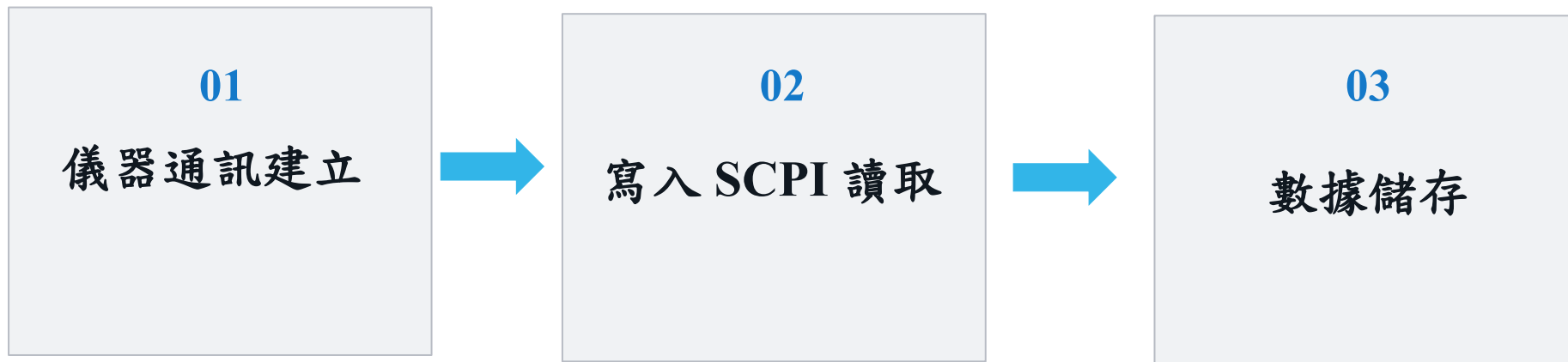
- 課程內容前置
- C#課程內容參考資源

## II C# 基礎語法

- 儀器通訊建立
- SCPI寫入與讀取
- 數據儲存

## III. 實作作業

## II. C# 基礎語法



## II. 儀器通訊建立

```
public partial class Form1 : Form
{
    private MessageBasedSession mbSession; // create Message based session

    1 個參考
    public Form1()
    {
        InitializeComponent();
    }

    1 個參考
    private void button1_Click(object sender, EventArgs e)
    {
        string strVISARsrc = textBox1.Text; // Get VISA resource ID from combobox
        try
        {
            mbSession = (MessageBasedSession)ResourceManager.GetLocalManager().Open(strVISARsrc); // 當按下 open 後執行傳輸線接上指定的儀器
        }
        catch (InvalidCastException)
        {
            MessageBox.Show("Resource selected must be a message-based session");
        }
        catch (Exception exp)
        {
            MessageBox.Show(exp.Message);
        }
    }
}
```



▲ open: button 1

VISA Resource ID

USB0::0x09C4::0x0400::DG1D151700931::INSTR

↑                    ↑                    ↑  
Vendor ID   Product ID   Serial Number  
(16進制)   (16進制)   (裝置序號)

當按下 open 後執行傳輸線接上指定的儀器

Catch塊檢視有無例外異常。可捕獲連線失敗等錯誤並跳出警告，避免程式直接閃退。

※mbSession(Message Based Session)

## II. 儀器通訊建立

```
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        mbSession.Dispose(); // Dispose the session
    }
    catch (Exception exp)
    {
        MessageBox.Show(exp.Message);
    }

    Application.Exit();
}
```

作用是釋放 mbSession 所佔用的資源，  
其中 Application.Exit() 則是離開應用程式



▲ open: button 1

▲ close: button 2

●.Dispose()：釋放連線資源



## II. 寫入 SCPI 與讀取

- String(字串)就是一串文字，用來儲存任何不需要直接進行數學運算的資料，例如儀器名稱、指令或是檔案路徑
- SCPI (Standard Commands for Programmable Instruments)



▲write: button 3

▲read: button 4

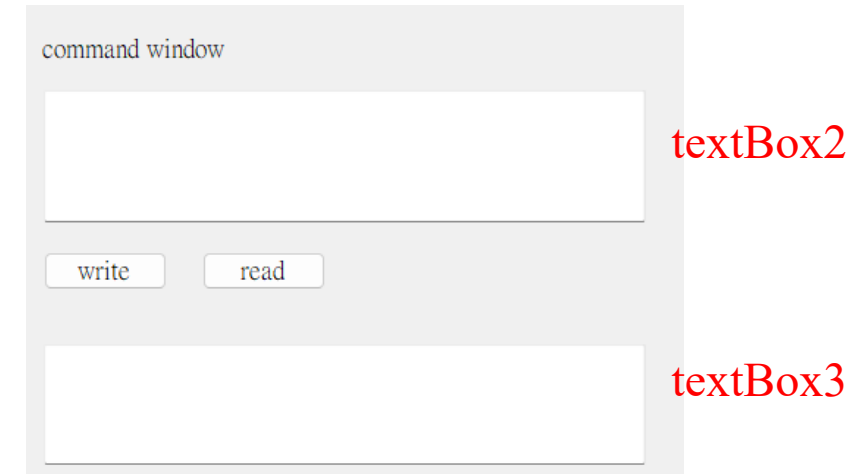
```
private void button3_Click(object sender, EventArgs e)
{
    string strWrite = textBox2.Text; // Get command from text box
    try
    {
        mbSession.Write(strWrite); // Send Write command to VISA resource
    }
    catch (Exception exp)
    {
        MessageBox.Show(exp.Message);
    }
}
```

← 建立字串，將使用者輸入的指令 (如 \*IDN?) 儲存起來準備發送

← 透過 mbSession 這個通訊橋樑，將儲存好的字串指令寫入並發送給示波器

## II. 寫入 SCPI 與讀取

- String(字串)就是一串文字，用來儲存任何不需要直接進行數學運算的資料，例如儀器名稱、指令或是檔案路徑



▲write: button 3

▲read: button 4

```
private void button4_Click(object sender, EventArgs e)
{
    string strRead = null;
    try
    {
        strRead = mbSession.ReadString(); // Read string from VISA resource
        textBox3.Text = strRead;
    }
    catch (Exception exp)
    {
        MessageBox.Show(exp.Message);
    }
}
```

指令發送後，透過 `mbSession.ReadString()` 讀取儀器回傳的反饋值，並賦予給準備好的 `strRead` 字串變數

## II. 數據儲存

textBox4

▲ 儲存: button 5

```
private void button5_Click(object sender, EventArgs e)
```

```
{
```

```
try
```

```
{
```

```
    // 檢查是否已連接儀器
```

```
    if (mbSession == null)
```

```
    {
```

```
        MessageBox.Show("請先連接儀器");
```

```
        return;
```

```
    }
```

```
    // 取得儲存路徑
```

```
    string savePath = textBox4.Text;
```

```
    // 檢查路徑是否為空
```

```
    if (string.IsNullOrEmpty(savePath))
```

```
    {
```

```
        MessageBox.Show("請輸入儲存路徑");
```

```
        return;
```

```
    }
```

```
    // 如果沒有副檔名，自動加上.csv
```

```
    if (!savePath.EndsWith(".csv", StringComparison.OrdinalIgnoreCase))
```

```
    {
```

```
        savePath += ".csv";
```

```
    }
```

利用 `if (mbSession == null)` 進行防呆。如果通訊橋樑根本沒建立（代表尚未連線），就立刻用 `return` 擋下，避免後續對空物件下指令導致崩潰。

將使用者輸入的路徑存入 `string savePath` 之後，用 `string.IsNullOrEmpty()` 檢查路徑是否為空白，確保檔案知道要存的路徑。

## II. 數據儲存

```
// 設定波形資料格式和來源 (適用於Tektronix MD03034)
mbSession.Write("DATA:SOURce CH1"); // 選擇通道1, 可改為CH2, CH3, CH4
mbSession.Write("DATA:ENCdg ASCII"); // 設定為ASCII格式
mbSession.Write("DATA:STARt 1"); // 從第一個資料點開始
mbSession.Write("DATA:STOP 10000"); // 讀取10000個點 (可調整)
```

儲存

textBox4

▲ 儲存: button 5

波形資料格式設定指令 (參考2-103 頁)

// 取得波形參數

```
mbSession.Write("WFMPRE:X INCR?");
```

查詢 X 軸 (時間) 每兩個資料點之間的間距(參考2-107 頁)。

```
string xIncrement = mbSession.ReadString().Trim();
```

```
mbSession.Write("WFMPRE:XZERO?");
```

查詢 X 軸 (時間) 第一個資料點的起始時間座標(參考2-107 頁)

```
string xZero = mbSession.ReadString().Trim();
```

```
mbSession.Write("WFMPRE:YMULT?");
```

查詢 Y 軸 (電壓) 的Scale Factor(參考2-107 頁)

```
string yMult = mbSession.ReadString().Trim();
```

```
mbSession.Write("WFMPRE:YOFF?");
```

查詢 Y 軸 (電壓) 在數位轉換上的垂直偏移量(參考2-107 頁)。

```
string yOffset = mbSession.ReadString().Trim();
```

```
mbSession.Write("WFMPRE:YZERO?");
```

查詢 Y 軸 (電壓) 的絕對中心零點偏移值(參考2-107 頁)。

```
string yZero = mbSession.ReadString().Trim();
```

※上述SCPI指令請參考Series Oscilloscopes Programmer Manual

## II. 數據儲存

```
// 讀取波形資料
mbSession.Write("CURVe?");
string waveformData = mbSession.ReadString();

// 解析資料點
string[] dataPoints = waveformData.Split(',');

// 轉換參數為數值
double xInc = double.Parse(xIncrement);
double xZ = double.Parse(xZero);
double yM = double.Parse(yMult);
double yOff = double.Parse(yOffset);
double yZ = double.Parse(yZero);

// 建立CSV內容
StringBuilder csvContent = new StringBuilder();
csvContent.AppendLine("Time(s),Voltage(V)"); // CSV標題
csvContent.AppendLine($"# X Increment: {xIncrement}");
csvContent.AppendLine($"# Y Multiplier: {yMult}");
csvContent.AppendLine($"# Y Offset: {yOffset}");
csvContent.AppendLine($"# Y Zero: {yZero}");
```

```
for (int i = 0; i < dataPoints.Length; i++)
{
    string cleanValue = dataPoints[i].Trim();
    if (!string.IsNullOrEmpty(cleanValue))
    {
        try
        {
            // 計算實際時間和電壓值
            double time = xZ + (i * xInc);
            double rawValue = double.Parse(cleanValue);
            double voltage = yZ + yM * (rawValue - yOff);

            csvContent.AppendLine($"{time:E6},{voltage:E6}");
        }
        catch
        {
            // 跳過無法解析的資料點
        }
    }
}

// 儲存CSV檔案
File.WriteAllText(savePath, csvContent.ToString());

MessageBox.Show($"波形資料已成功儲存至: {savePath}\n資料點數: {dataPoints.Length}",
    "儲存成功", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
catch (Exception exp)
{
    MessageBox.Show($"儲存失敗: {exp.Message}", "錯誤",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

# Outline

## I 課程內容

- 課程內容前置
- C#課程內容參考資源

## II C# 基礎語法

- 儀器通訊建立
- SCPI寫入與讀取
- 數據儲存

## III. 實作作業

# III. 實作作業

1. 使用下方SCPI指令寫入完成Average 功能，並將數據儲存至CSV檔後輸出波型圖形  
(初始先用產波器輸出1 kHz 正弦波，1 Vpp)

※盡量一行一行寫入，避免一次輸入多條指令造成後面程式未能寫入示波器

確認設備身份

\*IDN?

設置示波器基本參數

:SElect:CH1 ON # 開啟通道 1

:CH1:SCALe 0.5 # 垂直刻度 0.5V/div

:HORizontal:SCALe 0.0002 # 時基 200us/div

:TRIGger:MAIn:TYPE EDGE # 邊緣觸發

:TRIGger:MAIn:EDGE:SOURce CH1 # 觸發源為通道 1

:TRIGger:MAIn:LEVel 0.1 # 觸發電平 0.1V

啟用 Average 模式

:ACQuire:MODE AVErage # 擷取模式設為平均模式

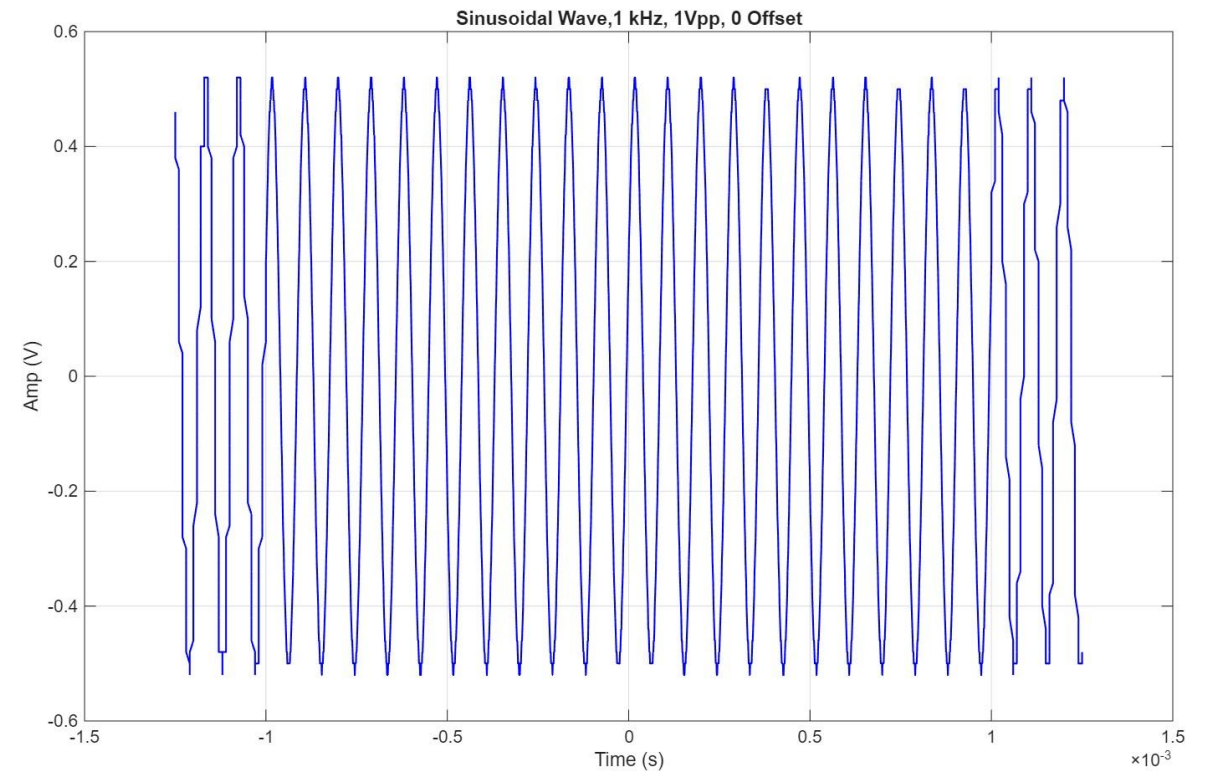
:ACQuire:NUMAVg <count> # <count>為平均次數

:ACQuire:STATE ON # 啟動資料擷取

# III. 實作作業



控制示波器之GUI



擷取csv的波形

# III. 實作作業

2. 在目前的示波器控制程式中，我們透過 NI-VISA 進行連線控制。原本的設計要求使用者手動輸入 VISA 資源名稱（Resource Name）。需將介面修改成程式可以在啟動時能自動掃描並列出目前連接的儀器之 Resource Name。

- 需做成報告給我，報告要有修改過程及邏輯、結果、程式碼....等。