

Tutorial - Hello World

陳峻暉

September 9, 2022

Contents

About This Document.....	1
A Hello World Form	2
Windows Form Designer	5

About This Document

本次練習介紹了兩個主題：(1) 如何在 Microsoft Visual Studio 中創建 Windows Forms 專案，以及 (2) 如何在 Forms 中使用 Buttons、TextBox 和其他控件。在視窗程式設計課程中，我們使用 C#作為我們的程式語言，以及使用 Windows Form 技術來構建 Windows Forms 應用程式。Microsoft Visual Studio 是一個強大的 Windows IDE，我們將在整個學期使用它。

A Hello World Form

Step 1 開啟 Visual Studio

開始 → 所有程式 → Microsoft Visual Studio 2019 → Visual Studio 2019。

Step 2 建立新專案 (Windows Forms 應用程式)

檔案 → 建立新專案 → 選擇Windows Forms應用程式 (Figure 1-1) → 輸入專案名稱及儲存位置 (Figure 1-2) → 選擇目標Framework(使用預設) → 建立。

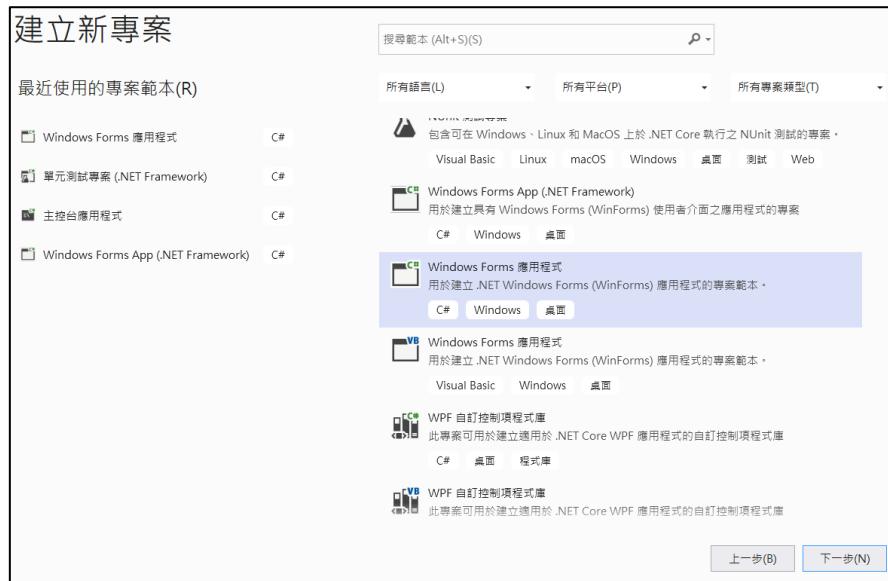


Figure 1-1 建立新專案



Figure 1-2 輸入專案名稱及儲存位置

Step 3 重設視窗配置

若你在操作專案時不小心更改了視窗的位置，可以點擊“視窗”→“重設視窗配置” (Figure 1-1) 將視窗位置復原成預設的樣子 (Figure 1-2)。

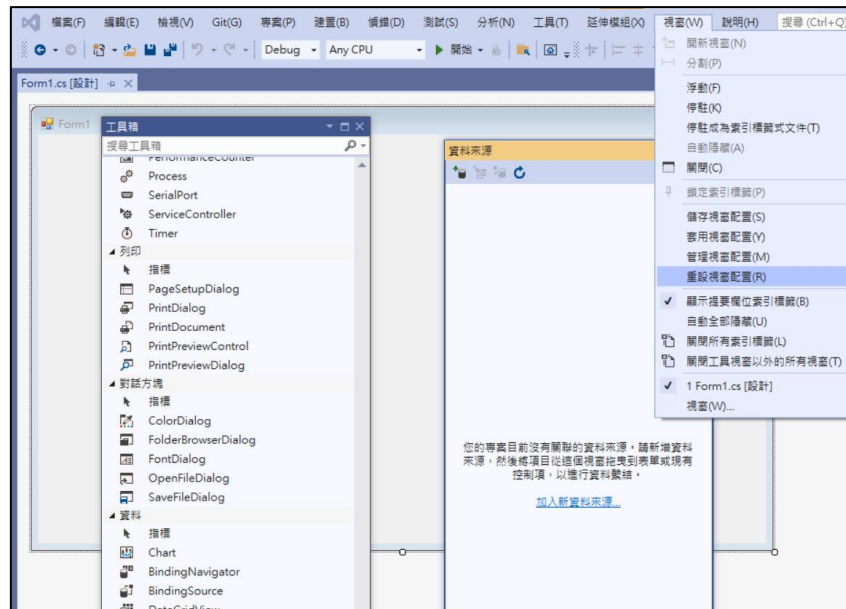


Figure 2-1 點擊重設視窗配置

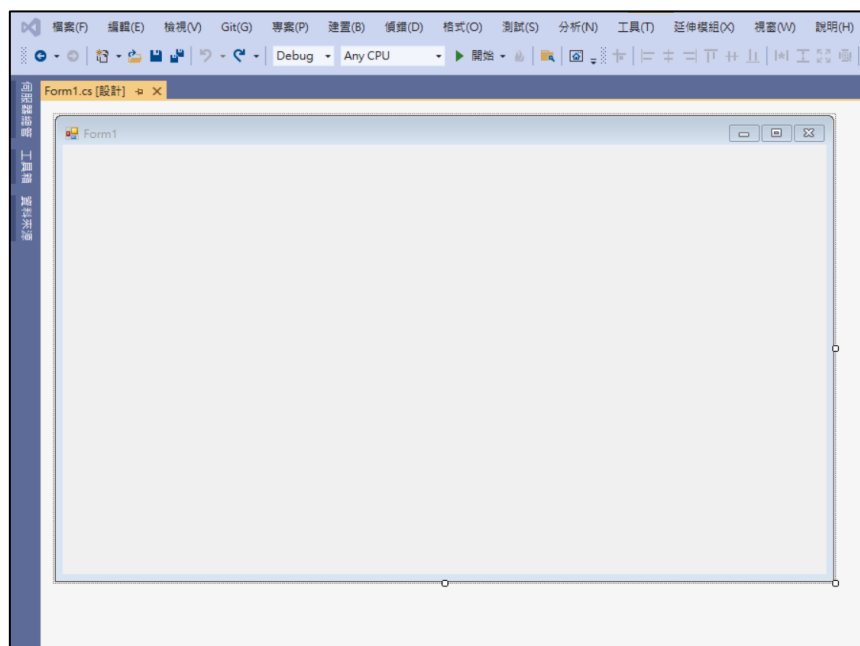


Figure 2-2 重設視窗配置結果

Step 4 修改 Form 標題

單擊起始畫面中的 Form，可以呼叫出此 Form 的屬性窗格，修改 Text 屬性為「Hello World」，修改後畫面中 Form 的標題也會同步顯示為 Hello World。

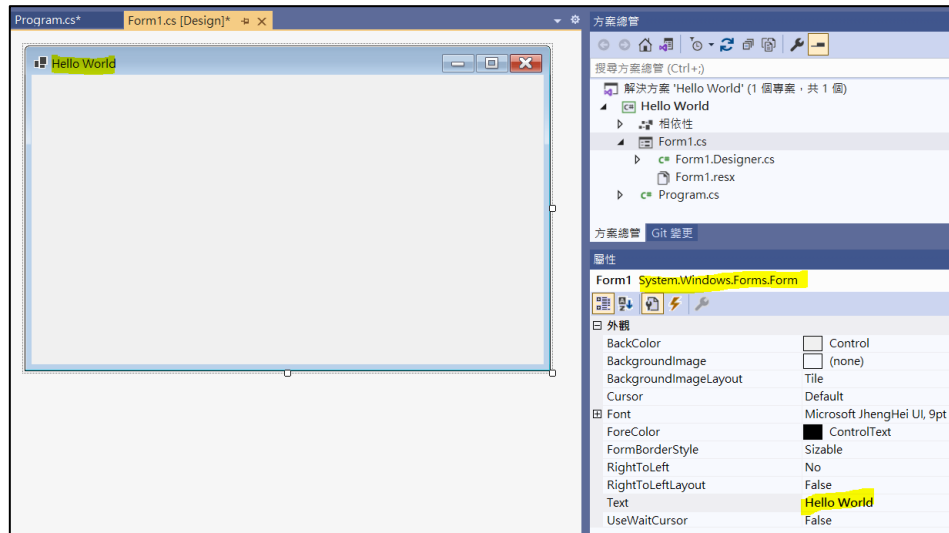


Figure 2 修改 Form 標題

按下“F5”鍵(開始偵錯)，如果沒有出錯，Visual Studio 將會編譯並執行程式，執行結果如 Figure 3，會出現一個空視窗，抬頭為“Hello World”，沒有任何功能，只能按“×”關閉視窗。

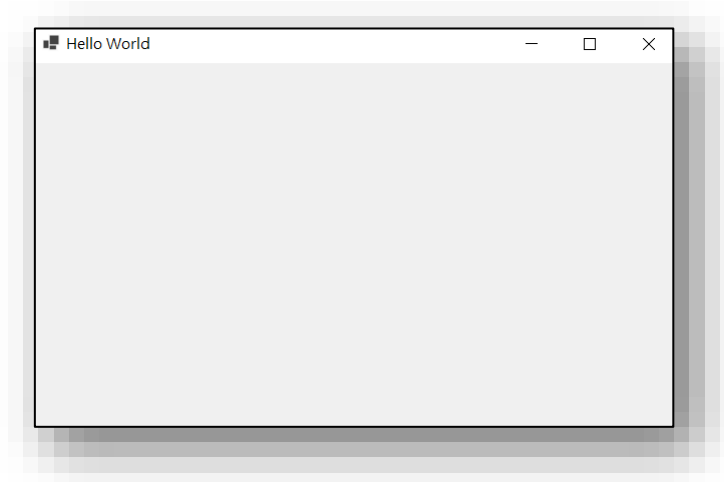


Figure 3 執行結果

Windows Form Designer

Step 5 開啟 Windows Form Designer

在 Visual Studio 中，Windows Form 應用程式提供我們一個簡易工具用來設計使用者介面，首先雙擊專案中的“Form1”項目 (Figure 4)，Visual Studio 就會直接開啟 **Windows Form Designer** (Figure 5)。

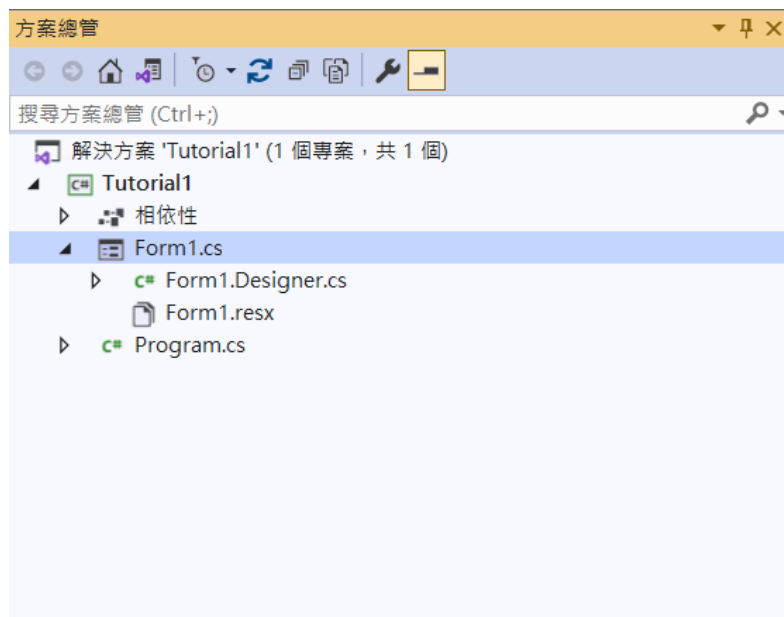


Figure 4 雙擊專案中的 "Form1" 項目

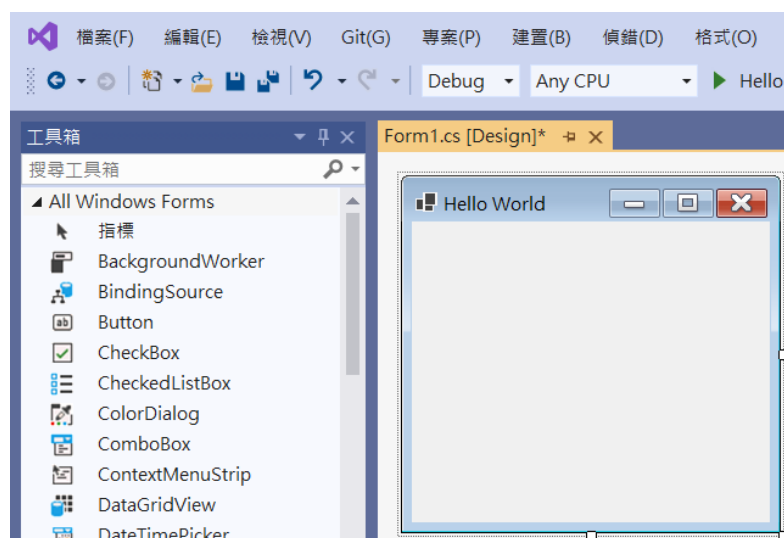


Figure 5 開啟 Windows From Designer

Step 6 工具箱元件

工具箱中有許多的元件可以使用，點擊 Visual Studio 最左側工具箱(Figure 6) 即可開啟工具箱，請依序將 1 個 Label、1 個 TextBox 以及 1 個 Button，拖曳至 Form 中擺放(Figure 7)，接著點擊上述元件開啟對應的屬性視窗，依序修改這 3 個元件的 Name 屬性改為 IDLabel、IDTextBox 和 addButton (Figure 8)，並將其 Text 屬性改為“Student ID”、“”，以及“Add Student”。

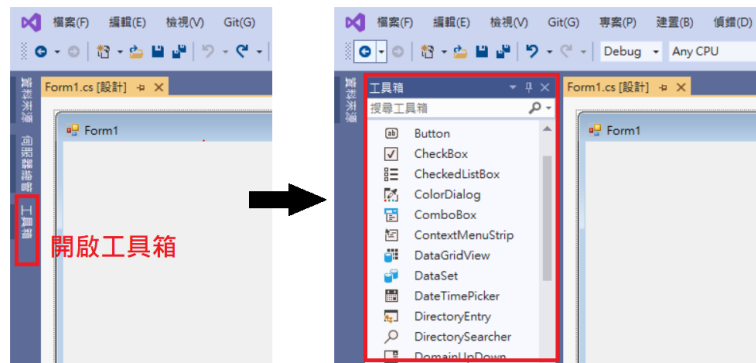


Figure 6 開啟工具箱

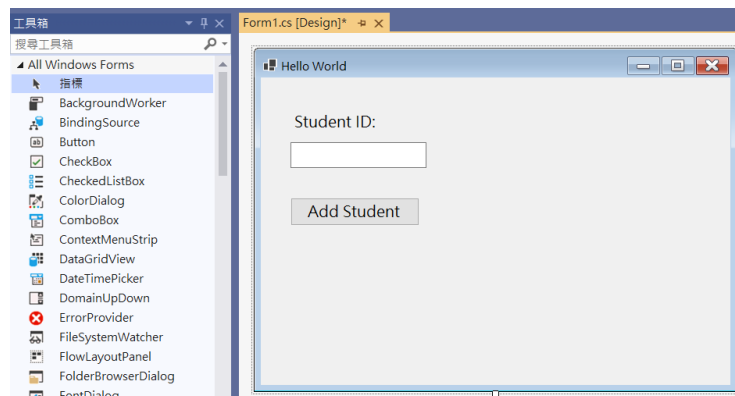


Figure 7 擺放工具箱元件

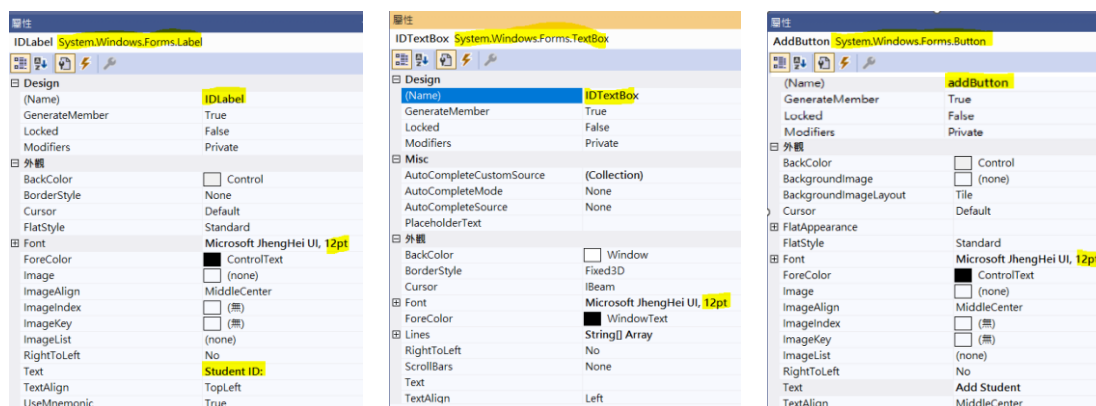


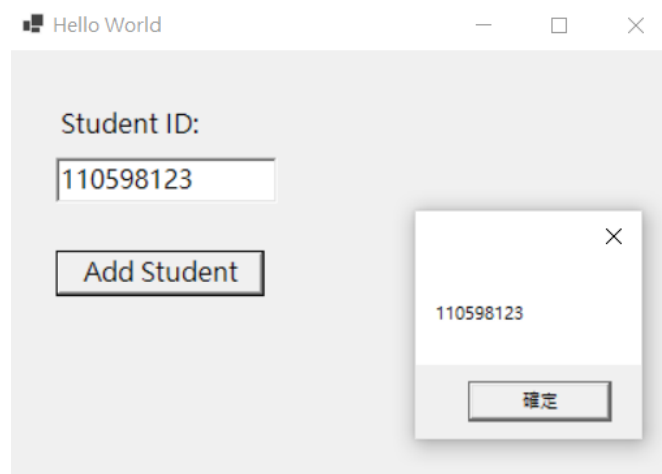
Figure 8 修改元件屬性

Step 7 按鈕點擊事件及 MessageBox

接著雙擊 "Add Student" 按鈕，Visual Studio 將會自動建立一個按鈕點擊事件的處理程式 (AddButton_Click)，並且切換到程式編輯頁面，我們在此處理程式中寫一個 MessageBox，用來顯示 TextBox 中的文字，程式如 Figure 9-(a)，接著編譯並執行程式 (按 F5)，當畫面出現後在 TextBox 輸入 student ID，再點擊 "Add Student" 按鈕，檢查結果是否如 Figure 9-(b)。



(a) AddButton_Click 處理程式



(b) 程式結果畫面

Figure 9 事件的處理程式

重複 Step 6 和 Step 7，再新增一個 Label 及一個 TextBox，如 Figure 10，其元件之 name 屬性分別設為 nameLabel 與 nameTextBox，並修改按鈕點擊事件程式碼(下圖)，使 MessageBox 顯示學生 ID 和 name，若修改正確，則執行結果應如圖 10 所示。

```
private void AddButton_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello, " + IDTextBox.Text + " " + nameTextBox.Text);
}
```

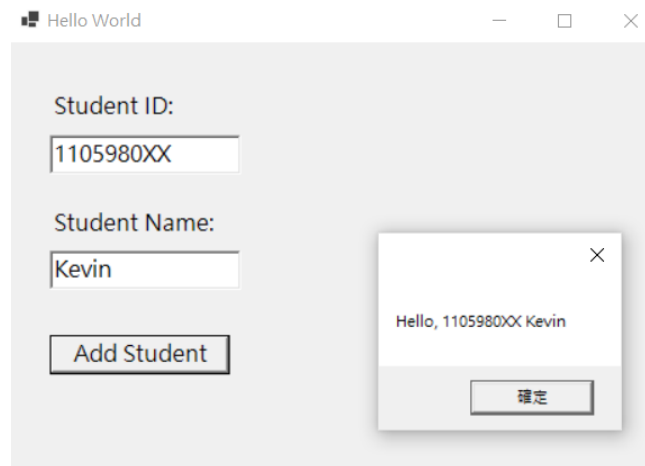


Figure 10 新增元件並修改 MessageBox 執行結果

Step 8 新增 DataGridView

當每一筆資料的屬性既多樣化，而且資料筆數越來越多時，我們可以透過 DataGridView 以表格的形式顯示資料，DataGridView 提供簡易的方式來綁定(顯示)靜態資料，也能夠動態新增、刪除表格中的資料。

從工具箱中找到 DataGridView 並拖曳至 Form 中，接著點選右上角的小按鈕打開 DataGridView Tasks，點選“Edit Column”我們可以在表格中新增 ID、Name 等欄位 (Figure 11-1~11-3)。

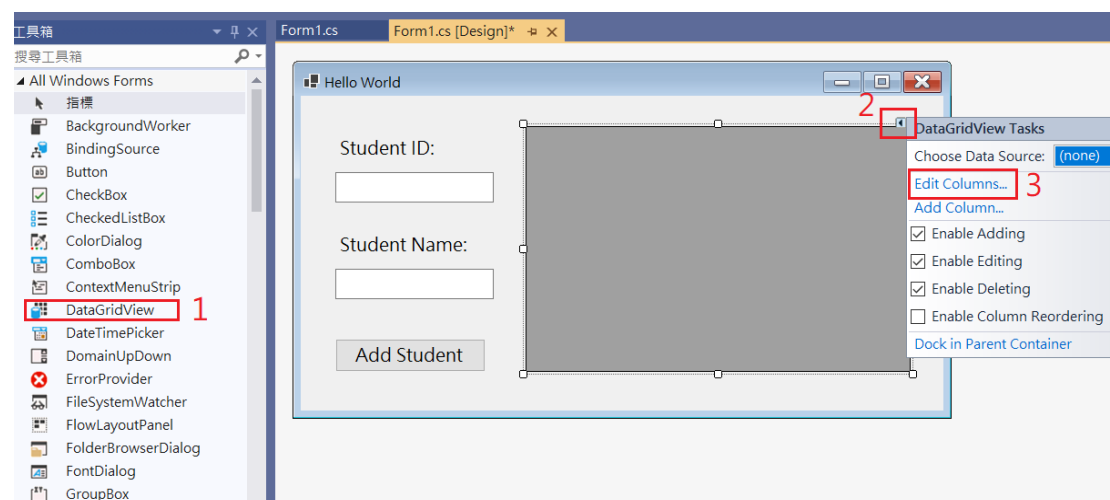


Figure 11-1 新增 DataGridView

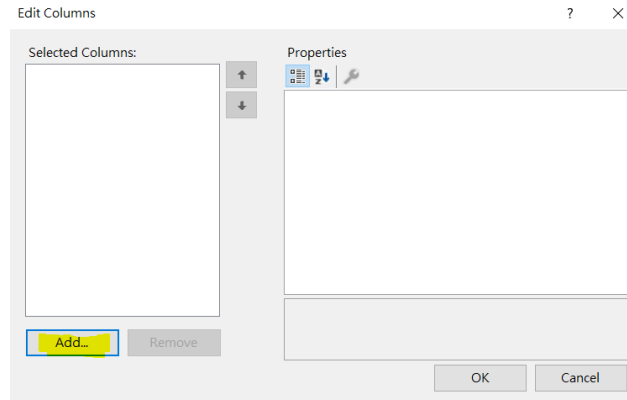


Figure 11-2 編輯 Columnm

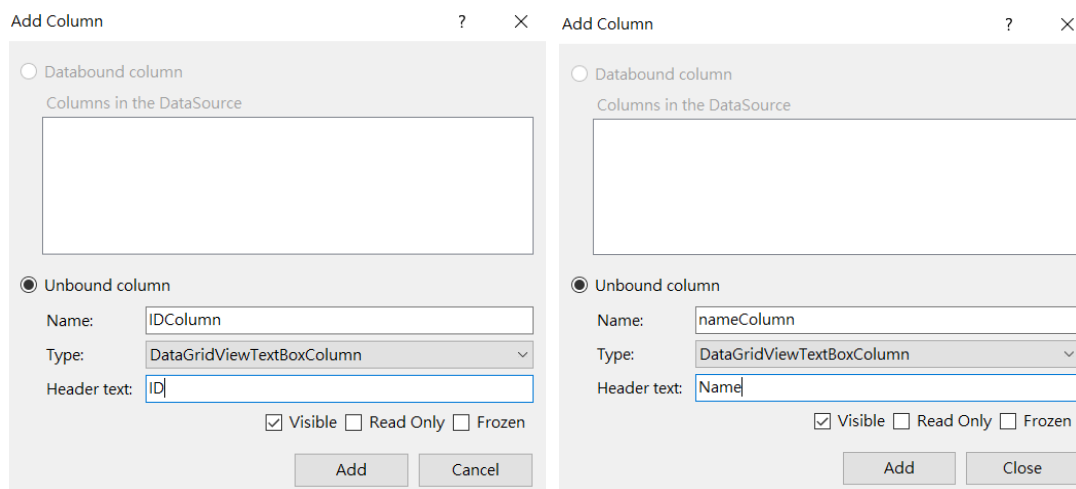


Figure 11-3 新增 ID、Name Columnm

Step 9 修改 DataGridView 屬性及顯示預設資料

在屬性窗格中，將 DataGridView 修改名稱為 studentDataGridView，並修改 AutoSizeColumnsMode 為 Fill(Figure 12) 即可將欄位標題延展填滿窗格 (Figure 13)。

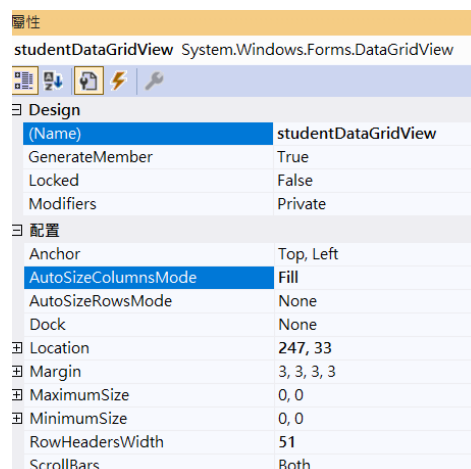


Figure 12 修改 DataGridView 屬性

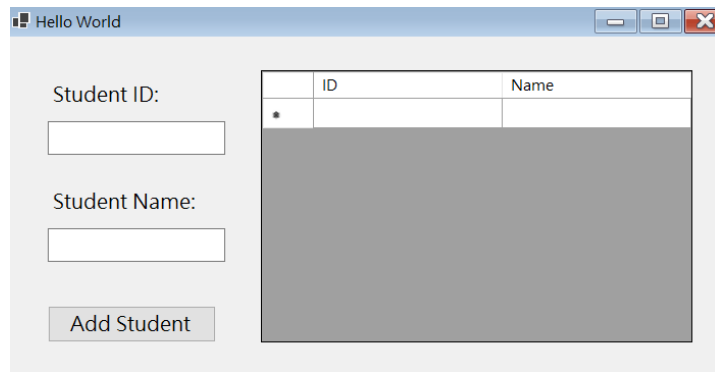


Figure 13 延展欄位標題

Step 10 靜態顯示預設資料

在 Form1.cs 中按**右鍵**並點選”檢視程式碼”進入程式編輯頁面，我們可以透過修改 Form1() 建構式，為 DataGridView 預先寫入一筆資料，程式如下圖，最後我們執行程式檢查畫面是否如 Figure 14 所示。

```
public Form1()
{
    InitializeComponent();
    studentDataGridView.Rows.Add("1105980XX", "Kevin");
}
```

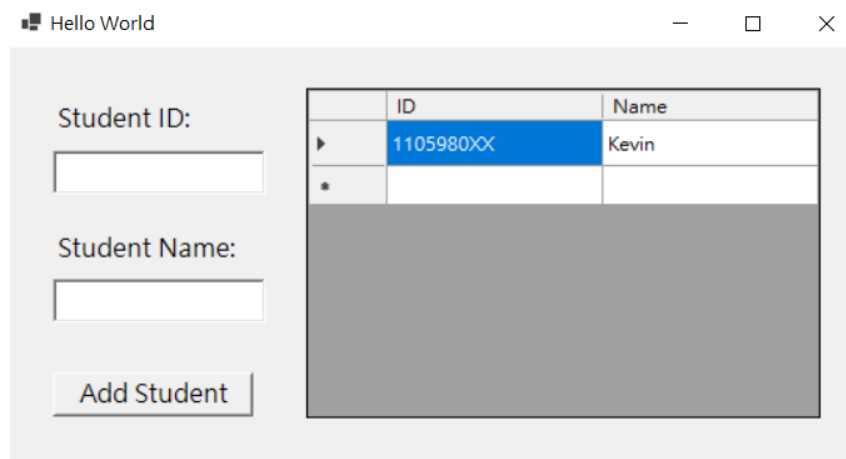


Figure 14 顯示資料結果

Step 11 動態新增資料

再次修改點擊按鈕事件程式，如下圖，當按下按鈕時，將 textBox 中的 ID、Name 資料新增至 DataGridView 中，修改後執行程式檢查結果是否如圖 Figure 15。

```
private void AddButton_Click(object sender, EventArgs e)
{
    studentDataGridView.Rows.Add(IDTextBox.Text, NameTextBox.Text);
}
```

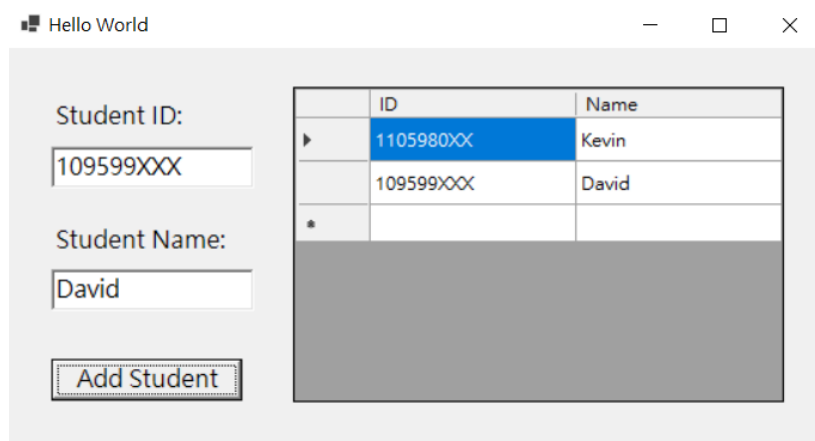


Figure 15 新增資料結果

Step 12 提示錯誤訊息

最後，在點擊按鈕事件中，我們增加簡單的**防呆邏輯**，當使用者在沒有輸入 ID、Name 的情況下點擊按鈕，我們希望使用者會接收到錯誤訊息的提示，修改點擊按鈕事件，如下圖，並重新編譯執行程式，當點擊“Add Student”按鍵時結果應如 Figure 16-18。

```
private void AddButton_Click(object sender, EventArgs e)
{
    if (IDTextBox.Text == "")
    {
        MessageBox.Show("Please enter ID");
    }
    else if (NameTextBox.Text == "")
    {
        MessageBox.Show("Please enter Name");
    }
    else
    {
        studentDataGridView.Rows.Add(IDTextBox.Text, NameTextBox.Text);
    }
}
```

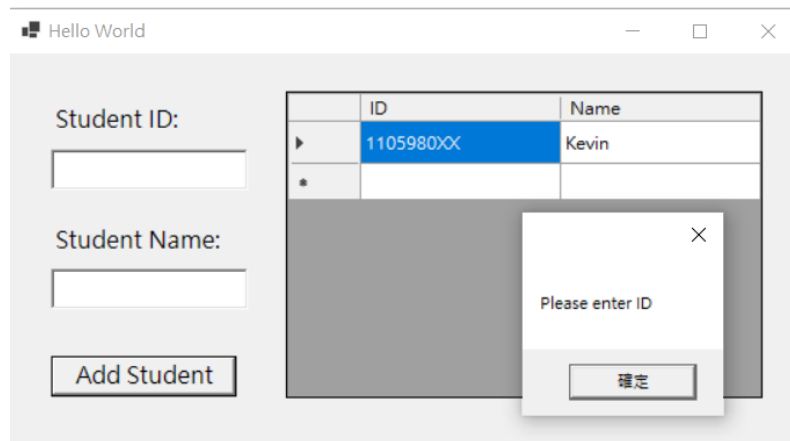


Figure 16 提示未輸入 ID 錯誤訊息

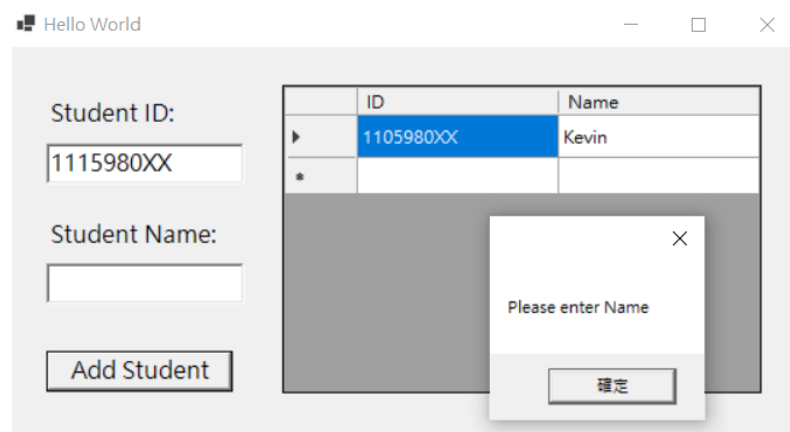


Figure 17 提示未輸入 Name 錯誤訊息

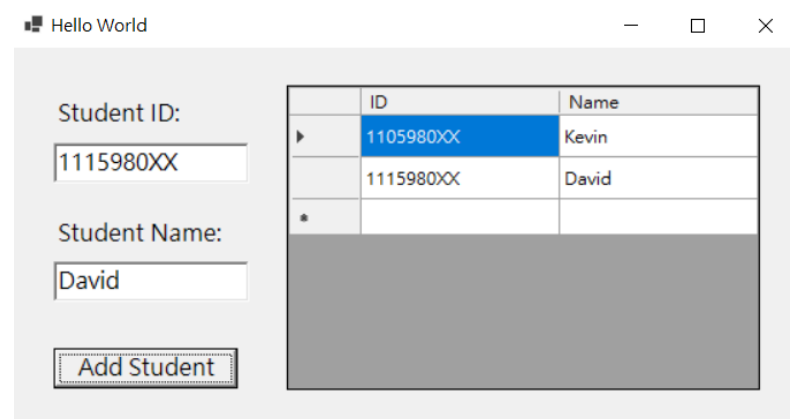


Figure 18 完整輸入成功新增資料

-- The End --