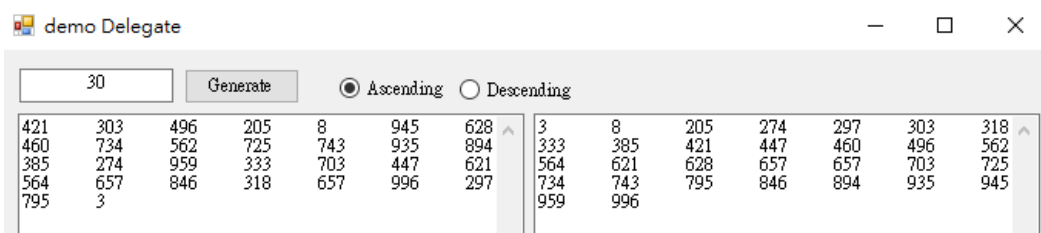
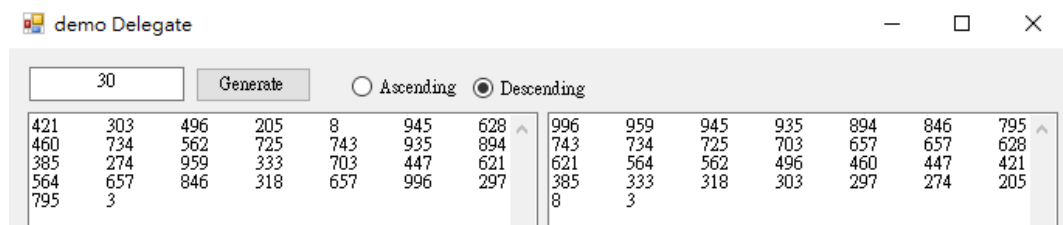
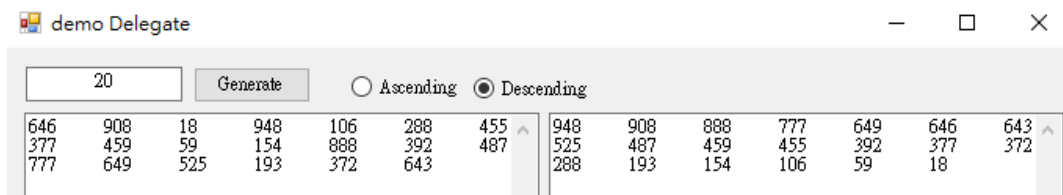
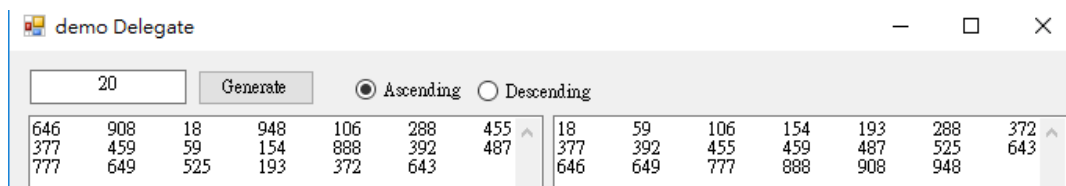
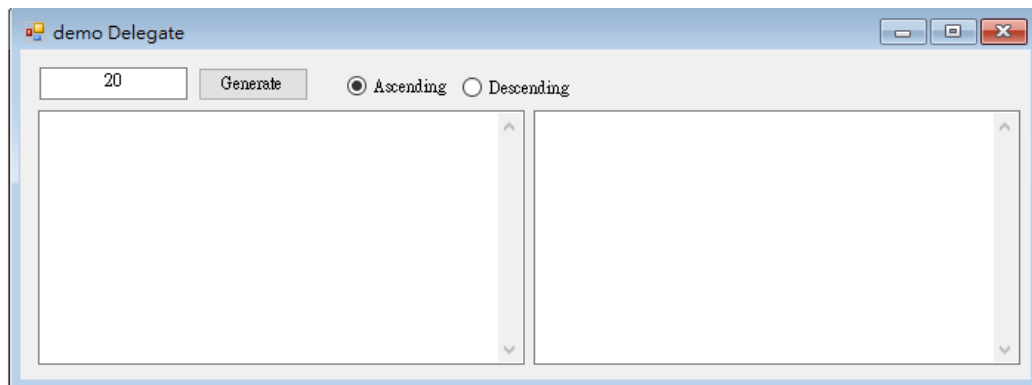


Topic 02: Introduction to Delegate

- ✧ 資料型別可以用來宣告變數或物件，例如 `int` 型別宣告出來的變數可以儲存數字、`string` 型別宣告出來的變數可以儲存文字
- ✧ **Delegate (委派)** 是用來儲存方法位址的資料型別。
 - 這種資料型別宣告出來變數所儲存的是方法的位址，一個儲存了方法位址的委派變數，可以用來呼叫該位址的方法
 - 由於變數的內容可以動態改變，因此只要將該變數內容改變為另一個方法的位址，該變數呼叫的就會是另一個方法。

Ex. 針對陣列資料指定排序方法



```

11 namespace Ex01_delegate
12 {
13     3 references
14     public partial class Form1 : Form
15     {
16         1 reference
17         public Form1()
18         {
19             InitializeComponent();
20         }
21         Random rnd;
22         uint N;
23         int[] data;
24         1 reference
25         private void Form1_Load(object sender, EventArgs e)
26         {
27             rnd = new Random((int)DateTime.Now.Ticks);
28         }
29         1 reference
30         private void btnGenerate_Click(object sender, EventArgs e)
31         {
32             if (uint.TryParse(txtBxN.Text, out N))
33             {
34                 data = new int[N];
35                 for (int i = 0; i < N; i++)
36                     data[i] = rnd.Next(0, 1000);
37                 txtBxRaw.Text = "";
38                 foreach (int dd in data)
39                     txtBxRaw.Text += string.Format("{0}\t", dd);
40             }
41             else
42             {
43                 MessageBox.Show("Please input a positive number!!", "Error");
44                 txtBxN.Focus();
45             }
46         }
47     }
48 }

```

```

43 bool IsSmaller(int x, int y)//step.1
44 {
45     return x < y;
46 }
47 2 references
48 bool IsBigger(int x, int y)//step.1
49 {
50     return x > y;
51 }
52 delegate bool CompareFunc(int x, int y);//step.2
53 CompareFunc myCompare;//step.3
54 1 reference
55 void swap(ref int x, ref int y)
56 {
57     int tmp;
58     tmp = x;
59     x = y;
60     y = tmp;
61 }

```

```

60 void mySort(int[] data0, CompareFunc CompareMethod)
61 {
62     int i, j;
63     for (i=0;i<=data0.Length-2;i++)
64     {
65         for (j=i+1;j<=data0.Length-1;j++)
66         {
67             if (CompareMethod(data0[j], data0[i]))
68                 swap(ref data0[j], ref data0[i]);
69         }
70     }
71 }
72 private void radioButton1_Click(object sender, EventArgs e)
73 {
74     if (myCompare != IsSmaller)
75         myCompare = new CompareFunc(IsSmaller); //step.4
76     mySort(data, myCompare);
77     txtBxSort.Text = "";
78     foreach (int dd in data)
79         txtBxSort.Text += string.Format("{0}\t", dd);
80 }
81 private void radioButton2_Click(object sender, EventArgs e)
82 {
83     if (myCompare != IsBigger)
84         myCompare = new CompareFunc(IsBigger); //step.4
85     mySort(data, myCompare);
86     txtBxSort.Text = "";
87     foreach (int dd in data)
88         txtBxSort.Text += string.Format("{0}\t", dd);
89 }
90 }
91 }

```