1. **請從技術面及使用面解釋interface及abstract的差異**

**abstract:**  
1.可以定義字段、屬性、方法實現

2.只能被單一繼承  
3定義在緊密相關的類別間

**interface:**  
1.只能定義屬性、索引器、事件、和方法聲明，不能包含字段

2.可以繼承多個介面

3.定義在關系疏松但都實現某一功能的類別中

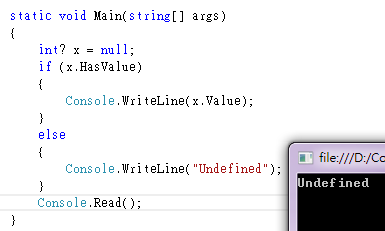
1.設計大的功能單元，使用抽象類；如果設計小而簡練的功能，則使用介面

2.需要創建組件的多個版本，則創建抽象類，因介面一旦創建就不能更改，如需新版本，必須創建一個全新的介面。

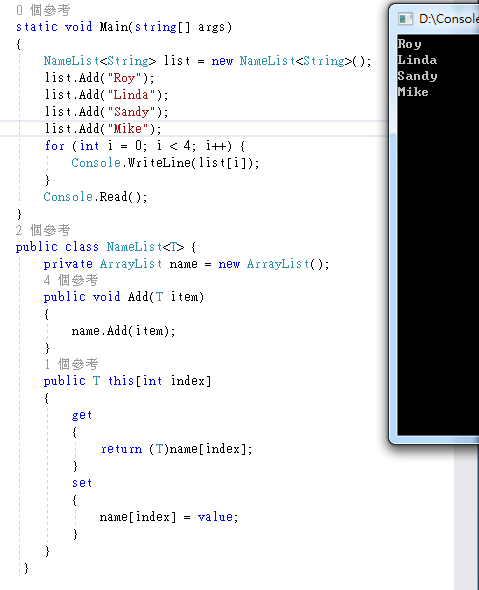
2. **什麼是泛型 (Generic Types)，特性有哪些？請示著宣告一個可為Null的實值型別變數。(10%)**

定義泛型類別時不指定某些變數的具體類型，而是用一個類型參數代替，使用這個類別時，這個類型的參數會由一個具體的類型所代替

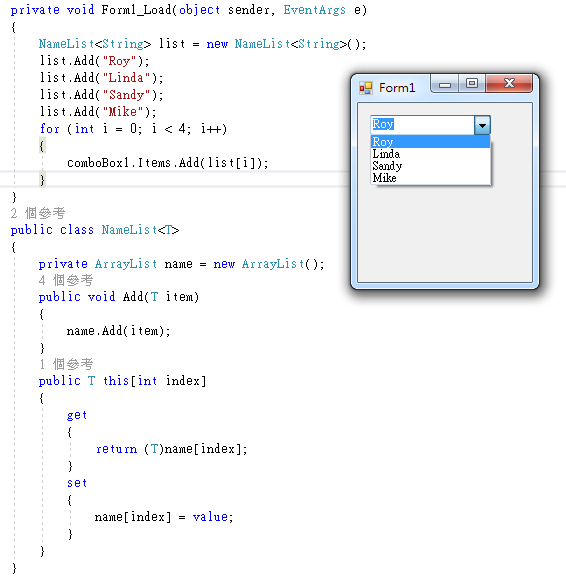
特性:重複使用性、型別安全、高效率



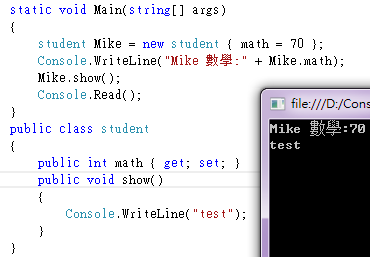
3.**承1，請試著使用.NET Framework既有泛型，使用Console.WriteLine(string s)，列出任意自建字串清單內容(例如英文人名清單)。(10%)**



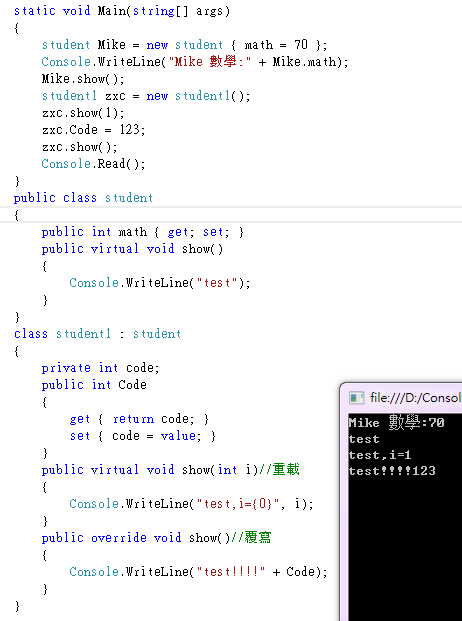
4. 承2，請改寫為WPF、Windows Forms或是ASP.NET/C#(擇一)，使用清單類型的控制項(Control)，顯示

該字串清單。(20%)

5. 請示著自行定義一個泛型類別(Class)及包含至少一個泛型方法(Method)，並使用該自定義類別宣告並初始化一個物件。(15%)

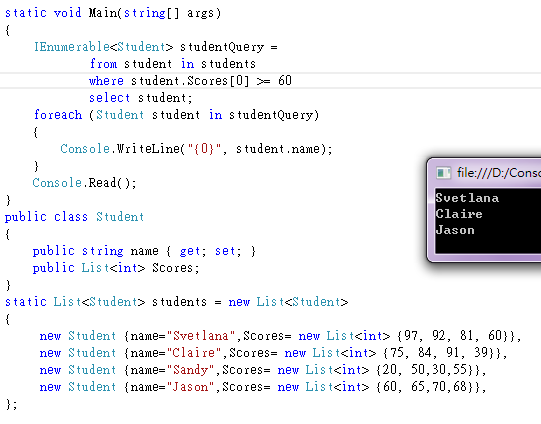


6. 承5，請繼承該自定義類別，覆寫(overriding)至少一個方法(提示：virtual、override)，加入一個重載(overloading)方法；並加上Property及Field混搭應用。(20%)



7. 加分題，試著寫一個Linq的語法來查詢泛型(Generic Types)：(25%)

1.使用Query Expressions寫一個範例



1. 使用Lambda寫一個範例

