

HCCLAB-人本計算實驗[體感篇]

DAY1-C# TUTORIAL

I. C# TUTORIAL

1. C# INTRODUCTION

C#(C-sharp)是微軟推出的一種基於.NET 框架的、物件導向的高階程式語言。C#由 C 語言和 C++衍生而來，繼承了其強大的效能，同時又以.NET 框架類別庫作為基礎，擁有類似 Visual Basic 的快速開發能力。C#由安德斯·海爾斯伯格主持開發，微軟在 2000 年發行了這種語言，並希望藉助這種語言來取代 Java。C#已經成為 Ecma 國際和國際標準組織的標準規範。

2. DESIGN GOALS OF C#

- ✓ C#旨在設計成為一種「簡單、現代、通用」，以及物件導向的程式語言
- ✓ 此種語言的實作，應提供對於以下軟體工程要素的支援：Strongly type 的檢查、陣列維度檢查、未初始化的變數參照檢測、自動垃圾收集 (Garbage Collection，指一種記憶體自動釋放技術)。軟體必須做到強大、持久，並具有較強程式開發的生產力。
- ✓ 此種語言為在分散式環境中的開發提供適用的元件開發應用。
- ✓ 為使程式設計師容易遷移到這種語言，原始碼的可移植性十分重要，尤其是對於那些已熟悉 C 和 C++的程式設計師而言。
- ✓ 對國際化的支援
- ✓ C#適合為獨立和嵌入式的系統編寫程式，從使用複雜作業系統的大型系統到特定應用的小型系統均適用。
- ✓ 雖然 C#程式在儲存和操作能力需求方面具備經濟性，但此種語言並不能在效能和程式大小方面與 C 語言或組合語言相抗衡。

3. C# COMPILER?

- Visual C#(for Windows)
- Mono C#(for Unix-like OS)

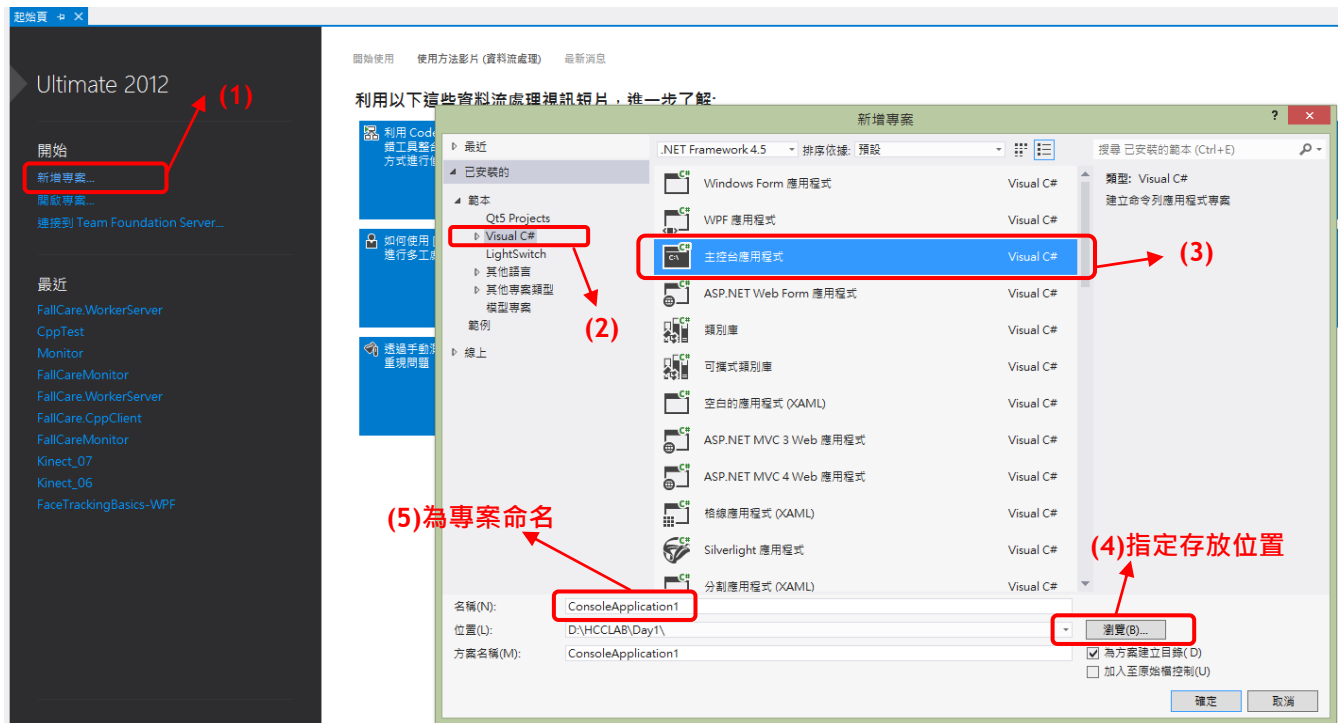
4. ENVIRONMENT FOR THIS COURSE

Visual Studio 2015 community → You can download it online

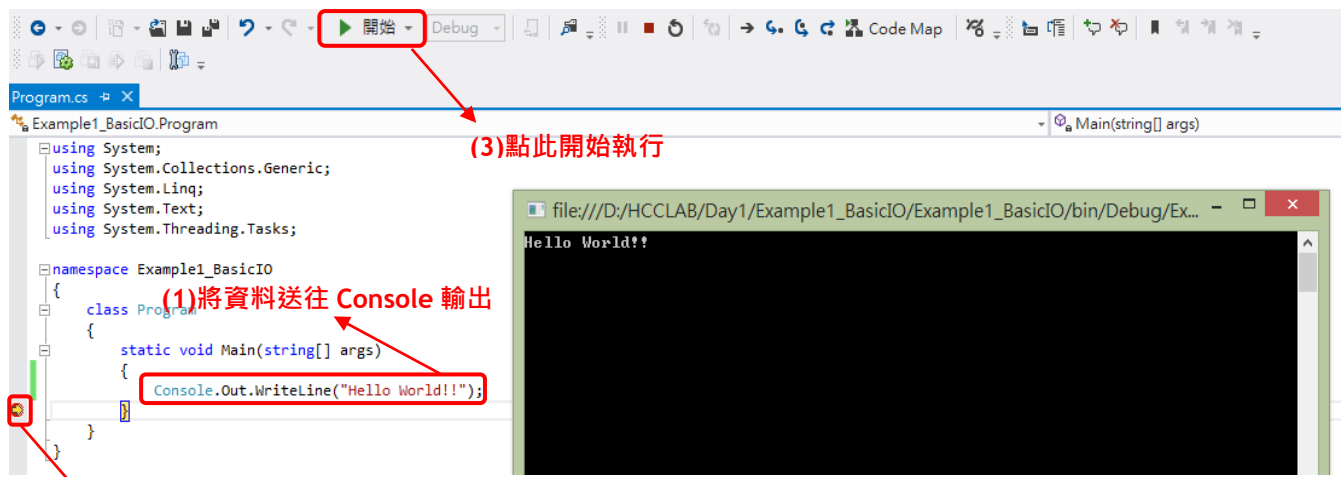
We suggest that all of you install it to your own PC not only for your HW, but also for your Project.

5. C# PROGRAMMING GUIDE

A. COMMAND LINE BASIC IO



*建議同學將今天所練習的程式的存放位置放在同一個資料夾內，方便打包回家複習



EXERCISE.1

重現上圖，並請助教 DEMO 通過即可。

B. PRIMITIVE TYPE INTRODUCTION

C#的常見型別有 char, int, float, double, bool, string...，雖然在名稱上與 C/C++或是 Java 等語言相似，但實作方式相異。C#的上述型別是利用 Struct 實作，並利用 alias 的方式重新命名以符合大多數程式設計師的習慣，而 C/C++, Java 的 primitive type 則否。

以下我們會利用程式說明示範 C#在 Console I/O 中不同型態的輸入差異。使用者會輸入 ID 與名字，而 Console 會輸出 Hello,{ID}{名字}，比如 ID=1234567,名字是王大明，則程式會輸出：

Hello,1234567 王大明

而程式如下：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Example2_BasicType
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Input your ID:"); //Console.WriteLine會換行 Console.Write則否
            int userID;
            if (!int.TryParse(Console.ReadLine(), out userID)) //輸入進來時type為string，在此使用int.TryParse轉型
            {
                Console.WriteLine("ID is invalid");
                return; //若錯誤，則印出錯誤訊息並跳出
            }
            Console.Write("Input your name:");
            string userName = Console.ReadLine();
            Console.WriteLine("Hello," + userID + " " + userName); //userID被隱含轉換成string(透過ToString方法)
        }
    }
}
```

EXERCISE.2

寫一個小程序用來計算人的 BMI，身高的 type 須為 int，體重的 type 須為 float/double。

範例：

輸入您的身高：182

輸入您的體重：75.3

您的 BMI 為 22.6421929

寫完請助教 Demo 通過即可。

C. ARRAY

C# 當中的陣列跟 C/C++ 一樣是用中括號的方式宣告的，而 index 也繼承了 C/C++ 的特性是以 0 作開始，(length-1) 作結束。但多維陣列的宣告方式略有不同。以下我們會利用程式說明示範 C# 中陣列的型別與他們獨特的初始化方式。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Example3_Array
{
    class Program
    {
        static void Main(string[] args)
        {
            //陣列的三種初始化方式

            int[] aArray = new int[10]; //利用new來初始化陣列
            int[] bArray = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }; //若已知array中欲初始化的值，可利用大括號對{}來初始化陣列
            int[] cArray = bArray; //此種方式為參考複製，若更動到bArray, cArray其中之一的元素，則另一陣列也會隨之改變

            //多維陣列的兩種type

            int[][] a1Matrix = new int[10][]; //type 1. 類似C/C++的多維陣列，好處是可以動態更改第二維的長度
            for (int i = 0; i < 10; i++)
                a1Matrix[i] = new int[20];
            int[][] a2Matrix = { new int[] { 1, 2, 3 }, new int[] { 1, 2 } }; //必須加new int[]

            int[,] b1Matrix = new int[10, 20]; //type 2. 很方便地宣告方式，壞處是每個第二維的長度都必須一樣
            int[,] b2Matrix = { { 1, 2, 3 }, { 1, 2, 3 } }; //不可加new int[]
        }
    }
}
```

EXERCISE.3

分別用兩種方式創造一個三維 3*3*3 陣列，並將內容初始化為

```
1 1 1  2 2 2  3 3 3
1 1 1  2 2 2  3 3 3
1 1 1  2 2 2  3 3 3
```

寫完請助教 Demo 通過即可。

D. FLOW CONTROL

C#的流程控制方式與其他語言相似，在條件判斷部分使用 `if-else` 或是 `switch-case`，而在迴圈部分使用 `for`, `while`, `do-while`，而除了這些 C/C++ 相同的部分，C# 也有在 Java 中被稱為 `Enhanced for` 的增強型 (簡便型) 迴圈 `foreach`。

以下我們會利用程式說明示範 C# 的流程控制，我會印出亂數陣列 (`randArray`) 中所有的數字以及其中的奇數。假設亂數陣列中的值為 3, 22, 97, 99, 56, 47, 2, 38, 42, 46，則程式會輸出：

3, 22, 97, 99, 56, 47, 2, 38, 42, 46

奇數為：3 97 99 47

而程式如下：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Example4_FlowControl
{
    class Program
    {
        static void Main(string[] args)
        {
            const int arrayLength=10; //設定array長度為10
            Random randomGenerator = new Random(); //產生Random實體
            int[] randArray = new int[arrayLength];
            for (int i = 0; i < arrayLength; i++)
            {
                randArray[i] = randomGenerator.Next(100); //randomGenerator產生最大值為100的非負整數亂數
            }
            //印出亂數陣列
            foreach (int i in randArray)
            {
                Console.Write(i + " ");
            }
            //印出奇數
            Console.WriteLine(Environment.NewLine+"奇數："); //字串中的換行符號標準使用Environment.NewLine
            for (int i = 0; i < arrayLength; i++)
            {
                if (randArray[i] % 2 == 1) //判定是否為奇數
                    Console.Write(randArray[i] + " ");
            }
        }
    }
}
```

EXERCISE.4

利用「埃拉托斯特尼篩法(Sieve of Eratosthenes)」來找出正整數 N 以下的所有質數， N 必須給使用者輸入。範例：

Input N: 20
2 3 5 7 11 13 17 19

提示：求方根函式為 `Math.Sqrt`。寫完請助教 Demo 通過即可。

E. FUNCTION AND RECURSION

C#的函數分為靜態函數(static member function)與成員函數(member function)兩類，靜態函數附屬於類別，呼叫時可以直接指定類別名稱即可。成員函數附屬於物件，呼叫時必須透過物件變數進行呼叫。在此我們會先介紹靜態函數，而成員函數留待下一章節再作介紹。

以下我將會利用程式說明示範 C#的函數宣告及使用，我會使程式擷取使用者輸入 N,並計算 N!且顯示在螢幕上。假設輸入為 5，則程式會輸出：

```
Input N:5
N!=120
```

而程式如下：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Example5_FunctionAndRecursion
{
    class Program
    {
        static void Main(string[] args)//Main也是一個static member function
        {
            Console.Out.Write("Input N:");
            uint N;
            if (!uint.TryParse(Console.ReadLine(), out N))
            {
                Console.WriteLine("N is invalid.");
                return;
            }
            Console.Out.WriteLine("N!="+getFactorial(5));
        }
        //在C#中 函數的宣告不受前後限制 宣告不須在使用之上
        static uint getFactorial(uint n)//uint為unsigned int在C#中之寫法
        {
            return n == 0 ? 1 : n * getFactorial(n - 1);
            //利用?:運算符來做回傳值控制
            //意同if(n==0) return 1; else return n * getFactorial(n - 1);
        }
    }
}
```

EXERCISE 5.

利用上述所學寫出計算費波那契數列(Successione di Fibonacci)的程式，程式必須輸出從 0 開始數 N 個的所有費波那契數，N 必須為使用者輸入。範例：

```
Input N:10
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
```

寫完請助教 Demo 通過即可。

F. CLASS AND OBJECT

C#為一個完全物件導向的語言，其中除了 **Object** 類別(class)以外的所有類別都繼承自 **Object**。而其類別宣告方式、繼承方式等等皆與 C++大同小異，接下來我們會利用一個小程序來看 C#中 OO 的實作方法與特性。

下列範例程式會輸出：

吳小華是女的

馬小九是男的

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Example6_ClassAndObject
{
    class Program
    {
        static void Main(string[] args)
        {
            Person[] friends = new Person[2];
            friends[0] = new Woman("吳小華");
            friends[1] = new Man("馬小九");
            foreach (Person person in friends)//將陣列或容器內所有元素依序存取一次
            {
                printPersonStatus(person);
            }
            static void printPersonStatus(Person argPerson)
            {
                Console.Out.WriteLine(argPerson.name + "是" + argPerson.getGender() + "的");
            }
        }
        abstract class Person//抽象化類別必須加上abstract修飾詞
        {
            public string name { get; private set; };//利用get跟set符管理存取權限 且 set符合會自動重載operator=
            protected Person(string argName)
            {
                name = argName;
            }
            abstract public string getGender();
        }
        class Woman : Person
        {
            public Woman(string argName): base(argName)//利用base()呼叫父類別之建構式
            {
            }
            public override string getGender();//重載函式必須使用override修飾詞
            {
                return "女";
            }
        }
        class Man : Person
        {
            public Man(string argName) : base(argName)
            {
            }
            public override string getGender()
            {
                return "男";
            }
        }
    }
}
```

EXERCISE 6.

利用上述所學將範例程式加入身高、體重、愛吃的東西等等(至少三樣)因素，並多一個欄位為「伴侶」，型別為 **Person**。在 **Main** 中新增一個長度為 6 的 **Person Array**，自己定義其中內容並顯示他們的資訊(至少要有兩對伴侶)。(提示：設定伴侶可利用 **Static member function**) 寫完請助教 Demo 通過即可。

G. CONTAINERS

C#支援許多主流的 Containers。在此，我們只列出目前較常用的 Containers，若同學有興趣可以上 [http://msdn.microsoft.com/en-us/library/gg145035\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/gg145035(v=vs.110).aspx) 上面有各個 Containers 的用法及特性。而最基礎的 List, Queue, Stack 等架構也希望同學可以熟練，在之後的課程中會大量地使用到。

一般型別	泛型型別	說明
ArrayList	List< >	串列，以陣列實作的串列結構
Queue	Queue< >	佇列，先進先出的結構
Stack	Stack< >	堆疊，後進先出的結構
Hashtable	Dictionary< >	雜湊表格，快速用 key 查找 value
SortedList	SortedList< >	排序串列，使用排序與二分搜尋法的結構
ListDictionary	Dictionary< >	字典，快速用 key 查找 value
HybridDictionary	Dictionary< >	小集合採用 ListDictionary，集合變大時，會自動改用 Hashtable 的一種字典
OrderedDictionary	Dictionary< >	比 SortedList 類別多了一些功能的類別
SortedList	SortedList< >	鍵值一定是字串，用法跟 Hashtable 相似
NameValueCollection	Dictionary< >	NameValueCollection 可以單一索引鍵對應多重值
DictionaryEntry	KeyValuePair< >	字典中的一個項目，(Key, Value) 的結構
StringCollection	List<String>	用法跟 ArrayList 相似
StringDictionary	Dictionary<String>	字串字典

而經過中斷點的設立，可以即時監看 Container 內的值，希望同學多加熟悉利用。

```
static void Main(string[] args)
{
    List<string> aList = new List<string>();
    for (int i = 0; i < 10; i++)
    {
        aList.Add("第" + i + "號位置");
    }
}
```

即時監看變數值

名稱	值	型別
args	{string[0]}	string[]
aList	Count = 10	System.Collections.Generic.List<string>
[0]	"第0號位置"	string
[1]	"第1號位置"	string
[2]	"第2號位置"	string
[3]	"第3號位置"	string
[4]	"第4號位置"	string
[5]	"第5號位置"	string
[6]	"第6號位置"	string
[7]	"第7號位置"	string
[8]	"第8號位置"	string
[9]	"第9號位置"	string

H. BASIC THREAD

C#原生支援執行緒(而 C++11 後才原生支援) · 在 Thread 的使用上不論是網路應用程式或是視窗應用程式都是不可或缺的一部份。在網路應用程式的部分必須要使用 Thread 才能在等待網路的資料的時候同時進行其他工作；而在視窗應用程式中，若是將高計算負載的工作或是需要等待的工作放在主執行緒中，則會造成畫面的停頓(也就是俗稱的 LAG)。我們將在下面以程式範例進行 C# Thread 的基礎介紹：

//Bank.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace Example8_Thread
{
    class Bank
    {
        public static int money=0;
        public static object moneyLock = new object();

        static void Main(string[] args)
        {
            List<Action> actionList=new List<Action>();
            actionList.Add(new Deposit(3000, 200)); //每3秒存200塊
            actionList.Add(new Deposit(4000, 800)); //每4秒存800塊
            actionList.Add(new Deposit(1000, 50)); //每1秒存50塊
            actionList.Add(new Draw(2000, 100)); //每2秒領(借)100塊
            actionList.Add(new Draw(12000, 2000)); //每12秒領(借)2000

            foreach (Action action in actionList)
            {
                new Thread(action.Start).Start(); //新增執行緒並開始執行
            }
        }
    }
}
```

//Part of Action.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace Example8_Thread
{
    abstract class Action
    {
        protected int frequency { get; private set; }
        protected int amount { get; private set; }
        protected Action(int argFrequency, int argAmount)
        {
            frequency = argFrequency;
            amount = argAmount;
        }
        abstract public void Start();
    }

    class Deposit : Action
    {
        public Deposit(int argFrequency, int argAmount)
            : base(argFrequency, argAmount)
        {
        }
        public override void Start()
        {
            while (true)
            {
                lock (Bank.moneyLock) //避免造成同步問題
                {
                    Bank.money += amount;
                    Console.WriteLine("$" + amount + " is deposited. Now there is " + Bank.money);
                }
                Thread.Sleep(frequency);
            }
        }
    }
}
```

I. IDISPOSIBLE & USING

C#在 GC 方面有著與 JAVA 相近的機制，但對於 IO (File, Socket)之類的物件若無法及時解構的話可能會造成後續程式錯誤的結果。因此 C#定義了 IDisposable(一次性的)介面來解決這個問題。並配合 using 陳述式，來達成快速解構的目標。

J. LINQ & IENUMERABLE

Language-Integrated Query (LINQ) 是 Visual Studio 2008 和 .NET Framework 3.5 版中引進的創新技術，用來填補物件與資料之間的缺口。

傳統上，資料查詢是以簡單的字串表示，既不會在編譯時期進行型別檢查，也不支援 IntelliSense。此外，您還必須針對每種資料來源學習不同的查詢語言：SQL 資料庫、XML 文件、各種 Web 服務等等。LINQ 將「查詢」(Query) 變成 C# 和 Visual Basic 中第一級的語言建構。只要使用語言關鍵字和您熟悉的運算子，就可以針對強型別的物件集合撰寫查詢。

```
public class Program
{
    static void Main(string[] args)
    {
        var testData = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 };
        using (StreamWriter sw = new StreamWriter(new FileStream("out.txt", FileMode.OpenOrCreate, FileAccess.Write)))
        {
            var odd = from number in testData
                      where number % 2 == 1
                      orderby number ascending
                      select number;

            foreach(var n in odd)
            {
                sw.WriteLine(n);
            }
        }
    }
}
```

EXERCISE.7

請利用政府公開平台資訊(新竹市政府)，並使用 IDisposable 介面與 LINQ 建立新竹市的 Wifi 查找系統

資料在 <http://opendata.hccg.gov.tw/dataset/056f201e-b4b8-403e-985d-bcb4f9f13b3f/resource/56857996-3f5a-4a39-b210-56bde7dcabb2/download/20160509143251980.csv>

Input: 東區,香山區...etc

Output: Wifi 名稱 & 地址

HW 1. 現有工具研讀

為了讓各位同學在未來實作專題的時候不用每個步驟都從頭打造，請各位同學(三人一組)回去各自研讀且使用以下項目，並且於溫老師課程的最後一周繳交報告至 E3，內容包含但不限於：如何使用、使用規範、最小可行性程式碼(類似 Hello World)、你的實作(小專題程式碼與解說)

1. ASP.NET MVC
2. Universal Windows Platform (須至少有一人電腦為 Win10)
3. Nancy Module (<http://nancyfx.org/>)
4. Django
5. Xamarin
6. Mongo DB
7. Rails
8. MS Bot framework (<https://dev.botframework.com/>)
9. Google translation API (包含發音)
10. BlockChain

採用霍格華茲評分標準、分為 O(Outstanding - 100)、E(Exceeds Expectations - 95)、

A(Acceptable - 85)、P(Poor - 75)、D(Dreadful - 70)、T(Troll - 0)六級，請各位同學加油

附上 TA 的聯絡方式 FB: www.facebook.com/behigheveryday

有任何問題可以直接密我比較快速