**2021/03/24 C# 測驗,請關閉通訊軟體 ,不開放上網查資料 , 作答完成後將檔案壓縮(zip), 寄出給 Hendry及組長**

1. Linq(共50%)
2. 實作LINQ (使用join、orderby、where 將兩張表join，查詢條件為className等於TWO，排序條件為StudentID反序列，輸出className跟StudentName欄位)(10%)

List<clsClass> cls = new List<clsClass>();

cls.Add(new clsClass("1", "ONE"));

cls.Add(new clsClass("2", "TWO"));

cls.Add(new clsClass("3", "THREE"));

List<clsStudent> stu = new List<clsStudent>();

stu.Add(new clsStudent("1", "ALEX", "006"));

stu.Add(new clsStudent("1", "JOHN", "005"));

stu.Add(new clsStudent("2", "WANG", "004"));

stu.Add(new clsStudent("2", "LEE", "003"));

stu.Add(new clsStudent("3", "LIN", "002"));

stu.Add(new clsStudent("3", "JAY", "001"));

1. selectmany，下面是一個商店的物件，有兩間商店seveneleven跟FamilyMart，使用selectmany找出所有的產品。(10%)

class Store

{

public string Name { get; set; }

public string[] Products { get; set; }

}

Store[] stores = new Store[]

{

new Store()

{

Name = "SevenEleven",

Products = new string[] {"pen", "candy", "bread"}

},

new Store()

{

Name = " FamilyMart",

Products = new string[] {"Biscuits", "Drink"}

}

};

1. 傳入值相同時，這兩個函式輸出的結果為何? (10%)

static void Main(string[] args)

{

List<Person> data = new List<Person>();

data.Add(new Person("40", "a"));

data.Add(new Person("30", "b"));

data.Add(new Person("30", "c"));

useorderby(data);

useorderbythenby(data);

}

static void useorderby(List<Person> data)

{

var item = data.OrderBy(a => a.age).OrderBy(a => a.name).FirstOrDefault();

Console.WriteLine($"{item.name}, Age:{item.age}");

}

static void useorderbythenby(List<Person> data)

{

var item = data.OrderBy(a => a.age).ThenBy(a => a.name).FirstOrDefault();

Console.WriteLine($"{item.name}, Age:{item.age}");

}

public class Person

{

public string age;

public string name;

public Person(string age, string name)

{

this.age = age;

this.name = name;

}

}

1. 實作linq，去除age欄位重複資料(10%)

public class Person

{

public string age;

public string name;

public Person(string age, string name)

{

this.age = age;

this.name = name;

}

}

List<Person> dataA = new List<Person>();

dataA.Add(new Person(40, "a"));

dataA.Add(new Person(30, "b"));

dataA.Add(new Person(30, "c"));

dataA.Add(new Person(50, "d"));

1. 請寫出輸出結果(10%)

private static void LinqExceptC()

{

List<string> oldList = new List<string>(new string[] { "1", "2", "3" });

List<string> newList = new List<string>(new string[] { "2", "4", "6" });

var diffList = newList.Except(oldList);

foreach (var item in diffList)

{

Console.WriteLine($"number:{item} ");

}

}

1. 請寫一個擴充方法，計算出兩個日期的差異天數(20%)

void Main()

{

var day1 = new DateTime(2019, 02, 12);

var day2 = new DateTime(2019, 02, 17);

var aaa =day2.AAA(day1);

Console.Write(aaa);

}

public static class MyClass

{

// Define other methods and classes here

public static int AAA(this DateTime test,DateTime test2)

{

return (test - test2).Days;

}

1. }使用dictionary，建立資料、移除資料、計算數量(10%)

void Main()

{

var MyDic = new Dictionary<string, string>( );

MyDic.Add( "Name", "Jack" );

MyDic.Add( "Blog", "Jack’s Blog" );

MyDic.Add( "Group", "KTV Group" );

MyDic.Remove( "Group");

Console.Write(MyDic.Count());

Console.Write('\n');

foreach(var obj in MyDic)

{

Console.Write(obj.Key+'\n');

}

}

1. 下列有 Fly 類別,請使用繼承方式設計 AirPlane 類別,呼叫結果請參考下圖(20%)

public class Fly

{

public int speed;

public string initial;

public Fly()

{

initial = "fly類別初始化";

Console.WriteLine(initial);

}

public Fly(string name)

{

initial = initial + name;

Console.WriteLine(initial);

}

public virtual void flyToString()

{

Console.WriteLine("這是fly原本的string方法");

}

}

呼叫代碼 (不可變更)

AirPlane c = new AirPlane();

c.type = "Jstar";

c.name = "kiki";

c.speed = 300;

c.flyToString();

c.Info();

