一、C#

1. API 回傳以下金額資訊，請您將金額 \* 利率0.33後，由大到小進行排序。 ( - 視為 0 )

string[] amountList = { "1.2", "1.4", "0.2", "-", "-0.005" };

見CSharpQ1.cs

1. 承上題，請實作單元測試

見UnitTest1.cs

1. 客戶傳入信用卡號，請進行信用卡卡號隱碼，信用卡號分為16碼與12碼隱碼規則 : 僅顯示末4碼，其餘以\*隱碼，並每四位以 - 分隔開Example :

0123456789012345 -> \*\*\*\*-\*\*\*\*-\*\*\*\*-2345

012345678901 -> \*\*\*\*-\*\*\*\*-8901

見CSharpQ3.cs

1. 請用泛型實作一個function滿足以下需求:

* 輸入參數型別可為 String、Int、DateTime…
* 輸出結果為字串型別
* 當輸入型別為DateTime時，輸出結果須轉為”xxxx/xx/xx”格式之年月日字串

例如

輸入數字888輸出字串型別"888",

輸入DateTime 2022/6/2 12:05:33輸出字串型別”2022/06/20”

見CSharpQ4.cs

1. 什麼是Linq的延遲執行? 何時會觸發實際執行?

LINQ 語法在被建立時並不會立即執行，而是等到需要計算最後結果時才執行以提高效率。當你呼叫 ToList()、ToArray()、ToDictionary()，或使用了需立即執行的方法(Count()、Sum()、First()…)將立即執行。

1. 使用繼承時有哪些相關的修飾詞? 請列舉及說明。

Public對任何使用者公開

Protected宣告類及子類可使用

Private只有宣告類可使用

Internal只有同assembly可使用

Virtual子類可覆寫

1. 請解釋async await是什麼? 有什麼優缺點?

async await為異步調用修飾詞，執行緒遇到await時不會阻塞等到await語句完成

優點為提高程式回應速度，降低效能消耗並提高可讀性(不須處理callback)

缺點為如果在異部調用區塊內使用同步程式碼可能造成死鎖

1. 下列程式的主控台輸出會是?

public class B0

{

public virtual string GetText()

{

return "Box0";

}

}

public abstract class B1 : B0

{

public override string GetText()

{

return "Box1";

}

}

public class B2 : B1

{

public override string GetText()

{

return "Box2";

}

}

static class Program

{

[STAThread]

static void Main()

{

B0 box0 = null;

B1 box1 = new B2();

box0 = box1;

string result = box0.GetText();

Console.WriteLine(result);

}

}

Box2

二、DB

1. 請使用SQL回答問題，DB中有以下三個Table

Students [ ClassID , StdID , StdName ]

Examiner [ ClassID , ExerID , ExerName ]

Exam [ ExamType , ExamSubject , StdID , Score ]

 列出每個班級的主考官姓名(ExaminerName)與學生人數

SELECT e.ClassID, e.ExerName AS ExaminerName, COUNT(s.StdID) AS StudentCount

FROM Examiner e

LEFT JOIN Students s ON e.ClassID = s.ClassID

GROUP BY e.ClassID, e.ExerName;

 列出ExamType為「FinalExam」的考試，每班所有科目分數相加後，

總分最高的學生姓名

WITH TotalScores AS (

SELECT e.ClassID, e.StdID, SUM(Score) AS TotalScore

FROM Exam e

WHERE e.ExamType = 'FinalExam'

GROUP BY e.ClassID, e.StdID

)

SELECT s.StdName

FROM TotalScores t

INNER JOIN Students s ON t.StdID = s.StdID

WHERE t.TotalScore = (

SELECT MAX(TotalScore)

FROM TotalScores

WHERE ClassID = t.ClassID

);

1. 使用預存程序有什麼好處跟缺點? 使用ORM有什麼好處跟缺點?

預存程序好處為提高效能、安全性與可重用，缺點為不易除錯

ORM的優點是自動生成對應物件與/或SQL語法提高開發效率，同時可以對應多種後端資料庫，缺點是有性能要求且因底層SQL被隱藏不易除錯

1. 設計資料表時, 你會注意到那些事項? 請列舉並解釋你的考量點。

是否可能阻斷，如果有阻斷就有可能死鎖

是否能正規化減少重複資料

索引與主鍵設置是否得宜

Where條件是否皆落於索引與主鍵上

欄位格式(型別、大小、範圍)與命名是否符合需求

是否可NULL，如不可預設值是什麼

1. 如果你需要調校SQL陳述式，而執行計畫顯示Table Scan是造成執行緩慢的原因，那你會怎麼做?

Where條件是否皆落於索引與主鍵上，是否須重新設計或增加索引

是否需進行資料表分割

1. 有一個Table定義如下：

CREATE TABLE [dbo].[User](

[UserID] [int] NOT NULL,

[UserName] [varchar](100) NULL,

[DateCreated] [datetime2](7) NULL

)

CREATE NONCLUSTERED INDEX IX\_User\_UserName ON [dbo].[User]

([UserName])

下列哪一段語法能有效使用Index Seek?

* 1. SELECT UserName FROM [dbo].[User] WHERE UserName Like 'min%'
  2. SELECT UserName FROM [dbo].[User] WHERE LEFT(UserName,3) ='min'
  3. SELECT UserName FROM [dbo].[User] WHERE UserName like '%min'
  4. SELECT UserName FROM [dbo].[User] WHERE UserName like '%min%'
  5. SELECT UserName FROM [dbo].[User] WHERE Upper(UserName) like 'MIN%'

B

三、ASP.NET

1. ASP.NET MVC中的Filter是什麼? 請試著列舉你所知道的Filter，並舉例說明有什麼用處

依執行順序排列(都有設定的話)

Global Filter：註冊在應用程式啟動時，所有action受影響

Authorization Filter：判定使用者有無權限訪問指定的action或controller

Action Filter：用於在執行指定action之前或之後執行特定的邏輯

Result Filter：用於修改或擴展action返回的結果。在action執行後，結果返回前執行。

Exception Filter：用於修改、擴展action執行期間發生的例外。在action中例外發生時執行。

1. 如果要在ASP.NET MVC或Web API中記下每個request的log資訊，你會怎麼做? 同樣的機制在ASP.NET Core怎麼做?

MVC或API用action filter

Core用middleware

1. 網站實做快取時有哪幾種方式, 請列舉及說明

Output Cache:快取整個或部分網頁(如腳本、圖片、樣式表)於記憶體。指定時間內返回相同結果

Data Cache: 快取資料庫或API回傳資料於記憶體，可用MemoryCache類實現

Client-Side Cache: Cookie, Local storage, Session storage…

Distributed Cache: 使多個微服務或伺服器可以共享同一份快取，如Redis

1. 在 ASP.NET Core 中，原本 ASP.NET 的程式進入點 Application\_Start() 由什麼取代?

Startup.ConfigureService()

1. 我們可以將一個 WebAPI 的 Request/Response 生命周期簡化為下圖，請依序在空格中填入適當答案。



Answer 1：ActionFilter

Answer 2：MessageHandler

1. 我們可以將一個 ASP.NET MVC 的 Request/Response 生命周期簡化為下圖，請依序在空格中填入適當答案。



Answer 1：Routing

Answer 2：Controller Initialization

Answer 3：Action Filters

Answer 4：Action Result

Answer 5：View Engine