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" };

public class Answers

{

public static List<double> GetAnswer1(List<string> stringList, double interestRate)

{

return stringList

.Select(amount => amount == "-" ? 0 : Math.Round(Convert.ToDouble(amount) \* interestRate, 2))

.OrderByDescending(amount => amount)

.ToList();

}

}

1. 承上題，請實作單元測試

[Fact]

public void Test1()

{

var amountList = new List<string> { "1.2", "1.4", "0.2", "-", "-0.005" };

var interestRate = 0.33;

var expected = new List<double> { 0.46, 0.4, 0.07, 0, 0 };

var result = Answers.GetAnswer1(amountList, interestRate);

Assert.Equal(expected, result);

}

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

public static string GetAnswer3(string cardNumber)

{

string result = "Invalid";

if (cardNumber.Length == 12)

{

result = $"\*\*\*\*-\*\*\*\*-{cardNumber.Substring(8)}";

}

else if (cardNumber.Length == 16)

{

result = $"\*\*\*\*-\*\*\*\*-\*\*\*\*-{cardNumber.Substring(12)}";

}

return result;

}

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”

public static string GetAnswer4<T>(T input)

{

try

{

return input is DateTime ? ((DateTime)(object)input).ToString("yyyy/MM/dd") : $"{input}";

}

catch (Exception ex)

{

return string.Empty;

}

}

1. 什麼是Linq的延遲執行? 何時會觸發實際執行?

Linq延遲執行只有在需要結果的時候才執行，例如foreach或.tToList()，在Entity Framework中當我們實際去訪問導航屬性的時候才會去加載也是相同的概念。

1. 使用繼承時有哪些相關的修飾詞? 請列舉及說明。

Private: 只有自己可以存取  
protectd: 自己或衍伸的類別可以存取  
internal: 專案內可存取  
protected internal: 專案內“或”衍伸類別可存取

Public: 公開

1. 請解釋async await是什麼? 有什麼優缺點?

當method 有使用await便需要定義為async function, 回傳Task or Task<T>。

當程式執行到await時會先將控制權交還給呼叫端，直到await的結果回傳後繼續執行後續的程式碼。

優點：提升性能，因為可以阻塞的需要長時間的操作例例如I/O, 舉例來說當我們呼叫其他API或是等待資料庫時可以先繼續執行其他動作，等結果回傳之後再繼續對應的動作。  
缺點：程式碼相對複雜。有淺在死結的風險，另外如果使用不當容易造成過度大量的資源損耗

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);

}

}

Answer: “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, COUNT(s.StdID) AS StudentCount

FROM Examiner e

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 RankedStudents AS (

SELECT S.ClassID, S.StdName, SUM(E.Score) as TotalScore,

DENSE\_RANK() OVER (PARTITION BY S.ClassID ORDER BY SUM(E.Score) DESC) as rank

FROM Exam E

JOIN Students S ON E.StdID = S.StdID

WHERE E.ExamType = 'FinalExam'

GROUP BY S.ClassID, S.StdID, S.StdName

)

SELECT ClassID, StdName, TotalScore

FROM RankedStudents

WHERE rank = 1

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?

ＳＰ

優點：

1. 可重複使用，在多個應用中重複使用保持一制性
2. 安全，因為可以最小化sql注入攻擊的風險

缺點：

1. 過多的商業邏輯可能會使Sp難以維護
2. 無法版本控制，誰修改了哪些地方難以追蹤

ＯＲＭ

優點：

1. ORM提供了更高級的抽象，讓開發者可以用類似物件導向的方式來跟資料庫互動
2. 可移植性高
3. 可讀性更高

缺點：

1. 性能不如手寫sql
2. 對於非常複雜的sql操作有時候ORM難以做到
3. 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。
4. 首先會先仔細思考table的設計要能與實際的應用場景吻合
5. 個人習慣每張表的pk會用auto increasement的方式，但是會根據真實狀況設立unique constraint
6. 根據表之間的關聯建立fk
7. 依據實際狀況來建立index
8. 正規劃，盡量減少重複的欄位來保持資料的一致性
9. 某些欄位是否允許null或是設定預設值要考慮清楚
10. 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?
11. 根據狀況增加index
12. 檢視sql語法重寫查詢，看能否更有效率的join或是減少不必要的column
13. 考慮建立一些view
14. 有時候會再利用程式碼做一些暫存減少打DB的次數
15. 最後是看有沒有可能在硬體上做加強
16. 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%'

Answer: 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

簡單的說就是一種middleware, 常見的有

Authorization filter: 檢查權限

Actoin filter: 可以在Action執行前、後做一些處理

Exception filter: 捕捉例外，執行一些自訂的例外處理

Result filter: 返回結果之前、後進行一些動作

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?

.net mvc的話可以自訂一個action filter, 在OnActionExecuting or OnActionExecuted中紀錄log

.net core的話可以自訂義一個middleware然後在Startup.cs的Configure註冊

app.UseMiddleware<MyLogMiddleware>();

1. 網站實做快取時有哪幾種方式, 請列舉及說明
2. OutputCache or ResponseCache
3. MemoryCache or Redis
4. CDN節點，可以將一些靜態資源暫存
5. 在 ASP.NET Core 中，原本 ASP.NET 的程式進入點 Application\_Start() 由什麼取代?

由Startup 裡面的Configure() and ConfigureServices()取代

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：Action

Answer 2：Routing

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：Rouing

Answer 2：Controller Initialization

Answer 3：Action filter

Answer 4：Action result

Answer 5：View engine