

您正在建立用以儲存測驗結果的應用程式。

您需要記錄每個答案正確或不正確。

您還需要將儲存資料所需的記憶體數量減到最少。

您應該使用哪一種資料型別？

Explanation:

題目解析 boolean在C#中是1位元組 double是雙精度浮點數，通常需要8個位元組 integer是整數，通常4位元組 string是字串型別，佔用的記憶體更大。

你建立了一個應用程式，其中包含一個名為 Red 的函式。

Red 會呼叫一個名為 Blue 的函式。

在 Blue 執行完成之後，編譯器如何返回程式碼路徑中的正確位置？

Explanation:

1. 函式呼叫與堆疊的關係 在程式中，當一個函式（如 Blue）被另一個函式（如 Red）呼叫時，會在堆疊（Stack）上建立一個「執行上下文」（Execution Context），包含：2. 呼叫者（Red）的返回地址（即執行完 Blue 後要返回的程式位置）3. Blue 的區域性變數、引數等資料 4. 其他執行所需的元資料 堆疊是一種後進先出（LIFO）的記憶體結構，用於管理函式呼叫的巢狀層級。1. 執行完 Blue 後的處理 2. 執行資料清除：當 Blue 執行完成後，其佔用的堆疊空間會被彈出(Pop)，釋放記憶體。3. 最後一個指示保留在頂端：堆疊頂端（Top of Stack）會保留最後一個未完成的函式呼叫的返回地址，以便執行完 Blue 後能正確返回到 Red 的執行位置。

你執行了下列程式碼：

```
int a = 10;
int b = 20;
int c = 30;
int result = 0;
if (a <= b || c > a)
    result = 10;
else if (a <= b || c <= a)
    result = 20;
else
    result = 30;
```

程式碼執行完畢之後，result 的值為何？

Explanation:

你執行了下列程式碼：

```
int a = 10;
int b = 20;
int c = 30;
int result = 0;
if (a <= b && c <= a)
    result = 10;
else if (a <= b && b >= c)
    result = 20;
```

```
else
```

```
result = 30;
```

程式碼執行完畢之後， result 的值為何？

Explanation:

在例外狀況處理常式中， catch 區段的用途為何？

Explanation:

有一個網頁顯示了 clock 元素。

你需要完成 JavaScript 程式碼以每秒更新 clock 元素一次。

請選取正確的選項以完成程式碼：

```
_ (Q1) _ _ (Q2) _ _ (Q3) _ ;  
function updateClock() {  
let d = new Date();  
  
document.getElementById("clock").innerHTML =  
d.getHours() + ":" +  
d.getMinutes() + ":" +  
  
d.getSeconds();  
}
```

Q1該寫入？

Explanation:

題目核心需求：題目要求「每秒更新 clock 元素一次」，JavaScript 中處理「定期執行」 正確答案：B. setInterval(//因為 setInterval() 是唯一能實現「每秒重複執行」的語法，符合題目需求。

有一個網頁顯示了 clock 元素。

你需要完成 JavaScript 程式碼以每秒更新 clock 元素一次。

請選取正確的選項以完成程式碼：

```
_ (Q1) _ _ (Q2) _ _ (Q3) _ ;  
function updateClock() {  
let d = new Date();  
  
document.getElementById("clock").innerHTML =  
d.getHours() + ":" +  
d.getMinutes() + ":" +  
  
d.getSeconds();  
}
```

Q2該寫入?

Explanation:

題目核心要點 1. 語法結構: `setInterval(Function, Interval)`, 其中: Function 是函式的引用 (不加括號) 。 2. Interval 是以毫秒為單位的時間 (1000 毫秒 = 1 秒) 。

有一個網頁顯示了 clock 元素。

你需要完成 JavaScript 程式碼以每秒更新 clock 元素一次。

請選取正確的選項以完成程式碼:

```
_ (Q1) _ _ (Q2) _ _ (Q3) _ ;
function updateClock() {
let d = new Date();

document.getElementById("clock").innerHTML =
d.getHours() + ":" +
d.getMinutes() + ":" +
d.getSeconds();
}
```

Q3該寫入?

Explanation:

`setInterval`的引數是毫秒, 1000毫秒 = 1 秒 符合題目要求

您撰寫了以下這段程式碼:

```
public delegate void Notification();
public Notification DelegateA;
DelegateA = MethodA;
DelegateA += MethodA;
public delegate void Notification();
public event Notification EventA;
EventA = EventHandlerA;
EventA += EventHandlerA;
```

請問哪一個陳述式會讓程式無法正確執行?

Explanation:

題目解析重點 1.DelegateA是一個普通的委託, 而EventA是一個事件。 2.delegate 欄位可以直接賦值 (=) 或用 +=/-= 來加減委派方法。 3.event 在外部只能用 += 或 -= 來訂閱/取消事件, 不能直接用 = 指派事件處理器。 4.=只允許在事件宣告類別的內部使用 (或在事件自訂 add/remove accessor 時) 。

您正在將軟體規格轉換為程式碼元件。

規格需要無法具現化的類別, 但是可以指定必須在實體繼承類別中實作的方法。

您應該建置哪一種類別?

Explanation:

抽象類別 (abstract class) 無法被直接具現化 (不能用 new 關鍵字建立例項)。
它可以用來定義抽象方法，這些方法必須由繼承自該抽象類別的非抽象子類別來實作。
這與題目的需求完全吻合。sealed 類別則相反，它無法被繼承。

有一個堆疊包含整數值，這些值會按照下列順序推入堆疊：2、4、6、8。

您執行了下列作業序列：

Pop

Push 3

Pop

Push 4

Push 6

Push 7

Pop

Pop

Pop

這些作業執行完畢之後，最上層元素的值為何？

Explanation:

依照堆疊操作後，最上層依序彈出 8 → 3 → 7 → 6，剩下的是 2、4、6，其中 6 為最上層。

堆疊是後進先出 (LIFO) 的資料結構。初始堆疊 (Stack) : [2, 4, 6, 8] (8在最上面)。Pop: 移除 8。堆疊: [2, 4, 6]。Push 3: 堆疊: [2, 4, 6, 3] (3在最上面)。Pop: 移除 3。堆疊: [2, 4, 6]。Push 4: 堆疊: [2, 4, 6, 4]。Push 6: 堆疊: [2, 4, 6, 4, 6]。Push 7: 堆疊: [2, 4, 6, 4, 6, 7]。Pop: 移除 7。堆疊: [2, 4, 6, 4, 6]。Pop: 移除 6。堆疊: [2, 4, 6, 4]。Pop: 移除 4。堆疊: [2, 4, 6]。最後，最上層的元素是 6。

您任職於某個應用程式開發團隊。應用程式存放於 Git 檔控制系統中。您計劃將一項新功能加入至應用程式，這項功能需要數週的時間來完成。其他團隊成員所進行的應用程式維護作業不得影響您的工作。請問您該怎麼做？請將清單中適當的詞彙移至正確位置：

詞彙：

1. 合併 (merge)
2. 提取要求 (pull request)
3. 推送 (push)
4. 分支 (Branch)
5. 認可 (commit)
6. 複製

您應該先建立原始程式碼的 A。

當您在本機工作並確認變更時，需要執行 B，如此您的變更才能儲存在本機儲存庫中。

然後，您應該執行 C，以將您的變更儲存在遠端儲存庫中。

為了通知其他開發人員您所做的變更，您應該建立 D。

在您的功能完成之後，其他開發人員應該執行 E。

Explanation:

依照堆疊操作後，正確的順序應該是： 1. 建立分支 (4) //避免影響主程式碼 2. 認可 (5)

//在本機完成修改並儲存變更 3. 推送 (3) //將本機變更送到遠端儲存庫 4. 提取要求 (2) //通知其他人你的變更
5. 合併 (1) //完成後由其他人將變更合併至主分支

請將清單中的每個版本控制系統命令，移至正確的描述位置：

命令：

1. branch
2. checkout
3. commit
4. merge
5. rollback

描述：

- A. 儲存新的專案快照
 - B. 管理開發程式碼行
 - C. 更新工作專案的全部或其中一部分
 - D. 將您的本機專案還原回先前的狀態
 - E. 將另一個專案的內容匯入目前的專案
- 哪一個選項的順序正確？

Explanation:

您正在為應用程式設計一個 C# 類別。

您需要將 accessCount 成員變數的可用性限制為基底類別以及衍生自基底類別的任何類別。

請問您應該使用哪個存取修飾詞？

Explanation:

哪一個詞能描述將物件的實值型別轉換成參考型別的程式？

Explanation:

您有一個具有屬性的類別。

您需要確保該類別的取用者可以寫入屬性的值。

請問您應該使用哪一個關鍵字？

Explanation:

哪一個詞能描述自現有類別繼承功能的 C# 類別？

Explanation:

類別 C 和類別 D 繼承自類別 B。

類別 B 繼承自類別 A。

這些類別的方法如下所示：

所有類別都有受保護的範圍。

類別 C 可以存取哪些方法？

Explanation:

哪一個詞能描述將物件轉換成較一般型別的程式？

Explanation:

衍生類別可以覆寫哪一種函式？

Explanation:

下列哪一項敘述正確描述了在基底類別中宣告為虛擬的方法？

Explanation:

您正在撰寫一個儲存及顯示各種賽跑結果的應用程式。此應用程式包含下列定義：

```
public interface IDisplayResult {  
    void Display();  
}  
  
public class Race {  
    public int Seconds;  
    public virtual void Display() {  
        Console.WriteLine(Seconds);  
    }  
}
```

下列程式碼會顯示競賽的結果：

```
var r1 = new FootRace();  
var r2 = r1 as IDisplayResult;  
r1.Seconds = 99;  
r1.Display();  
r2.Display();  
r1.DisplayRaw();
```

此程式碼必須輸出下列結果至主控臺：

99 seconds
1.65 minutes
99

請完成 FootRace 類別的實作，將下列宣告依適當順序放入程式碼中：
宣告：

(1)public override void Display(){
(2)public void DisplayRaw() {
(3)void IDisplayResult.Display() {

(4)private void Display()

作答區：

```
public class FootRace : Race, IDisplayResult  
{
```

_ (A) _

```
Console.WriteLine(Seconds + " seconds");  
}
```

_ (B) _

```
base.Display();  
}
```

_ (C) _

```
Console.WriteLine(Seconds / 60d + " minutes");  
}  
}
```

Explanation:

您需要將所有樣式設定分組成個別的檔案，以便套用至 Web 應用程式的所有頁面。請問您該怎麼做？

Explanation:

您正在開發一個 Web

程式，您需要使用階層式樣式表（CSS）建立上列圖形：綠色漸層至黑色的矩形。
請選擇正確的選項以完成句子。

對於 background-image 樣式，您應該使用此值：

Explanation:

您正在開發一個 Web 程式，您需要使用階層式樣式表（CSS）建立上列圖形：

請選擇正確的選項以完成句子。

對於 background-image 樣式，您應該使用此值：

Explanation:

您正在建立一個 Web 應用程式。即將存取該應用程式的使用者端電腦會執行各種網頁瀏覽器。哪一個詞能描述讓應用程式可供使用者端電腦存取的程式？

Explanation:

請選取正確的選項以完成句子。所有網頁伺服器都有__位址和網域名稱，用以在網際網路上識別其身分。

Explanation:

請選取正確的選項以完成句子。所有網頁伺服器會經由__通訊協定接受使用者端要求，然後以網頁的形式將資料傳回使用者端來回應。

Explanation:

請選取正確的選項以完成句子。擁有存取權的使用者端可以經由__通訊協定，修訂或移除裝載在網頁伺服器上的內容。

Explanation:

哪一個程式會將資料轉換為 Web 服務所用的 XML 字串？

Explanation:

請選取正確的選項以完成圖表。請在圖表中指出 MVC 架構模式的正確元件對應順序。

Explanation:

請完成圖表以解說 MVC 架構的核心功能。請將每個動作移至正確的位置。

1. 將事件轉換成要求
2. 顯示資料
3. 解譯並傳遞資料
4. 使用者點選按鈕
5. 驗證要求、儲存資料

Explanation:

您正在檢閱一項資料庫設計。此設計的其中一部分如下圖所示。哪一個詞能描述 Customer 與 Order 之間的關聯性？

Explanation:

有一個名為 Items 的資料表，其欄位包括：

- ID (integer, 主索引鍵, 自動產生)
- Description (text)
- Completed (Boolean)

您只需要刪除 Completed 欄位設為 True 的資料列。請問您應該使用哪一個陳述式？

Explanation:

請設定交易以符合下列需求：

- 如果客戶只有一張發票，請刪除該發票。
- 如果客戶有多張發票，請取消交易。

請問 A 部分應填入下列何者程式碼？

-- A 部分

```
DECLARE @invoice_count INT;
```

```
SELECT @invoice_count = COUNT(*)  
FROM Invoices
```

```
WHERE CustomerID = 123;
```

```
IF @invoice_count = 1  
BEGIN  
DELETE FROM Invoices
```

```
WHERE CustomerID = 123;
```

```
-- C 部分  
END  
ELSE  
BEGIN  
-- B 部分  
END
```

Explanation:

請設定交易以符合下列需求：

- 如果客戶只有一張發票，請刪除該發票。
- 如果客戶有多張發票，請取消交易。

請問 B 部分應填入下列何者程式碼？

-- A 部分

```
DECLARE @invoice_count INT;
```

```
SELECT @invoice_count = COUNT(*)  
FROM Invoices
```

```
WHERE CustomerID = 123;
```

```
IF @invoice_count = 1  
BEGIN  
DELETE FROM Invoices
```

```
WHERE CustomerID = 123;
```

```
-- C 部分  
END  
ELSE  
BEGIN  
-- B 部分  
END
```

Explanation:

請設定交易以符合下列需求：

- 如果客戶只有一張發票，請刪除該發票。
- 如果客戶有多張發票，請取消交易。

請問 C 部分應填入下列何者程式碼？

-- A 部分

```
DECLARE @invoice_count INT;
```

```
SELECT @invoice_count = COUNT(*)
```

FROM Invoices

WHERE CustomerID = 123;

IF @invoice_count = 1
BEGIN
DELETE FROM Invoices

WHERE CustomerID = 123;

-- C 部分

END

ELSE

BEGIN

-- B 部分

END

Explanation:

哪一個 LINQ 運算元會壓平合併序列？

Explanation:

問題：哪一個 LINQ 運算元會「壓平合併序列」？ 正解說明 SelectMany 是唯一一個能將巢狀序列「壓平」並合併成單一序列的 LINQ 運算元。例如：`var nested = new List<List<int>> { new List<int> { 1, 2 }, new List<int> { 3, 4 } }; var flattened = nested.SelectMany(list => list).ToList(); // 輸出: [1, 2, 3, 4]`

如何設定 NoSQL 檔案資料庫的資料結構？

Explanation:

您正在為應用程式建立必要的變數。

您將儲存在這些變數中的資料具有下列特性：

由數字組成

包含小數點的數字

需要超過七位數的精確度

您需要將儲存資料所需的記憶體數量減到最少

您應該使用哪一種資料型別？

Explanation:

您正在為應用程式建立變數。

您將儲存在這個變數中的資料具有下列特性：

由數字和字元組成 (例如："123abc" 或 "123.45abc")

含有小數點的數字 (例如："123.45" 或 "123.45abc")

您應該使用哪一種資料型別？

Explanation:

您執行了下列程式碼：

```
bool beakerFull = true;
bool flameOn = false;
int iResult = 0;
if (beakerFull) {
    if (flameOn) {
        iResult = 1;
    } else {
        iResult = 2;
    }
} else {
    iResult = 3;
}
```

程式碼執行完畢之後，哪一個值會儲存在 iResult 變數中？

Explanation:

您有下列虛擬程式碼邏輯運算式：

(A > B) AND (C < D)

您使用下列值評估了該運算式：

A = 3

B = 4

C = 4

D = 5

請問結果為何？

Explanation:

在例外狀況處理程式中，finally 段的用途為何？

Explanation:

有一個 C# 程式包含下列程式碼：

```
public delegate string CheckCodingStyle(string code);
public void CheckCode(string code, CheckCodingStyle algorithm)
{
}
```

您需要在第 05 行撰寫一個陳述式以呼叫 CheckCodingStyle 委派。

請問您應該使用哪個陳述式最恰當？

Explanation:

請檢視 Winter Sports Corporation 的下列軟體需求規格：
根據所需的系統功能，您應該選擇哪一種架構？

Explanation:

有一個名為 theStack 的堆疊包含整數值，這些值會按照下列順序推入堆疊：

1 → 4 → 1 → 6 → 3 → 5 → 6

推入堆疊的第一個數字是 1。

然後執行下列程式碼：

```
for (int i = 0; i < 3; i++)  
theStack.pop();
```

請問程式執行完畢之後，theStack 將包含什麼？

Explanation:

請根據 Git 的常見操作流程，將下列詞語移至正確位置。

詞語：

1.合併 2.提取要求 3.推送

4.分支 5.認可 6.複製

A. 您應該先建立原始的程式碼的 _____ 副本。

B. 當您在本機工作並確認變更時，需要執行 _____，如此您的變更才能儲存在本機版本庫中。

C. 然後，您應該執行 _____，以將您的變更儲存在遠端存放庫中。

D. 為了通知他們有人修改所的變更，您應建立 _____。

E. 在您的功能完成後，其他開發人員應該執行 _____ 將其納入主要分支。

Explanation:

您的團隊使用了 GitHub 做為版本控制系統。

您想要正確地將頁面填入完成的碼

可回報您的變更之前，必須先執行？

Explanation:

您的團隊使用了 GitHub 做為版本控制系統。

若要以非複製模式方式將遠端原始碼專屬變更複製到您的設定檔

可以執行？

Explanation:

您的團隊使用了 GitHub 做為版本控制系統。

若要獲知儲存庫中取得專案的最新修訂，請使用選項內那個指令？

Explanation:

請選擇正確的選項以完成有關資料化加密的句子：
數位簽章會使用？

Explanation:

請選擇正確的選項以完成有關加密解密的句子：
數位簽章需要？

Explanation:

在類別中，建構函式的用途為何？

Explanation:

您正在為應用程式設計一個 C# 類別。您需要讓 AccessCount 欄位變數只有在類別內可用。請問應該使用哪個存取修飾詞？

Explanation:

下列程式碼會追蹤與競賽選手排名的資訊。（一般先名字、名次、隊伍）

```
public class RankedPlayer
{
    private static string _defaultTeam;
    private int _rank;
    private string _name;
    public RankedPlayer(string name, int rank):this(name)
    {
        _rank = rank;
    }
    public RankedPlayer(string name, string team, int rank):this(name, rank)
    {
        _defaultTeam = team;
    }
    public RankedPlayer(string name)
    {
        _name = name;
    }
}
```

呼叫包含三個引數的建構函式時，指派欄位的順序為何？

請選出下列欄位位置正確排序：

1. _defaultTeam
2. _rank
3. _name

Explanation:

您正在開發一個追蹤網球比賽的應用程式。

比賽由下列 Match 類別代表：

```
public class Match {  
    public Match() {  
        Location = "unknown";  
    }  
    public string Location { get; set; }  
    public DateTime MatchDate { get; set; }  
}
```

比賽是使用下列程式碼來建立：

```
Match myMatch = new Match();  
myMatch.Location = "north region";  
myMatch.MatchDate = new DateTime(2021, 4, 14);
```

新建立 Match 類別的 Location 屬性指派了幾次？

Explanation:

有一個名為 Mammal 的類別衍生自一個名為 Animal 的父類別。

Mammal 類別則具有 Animal 類別的變數和方法。

哪一個能描述這種動物分類的處理過程？

Explanation:

下列何者必須存在，才能從特定類別推匯出繼承性？

Explanation:

有一個名為 Glass 的類別繼承自 Window 類別。Window 類別中包含一個名為 break() 的受保護方法。請問應從 Glass 類別中呼叫 break() 方法的語法是？

Explanation:

您建立了一個ANumber型別的物件並將類別定義如下：

```
public class ANumber {  
    private int _number = 7;  
    public ANumber() {  
    }  
    public ANumber(int number) {  
        _number = number;  
    }
```

您執行了下列程式碼：

```
ANumber mynumber = new ANumber(3);
```

程式碼執行完畢之後,_number的值為何?(1)num1 (2)0(3)3 (4)7

Explanation:

Circle 和 Square 這兩個類別繼承自 Shape 類別，並且都繼承並以不同方式實作 Area 方法。請問這代表哪一種物件導向的概念？

Explanation:

您正在擴充一個儲存及顯示各種賽跑結果的應用程式。此應用程式包含下列定義：

```
public interface IDisplayResult {  
    void Display();  
}  
  
public class Race {  
    public int Seconds;  
    public virtual void Display()  
    {  
        Console.WriteLine(Seconds);  
    }  
}
```

下列程式碼會顯示競賽的結果：

```
var r1 = new FootRace();  
var r2 = r1 as IDisplayResult;  
r1.Seconds = 99;  
r1.Display();  
r2.Display();  
r1.DisplayRaw();
```

此程式碼必須輸出下列結果至主控臺：

99 seconds
1.65 minutes
99

請完成 FootRace 類別的實作，將下列宣告依適當順序放入程式碼中：
宣告：

```
(1)public override void Display(){  
(2)public void DisplayRaw() {  
(3)void IDisplayResult.Display() {  
(4)private void Display() {
```

作答區：

```
public class FootRace : Race, IDisplayResult  
{
```

_ (A) _

```
Console.WriteLine(Seconds + " seconds");
}
```

_ (B) _

```
base.Display();
}
```

_ (C) _

```
Console.WriteLine(Seconds / 60d + " minutes");
}
}
```

Explanation:

您正在撰寫一個處理訂房要求的 web 應用程式。

您需要確認每位訪客所選取的房間尚未被其他訪客預定。

您應該使用哪一種程式設計來判斷建立要求時房間仍然可用？

Explanation:

您正在開發一個 Web 程式。需要使用階層式樣式表 (CSS) 建立下列圖形：

請選擇正確的選項以完成句子。

對於 border-radius 式，您應該使用此值？

Explanation:

您正在開發一個 Web 程式。需要使用階層式樣式表 (CSS) 建立下列圖形：

請選擇正確的選項以完成句子。

對於 border-radius 式，您應該使用此值？

Explanation:

請從清單中選取正確的選項以完成句子。

所有網頁伺服器都有 _ _ 位址和網域名稱，用以在網際網路上識別其身分。

Explanation:

請從清單中選取正確的選項以完成句子。

所有網頁伺服器會經由 _ _

通訊協定接受使用者端要求，然後以網頁的形式將資料傳回使用者端來回應。

Explanation:

請從清單中選取正確的選項以完成句子。

擁有存取權的使用者端可以經由 _ _ 通訊協定，修訂或移除裝載在網頁伺服器上的內容。

Explanation:

您的網站 (sunsetweb.org) 需要 SSL。

您使用如下圖所示的設定建立了一個虛擬目錄。

您將 Overview.jpg 檔案儲存在虛擬目錄中。

您應該使用哪一個 URL 來存取該檔案？

Explanation:

下列 REST API Web 服務程式碼會在 www.sunsetweb.com 網域上執行：

```
public class CarService
{
    private static List lst = new List(new String[] { "Oil", "Engine", "Wheels", "Paint" });

    #所以 lst = ["Oil", "Engine", "Wheels", "Paint"]
    [WebGet(UriTemplate = "/CarServices")]

    public String GetAllCarServices()
    {
        int count = lst.Count;
        String ServiceList = "";
        for (int i = 0; i < count; i++)
            ServiceList = ServiceList + lst[i] + ",";
        return ServiceList;
    }

    [WebGet(UriTemplate = "/CarServices/{ServiceListid}")]
    public String GetCarServicebyID(String ServiceListid)
    {
        int id;
        Int32.TryParse(ServiceListid, out id);

        return lst[id]; #回傳 lst[3]
    }
}
```

您對這個 web 服務發出了如下列呼叫：

www.sunsetweb.com/carservices/3

請問輸出為何？

Explanation:

請在圖表中指出MVC架構模式的正確元件。請從清單中選取正確的這項以完成圖表(依照ABC順序回答)

Explanation:

請完成圖表以解說MVC 架構的核心功能。請將每個動作

Explanation:

哪一個正規化要求資料表的屬性僅相依於主索引鍵?

Explanation:

您需要建立一個傳入人員 UserName 和 Age 的預存程式。

Age 必須是以年為單位的整數。

UserName 至少必須有三個字元。

您應該使用哪一個陳述式來建立此預存程式?

Explanation:

在更新資料庫的多個資料表時，使用交易有何優點?

Explanation:

您有一部限制資料連線數目的伺服器。您應該使用下列何者，在使用者數目超過可用的連線數目時將連線能力最佳化?

Explanation:

您應該在有效的單元測試案例中納入哪兩種資訊? (請選擇 2 個答案)

Explanation:

下列哪兩項敘述正確描述了 JSON 物件? (請選擇 2 個答案)

Explanation:

下列哪兩項敘述正確描述了將資料庫正規化的優點? (請選擇 2 個答案)

Explanation:

連線共用有哪三項優點? (請選擇 3 個答案。)

Explanation:

哪兩個程式碼片段可用來確保資源釋出？（請選擇 2 個答案。）

Explanation:

您應該在有效的單元測試案例中納入哪兩種資訊？（請選擇 2 個答案。）

Explanation:

下列哪兩項敘述正確描述了JSON物件？（請選擇 2 個答案。）

- (A) JSON物件可使用名稱/直對來建置。
- (B) JSON物件可使用已排序的值清單來建置。
- (C) JSON物件可運用名稱空間。
- (D) JSON物件可延伸。
- (E) 以上皆是
- (F) 以上皆非

Explanation:

連級共用有那三項優點？（請選擇 3 個答案。）

- (1) 提升效能
- (2) 提升可據能性
- (3) 縮短建立連線的時間
- (4) 不需要任何設定
- (5) 提升安全性
- (6) 以上皆非

Explanation:

請回答有關單元測試的敘述：

單元測試會測試整個應用程式的整合與相依性。

Explanation:

單元測試方法會呼叫目前測試的方法、傳遞資料，然後檢查傳回值是否正確。

Explanation:

單元測試的主要目標是取出應用程式中最小的可測試軟體部分，將此部分與其於程式碼隔離，然後判斷其行為表現是否完全符合預期。

Explanation:

連結清單可進行排序。

Explanation:

連結清單最多包含 100 個節點。

Explanation:

您可以在連結清單(Linked List)內任何位置加入新的節點。

Explanation:

數位簽章會使用寄件者的公開金鑰來加密訊息。

Explanation:

關鍵點：數位簽章的加密（簽章）是使用私鑰，而非公鑰。 2. 加密（Encryption）與數位簽章的區別

CSRF（跨網站偽造要求）權杖是用來防止攻擊。

Explanation:

防止 SQL 插入式攻擊的其中一種方式就是逸出使用者輸入。

Explanation:

SQL Injection 攻擊 攻擊者透過在輸入欄位或 URL 中注入惡意 SQL 語法，讓系統執行非預期的資料庫操作。例如輸入 "' OR 1=1 --"，就可能繞過登入檢查。防禦方法之一：輸入逸出 (Escaping User Input) 逸出 (Escape) 的意思是把使用者輸入中的特殊字元（如 '，;）轉換為無害字元，避免被解釋為 SQL 語法的一部分。雖然這樣做可以降低風險，但要注意它不是最安全、最推薦的方法。更佳的防禦方式：引數化查詢 (Parameterized Query, Prepared Statement) → 最標準的解法。最小許可權原則 (Least Privilege) → 限制資料庫帳號許可權。輸入驗證 (Input Validation) → 過濾非預期的輸入內容。

Person 類別中的方法可以存取 color 屬性。

Explanation:

#Person 類別中的方法可以訪問該類的屬性color，只要該屬性是公開的或者在類的範圍內。

某個專案包含一個名為 Person 的類別，以及一個名為 Tree 類別。Tree 類別具有一個名為 color 的屬性，以及一個名為 NumberOfLeaves 的受保護屬性。請選取 [正確] 或 [錯誤]。Person 類別中的方法可以存取 NumberOfLeaves 屬性。

Explanation:

※因Person類別與Tree類別無繼承關係，且未明確在同一包內，Person類別的方法無法訪問Tree的protected屬性。因此正確答案為 B. No.

某個專案包含一個名為 Person 的類別，以及一個名為 Tree 類別。

Tree 類別具有一個名為 color 的屬性，以及一個名為 NumberOfLeaves 的受保護屬性。

請選取 [正確] 或 [錯誤]。

Tree 類別中的方法可以存取 Person 中的所有私用屬性。

Explanation:

在軟體開發中，類別之間的私有屬性無法被其他類別直接訪問，因此 Tree 類別中的方法無法存取 Person 中的所有私用屬性，正確答案為 B. No。題目關鍵點分析 1. 類別與屬性範圍 Person 類別與 Tree 類別為獨立的類別，無繼承關係。Tree 類別中包含 color 屬性(未標註訪問修飾符，預設為私有)與

NumberOfLeaves 受保護屬性。 Person 類別的私有屬性（如 private int age;）僅能被 Person 自身的成員方法訪問，無法被其他類別（如 Tree）直接存取。 2. 訪問控制機制
私有(private)：僅能被定義該屬性的類別內部方法訪問。
受保護(protected)：可被同一類別或其子類別訪問，但 Tree 類別未繼承 Person，因此無法訪問 Person 的私有屬性。 公有 (public)：可被所有類別訪問，但題目未提及 Person 有公有屬性。 3. 跨類別的訪問限制即使 Tree 類別與 Person 類別在相同包中，或有其他關聯，類別之間的私有屬性無法被直接存取，除非透過公開的 getter 方法或繼承關係。

請回答有關程式碼的敘述：
加上行號僅為參考之用。

```
01 var xhr = new XMLHttpRequest();
02 xhr.onload = function() {
03   if(xhr.status === 200) {
04     responseObject = JSON.parse(xhr.responseText);
05     var newContent = '';
06     for (var i = 0; i < responseObject.menu.length; i++) {
07       newContent += '<p>' + responseObject.menu[i].Entree + '<br>' + responseObject.menu[i].Price + '</p>';
08     }
09   }
10   document.getElementById('content').innerHTML = newContent;
11 }
12 };
13 xhr.open('GET', 'menu.json', true);
14 xhr.send(null);
```

在第 02 行，onload 物件會將要求傳送至伺服器。

Explanation:

請回答有關程式碼的敘述：
加上行號僅為參考之用。

```
01 var xhr = new XMLHttpRequest();
02 xhr.onload = function() {
03   if(xhr.status == 200) {
04     responseObject = JSON.parse(xhr.responseText);
05     var newContent = '';
06     for (var i = 0; i < responseObject.menu.length; i++) {
07       newContent += '<p>' + responseObject.menu[i].Entree + '<br>' + responseObject.menu[i].Price + '</p>';
08     }
09   }
10   document.getElementById('content').innerHTML = newContent;
11 }
12 };
13 xhr.open('GET', 'menu.json', true);
14 xhr.send(null);
```

在第 03 行，status 是由處理要求的伺服器傳回。

Explanation:

請回答有關程式碼的敘述：

加上行號僅為參考之用。

```
01 var xhr = new XMLHttpRequest();
02 xhr.onload = function() {
03   if(xhr.status === 200) {
04     responseObject = JSON.parse(xhr.responseText);
05     var newContent = <div>;
06     for (var i = 0; i < responseObject.menu.length; i++) {
07       newContent += <p>Entree: </p> + responseObject.menu[i].Entree +
08       <br>+<p>Price: </p> + responseObject.menu[i].Price + </div>;
09     }
10     document.getElementById('content').innerHTML = newContent;
11   }
12 };
13 xhr.open('GET', 'menu.json', true);
14 xhr.send(null);
```

在第 13 行，true 引數會指定非同步處理。

Explanation:

請回答有關佇列的敘述：

佇列的專案數有限制。

Explanation:

請回答有關佇列的敘述：

佇列是一種先進先出 (FIFO) 資料結構。

Explanation:

請回答有關佇列的敘述：

佇列的專案是按照加入的順序儲存。

Explanation:

請回答有關 NoSQL 資料庫的敘述：

檔案資料庫會將資料儲存在類似 JSON 的檔案中。

Explanation:

請回答有關 NoSQL 資料庫的敘述：
檔案資料庫的設計目的是以預先定義的關聯處理資料。

Explanation:

請回答有關 NoSQL 資料庫的敘述：
索引鍵值資料庫會將資料儲存成您可以修改或查詢其屬性的物件。

Explanation:

實體關聯圖表 (ERD) 是專為物件導向的資料庫所設計。

Explanation:

實體關聯圖表 (ERD) 包含實體、關聯和屬性。

Explanation:

基數是指資料表的互相連結方式，例如一對多。

Explanation:

【OOP-繼承】在物件導向程式設計中，當一個類別 (Class) 衍生自另一個類別時，這稱為繼承。例如 `Dog` 類別繼承自 `Animal` 類別。請問這代表了什麼關係？

Explanation:
繼承 (Inheritance) 建立的是 **Is-a** 的關係。例如：狗 (Dog) **是一個 (is an)** 動物 (Animal)。這是判斷是否該使用繼承的主要依據。

【OOP-封裝】您正在設計一個銀行帳戶類別 `BankAccount`，其中的 `balance` (餘額) 欄位不應該讓外部程式碼直接修改，而應該透過 `Deposit()` 或 `Withdraw()` 方法來變更。請問您應該將 `balance` 設定為哪種存取修飾詞？

Explanation:
private (私有) 成員只能在類別內部存取，這是**封裝 (Encapsulation)** 的核心概念，用於保護資料不被外部不當修改。

【OOP-多型】在 C# 中，如果父類別有一個方法 `Draw()`，而子類別想要提供該方法的不同實作（例如圓形和矩形的畫法不同），子類別應該使用哪個關鍵字？

Explanation:
override (覆寫) 用於在子類別中重新定義父類別的虛擬 (virtual) 或抽象 (abstract) 方法，這是實現**多型 (Polymorphism)** 的方式。

【OOP-類別與物件】關於類別 (Class) 與物件 (Object) 的敘述，下列何者正確？

Explanation:

類別 (Class) 像是一個**模具或藍圖**，定義了屬性與行為；而物件 (Object) 是使用 `new` 關鍵字根據類別建立出來的具體**實體 (Instance)**。

【ALM-生命週期】在應用程式生命週期管理 (ALM) 中，哪個階段主要負責定義「系統必須做什麼」以及蒐集使用者需求？

Explanation:

需求分析階段的目標是釐清使用者的痛點與需求，產出規格書，確認「要做什麼 (What)」。設計階段則是規劃「如何做 (How)」。

【ALM-測試】開發人員在寫完一個函式後，撰寫了一小段程式碼來驗證該函式的邏輯是否正確。這種測試稱為？

Explanation:

單元測試是針對軟體中最小的可測試單元（通常是函式或方法）進行驗證，通常由開發人員自己撰寫。

【Web-基礎結構】在網頁開發中，負責定義網頁「結構」與「內容」的是哪種語言？

Explanation:

HTML (HyperText Markup Language) 負責網頁的骨架與內容； **CSS** 負責樣式與美觀； **JavaScript** 負責行為與互動。

【Web-CSS】如果您希望網頁上的某個按鈕背景變成藍色，字型變成白色，您應該使用哪種技術？

Explanation:

CSS 用於描述 HTML 元素的外觀樣式（如顏色、字型、排版）。

【Web-執行環境】JavaScript 程式碼通常是在哪裡執行的？

Explanation:

雖然現在有 Node.js，但在傳統 Web 開發語境下，**JavaScript** 主要在使用者的**瀏覽器 (Client-side)** 上執行，負責動態互動效果。

【Web-伺服器】哪一個術語是用來描述負責託管 (Hosting) 網頁應用程式並回應 HTTP 請求的軟體（如 IIS 或 Apache）？

Explanation:

Web Server (網頁伺服器) 負責儲存、處理並傳送網頁檔案給客戶端。

【OOP-抽象】關於抽象類別 (Abstract Class) 與介面 (Interface)，下列敘述何者正確？

Explanation:

C# 支援**單一繼承**（只能繼承一個類別），但可以**實作多個介面**。抽象類別不能被直接實體化，必須被繼承。

【C#-記憶體】在 C# 中，參考型別 (Reference Type) 的物件實體通常儲存在記憶體的哪個區域？

Explanation:

參考型別（如 Class）的實體儲存在 **Heap**；而實值型別（如 int, bool）的值通常儲存在 **Stack**（當作為區域變數時）。

【ALM-排序】請排列軟體開發生命週期的正確順序：1.開發 2.需求分析 3.測試 4.設計 5.維護

Explanation:

標準瀑布式流程：**需求分析** -> **設計** -> **開發 (實作)** -> **測試** -> **維護** (部署後)。

【C#-集合】您需要儲存一個沒有重複元素的數列，應該使用哪種集合型別？

Explanation:

HashSet 是一種集合，其特性是**不允許重複元素**，且存取速度快。

【Web-狀態】HTTP 協定本身是無狀態的

(Stateless)。為了在不同頁面間記住使用者的登入狀態，通常會使用什麼技術？

Explanation:

Cookies 和 **Session** 是 Web 應用程式用來追蹤使用者狀態（如購物車、登入資訊）的主要機制。