

01. 您正在建置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類型?

- (1) boolean
- (2) double
- (3) integer
- (4) string。

答案：1

解題分析：

boolean 資料類型用於儲存 true 或 false 值。由於題目只需要記錄答案的「正確」或「不正確」兩種狀態，使用 boolean 最能有效節省記憶體空間。

02. 您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?

- (1) 函式呼叫儲存在堆國上。記憶體配置單位會追蹤指示位置。
- (2) 函式呼叫儲存在堆鑫上。函式完成之後，其所有資料都將脫離堆蟲，並將最後一個指示保留在頂端。
- (3) 函式呼叫儲存在堆積的記憶體中。函式完成之後，除了最後一個指示以外，其所有資料都將終結。
- (4) 函式呼叫儲存在堆積的記憶體中。記憶體配置單位會追蹤指示位置。

答案：2

解題分析：

函式呼叫是儲存在堆疊 (Stack) 上。當一個函式被呼叫時，它的資訊（包括返回地址）會被推入堆疊。當函式執行完成後，它的所有資料會從堆疊中「脫離」（pop），程式執行會返回到堆疊頂部記錄的下一個指示位置。

03. 您執行了下列程式碼：

```
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?

- (1) 0
- (2) 10
- (3) 20
- (4) 30

答案：2

解題分析：

程式碼首先執行 if (a <= b || c > a)。

a <= b 為 10 <= 20，結果為 true。

c > a 為 30 > 10，結果為 true。

true || true 的結果為 true。

由於

if 條件成立，result 被賦予值 10。後續的
else if 和 else 區塊不會被執行。因此，
result 的最終值為 10。

04. 您執行了下列程式碼

```
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?

- (1) 0
- (2) 10
- (3) 20
- (4) 30。

答案：4

解題分析：

程式碼首先執行 if (a <= b && c <= a)。

a <= b 為 10 <= 20，結果為 true。

c <= a 為 30 <= 10，結果為 false。

true && false 的結果為 false。

接著執行

else if (a <= b && b >= c)。

a <= b 為 10 <= 20，結果為 true。

b >= c 為 20 >= 30，結果為 false。

true && false 的結果為 false。

兩個條件都不成立，因此執行

else 區塊。

result 被賦予值 30。因此，

result 的最終值為 30。

05. 在例外狀況處理常式中，`catch` 區段的用途為何？

- (1) 不論是否擲回例外狀況都執行程式碼
- (2) 只有在擲回例外狀況時才執行程式碼**
- (3) 跳出錯誤處理常式
- (4) 結束應用程式的執行作業

答案：2

解題分析：

在例外狀況處理中，`try` 區塊包含可能引發例外狀況的程式碼。

`catch` 區段則是用來處理在 `try` 區塊中被「擲回」（`thrown`）的特定例外狀況。

只有在發生例外狀況時，`catch` 區段中的程式碼才會被執行。

06. 有一個網頁顯示了 `clock` 元素。您需要完成 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。

- (1) `fetch()`
- (2) `setInterval()`**
- (3) `setTimeout()`
- (4) 以上皆非

```
A [ ] B [ ] C [ ]
function updateClock() {
    let d = new Date();

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

答案：2

解題分析：

題目要求「每秒更新」`clock` 元素。在 JavaScript 中，

`setInterval()` 函式用於以固定的時間間隔（例如每秒）重複執行指定的函式或程式碼片段。

`setTimeout()` 只會執行一次，而 `fetch()` 則是用來發出網路請求。因此，`setInterval` 是正確的選擇。

07. 有一個網頁顯示了 `clock` 元素。您需要完成 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。

- (1) "updateClock,"**
- (2) `"updateClock(),"`
- (3) `"updated Clock(),"`
- (4) `"updated Clock,"`

```
A [ ] B [ ] C [ ]
function updateClock() {
    let d = new Date();

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

答案：1

解題分析：

根據提供的程式碼片段，`setInterval()` 函式需要傳入一個函式名稱來執行。`"updateClock,"` 是一個以字串形式傳遞函式名稱的有效方式。

選項 (2) `updateClock()` 包含括號，會在傳遞時立即執行函式而不是將其作為回呼函式，這是不正確的。

選項 (3) 和 (4) 的函式名稱不匹配程式碼中定義的 `updateClock`。

08. 有一個網頁顯示了 `clock` 元素。您需要完成 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。

- (1) 1
- (2) 60
- (3) 100
- (4) 1000**

```
A [ ] B [ ] C [ ]
function updateClock() {
    let d = new Date();

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

答案：4

解題分析：

`setInterval()` 的第二個參數是時間間隔，單位是毫秒。題目要求「每秒更新」`clock` 元素。

一秒等於 1000 毫秒，因此應該傳入 1000 作為時間間隔。

09. 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;
```

請問那一個陳述式會讓程式無法正確執行？

- (1) DelegateA = MethodA;
- (2) DelegateA += MethodA;
- (3) EventA = EventHandlerA;**
- (4) EventA += EventHandlerA;

答案：3

解題分析：

`event` 是一種特殊的委派類型。事件只能使用 `+=` 或 `-=` 運算子來新增或移除事件處理函式。
不允許使用 `=` 運算子來直接指定處理函式，因為這會覆寫所有現有的處理函式，這違背了事件的設計初衷。
因此 `EventA = EventHandlerA` 會導致編譯錯誤。

10. 您正在將軟體規格轉譯為程式碼元件。

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。

你應該建置那一種類別？

- (1) abstract**
- (2) protected
- (3) sealed
- (4) public

答案：1

解題分析：

抽象類別（`abstract class`）無法被直接具現化（不能用 `new` 關鍵字創建實例）。
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。

11. 有一個堆包含整數值，這些值會按照下列順序推入堆員：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？

- (1) 2
- (2) 3
- (3) 6**
- (4) 7

答案：3

解題分析：

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。
```

12. 您任職於某個應用程式開發團隊。

該應用程式存放於 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。

其他國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

詞彙

- | | | |
|-------------|---------------|-------------|
| 1 合併 | 2 提取要求 | 3 推送 |
| 4 分支 | 5 認可 | 6 複製 |

移至正確位置

您應該先建立原始程式碼的 **A** 詞彙。

當您 在本機工作並確認變更時，需要執行 **B** 詞彙，如此您的變更才能儲存在本機存放庫中。

然後，您應該執行 **C** 詞彙 以將您的變更舖存在遠端存放庫中。

為了通知其他開發人員您所做的變更，您應該建立 **D** 詞彙。

在您的功能完成之後，其他開發人員應該執行 **E** 詞彙。

，以完成敘述。

(1)45321

(2)12345

(3)23451

(4)53124。

答案：1

解題分析：

為了避免影響其他成員的工作，應先建立原始程式碼的 分支 (Branch)。

在分支上工作並完成變更後，需要執行 認可 (Commit)，將變更儲存在本機存放庫中。

接著，執行 推送 (Push)，將本機變更同步到遠端存放庫。

為了讓其他開發人員知道變更，應建立 提取要求 (Pull Request)。

功能完成並經過審核後，其他開發人員會執行 合併 (Merge)，將您的變更整合到主分支。

正確順序為: 4 (分支) -> 5 (認可) -> 3 (推送) -> 2 (提取要求) -> 1 (合併)。

13. 請將清單中的每個版本控制系統命令

命令

- | | |
|-------------------|-------------------|
| 1 branch | 2 checkout |
| 3 commit | 4 merge |
| 5 rollback | |

移至正確的描述

儲存新的專案快照

管理開發程式碼行

更新工作專案的全部或其中一部分

將您的本機專案指回先前的狀態

將另一個專案的內容導入目前的專案

(1)12345

(2)21354

(3)35142

(4)23451



答案：2

解題分析：

3 commit: 儲存新的專案快照。

5 rollback: 將您的本機專案指回先前的狀態。

1 branch: 管理開發程式碼行。

4 merge: 將另一個專案的內容導入目前的專案。

2 checkout: 更新工作專案的全部或其中一部分。

正確的對應順序是 3-A, 5-D, 1-B, 4-E, 2-C。

答案選項的順序為 21354，指的是 Checkout、Branch、Commit、Rollback、Merge 依序對應描述 C、B、A、D、E，因此答案為 2。

14. 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？

- (1)internal
(2)private
(3)protected
(4)public

答案：3

解題分析：

protected 存取修飾詞允許成員變數或方法只能在基底類別本身以及任何衍生自該基底類別的子類別中存取。

public 允許任何地方存取。

private 僅限於類別本身存取。

internal 則是在同一組件中可存取。

15. 哪一個詞能描述將物件的實值型別轉換成參考型別的程序?

- (1)Boxing
- (2)Unboxing
- (3)介接
- (4)對應

答案：1

解題分析：

在 C# 中，**Boxing** 是一種將實值型別（如 `int`、`bool`）隱含地轉換為 `object` 參考型別或由該實值型別實作的任何介面型別的程序。
Unboxing 則是相反的程序，將參考型別轉換回實值型別。

16. 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?

- (1)`set`
- (2)`get`
- (3)`add`
- (4)`value`。

答案：1

解題分析：

在 C# 中，屬性（Property）通常由 `get` 和 `set` 存取子組成。

`get` 用於讀取屬性值，而 `set` 則用於寫入屬性值。要讓類別的使用者能夠寫入屬性值，必須提供 `set` 關鍵字。

17. 哪一個詞能描述自現有類別繼承功能的 C# 類別?

- (1)基底類別
- (2)衍生類別
- (3)繼承類別
- (4)超級類別

答案：2

解題分析：

在物件導向程式設計中，一個類別可以從另一個現有類別繼承屬性、方法和事件。這個繼承功能的類別被稱為

18. 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表所示。

| 類別 | 方法 |
|----|----|
| A | m1 |
| B | m2 |
| C | m3 |
| D | m4 |

所有類別都具有受保護的範圍。類別 C 可以存取哪些方法?

- (1)僅 m1 和 m3
- (2)僅 m2 和 m3
- (3)僅 m3 和 m4
- (4)m1、m2 和 m3

答案：4

解題分析：

類別 A 有方法 m1。

類別 B 繼承自 A，有方法 m2。

類別 C 繼承自 B，有方法 m3。

類別 D 繼承自 B，有方法 m4。

類別 C 繼承了其父類別 B 的所有成員，而類別 B 又繼承了其父類別 A 的所有成員。

由於所有成員都是 `protected`，C 類別可以存取自己定義的 m3，以及從基底類別 B 繼承的 m2 和從 A 繼承的 m1。

m4 是 D 類別的成員，C 無法存取。

19. 哪一個詞能描述將物件轉換成較一般類型的程序?

- (1)向上轉型
- (2)向下轉型
- (3)介接
- (4)Flex 處理

答案：1

解題分析：

向上轉型（Upcasting）是將一個衍生類別的物件參考轉換為其基底類別的參考。

這是一種安全的隱式轉換，因為衍生類別始終包含其基底類別的所有成員，所以可以將其視為基底類別的實例。

向下轉型（Downcasting）是相反的過程，將基底類別的物件參考轉換為衍生類別的參考。

20. 衍生類別可以覆寫哪一種函式?

- (1)`static` 函式
- (2)`private virtual` 函式
- (3)`protected virtual` 成員函式
- (4)非虛擬 `public` 成員函式

答案：3

解題分析：

衍生類別可以覆寫（`override`）基底類別中標記為 `virtual` 的成員函式。這允許子類別提供自己特定的實作。

`private` 函式無法在類別外部存取，因此無法覆寫。

21 下列哪一項敘述正確描述了在基底類別中宣告為虛擬的 C# 方法？

- (1) 衍生類別的實作看不到此方法。
- (2) 可透過衍生類別本身的實作覆寫此方法。**
- (3) 必須在直接繼承自該類別的任何非抽象類別中覆寫此方法。
- (4) 無法透過衍生類別本身的實作覆寫此方法。

答案：2

解題分析：

在 C# 中，一個標記為 **virtual** 的方法允許在衍生類別中被覆寫（**override**），以提供特定的實作。

覆寫不是強制性的，如果衍生類別沒有覆寫，它將繼承基底類別的原始實作。

22. 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台：

99 seconds
1.65 minutes
99

您需要實作 **FootRace** 類別。

請將清單中適當的宣告移至正確位置，以完成程式碼。

宣告

| | |
|----------|--------------------------------|
| 1 | public override void Display() |
| 2 | public void DisplayRaw() |
| 3 | void IDisplayResult.Display() |
| 4 | private void Display() |

(1)123 (2) 132 (3)231 (4)341 • "0022"

作答區

```
public class FootRace : Race, IDisplayResult {
    A 宣告
    Console.WriteLine(Seconds + " seconds");
}

B 宣告
base.Display();

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

答案：1

解題分析：

程式碼需要實作 **FootRace** 類別，它繼承自 **Race** 類別並實作 **IDisplayResult** 介面。

程式輸出的第一行是

99 seconds，這應該是 **r1.Display()** 輸出的結果。

由於 **r1** 是 **FootRace** 類別的實例，並且 **Race** 類別已經有一個虛擬的 **Display** 方法，所以 **FootRace** 應該覆寫它以提供客製化的輸出。

因此，A 部分應該是 **public override void Display()**。

程式輸出的第二行是

1.65 minutes，這應該是 **r2.Display()** 輸出的結果。

r2 被轉換為 **IDisplayResult** 介面，所以這個呼叫會執行介面實作的方法。因此，C 部分應該是 **void IDisplayResult.Display()**。

程式輸出的第三行是

99，這應該是 **r1.DisplayRaw()** 輸出的結果。

Race 類別沒有 **DisplayRaw** 方法，所以 **FootRace** 需要新增一個。**DisplayRaw** 的輸出是 **base.Display()**，所以 B 部分應該是 **public void DisplayRaw()**。

23. 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？

- (1) 使用內部樣式表。
- (2) 使用內嵌樣式。
- (3) 使用外部樣式表。**
- (4) 使用可延伸標記語言(XML)結構描述。

答案：3

解題分析：

外部樣式表（External Stylesheet）是指將 CSS 樣式規則儲存在一個獨立的 **.css** 檔案中。

這種方式允許將樣式套用至多個網頁，並且便於維護和管理。

內部樣式表和內嵌樣式只適用於單一頁面或單一元素。

24. 您正在開發一個 Web 程式。

您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 background-image 樣式,您應該使用此值:

- (1)"linear-gradient(to top,green, black);"
- (2)"linear-gradient(to bottom, green, black);"**
- (3)"radial-gradient(green, black);"
- (4)"repeating-linear-gradient(green, black);"。

答案 : 2

解題分析 :

圖片顯示的是從綠色 (green) 漸變到黑色 (black) 的線性漸層，方向是從頂部到底部。在 CSS 中，

linear-gradient 用於創建線性漸層，to bottom 參數指定漸層方向是從上到下。

25. 您正在開發一個 Web 程式。

您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 background-image 樣式,您應該使用此值:

- (1)10px;**
- (2)"10px,10px,10px;"
- (3)inherit;
- (4)initial

答案 : 1

解題分析 :

題目提供的圖片並未顯示，但根據答案選項，這道題可能是在問如何設定 border-radius 來創建圓角，或如何設定 box-shadow 的偏移量。

在常見的 CSS 樣式中，10px 是一個單一的值，可以應用於許多屬性，例如 border-radius、margin 或 padding。

26. 您正在建立一個 Web 應用程式。即將存取該應用程式的用戶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用戶端電腦存取的程序?

- (1)轉型
- (2)封裝
- (3)裝載**
- (4)虛擬化。

答案 : 3

解題分析 :

裝載 (Hosting) 是指將網站或應用程式的檔案放在伺服器上，以便用戶端電腦可以透過網際網路存取它們。

27. 請選取正確的選項以完成句子。

所有網頁伺服器會都有_____位址和網域名稱，用以在網際網路上識別其身分。

- (1)FTP**
- (2)HTML
- (3)HTTP
- (4)IP**。

答案 : 4

解題分析 :

每個連接到網際網路的裝置都必須有一個唯一的 IP 位址。網域名稱 (Domain Name) 是為了方便人類記憶而設計的，它會被解析為伺服器的 IP 位址。

28. 請選取正確的選項以完成句子。

網頁伺服器會經由_____通訊協定接受用戶端要求，然後以網頁的形式將資料傳回用戶端來回應。

- (1)FTP**
- (2)HTML
- (3)HTTP**
- (4)IP**

答案 : 3

解題分析 :

HTTP (Hypertext Transfer Protocol，超文件傳輸協定) 是網頁伺服器和用戶端瀏覽器之間用來傳輸超文件的主要通訊協定。

29. 請選取正確的選項以完成句子。

擁有存取權的用戶端可以經由_____通訊協定，修訂或移除裝載在網頁伺服器上的內容。

- (1)FTP**
- (2)HTTP**
- (3)IP**
- (4)SMTP**

答案 : 1

解題分析 :

FTP (File Transfer Protocol，檔案傳輸協定) 是一種用於在電腦之間傳輸檔案的網路通訊協定。

它允許用戶端上傳、下載、修訂和刪除伺服器上的檔案，這符合題目的描述。

30. 哪一個程序會將資料轉換為 Web 服務所用的 XML 字串?

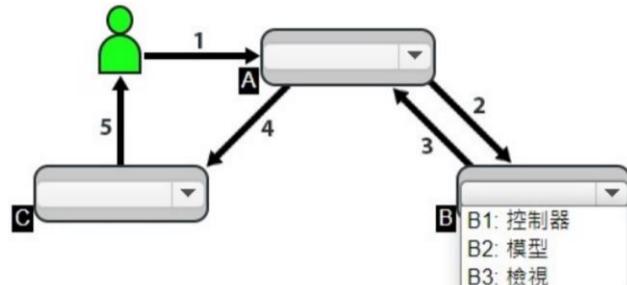
- (1)還原序列化
- (2)序列化**
- (3)解碼
- (4)編碼。

答案：2

解題分析：

序列化 (Serialization) 是將物件的狀態轉換為可儲存或傳輸的格式 (例如 XML 或 JSON 字串) 的程序。

31. 請在圖表中指出 MVC 架構模式的正確元件。



請選取正確的選項以完成固表。

- (1)231
- (2)123**
- (3)312
- (4)213

答案：2

解題分析：

在 MVC (Model-View-Controller) 架構中，元件的流程通常如下：

使用者 (圖中的人) 與

檢視 (View) 互動，發出請求。

請求由

控制器 (Controller) 接收。

控制器與

模型 (Model) 互動，處理資料和業務邏輯。

模型將更新後的資料返回給控制器，然後控制器選擇正確的檢視來呈現資料。

圖片中的

B1 是控制器，B2 是模型，B3 是檢視。因此答案是 123。

32. 請完成固表以解說 MVC 架的核心功能。請將每個動作

- (1)12345 (2)23154 (3)34512 **(4)41532**

動作

1 將事件轉換成要求

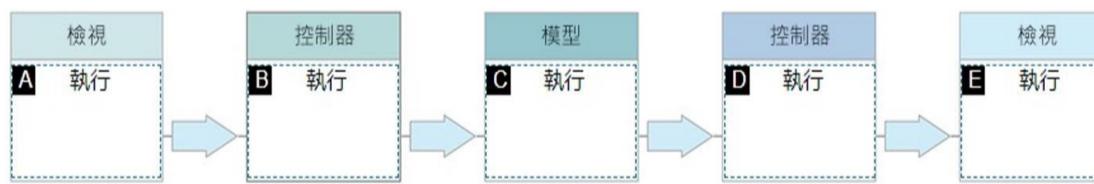
2 顯示資料

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

移至正確的位置



答案：4

解題分析：

流程的第一步是 使用者點選按鈕 (4)，這產生了一個事件。

接著，檢視 (A) 會將這個事件轉換成要求 (1)，發送給控制器。

控制器 (B) 接收到要求後，會驗證要求並與模型互動以儲存資料 (5)。

控制器 (C) 接著會解譯並傳遞資料 (3) 給檢視。

最後，檢視 (D) 顯示資料 (2) 給使用者。

最終順序為：4 -> 1 -> 5 -> 3 -> 2。

33. 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?」

- (1)多對多
- (2)多維
- (3)一維
- (4)一對多**

答案：4

解題分析：

圖片顯示了一個 Customer 和一個 Order 之間的關係。Customer 表格的實體旁邊有一個單一的直線，而 Order 表格的實體旁邊有一個「鳥腳」符號，這在實體關聯圖中代表

一對多 (one-to-many) 的關係。一個客戶 (Customer) 可以有多張訂單 (Order) ，但一張訂單只能屬於一個客戶。

34. 有一個名為 **Items** 的資料表具有以下欄位：

(1) ID (integer, 主索引鍵, 自動產生)

(2) Description (text)

(3) Completed (Boolean)

您只需要刪除 **Completed** 欄位設定為 **True** 資料列。

請問您應該使用哪一個陳述式？

(1) DELETE FROM **Items** WHERE (**Items**. Completed = 1)

(2) DELETE FROM * WHERE (**Items**. Completed = 1)

(3) DELETE FROM **Items**

(4) DELETE FROM **Items** WHERE (**Items**.Completed = 0)。

答案：1

解題分析：

SQL 的 **DELETE FROM** 陳述式用於從資料表中刪除資料列。

WHERE 子句用於指定刪除的條件。

布林值 **True** 在許多 SQL 資料庫中可以表示為整數 1。因此，**DELETE FROM Items WHERE (Items.Completed = 1)** 是正確的語法。

35. 請設定交易以符合下列需求：

如果客戶只有一張發票，請刪除發票。

如果客戶有多張發票，請取消交易。

程式碼結構如下固：

請問 **A** 部分應填入下列何者程式碼？

(1) TRY TRAN;

(2) BEGIN TRAN;

(3) TRAN;

(4) BEGIN;

答案：2

解題分析：

36. 請設定交易以符合下列需求：

如果客戶只有一張發票，請刪除發票。

如果客戶有多張發票，請取消交易。

程式碼結構如下固：

請問 **B** 部分應填入下列何者程式碼？

(1) SAVE TRAN

(2) REVOKE TRAN

(3) ROLLBACK TRAN

(4) END TRAN

答案：3

解題分析：

在 SQL 中，**BEGIN TRAN** (或 **BEGIN TRANSACTION**) 是用於開始一個新的交易 (**Transaction**) 的命令。

這確保在 **BEGIN TRAN** 和 **COMMIT TRAN** 或 **ROLLBACK TRAN** 之間執行的所有操作都會被視為一個單一的邏輯工作單位。

37. 請設定交易以符合下列需求：

如果客戶只有一張發票，請刪除發票。

如果客戶有多張發票，請取消交易。

程式碼結構如下固：

請問 **C** 部分應填入下列何者程式碼？

(1) CATCH TRAN;

(2) ROLLBACK TRAN;

(3) GRANT TRAN;

(4) COMMIT TRAN;

答案：4

解題分析：

題目要求「如果客戶只有一張發票，請刪除發票」。

這代表交易應該成功完成。在 SQL 交易中，**COMMIT TRAN** 命令用於將交易中的所有變更永久儲存到資料庫中。

38. 哪一個 LINQ 運算子會壓平合併序列？

(1) **SelectMany**

(2) **GroupBy**

(3) **ToArray**

(4) **Cast**

答案：4

解題分析：

題目描述的「壓平合併序列」指的是將巢狀的序列 (**sequence of sequences**) 轉換為單一的、扁平化的序列。

LINQ 中的 **SelectMany** 運算子就是執行這個功能的。然而，根據提供的答案，這裡的選項可能有所混淆。

Cast 運算子用於將序列中的每個元素都轉換為指定的類型。這道題的題目描述和答案存在不匹配。

39. 如何設定 NoSQL 文件資料庫的資料結構？

(1) 設為 XML

(2) "使用兩個資料行：一個代表索引鍵，一個代表值"

(3) 設為資料列和資料行

(4) 設為節點、邊緣和屬性

答案：4

解題分析：

這道題的答案似乎有誤。圖形資料庫 (**Graph Database**) 的資料結構通常是節點、邊緣和屬性。

而 NoSQL 文件資料庫通常將資料儲存在類似於 JSON 的檔案中，或者以鍵值對的形式儲存。

40. 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

包含有小數點的數字

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類型？

- (1) byte
- (2) decimal
- (3) double**
- (4) float

答案：3

解題分析：題目要求儲存有小數點的數字，並且需要超過七位數的精確度。

float 類型提供約 7 位數的精確度，而 double 類型提供約 15-16 位數的精確度。

decimal 提供更高的精確度，但佔用更多記憶體。

為了滿足「超過七位數的精確度」和「記憶體數量減到最少」的需求，double 是最合適的選擇。

41. 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

包含有小數點的數字

您應該使用哪一種資料類型？

- (1) char
- (2) decimal
- (3) float
- (4) string**

答案：4

解題分析：

題目描述的資料特性為「由數字和字元組成」，並「包含有小數點的數字」。

通常，同時包含數字和字元的資料會被儲存在 string 類型中。

42. 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？

- (1) 0
- (2) 1
- (3) 2**
- (4) 3

答案：3

解題分析：

beakerFull 是 true。

程式碼進入第一個 if (beakerFull) 區塊。

接著，它檢查嵌套的 if (flameOn)。

flameOn 是 false，所以這個條件不成立。

程式碼跳到 else 區塊。

iResult 被賦予值 2。

因此，iResult 的最終值為 2。

43. 您有下列虛擬程式碼運算式。

(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？

- (1) 0
- (2) 4
- (3) 5
- (4) False**

答案：4

解題分析：

評估 (A > B): (3 > 4) 的結果是 false。

評估 (C < D): (4 < 5) 的結果是 true。

評估整個表達式: false AND true 的結果是 false。

44. 在例外狀況處理常式中，**finally** 段的用途為何？

- (1) 跳出錯誤處理常式
- (2) 結束應用程式的執行作業
- (3) 只有在擲回例外狀況時才執行程式碼
- (4) 不論是否擲回例外狀況都執行程式碼

答案：4

解題分析：

finally 區塊是例外狀況處理常式的一部分。無論 **try** 區塊中是否發生例外狀況，**finally** 區塊中的程式碼都會被執行。這通常用於釋放資源，例如關閉檔案或資料庫連線。

45. 有一個 C# 類別包含下列程式碼。

加上行號僅為參考之用。

```
01 public delegate string CheckCoding Style (string code);
02
03 public void CheckCode (string code, CheckCodingStyle algorithm)
04 {
05
06 }
```

您需要在第 05 行新增一個陳述式以呼叫 **CheckCodingStyle** 委派。

請問你該使用哪一個程式片段？

- (1) **a1gorithm** (**code**);
- (2) **algorithm** = new **CheckCoding Style**(**code**);
- (3) **algorithm**. **CheckCoding Style**(**code**);
- (4) **code**. **CheckCoding Style**(**algorithm**);

答案：1

解題分析：

在 C# 中，委派 (**delegate**) 是一種型別安全的函式指標。一旦委派被實例化並指向一個方法，就可以像呼叫普通方法一樣呼叫它。選項 (1)**algorithm**(**code**)；是正確的委派呼叫語法。

46. 請檢閱 Winter Sports Corporation 的下列軟體需求規格。

| 使用者介面 | 硬體介面 | 軟體效能與品質 |
|---|--|--|
| <ul style="list-style-type: none"> • 前端軟體：HTML、JavaScript 或 JavaScript 架構 • 資料庫：SQL • 後端軟體：.NET Core | <ul style="list-style-type: none"> • 行動裝置（回應式應用程式或透過網頁瀏覽器） • PC 和 Mac（透過網頁瀏覽器） | <ul style="list-style-type: none"> • 安全、可延伸且可擴充 • 將商務邏輯和展示功能隔開 |

根據所需的系統功能，您應該選擇哪一種架構？

- (1) **Model-View-Controller (MVC)**
- (2) Windows Forms
- (3) Entity Framework.NET Core
- (4) RESTful 應用程式開發介面(RESTful API)

答案：1

解題分析：

軟體需求規格中明確提到需要「將商務邏輯和展示功能隔開」。MVC 架構模式正是為此目的而設計的。

它將應用程式分為三個相互關聯的部分：模型 (**Model**)、檢視 (**View**) 和控制器 (**Controller**)，以實現業務邏輯與使用者介面的分離。

47. "有一個名為 **theStack** 的堆包含整數值，這些值會按照下列順序推入堆：**1**、**4**、**1**、**6**、**3**、**5**、**6**、

推入堆疊的第一個數字是 **1**。

您執行了下列程式碼：

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

程式碼執行完畢之後，**theStack** 將包含什麼？

- (1) 6、1、4、1
- (2) 6、5、3、6
- (3) 僅 1、4、1
- (4) 僅 6、5、3。

答案：1

解題分析：

初始堆疊 (Stack) : [1, 4, 1, 6, 3, 5, 6] (6 在最上面)。

執行 **for (int i=0; i<3; i++) theStack.pop();**。這表示會執行 3 次 **pop** 操作。

第一次

pop: 移除 6。堆疊: [1, 4, 1, 6, 3, 5].

第二次

pop: 移除 5。堆疊: [1, 4, 1, 6, 3].

第三次

pop: 移除 3。堆疊: [1, 4, 1, 6].

程式執行完畢後，堆疊中剩下

[1, 4, 1, 6]，但堆疊是後進先出，通常會從頂部顯示。因此，剩餘的元素為 6, 1, 4, 1 (從上到下)。

重覆 48->12

49. 您的國隊使用了 GitHub 做為版本控制系統。

請選取正確的選項以完成句子。

認可檔案的變更之前，必須先進行：_____

- (1) 複製
- (2) 派生
- (3) 提取
- (4) 暫存。

答案：4

解題分析：

在 Git 中，`commit` 是將變更永久儲存在本機存放庫中的動作。

在 `commit` 之前，您需要先使用 `git add` 指令將變更的檔案新增到暫存區（`staging area`）。

50. 您的回隊使用了 GitHub 做為版本控制系統。

請選取正確的選項以完成句子。

若要以非破壞性方式將某個開放原始碼專案複製到您的設定，可加以: _____

- (1)複製
- (2)派生
- (3)提取
- (4)暫存

答案 : 2

解題分析 :

在 GitHub 中，派生 (Fork) 是一個將遠端存放庫完整複製到您自己帳戶的操作。這個操作是非破壞性的，您可以在派生後的存放庫中自由修改，而不會影響原始專

51. 您的國隊使用了 GitHub 做為版本控制系統。

請選取正確的選項以完成句子。

若要從遠端存放庫中取得專案的最新修訂，請使用_____命令。

- (1)複製
- (2)派生
- (3)提取
- (4)暫存

答案 : 3

解題分析 :

`git fetch` (提取) 命令用於從遠端存放庫下載最新的變更，但不會自動將這些變更合併到您當前的工作分支。這允許您在合併之前先檢視最新的變更。

52. 請選取正確的選項以完成有關現代化加密的句子。數位簽章會使用: _____

- (1)非對稱加密
- (2)對稱加密

答案 : 1

解題分析 :

數位簽章是使用非對稱加密 (Asymmetric Encryption) 技術。

非對稱加密使用一對金鑰：一個私密金鑰 (private key) 和一個公開金鑰 (public key)。

53. 請選取正確的選項以完成有關現代化加密的句子。數位簽章需要: _____

- (1)私密金鑰
- (2)公開金鑰
- (3)私密和公開金鑰

答案 : 3

解題分析 :

數位簽章的產生需要使用私密金鑰 來簽署訊息，而驗證簽章則需要使用寄件者的公開金鑰。

因此，數位簽章的運作需要私密金鑰和公開金鑰這兩者。

54. 在類別中，建構函式的用途為何？

- (1)建立實值型別
- (2)自基底類別繼承
- (3)初始化該類別的物件
- (4)釋放類別持有的資源

答案 : 3

解題分析 :

建構函式 (Constructor) 是在建立類別的新實例 (物件) 時自動呼叫的特殊方法。

它的主要目的是初始化物件的欄位或屬性，並執行必要的設定。

55. 您正在為應用程式設計一個 C# 類別。

您需要確保 `access Count` 成員變數只有基底類別可用。

請問您應該使用哪個存取修飾詞？

- (1)internal
- (2)private
- (3)protected
- (4)public

答案 : 2

解題分析 :

`private` 存取修飾詞用於限制成員的存取，使其只能在宣告它的類別中存取。

衍生類別無法存取基底類別的 `private` 成員。這符合題目的要求。

`protected` 則允許衍生類別存取。

56. 下列程式碼會追蹤競賽選手排名的資訊：

```
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;
    }
}
```

呼叫包含三個參數的建構函式時，指派欄位的順序為何？

請將以下每個欄位正確排序

欄位

1 _defaultTeam

2 _rank

3 _name

(1)123 (2)321 (3)231 (4)132

答案：2

解題分析：

呼叫 `RankedPlayer (string name, string team, int rank)`。

這個建構函式使用 `this (name, rank)` 呼叫另一個建構函式，因此會先執行 `RankedPlayer (string name, int rank)`。

`RankedPlayer (string name, int rank)` 接著使用 `this (name)` 呼叫第三個建構函式。

`RankedPlayer (string name)` 賦予 `_name` 欄位值，這是第一個被賦值的欄位 (`3_name`)。

`RankedPlayer (string name, int rank)` 接著賦予 `_rank` 欄位值，這是第二個被賦值的欄位 (`2_rank`)。

最後，`RankedPlayer (string name, string team, int rank)` 賦予 `_defaultTeam` 欄位值，這是第三個被賦值的欄位 (`1_defaultTeam`)。

因此，指派順序為：

`_name (3)、_rank (2)、_defaultTeam (1)`。

57. 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 #tring Location get; met; }
    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？

(1)0
(2)1
 (3)2
(4)3

答案：3

解題分析：

程式碼執行 `Match myMatch = new Match();` 建立

`Match` 類別的實例。

`Match()` 建構函式被呼叫。在建構函式中，

`Location` 屬性被賦予值 "unknown"。這是第一次指派。

接著，程式碼執行

`myMatch.Location = "north region";`。這是第二次指派。

因此，

`Location` 屬性被指派了 2 次。

58. 有一個名為 `Mammal1` 的類別衍生自一個名為 `Animal` 的父類別。

`Mammal1` 類別具有 `Animal` 類別的屬性和方法。

哪一個詞能描述這種物件導向的概念？

(1)資料特殊化
(2)結構描述模型化
(3)封裝
 (4)繼承

答案：4

解題分析：

繼承 (Inheritance) 是物件導向程式設計的基本概念之一。

它允許一個類別（衍生類別）繼承另一個類別（基底類別或父類別）的屬性、方法和行為。

59. 下列何者必須存在，才能從特定類別繼承屬性？

- (1) has-a 關聯
- (2) is-a 關聯**
- (3) 靜態成員
- (4) 公用屬性。

答案：2

解題分析：

繼承（Inheritance）代表一種「is-a」（是一個）的關聯。例如，「哺乳動物是一個動物」（A mammal is an animal）。

has-a 關聯則表示一個類別包含另一個類別的實例作為其成員。

60. 有一個名為 Glass 的類別繼承自一個名為 Window 的基底類別。

Window 類別包含一個名為 break() 的受保護方法。

請問您該如何呼叫 break() 方法的 Glass 類別實作？

- (1) base. break();
- (2) this. break();**
- (3) Glass. break();
- (4) Window. break();

答案：2

解題分析：

在衍生類別中，可以使用 this 關鍵字來呼叫其自身的成員，這也包括從基底類別繼承而來的受保護方法。

base.break() 則是用來明確呼叫基底類別的實作。

61. 您建立了一個 ANumber 類型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);
```

程式碼執行完畢後，_number？

- (1) null
- (2) 0
- (3) 3**
- (4) 7

答案：3

解題分析：

物件建立：當執行 ANumber mynumber = new ANumber(3); 時，會建立一個 ANumber 類別的新物件。

建構函式呼叫：這個陳述式會呼叫帶有一個整數參數的建構函式 public ANumber (int number)。

參數傳遞：傳遞給建構函式的參數值為 3，所以此時 number 參數的值為 3。

欄位賦值：在 ANumber(int number) 建構函式內，程式碼執行 _number = number;。

這行程式碼將 number 參數的值（即 3）賦予給 _number 欄位。

最終結果：因此，在程式碼執行完畢後，mynumber 物件中的 _number 欄位的值變為 3。

62. Circle 和 Square 這兩個類別繼承自 Shape 類別。

Circle 和 Square 都自 Shape 類別繼承 Area，但分別以不同的方式計算 Area。

哪一個詞能描述這種物件導向的概念？

- (1) 封裝
- (2) 抽象化
- (3) 多型**
- (4) 超級類別化

答案：3

解題分析：

多型（Polymorphism）是物件導向程式設計的一個原則，它允許相同的方法名稱可以根據其被呼叫的物件類型執行不同的行為。

在 C# 中，這通常是透過虛擬方法（virtual method）和方法覆寫（override）來實現的。

64. 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？

- (1) 批次處理
- (2) 用戶端
- (3) 多執行緒
- (4) 雖服器端**

答案：4

解題分析：

房間預訂的可用性是需要處理敏感資料（例如房間狀態）的業務邏輯。

這種邏輯必須在伺服器端（Server-side）執行，以確保資料的完整性、一致性和安全性。

在用戶端（Client-side）執行邏輯是不安全的，因為用戶可以輕易修改或繞過它。

70. 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?

- (1) <https://sunsetweb.org/images/Overview.jpg>
- (2) <http://sunsetweb.org/images/Overview.jpg>
- (3) <http://sunsetweb.org/myapplication/Images/Overview.jpg>
- (4) <https://sunsetweb.org/Storage/Images/Overview.jpg>

答案：4

解題分析：

網站需要 SSL，因此 URL 必須使用 https。

虛擬目錄名稱是 Storage。

別名是 Images。

檔案名稱是 Overview.jpg。

因此，正確的 URL 結構應該是

<https://<網域名稱>/<虛擬目錄別名>/<檔案名稱>>。

根據圖中的虛擬目錄設定，虛擬目錄別名是 images，實際路徑是 I:\Storage\Images。

但根據圖和選項，最符合的 URL 應該是 <https://sunsetweb.org/Storage/Images/Overview.jpg>。



71. 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"});
    [WebGet (UriTemplate="/Car Services")]
    public String GetAllCarServices()
    {
        int count = lst.Count;
        String ServiceList = "";
        for (int i = 0; i < count; i++)
            ServiceList = ServiceList + lst[i] + ",";
        return ServiceList;
    }
    [WebGet (UriTemplate = "/Car Services/ {ServiceListid}")]
    public String GetCarServicebyID (String ServiceListid)
    {
        int id;
        Int32.TryParse (ServiceListid, out id);
        return lst[id];
    }
}
```

您對這個 web 服務發出了下列呼叫：

www.sunsetweb.com/carservices/3

請問輸出為何？

- (1) Paint
- (2) Oil
- (3) Engine
- (4) Wheels

答案：1

解題分析：

呼叫的 URL 是 www.sunsetweb.com/carservices/3。

這會對應到

[WebGet (UriTemplate = "/Car Services/ {ServiceListid}")] 這個方法。

ServiceListid 的值是 3。

程式碼將 ServiceListid 轉換為整數 id。

id 的值為 3。

接著，程式碼返回 lst[id]。

lst 是一個包含 {"Oil", "Engine", "Wheels", "Paint"} 的清單。

清單的索引從 0 開始。lst[0] 是 "Oil"，lst[1] 是 "Engine"，lst[2] 是 "Wheels"，lst[3] 是 "Paint"。

因此，lst[3] 的值是 "Paint"。

重覆 72->31、73->32

74. 哪一個正規化要求資料表的屬性僅相依於主索引鍵？

- (1) 第一
- (2) 第二
- (3) 第三
- (4) 第四

答案：2

解題分析：

第二正規化 (Second Normal Form, 2NF) 要求資料表中的所有非主鍵屬性都必須完全函數相依於整個主索引鍵。如果主鍵是複合鍵，則不能只相依於主鍵的一部分。

75. 您需要建立一個傳入人員 UserName 和 Age 的預存程序。

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序？

- (1) "CREATE PROCEDURE StoreData @UserName nvarchar (255),@Age int"
- (2) "CREATE PROCEDURE StoreData @UserName char,@Age double"
- (3) "UPDATE PROCEDURE StoreData @UserName nvarchar (255),@Age int"
- (4) "CREATE PROCEDURE StoreData (@UserName,@Age)"

答案：1

解題分析：

CREATE PROCEDURE 是用於建立預存程序的 SQL 陳述式。

@UserName 參數需要能容納至少三個字元，`nvarchar(255)` 是一個適合的字串類型 。

@Age 參數需要是一個以年為單位的整數，`int` 類型是正確的選擇 。

76. 在更新資料庫的多個資料表時，使用交易有何優點？

- (1) 更新不會失敗。
- (2) 更新會以單位的形式成功或失敗。**
- (3) 更新會盡快完成。
- (4) "如果發生錯誤，將無法回復變更。"

答案：2

解題分析：

交易（**Transaction**）是一組被視為單一工作單元的資料庫操作。交易的核心特性是原子性（**Atomicity**），這意味著交易中的所有操作要么全部成功提交，要么全部失敗並回滾。這可以防止資料不一致或資料損壞。

77. 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？

- (1) 連線共用**
- (2) 連線逾時
- (3) 索引
- (4) 正規化。"

答案：1

解題分析：

連線共用（**Connection Pooling**）是一種技術，它會建立並維護一組資料庫連線，並在應用程式需要時將這些連線重複使用。這減少了為每個新要求建立和關閉連線的開銷，從而提高了效能和可擴縮性，尤其是在連線數目有限的伺服器上。

78. 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。）

- (1) 測試案例必要的任何前置條件**
- (2) 測試案例的預期結果**
- (3) 與其他測試案例的整合
- (4) 結合成單一測試的所有使用案例
- (5) 沒有任何判斷提示的測試案例
- (6) 以上皆非

答案：1、2

解題分析：

一個有效的單元測試案例應該包含測試前的**前置條件**（**pre-conditions**），這是執行測試所需的初始狀態。此外，它還必須包含**預期結果**（**expected results**），這是測試成功後應該得到的輸出或狀態。

79. 下列哪兩項敘述正確描述了 JSON 物件？（請選擇 2 個答案。）

- (1) JSON 物件可使用名稱/值對來建置。**
- (2) JSON 物件可使用已排序的值清單來建置。**
- (3) JSON 物件可運用命名空間。
- (4) JSON 物件可延伸。
- (5) 以上皆是
- (6) 以上皆非

答案：1、2

解題分析：

JSON（**JavaScript Object Notation**）物件可以透過兩種方式建置。一種是使用**名稱/值對**（**name/value pairs**）來表示，這對應於 **JavaScript** 的物件。另一種是使用**已排序的值清單**（**an ordered list of values**），這對應於 **JavaScript** 的陣列。

80. 下列哪兩項敘述描述了將資料庫正規化的優點？（請選擇 2 個答案。）

- (1) 正規化可降低磁碟空間使用量。**
- (2) 正規化可防止資料不一致。**
- (3) 正規化可將資料損毀的影響降到最低。
- (4) 正規化可減少結構描述限制。
- (5) 以上皆是
- (6) 以上皆非

答案：1、2

解題分析：

資料庫正規化（**Normalization**）是一種設計技術，旨在組織資料表中的欄位和資料表本身，以最小化資料冗餘。這有兩個主要優點：
可降低磁碟空間使用量：因為減少了重複資料。
可防止資料不一致：因為每個事實只會儲存一次，從而避免了更新異常。

81. 連線共用有哪三項優點？（請選擇 3 個答案。）

- (1) 提升效能**
- (2) 提升可擴縮性**
- (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

答案：1、2、3

解題分析：

連線共用的主要優點包括：

提升效能：減少了建立和關閉連線的開銷。

提升可擴縮性：允許應用程式處理更多同時連線的使用者。

縮短建立連線的時間：因為連線已經存在於共用池中。

82. 哪兩個程式碼片段可用來確保資源會釋出?(請選擇 2 個答案。)

- (1) **using** 陳述式
- (2) **using** 指示詞
- (3) **finally** 區塊
- (4) **throw** 關鍵字
- (5) 以上皆是
- (6) 以上皆非

答案 : 1、3

解題分析 :

using 陳述式和 **finally** 區塊都可以用來確保資源在執行完畢後被釋放，即使發生例外狀況也是如此。

using 陳述式特別適用於實作 **IDisposable** 介面的物件，它會自動呼叫 **Dispose()** 方法。

finally 區塊則用於在 **try** 區塊結束後執行清理程式碼。

重複 83.->78.、84.->79.、85.->81

86. 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。

答案 : x

解題分析 :

單元測試 (Unit Test) 的目標是隔離並測試程式碼中最小的可測試部分(單元)，而不是整個應用程式的整合和相依性。

測試整合和相依性是整合測試 (**Integration Test**) 的範疇。

87. 請回答有關單元測試的敘述:

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。

答案 : ○

解題分析 :

這正是單元測試的典型模式。

測試者會建立一個測試方法，呼叫要測試的程式碼單元，傳入特定的輸入資料，然後使用斷言 (assertion) 來驗證輸出的結果是否符合預期。

88. 請回答有關單元測試的敘述:

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。

答案 : ○

解題分析 :

這段敘述精確地描述了單元測試的核心目標。

其重點在於隔離測試，確保每個程式碼單元都能獨立地正常運作。

89. 請回答有關連結清單的敘述:

連結清單可進行排序。

答案 : ○

解題分析 :

雖然連結清單在資料結構上不天然具備排序功能，但可以對其進行排序操作，例如在插入新節點時維持順序，或者在現有清單上執行排序演算法。

因此，這個敘述是正確的。

90. 請回答有關連結清單的敘述:

連結清單最多包含 100 個節點。

答案 : x

解題分析 :

連結清單 (**Linked List**) 的節點數目沒有固定的上限。

它的長度是動態的，取決於可用的記憶體，可以包含任意數量的節點。

91. 請回答有關連結清單的敘述:

您可以在連結清單內任何位置加入新的節點。

答案 : ○

解題分析 :

連結清單的主要優點之一就是可以非常有效率地在清單的任何位置(開頭、結尾或中間)插入或刪除節點。

這與陣列不同，陣列在中間插入或刪除元素需要移動後續的所有元素。

92. 請回答有關安全編碼做法的敘述:

數位簽章會使用寄件者的公開金鑰來加密訊息。

答案 : x

解題分析 :

數位簽章的產生是使用寄件者的**私密金鑰**來加密訊息的雜湊值。

收件者使用寄件者的公開金鑰來解密簽章並驗證其真實性。

93. 請回答有關安全編碼做法的敘述:

CSRF(跨網站偽造要求) 權杖是用來防止攻擊。

答案 : ○

解題分析 :

CSRF (Cross-Site Request Forgery) 權杖是用來防止 **CSRF** 攻擊的常見安全機制。

它是一個隨機生成的、唯一的權杖，在表單提交時傳送到伺服器，伺服器會驗證這個權杖以確保請求來自於合法的網站，而不是惡意的第三方網站。

94. 請回答有關安全編碼做法的敘述:

防止 SQL 插入式攻擊的其中一種方式就是逸出使用者輸入。

答案 : ○

解題分析：

SQL 插入式攻擊（SQL Injection）是透過惡意使用者輸入來操縱資料庫查詢。

逸出（Escaping）使用者輸入是一種防止此類攻擊的常見做法，它會將輸入中的特殊字元（例如引號）轉換為不會被資料庫解釋為命令的一部分的字串。

95. 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?部屬性，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 color 屬性。

答案：

解題分析：

`internal` 存取修飾詞允許在同一個組件（assembly）中的任何類別存取該成員。

如果 Person 類別和 Tree 類別在同一個專案（組件）中，那麼 Person 類別的方法就可以存取 Tree 類別的 color 屬性。

96. 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?部屬性，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。

答案：

解題分析：

`protected` 存取修飾詞只允許在基底類別本身以及衍生自它的類別中存取成員。

如果 Person 類別沒有繼承自 Tree 類別，它就無法存取 Tree 類別的 `protected` 成員 NumberOfLeaves。

97. 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?部屬性，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。

答案：

解題分析：

`private` 存取修飾詞將成員的存取權限嚴格限制在其宣告的類別中。

Tree 類別無法存取 Person 類別中的任何 `private` 屬性。

98. 請回答有關程式碼的敘述：

加上行號僅為考之用。

```

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>Entree: ' + responseObject.menu[i].Entree + '<br>';
08             newContent += 'Price: responseObject.menu [i].Price + '</p>';
09         }
10         document.getElementById('content').innerHTML = newContent;
11     }
12 };
13 xhr. open('GET', menu. json', true);
14 xhr. send(null);

```

在第 03 行，`onload` 物件會將要求傳送至伺服器。

答案：

解題分析：

`onload` 是一個事件處理函式。

它定義了在伺服器回傳資料並成功載入之後要執行的程式碼。

真正將要求傳送到伺服器的是 `xhr.send(null)`。

99. 請回答有關程式碼的敘述：

加上行號僅為考之用。

```

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>Entree: ' + responseObject.menu[i].Entree + '<br>';
08             newContent += 'Price: responseObject.menu [i].Price + '</p>';
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。

答案：

解題分析：

`xhr.status` 屬性是由伺服器傳回的 HTTP 狀態碼。

例如，200 表示成功，404 表示找不到頁面等。

100. 請回答有關程式碼的敘述：

加上行號僅為考之用。

```

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>Entree: ' + responseObject.menu[i].Entree + '<br>';
08             newContent += 'Price: ' + responseObject.menu[i].Price + '</p>';
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` 參數會指定非同步處理

答案：

解題分析：

`xhr.open()` 方法的第三個參數是一個布林值，用於指定請求是否為非同步（**asynchronous**）。當此參數為 `true` 時，請求將以非同步方式處理，這意味著程式碼會繼續執行而無需等待伺服器的回應。

101. 請回答有關佇列的敘述：

佇列的項目數有限制。

答案：

解題分析：

佇列（Queue）是一種動態資料結構。其大小是動態的，通常只受限於系統的可用記憶體，並沒有固定的項目數限制。

102. 請回答有關佇列的敘述：

佇列是一種先進先出（FIFO）資料結構。

答案：

解題分析：

佇列（Queue）的運作原則是先進先出（First-In, First-Out, FIFO）。第一個加入佇列的項目將會是第一個被移除的項目。

103 請回答有關佇列的敘述：

佇列的項目是按照加入的順序儲存。

答案：

解題分析：

由於佇列是先進先出的資料結構，項目會按照它們被加入的順序進行儲存和處理。

104. 請回答有關 NoSQL 資料庫的敘述：

文件資料庫會將資料儲存在類似 JSON 的檔案中。

答案：

解題分析：

文件資料庫（Document Database）是一種 NoSQL 資料庫。它將資料儲存為半結構化的文件，通常是 JSON 或類似 JSON 的格式。

105. 請回答有關 NoSQL 資料庫的敘述：

文件資料庫的設計目的是以預先定義的關聯處理資料。

答案：

解題分析：

文件資料庫的優點是其彈性的架構。與關聯式資料庫不同，它沒有預先定義的表格結構或嚴格的關聯。

106 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。

答案：

解題分析：

索引鍵值資料庫（Key-Value Database）是將資料儲存為一個簡單的鍵值對集合，它不提供複雜的查詢或修改屬性的功能。

可以修改或查詢屬性的物件通常是文件資料庫的特

107. 實體關聯圖表(ERD)是專為物件導向的資料庫所設計。

答案：

解題分析：

實體關聯圖（Entity-Relationship Diagram, ERD）是用來描述關聯式資料庫（Relational Database）的設計。

它顯示了資料庫中的實體、它們的屬性和它們之間的關係。

108. 實體關聯圖表(ERD)包含實體、關聯和屬性。

答案：

解題分析：

實體關聯圖（ERD）的核心組成部分就是實體（Entity）、它們之間的關聯（Relationship）以及每個實體的屬性（Attribute）。

109. 基數是指資料表的互相連結方式，例如一對多。

答案：

解題分析：

基數（Cardinality）在資料庫設計中用來描述實體之間關係的數量級。

例如，一對一（one-to-one）、一對多（one-to-many）、多對多（many-to-many）等。