

## 111 年特種考試地方政府公務人員考試試題

等 別：三等考試  
類 科：資訊處理  
科 目：系統分析與設計

曹勝老師 解題

一、內聚力 (Cohesion) 與耦合力 (Coupling) 是評估系統設計的重要指標，請使用結構圖 (structure chart) 分別繪製高內聚力 (high cohesion)、鬆散耦合 (loosely coupled) 的例子，並用文字說明這些例子。(20 分)

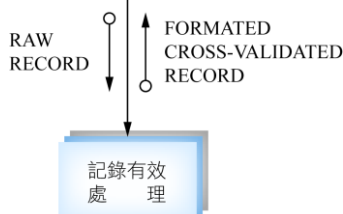
1. 《考題難易》：★★

2. 《解題關鍵》：系統設計概念題，只要了解各種內聚力與耦合力狀況即可作答。

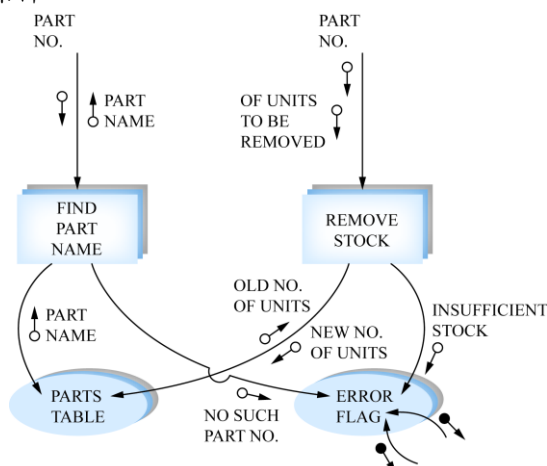
【擬答】：

(一)高內聚力如循序內聚力 (sequential cohesion)，是在同一模組內包含有兩個 (含) 以上的功能或任務，它們間的關係有如生產線上的連續動作，前一功能的輸出恰好是下一功能的輸入，彼此同心協力，一起完成一些性質類似而又不能劃歸於同一功能的工作。循序內聚力之模組有好的耦合力，將來容易維護，但它並未如功能內聚力那般富有可讀性 (readability)、維護性 (maintainable) 以及重覆使用性 (reusable)，因為它所執行的功能並非系統內其他部份均能恰好同時共用的。下圖中 uses raw record、format raw record、cross validate fields in raw record 以及 returns formatted cross validated record 均在“記錄有效處理”這個模組內順序完成而不能倒置。

圖 循序內聚力



(二)鬆散耦合如(四)共用耦合力 (common coupling)，兩個模組如果共同使用到一個資料區，可稱其為共用耦合力。有些語言允許程式內定義一共用區域 (Common Area)，給自己和和其他模組使用，透過共用區域，模組間也可以互傳資料。如 COBOL 的資料部 (DATA DIVISION)，它會被程序部 (PROCEDURE DIVISION) 內許多的段落 (PARAGRAPH) 所共用。下圖即為共用耦合力例子，其中 PARTS TABLE 與 ERROR FLAG 同為上面二個模組共用，即合乎此條件。



- 二、擔任軟體採購規劃的工作，必須著重委外資安議題的思考，(一)請舉例詳細說明服務水準協議 SLA (Service Level Agreement) 在資訊系統服務與營運中扮演的角色。(10 分)(二)請舉 5 個重要措施以降低委外開發、測試及運作時的風險。(10 分)

1.《考題難易》：★★

2.《解題關鍵》：委外資安應用題，了解委外資安 SLA 與控制措施即可作答。

【擬答】：

(一)服務級別協定 (service-level agreement, SLA) 也稱服務等級協定、服務水平協定，是服務提供商與客戶之間定義的正式承諾。服務提供商與受服務使用者之間具體達成了承諾的服務指標——品質、可用性，責任。[1]SLA 最常見的組成部分是以合約約定向客戶提供的服務。例如，網際網路服務供應商 (ISP) 和電信公司通常在與客戶的合約條款內包含簡單定義的服務級別協定。在此事例下，SLA 通常定義有平均故障間隔 (MTBF) 或平均修復時間 (MTTR)；哪一方負責報告錯誤與支付費用；吞吐量；抖動；或類似的可衡量細節。定義明確的典型 SLA 將包含以下組件：要提供的服務類型、服務所需的性能水平，尤其是其可靠性和響應能力、監控流程和服務水平報告、報告服務問題的步驟、響應和問題解決時間範圍、服務提供商未履行其承諾的後果。例如骨幹互聯網供應商在其網站上明確聲明其 SLA 的情況並不少見。雲端運算的 SLA 側重於數據中心的特性，包括網絡特性以支持端到端 SLA。外包涉及將責任從組織轉移到供應商。這種新安排通過可能包含一個或多個 SLA 的合同進行管理。合同可能涉及經濟處罰以及如果始終未達到任何 SLA 指標則終止合同的權利。設置、跟踪和管理 SLA 是外包關係管理(ORM) 規程的重要組成部分。特定的 SLA 通常作為外包合同的一部分預先協商，並用作外包治理的主要工具之一。

(二)可以採用 ISO27001 的控制措施以降低委外開發、測試及運作時的風險

1. 文件化運作程序

運作程序應加以文件化，並使所有有需要之使用者均可取得。確認相關作業系統的相關政策或規範，如：更新政策、備份政策、效能監控...等等。

2. 變更管理

應控制對影響資訊安全之組織、營運過程、資訊處理設施及系統的變更。除了自行開發的應用系統(A.14.2.2) 程式上版的變更管理，其他相關作業系統、作業程序、防火牆、白名單軟體、軟體設定的變更都應計入變更管理。變更管理流程：

- (1)評估資產環境
- (2)發現新的補丁
- (3)測試驗證和計畫佈署
- (4)佈署已核准的補丁

3. 容量管理

各項資源之使用應受監視及調適，並對未來容量要求預作規劃，以確保所要求之系統效能。一般來說會做監控或是未來的容量需求規劃考量，如，網路傳輸量、CPU 使用率、硬碟空間、記憶體容量...任何需要關注的項目，且最好可以發出告警訊息大部份會使用 Network Monitor 來監控，如：Prometheus、PRTG

4. 開發、測試及運作環境之區隔

應區隔開發、測試及運作之環境，以降低對運作環境未經授權存取或變更的風險。正式區跟測試區有沒有區隔，並說明如果沒有區隔會造成什麼樣的風險。在上更新時，如果

直接上可能會導致作系統存可用性失效，所以會在測試區測試完再上正式區。如果沒有測試區，那麼會有虛擬環境 或是其他上版前測試的機制嗎？如 Docker？或開發環境？

#### 5. 資訊備份

應依議定之備份政策，定期取得資訊、軟體及系統的影像備份複本，並測試之。實際驗證是否依**政策或作業規範**抽驗實際上是否完成備份，如果有一併驗證【營運持續管理】，會進一步確認是否有相應的回復演練流程？



志光保成學儒

## 真的有輕鬆考取的方法！

### 掌握上榜8大招

|                                                                                                                       |                                                                                                                   |                                                                                                                       |                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <br><b>法科架構班</b><br>結合實務例子<br>建構法科概念 | <br><b>扎實正規班</b><br>完整堂數<br>循序漸進 | <br><b>工科全科班</b><br>公職+國營<br>一次到位    | <br><b>作文實戰班</b><br>強化寫作架構<br>理清邏輯概念  |
| <br><b>主題題庫班</b><br>主題教學<br>考點分析     | <br><b>精華總複習</b><br>掌握考點<br>增強實力 | <br><b>全真模擬考</b><br>比照真實考試<br>檢視應考實力 | <br><b>考前關懷講座</b><br>名師最終提點<br>觀念更加清晰 |

三、你與軟體開發專案夥伴們，希望能追蹤從開始到定案的所有檔案版本，以確保能方便多人共用，順利完成專案。(一)請列出版本控制系統 (Version Control System) 重要的三個特色，並說明之。(10 分)(二)目前主流的版本控制系統的架構有集中式與分散式兩種，請分別舉例詳細說明。(20 分)

1. 《考題難易》：★★★

2. 《解題關鍵》：版本控制系統概念題，了解區間堆積資料結構始可作答。

【擬答】：

(一)在版本控制系統中，最重要的，莫過於「**倉庫(Repository)**」，在 Repository 中，會記錄所有的版本資訊、**分支(Branch)**、**合併(Merge)** 等等資料。而版控所有的指令，都是建立於系統對 Repository 的操作行為。版本控制系統，也一定存在**同步、追溯與檔案備份**，這三種特性：

1. 同步 - 確保每個使用者，最終取回的內容都是相同的。

2. 追溯 - 可以知道所有版本間的變動項目與原因，也能回復到歷史版本中的任一版本。

3. 檔案備份 - 因為將所有的變動都同步到 Repository，間接的達到備份的功用。

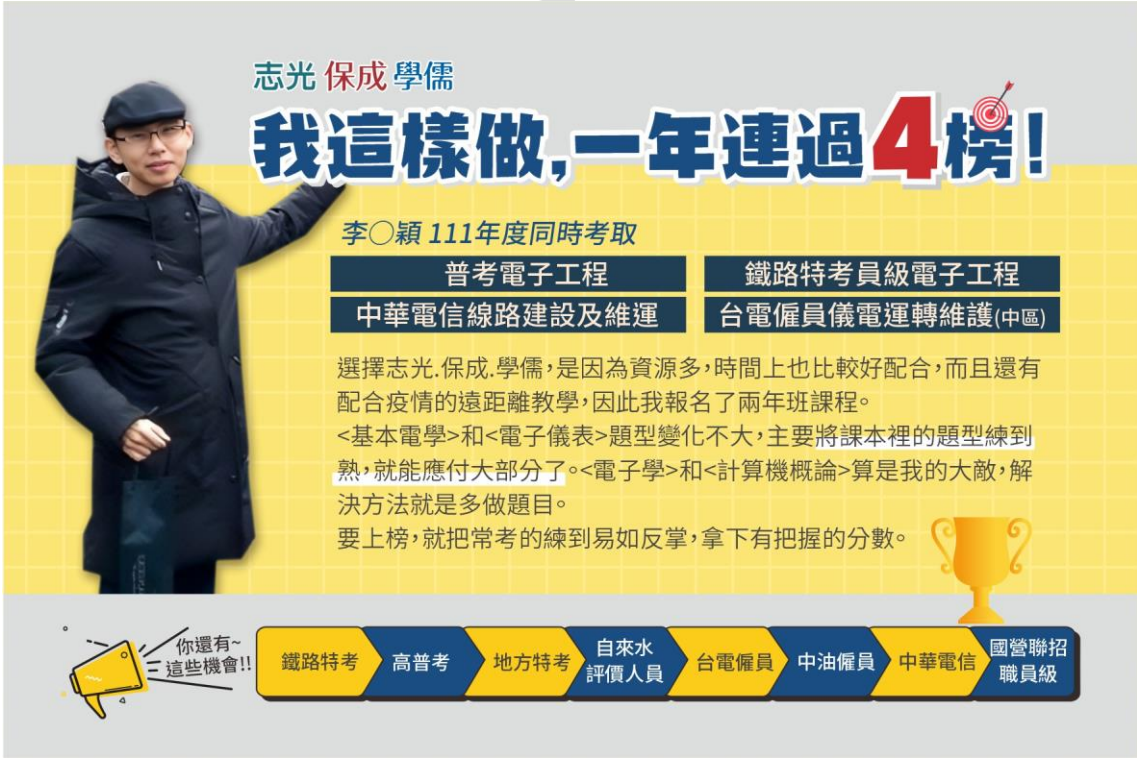
(二)集中式的版本控制系統，就是基於同個團隊之間合作的方式，是共用同一個 Repository。系統必須確保每位使用者都需與 Repository 保持一致。為要維持使用者之間保持同步的狀態，分為鎖定模式與合併模式兩種作法。鎖定模式是當使用者想要修改某檔案、簽出該檔案後，該檔案便會進入鎖定狀態，其他使用成員便無法加以修改，直到簽出者將該檔簽回為止。對於維持同步來說，這當然是一個十分保險的作法，因為永遠不會有兩個或以上的使用者同時修改同一個檔案。只是，這種方法造成了使用者對於檔案修改的互斥效應，使得使用效率受到影響。合併模式則允許多位使用者同時針對同一檔案進行修改，當他們分

## 公職王歷屆試題 (111 地方特考)

別將檔案提交回集中的檔案庫時，若發生衝突的情況，便會自動進行合併，而若自動合併失敗，再要求人工進行衝突的調解。如 CVS 即為集中式版本控制系統。

### (三)分散式版本控制系統 (Distributed VCS)

分散式的版本控制系統允許 Repository 存在一份或多份。每個使用者，都可以在自己電腦中，建立 Repository。對於分散式版本控制系統而言，讓每個使用者都可以修改各自的 Repository，享受版本控制系統的支援，而不受其他因素的限制。同時，運用 Push、Pull 的動作，使用者之間可分享自己的變更內容，達到同步的結果。因此，為了更有效的同步版本內容，分散式版本控制系統就更重視 Branch、Merge 的支援與功能。如 Git 就是最有名的開放原始碼分散式版本控制系統。



志光 保成 學儒

# 我這樣做,一年連過4榜!

李○穎 111年度同時考取

|             |                |
|-------------|----------------|
| 普考電子工程      | 鐵路特考員級電子工程     |
| 中華電信線路建設及維護 | 台電僱員儀電運轉維護(中區) |

選擇志光.保成.學儒,是因為資源多,時間上也比較好配合,而且還有配合疫情的遠距離教學,因此我報名了兩年班課程。

<基本電學>和<電子儀表>題型變化不大,主要將課本裡的題型練到熟,就能應付大部分了。<電子學>和<計算機概論>算是我的大敵,解決方法就是多做題目。

要上榜,就把常考的練到易如反掌,拿下有把握的分數。

你還有~這些機會!!

鐵路特考 高普考 地方特考 自來水評價人員 台電僱員 中油僱員 中華電信 國營聯招職員級

四、假設 ABC 專案採用敏捷方法開發，而你必須向其他成員說明敏捷開發的需求收集方式，請詳細說明下列三者，何謂使用者故事（user story）、情境（scenario）、故事板（storyboard）？並分別舉一實例。（30 分）

1. 《考題難易》：★★★★

2. 《解題關鍵》：敏捷方法需求收集實作題，必須了解使用者故事、情境、故事板等方法始可作答。

【擬答】：

#### (一)用戶故事 (User story)

是軟體開發和專案管理中的常用術語。主旨是以日常語言或商務用語撰寫句子，是一段簡單的功能表述。以客戶或使用者的觀點撰寫下有價值的功能、引導、框架來與使用者進行互動，進而推動工作進程。可以被認為是一種規格文件，但更精確而言，它代表客戶的需求與方向。以該用戶故事來反應對象在組織內的其工作職責、範圍、需要進行的任務等。用戶故事在敏捷開發方法中用來定義系統需要提供的功能和實現需求管理。例如用 epic story 方式建立篩選測驗：作為人力資源經理，我想創建一個篩選測驗，以便了解我是否想將可能的應聘者發送給職能經理。

#### (二)情境 (scenario)

描述了一個或多個人或組織如何與系統互動，描述了互動過程中發生的步驟、事件和/或動作。情境通常以不同的方式應用於多個開發過程。在統一過程 (UP) 的衍生產品中，它們

用於幫助從使用案例到循序圖，基本策略是識別通過使用案例的一部分的路徑，然後將情境編寫為該路徑的實例。例如，“提取資金”使用案例的文本將指示當一切正常時應該發生什麼，在這種情況下，資金存在於帳戶中並且 ATM 有資金。此為正常路徑。使用案例將包括替代路徑，描述發生錯誤時會發生什麼，例如帳戶資金不足或 ATM 機現金不足無法支付給客戶。此時的情境如下：

小華將她的銀行卡放入自動取款機。

小華使用她的個人識別號碼成功登錄了 ATM。

小華將薪水存入她的儲蓄帳戶。

小華從她的儲蓄帳戶中支付了電話費、電費、有線電視費和水費。

小華週末想從她的儲蓄帳戶中提取 100 美元，但發現她沒有足夠的資金

小華取出 40 美元並取回她的卡

### (三)故事板 (storyboard)

是一種可視化的溝通方法，將用戶故事（user stories）按照步驟畫出來，然後排列在一起，以此表達用戶如何使用一款產品（如網站或 APP）。製作故事板首先要確定三要素：角色、場景和情節。角色：確定一個故事的主人公，主人公的形象最好是你之前就整理好的用戶畫像，必須是基於真實的用戶建模而非臆想。他的行為、習慣、偏好、期望等因素都將決定着他們如何在使用產品時進行決策，因而人物的行為越是接近真實的用戶畫像，就越是能幫助你挖掘產品中的體驗問題。場景：角色不能是孤立存在的，需要一個承載角色形象，側面烘托人物性格的背景和環境，包括線上環境（網絡、應用程序等）和線下環境（地點，周圍的人物）。場景信息是故事板的背後承載信息，通常在故事板之前進行交代，而在故事板中則不需要詳細描述。情節：確定故事的主線劇情，通常會包含起因，經過和結果，即用戶目標、觸發事件、行為流程、行為結果等。在故事板中的劇情描述要簡單、清晰、易懂，並且要重點圍繞角色和角色主要目標展開。跟主要目標無關的情節都是耍流氓。在行為流程中，明顯的起承轉合能讓一個故事生動有趣，你可以借鑑一些講故事的套路，例如美國的超級英雄套路：主角是個普通人—大 Boss 發難要毀滅地球—主角被大 Boss 虐得體無完膚希望渺茫—忽然偶遇某些契機小宇宙爆發——打敗大 Boss 維護世界和平。

志光保成學儒 機械工程 | 電子工程 | 電力工程 | 資訊處理

# 一起站上工科勝利頂點

考取菁英 強勢佔榜

狀元

【全國狀元】111高 考電子工程-洪○銓  
 【竹苗區狀元】110地特四等電子工程-詹○凱  
 【台北市狀元】110地特四等資訊處理-于○  
 【台中市狀元】110地特四等電力工程-柯○訓  
 【金門縣狀元】110地特四等資訊處理-吳○展

【花東區第四】110地特三等資訊處理-羅○哲  
 【桃園市第四】110地特三等資訊處理-丁○妮  
 【高雄市第四】110地特四等電力工程-盧○源  
 【高雄市第六】110地特四等電力工程-蘇○禎

榜眼

【全國榜眼】111普 考資訊處理-羅○昌  
 【高雄市榜眼】110地特三等電力工程-江○展  
 【高雄市榜眼】110地特四等電子工程-曾○富  
 【台北市探花】110地特三等電力工程-黃○任  
 【台北市探花】110地特五等電子工程-柯○輝

探花

【全國第七】111普 考電子工程-卓○倫  
 【全國第七】111初 等 考電子工程-柯○輝  
 【桃園市第七】110地特三等電力工程-張○培  
 【全國第八】111高 考機械工程-江○禾

【全國第八】111普考電力工程-陳○璋  
 【全國第八】111普考電子工程-李○穎  
 【全國第九】111普考機械工程-施○佑

版面有限 無法一一刊登

單一年度優秀考取

高考資訊處理 賴○全; 高考資訊處理 郭○榕; 普考資訊處理 劉○廷; 普考資訊處理 賴○全; 高考電力工程 曾○倫; 高考電力工程 王○格  
 高考資訊處理 黃○迪; 高考資訊處理 廖○仲; 普考資訊處理 張○偉; 普考資訊處理 張○慧; 高考電力工程 鄧○駿; 高考電力工程 吳○瑋; 高考電子工程 莊○雪  
 高考資訊處理 張○偉; 高考資訊處理 羅○昌; 普考資訊處理 褚○華; 普考資訊處理 劉○銘; 高考電力工程 葛○宇; 高考電力工程 蔡○昇; 普考電子工程 馮○恩  
 高考資訊處理 郭○祐; 普考資訊處理 劉○廷; 普考資訊處理 李○庭; 普考資訊處理 陳○堂; 高考電力工程 陳○瑋; 普考電力工程 吳○瑋; 普考電子工程 蔣○森  
 高考資訊處理 胡○軀; 高考資訊處理 李○庭; 普考資訊處理 陳○明; 普考資訊處理 廖○仲; 高考電力工程 王○甯; 普考電力工程 吳○哲; 高考機械工程 黃○榮  
 高考資訊處理 許○傑; 高考資訊處理 于○瑋; 普考資訊處理 鄭○然; 高考電力工程 蔡○鎮; 高考電力工程 梁○豐; 普考電力工程 梁○豐; 普考機械工程 江○禾  
 高考資訊處理 陳○廷; 高考資訊處理 于○; 普考資訊處理 吳○翰; 高考電力工程 李○源; 高考電力工程 席○崇; 高考電力工程 卓○心; 普考機械工程 金○璋  
 高考資訊處理 陳○明; 普考資訊處理 黃○迪; 普考資訊處理 曾○瑋; 高考電力工程 丁○翔; 高考電力工程 吳○哲;

版面有限 無法一一刊登