

《系統分析與設計》

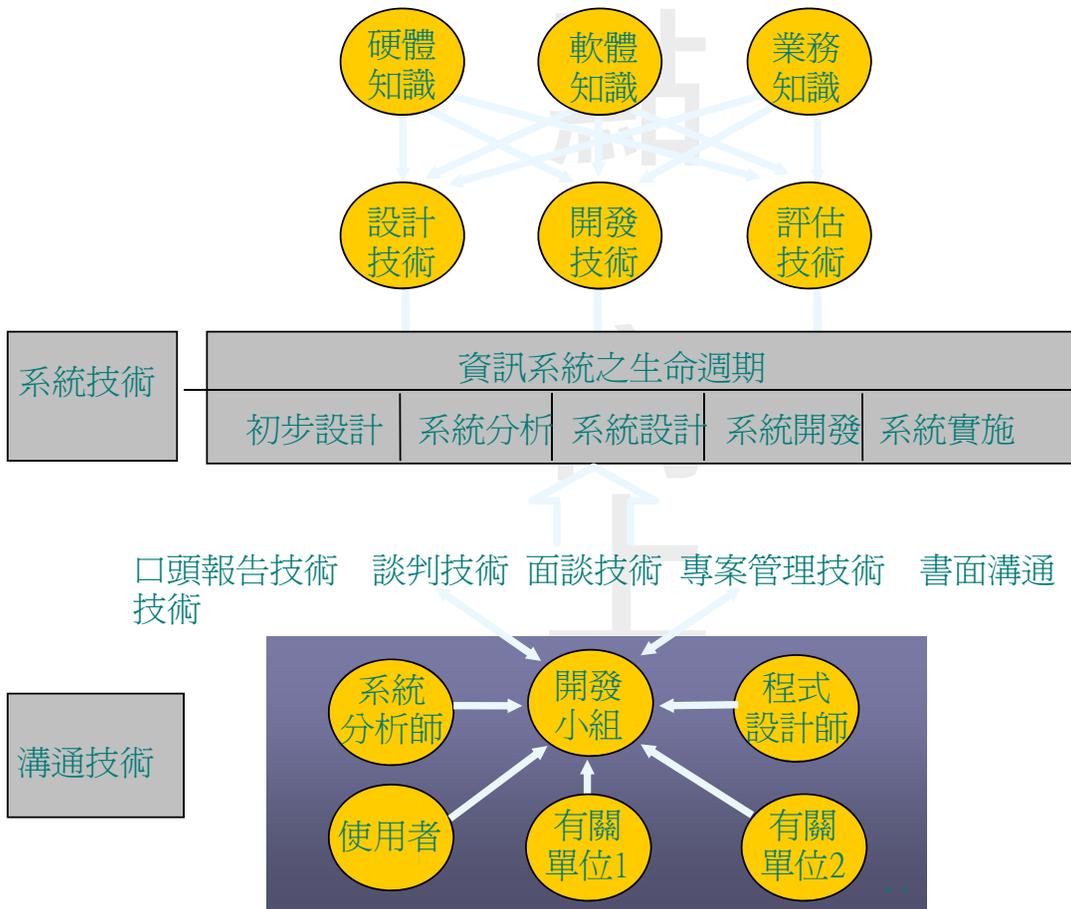
一、請舉例詳述一位理想的系統分析師 (System Analyst) 應具備那些面向的能力及其理由。此外，應如何進行資訊系統專案之成本效益分析 (Cost/Benefit Analysis) 之作法。(20分)

試題評析 本題為系統分析基礎跨專案管理，為記憶題型，可由講義內容著手作答。

考點命中 《高點系統分析與設計講義第一回》，張又中編撰，頁1-4~5及上課補充內容。

答：

系統分析師主要擔任資訊系統生命週期前三階段：初步設計、系統分析、系統設計。主要工作內容為確定資訊需求、選擇系統作業方式、定義系統規格等。所具備的技術分為系統技術以及溝通技術兩大類，如下圖所示：



資料來源：“資訊管理第六版”，謝清佳、吳琮璠

系統分析師不僅需要電腦技術，更重要的是如資訊系統設計、規劃、開發，以及評估等。由於相關工作龐雜，因此通常需要集合相關成員成立一個專案小組來完成。因此，其人際溝通技術亦相當重要，如專案管理、口頭溝通、書面溝通等。

常見的資訊系統成本效益分析方法如下：

1. 投資報酬率法 (Return on Investment Method)

將專案的利潤與成本互相比較，藉以瞭解專案所產生的效益，為一評估專案量化效益的系統化方法。

2.會計報酬率法(Accounting Rate of Return Method)

投資專案壽命期間的平均稅後淨利，除以該專案壽命期間中的原始投資額，計算出投資專案的會計報酬率。如投資專案的資金成本低於會計報酬率，則投資該專案。

3.內部報酬率法(Internal Rate of Return Method)

依據投資的資金換算為回收的利率，若投資專案的內部報酬率超過其資金成本，則投資該專案。

4.回收期間法(Payback Period Method)

又稱還本期間法，意指自投資專案的淨現金流入量中，回收該專案投資額所需的年數，回收期間愈短對公司愈有利，廣為一般公司採用。

5.淨現值法(Net Present Value Method, NPV)

考慮貨幣的時間價值來折算現金流量，若淨現值大於零，則投資該專案；若淨現值小於零，則拒絕投資。

二、請以任何以資料處理為主的資訊系統為例，以結構化技術完成以下工作及提供相關說明（可作任何假設）：

(一)試述該系統的目的，並繪製資料流程圖（Data Flow Diagram）；至少包含環境圖（Context Diagram）及至少下二階層的資料流程圖，每一階層至少含2個程序（Process）。（15分）

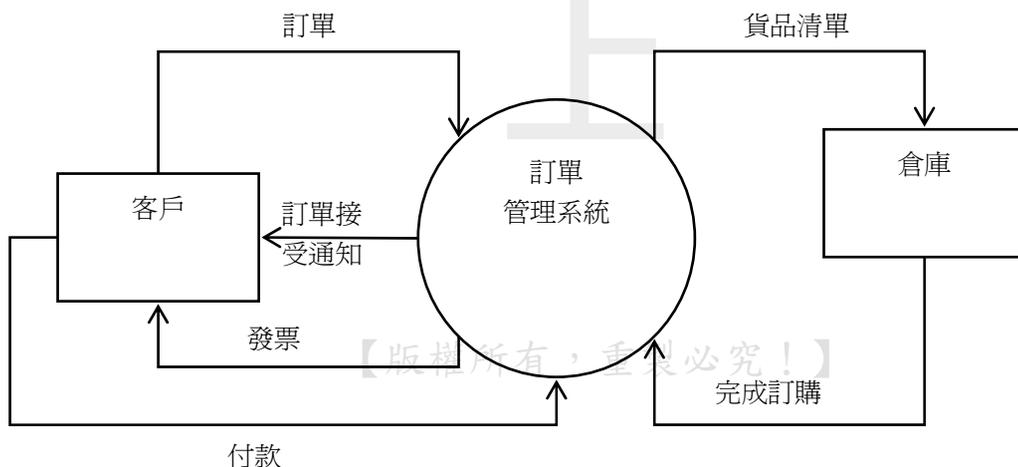
(二)詳述自前項資料流程圖轉化成結構圖（Structure Chart）的方法及其結果。（15分）

試題評析	本題為結構化分析與設計的DFD題型，由考生自行定義並繪圖，考題相當靈活，可由圖例切入作答。
考點命中	《高點系統分析與設計講義第四回》，張又中編撰，頁4-22、38~39及上課補充。

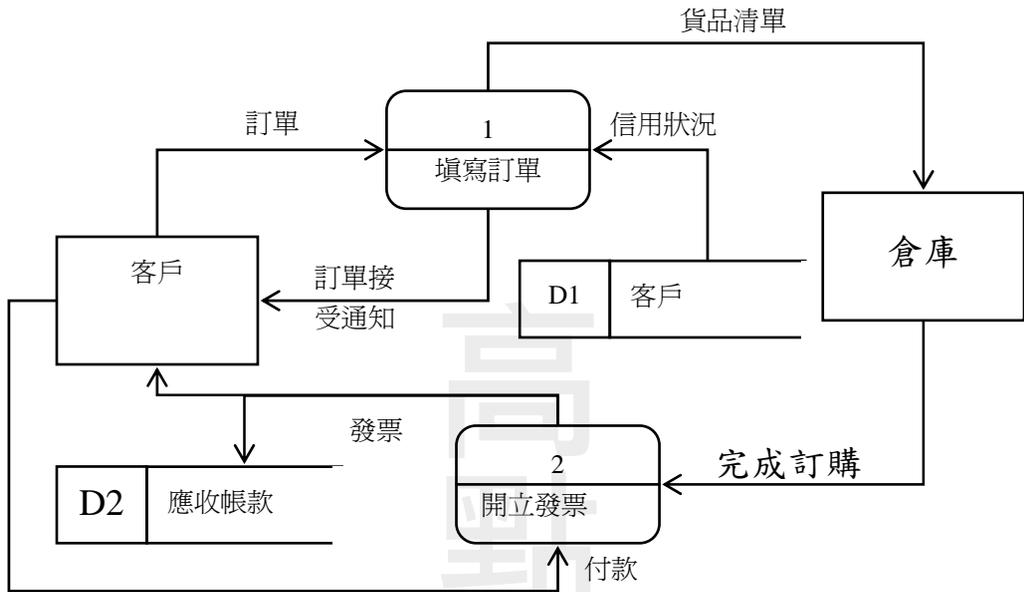
答：

設有一訂單管理系統，客戶可於該系統下訂單並付款，系統可查詢客戶信用狀況、產生訂單接受通知並傳送給倉庫貨品清單，倉庫確認後完成訂購程序，系統並開立發票給客戶。

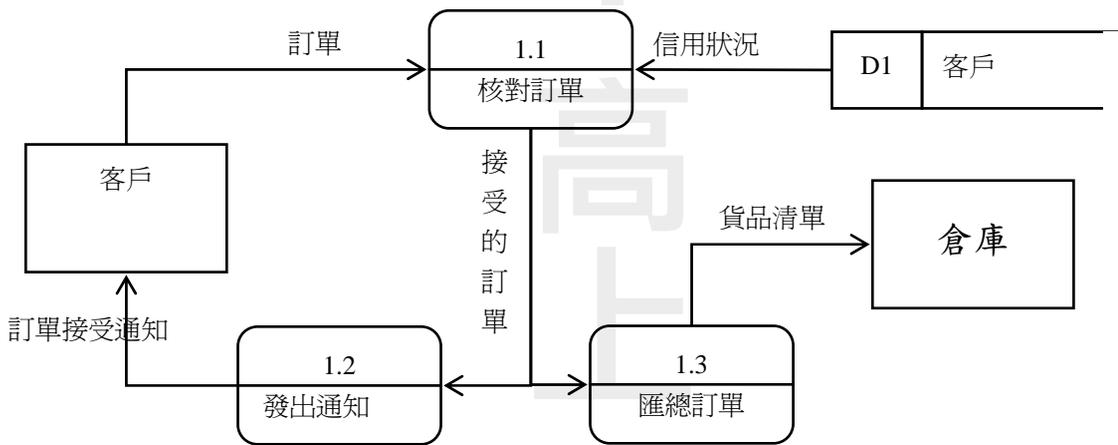
環境圖如下：



Level-0 DFD如下：



針對填寫訂單的Level-1 DFD

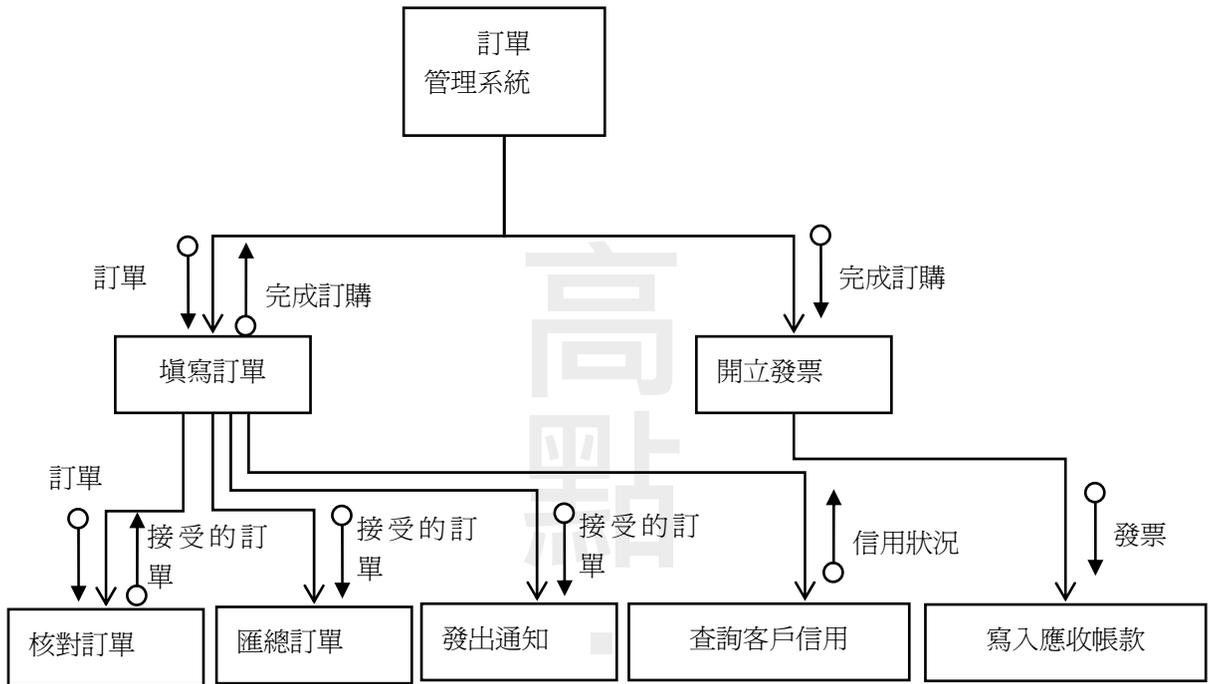


資料流程圖轉結構圖有下列四個步驟：

1. 設立總裁(President)與副總裁(Vice Presidents)
2. 設立較低層模組
3. 模組設計與結構圖修改
4. 進行評鑑

【版權所有，重製必究！】

根據DFD所轉成的結構圖如下：



三、請以圖書館管理資訊系統為例，採用UML符號完成以下工作及提供相關說明（可作任何假設）：

- (一) 試述該系統的目的，並繪製其使用案例圖（Use Case Diagram），至少含3個使用案例。（10分）
- (二) 繪製其類別圖（Class Diagram），至少含3種類別（含屬性與方法）。（10分）
- (三) 針對前項任何一種類別所產生的物件，繪製其狀態圖（State Machine Diagram），至少含3種狀態。（10分）

試題評析 本題為物件導向分析與設計的UML題型，同樣由考生自行定義與繪圖，可由圖例切入作答

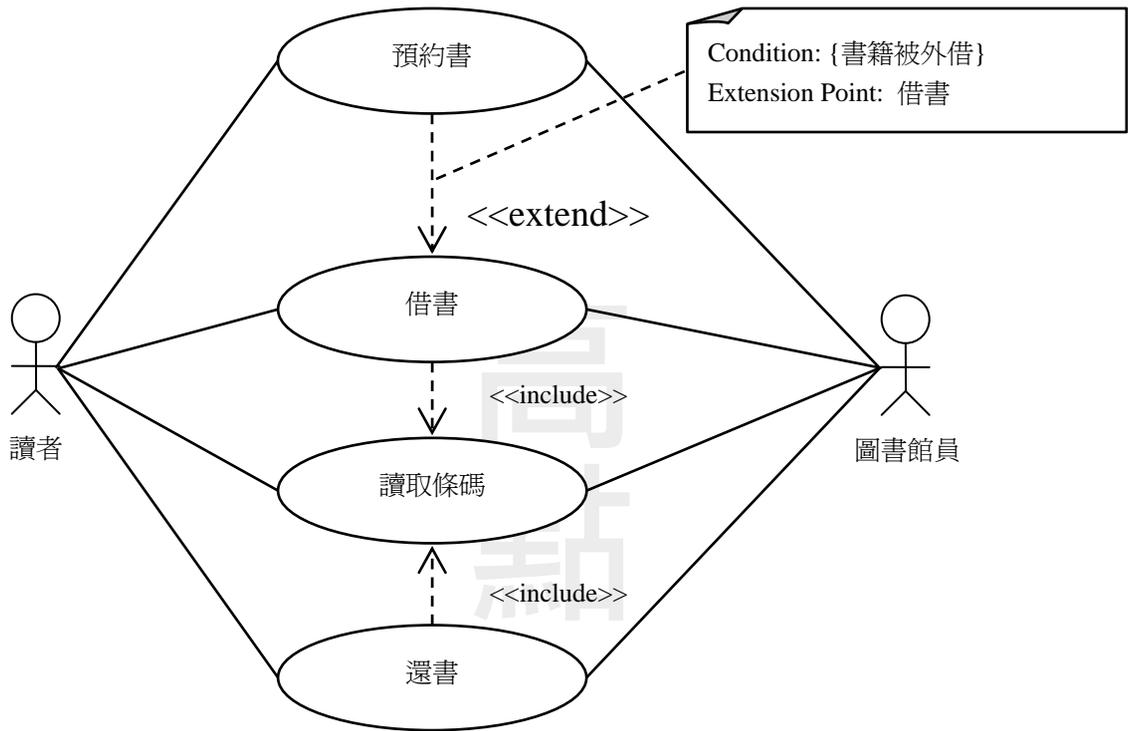
考點命中 《高點系統分析與設計講義第七回》，張又中編撰，頁7-11~13、15、18~19。

答：

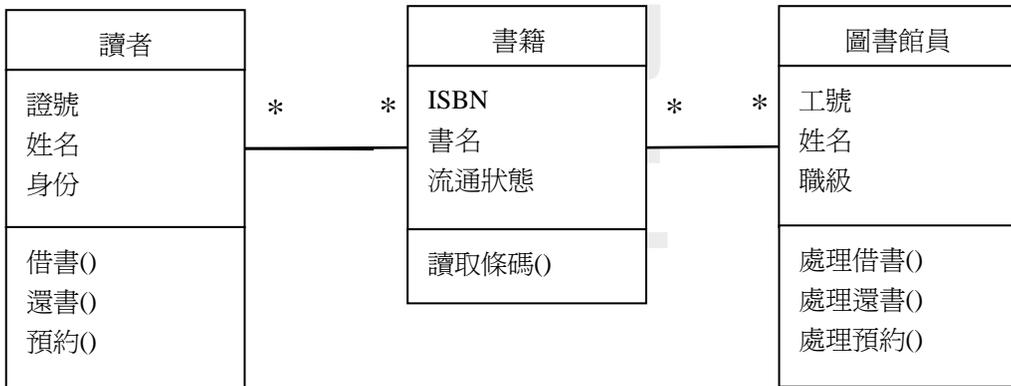
設有一圖書館管理資訊系統，讀者可借書、還書，若所需的書籍已被外借，還可預約書籍。圖書館員可以讀取條碼的方式來處理借、還書。系統可記錄讀者的證號、姓名與身份，每本書的ISBN、書名與流通狀態，以及圖書館員的工號、姓名與職級。

使用案例圖如下：

【版權所有，重製必究！】

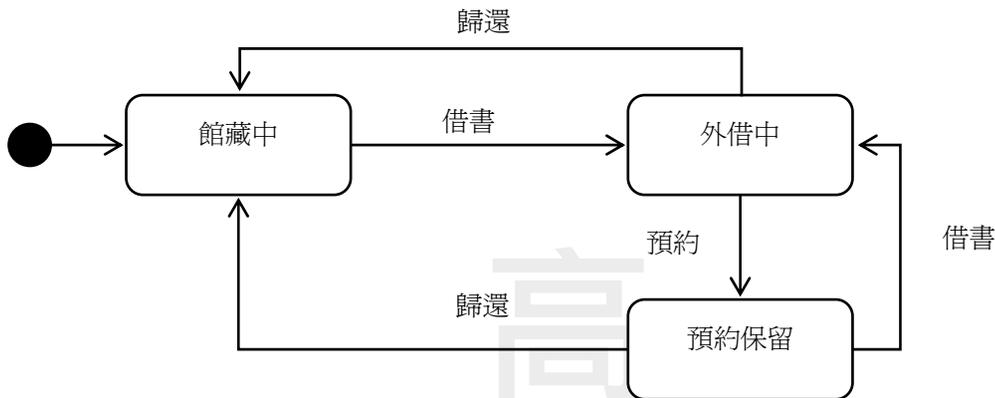


類別圖如下：



【版權所有，重製必究！】

書籍類別產生的物件之狀態圖如下：



四、何謂耦合力 (Coupling) 與內聚力 (Cohesion)？請分別針對結構化技術及物件導向技術，舉例說明如何利用耦合力與內聚力的概念，協助進行系統設計之相關工作？(20分)

試題評析 本題為內聚力與耦合力題型，為記憶題型，可由講義內容切入作答。

考點命中 《高點系統分析與設計講義第四回》，張又中編撰，頁4-15~21。

答：

內聚力被用來評估一個模組內部所處理事物的相關程度。模組的內聚力是衡量模組完成一件單一且定義清楚之工作的程度。可分為：

1.功能內聚力(Functional Cohesion)

當一個模組只做一件事情，即具有唯一功能為功能型內聚力。

2.順序內聚力(Sequential Cohesion)

指模組內具有多個功能或處理多件事情，且一項功能的輸出立即成為下一個功能的輸入，也就是共用相同的資料。

3.溝通內聚力(Communication Cohesion)

指模組內具有多個功能或處理多件事情，且這些功能使用相同的資料(輸入)，但其執行順序沒有相關性。

4.暫時內聚力(Temporal Cohesion)

模組內具有多個功能或處理多件事情，但是這些功能僅僅在時序上有所關聯，也就是必須在同一時間內執行完成。

5.程序內聚力(Procedural Cohesion)

指模組內具有多個功能或處理多件事情，這些功能必須按照一定的順序來執行，但不共用資料，這些功能群集在一個模組內僅為了確保其執行順序。

6.邏輯內聚力(Logical Cohesion)

指模組內具有多個邏輯上相關聯的功能。

7.偶發內聚力(Coincidental Cohesion)

若一個模組內部需要執行好幾件工作，且每一件工作都不相干，則該模組具有偶發內聚力。

耦合力是一種衡量模組間相互關聯強度的方法。當解決了一模組內的錯誤狀況，而在其他的模組內引起了新的錯誤，這種現象稱為漣波效應(Ripple Effect)。解決漣波效應之可行方法是盡量使一個模組不與其他模組糾結在一起，即讓每個模組盡量獨立，換言之，就是盡量降低模組間的耦合力，進而可提升模組之再利用。

耦合力可分為五類：

1.資料耦合力(Data Coupling)

指若模組間傳遞之參數為一些簡單型別的資料，則稱此模組間具有資料耦合力。

2. 資料結構耦合力(Data Structure Coupling)

指模組間以資料結構(Data Structure)型別作為程式的介面，但並非每個模組均用到該資料結構之所有欄位。

3. 控制耦合力(Control Coupling)

指的是當一模組傳遞旗標去控制另一個模組內的作業(內部邏輯)時，則稱這兩模組之間具有控制耦合力。

4. 共同耦合力(Common Coupling)

兩模組使用相同的資料區且都可讀寫資料區內之資料，則這兩模組具有共同耦合力。

5. 內容耦合力(Content Coupling)

是一個模組使用另一個模組內之部分程式碼或改變其他模組內的局部變數。

可接受的內聚力為功能、順序內聚力與溝通內聚力，耦合力部分則是資料、資料結構耦合力。良好的設計希望達到模組內的內聚力為功能內聚力，模組間的耦合力為資料耦合力。以結構化技術而言，可運用資料流程圖(Data Flow Diagram)、結構圖以及HIPO圖來衡量模組間的內聚力與耦合力；以物件導向技術而言，則是利用使用案例圖(Use Case Diagram)、循序圖(Sequence Diagram)以及類別圖(Class Diagram)來衡量之，期望系統能達到高內聚力與低耦合力。

點
·
高
上

【版權所有，重製必究！】