1. 解釋名詞(網路)
   1. CDMA : 一種無線通訊編碼技術，允許多個工作站在任何時間甚至同時傳輸且不互相干擾，再根據其各自所使用的方式解碼已獲得傳輸之內容
   2. ARP Poison Attack : 一種針對ARP與ARP Table的攻擊技術，以偽造封包來源與目的地MAC的方式，使其他在同一網域的網路設備，誤以為ARP Table中的所欲傳輸設備目的地MAC為正確位置，使惡意設備可以攔截封包或是癱瘓網路

[補充] 預防 : ARP一律改用靜態的方式；例如DHCP snooping，網路裝置可藉由DHCP保留網路上各電腦的MAC位址，在偽造的ARP封包發出時即可偵測到

* 1. ICMP : 一種用於管理網路之通訊協定用來偵測網路的狀況，位於OSI的網路層，提供可能發生在通訊環境中的各種回饋，通過這些資訊使管理者可以對發生之狀況做出診斷，然後採取適當的解決措施
  2. MPLS : 一種用於加速網路封包傳輸效率的技術，於封包中加入label，具有支援MLPS之設備收到此封包時，可以根據label內容做出判斷與行為而不需解開封包檢查IP內容後再封裝，因此可以加入網路封包傳輸效率
  3. Reverse Proxy Server : 用於作為原有伺服器之代理者，用戶端會先詢問proxy伺服器，如果有找到所需之資源的話，則直接回傳給客戶端；反之，未找到檔案則代替用戶端跟原有伺服器索取檔案再由傳給客戶端，客戶端只會知道proxy的存在，而不知道原有伺服器之存在

1. Hash
   * 1. 是非
2. X 🡪 The purpose of password hashing (in the context of a website) is not to protect the website from being breached, but to protect the passwords if a breach does occur.
3. O
4. O
5. O
6. O
7. O (不確定)
8. O 🡪 https://automationrhapsody.com/md5-sha-1-sha-256-sha-512-speed-performance/
9. X (不確定)
10. X (不確定)
11. X
    * 1. Salt

🡪指的是在雜湊前在任意固定位置插入特定的字串，其作用是為了讓加鹽後的雜湊結果和沒加鹽的結果不同以增加安全性

🡪以避免使用彩虹表與雜湊後的密碼反向查詢原本的密碼，即時查到了密碼也因為在雜湊前有使用加鹽的技術加入了不同之字串，大大減少密碼洩漏後獲取真實密碼的機率

1. OOP選擇題

🡪 B、D、E

🡪 A : 不同

🡪 B : <http://net-informations.com/java/cjava/abstract.htm>

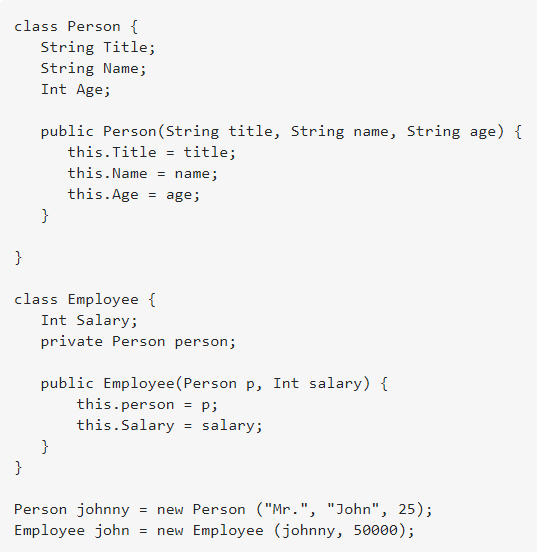
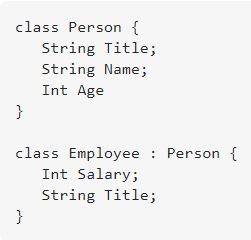
🡪 C : interface可多重繼承

🡪 D : <http://www.iitk.ac.in/esc101/05Aug/tutorial/java/interpack/createinterface.html>

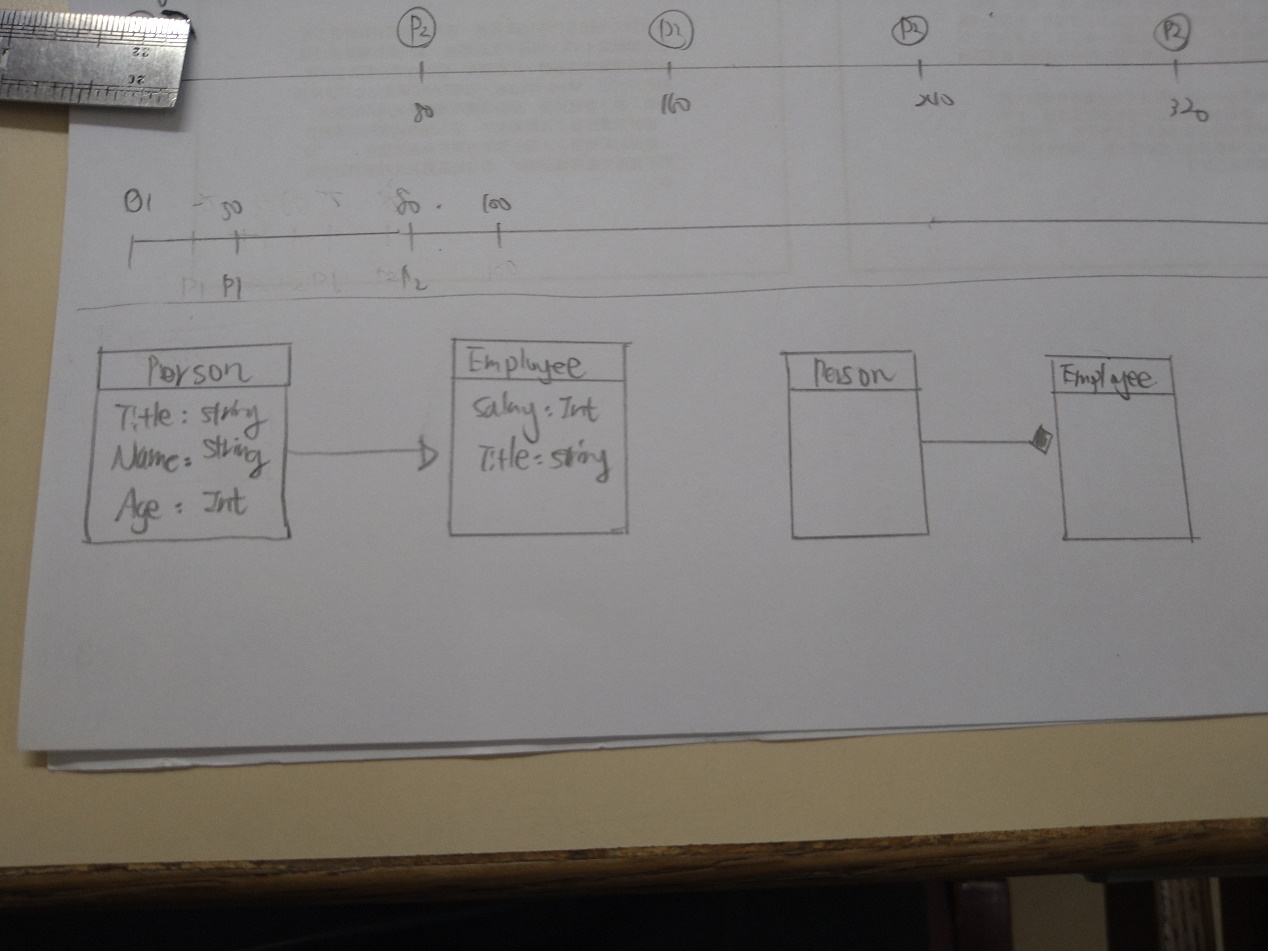
1. Composition 與繼承、優缺點、例子

[https://tw.twincl.com/programming/\*662v](https://tw.twincl.com/programming/*662v)

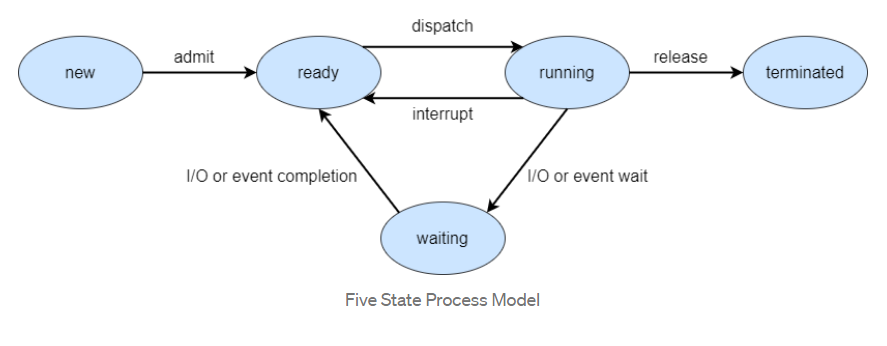
<https://stackoverflow.com/questions/49002/prefer-composition-over-inheritance>



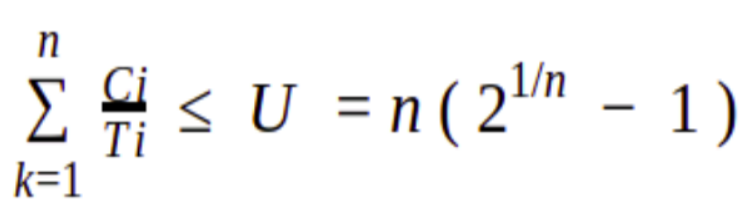
|  |  |  |
| --- | --- | --- |
|  | Composition | 繼承 |
| 優 | 彈性高、較輕巧 | 程式碼共享 |
| 缺 | 增加系統複雜度 | 父類別與子類別間tight coupling |



1. Process State (OS)

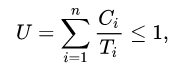


1. Deadlock
   1. 互斥存取 (mutual exclusion)
   2. Hold & Wait
   3. No preemptive
   4. Circular waiting
2. Belady’s Anomaly
   1. 當給定一process較多的page frame時，fault ratio不降反升稱之
   2. FIFO
3. Real-time 排班法 (OS)
   1. Rate Monotonic Scheduling : 是最常用的演算法，依據頻率高低決定優先權，週期短(頻率高)有高優先權，週期長(頻率低)較低優先權

🡪 根據以下公式計算是否可執行

🡪 + <= 2 \* ( – 1 )；0.5 + 0.4375 > 0.82；不可以

* 1. Earliest Deadline First Scheduling : 依據誰的deadline先到，誰的優先權就越高

🡪 

🡪 0.5 + 0.4375 < 1；可以

1. 記憶體存取計算 (OS)
   * 1. 80% \* (10 + 200) + 20% \* (10 + 200 \* (1 + 1)) = 250 ns
     2. 80% \* (10 + 200) + 20% \* (10 + 200 \* (3 + 1)) = 330 ns