

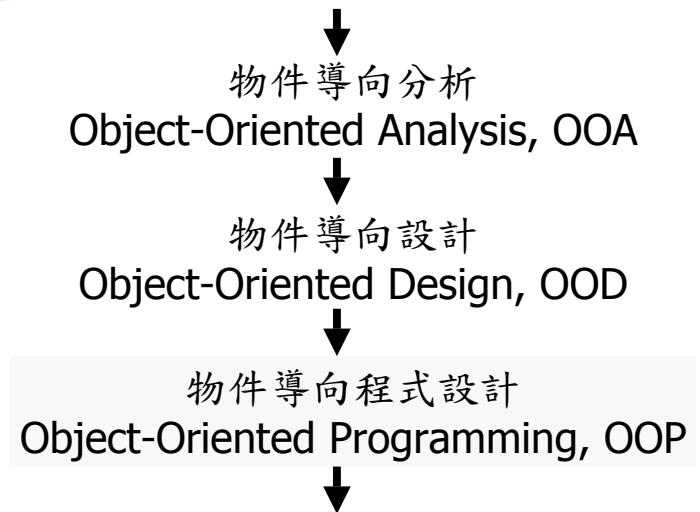
C++ 程式設計

丁培毅

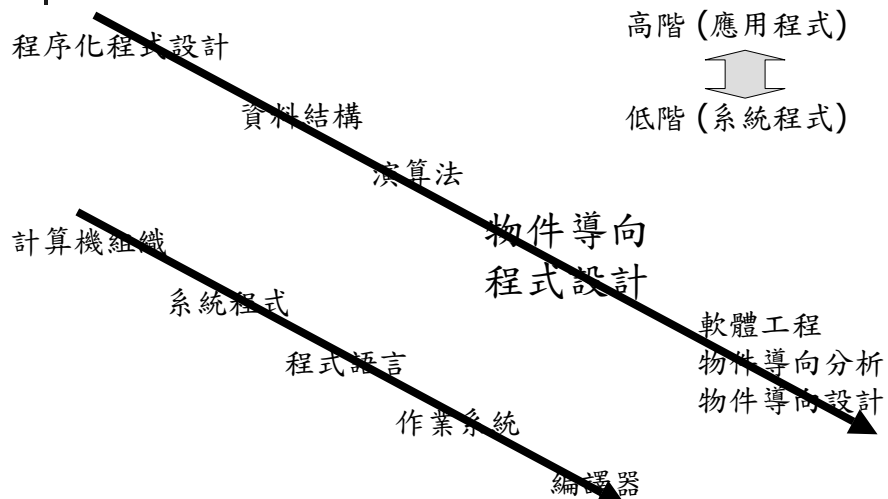
94/02 - 94/06

<http://squall.cs.ntou.edu.tw/cpp/>

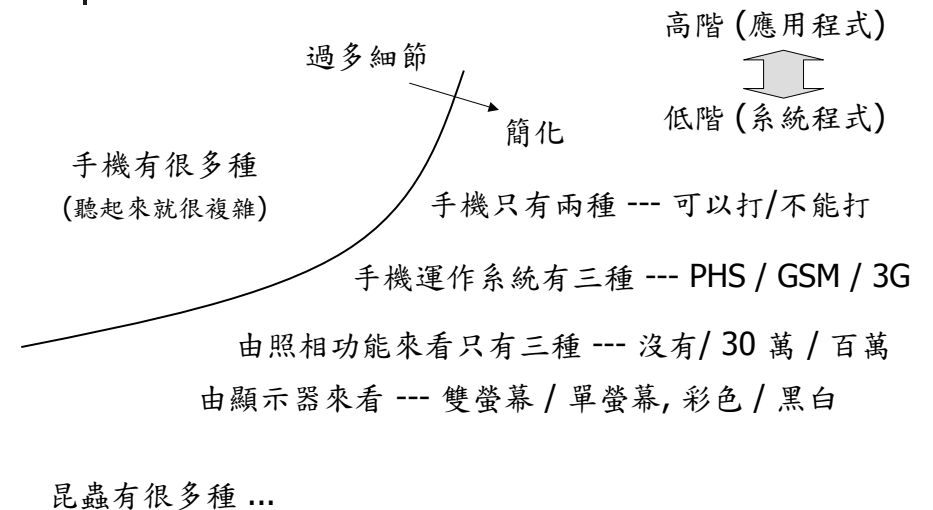
物件導向應用軟體開發過程



學習軟體設計的過程



抽象化 (簡化) 過程



抽象化 (簡化) 過程 (cont'd)

- 抽象化:
 - 去掉不在意的細節, 專注於所在意的性質
- 在 OOP 中以
 - 介面 (interface)
 - 來實作抽象化的過程

OOP 重用“軟體設計”

Reuse

- 縮短軟體的生產過程
- 減少不必要的錯誤
- 封裝 (Encapsulation)
 - 確保介面的實作, 降低軟體測試的成本
- 繼承 (Inheritance)
 - 重用軟體的架構 (程式碼與介面)
- 多型 (Polymorphism)
 - 一種基於介面, 撰寫處理不同實體物件的機制

教科書

- Thinking in C++, Bruce Eckel, 2nd Ed.,
<http://www.bruceeckel.com>
- 課程網頁
<http://squall.cs.ntou.edu.tw/squall/cpp/>
 - 參考書, 參考資料
 - 課程內容摘要
 - 作業
 - 實習
 - 討論群組
 - 過去課程網頁

如何上課

- 請預習
 - 課本
 - 中文參考書
 - 講義
 - 請準備問題
- 請複習
 - 不定期小考

程式學不會

沒興趣

我覺得熱門音樂都是一樣的 --- 都很吵

我覺得古典音樂都是一樣的 --- 都很沈悶

我覺得籃球沒什麼好看的 --- 一個球十個人搶來搶去

我覺得物理、數學很無聊 --- 式子一堆太繁瑣

那麼程式設計呢?? 蟲蟲危機?

興趣?

- 嚴格來說, 興趣不合應該都是藉口
 - 沒有花足夠多的時間, 去把東西弄清楚
 - 你花的時間有比玩 Game 的時間多嗎? (和你覺得有興趣的東西比較看看吧)
- 在大學裡學會一種東西, 其實收穫到的是“怎麼學會這樣東西的方法”, 如果你真的對某樣東西沒有興趣時再換就是了, 可以如果從來沒有學會過什麼東西, 一直在等待一個你可以不用付出就獲得很多的東西...

條條大陸通羅馬

沒有一條路、一個固定的模式適合所有的人

我確信 你的方法 才是對的, 對你最有用,

雖然該滿足的標準可能由不得你

不過切記 有準備的人才能掌握機會

怎麼調整你自己的方法?

- 老師、同學、甚至網友都是你的鏡子
- 覺得沒有效率或是沒有成效的話, 自己思考一下, 整理一下, 和別人討論一下
- 不要期待別人給你什麼實質的回饋, 自己一定會有新的體會

- 凡事總要自己提出自己的看法
- 再由同學、課本、課堂、實習中驗證

其它工程系所 VS. 資訊系所

- 機械、造船、土木
 - 將來做出來的東西很具體
 - 基礎學習過程比較辛苦, 不知道什麼時候可以真的踏進生產線
 - 模型
- 資訊
 - 成品有點抽象
 - 學習的過程中不斷地可以有成果出來
 - 模型?