資訊概論

劉承揚

程式語言

- 程式語言的分類
- 程式語言的用途
- 高階程式語言轉變成執行檔的歷程
- 程式設計的過程

程式語言的分類

- 依階層劃分
- 依使用模式劃分
- 依應用劃分

3

依階層劃分

- 機械語言：
  是指電腦硬體內部所使用的語言，也是電腦唯一能直接辨識的語言，是由許多0、1組合而成
- 例：
  0100011110101001001111000100010001101010101000010010010010001010111010100000100101010000100101001000010101010101010

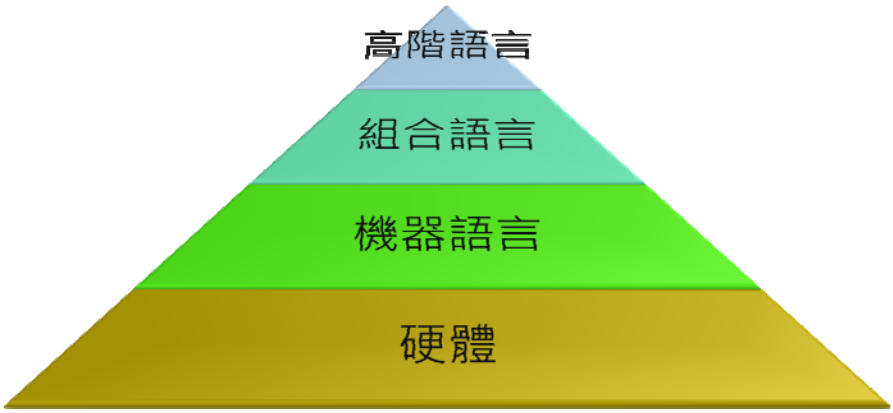
4

依階層劃分

- 組合語言 ：
  是最接近機器語言的程式語言，它是一種符號式語言，必須經過翻譯後，才能被電腦所接受
- 高階語言 ：
  又稱編譯語言，是指為各種應用程式而設計的語言，高階語言的文法比較接近日常生活用語，所以較為簡單易記

5

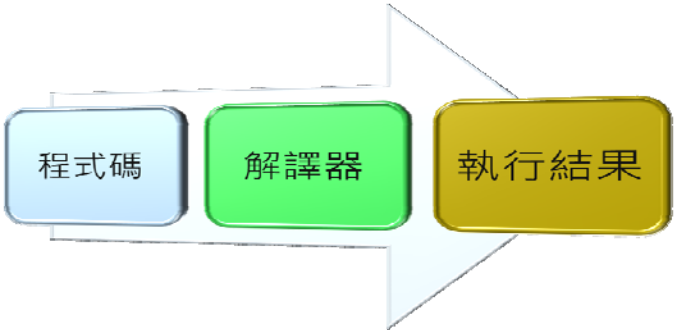
依階層劃分



6

依使用模式劃分

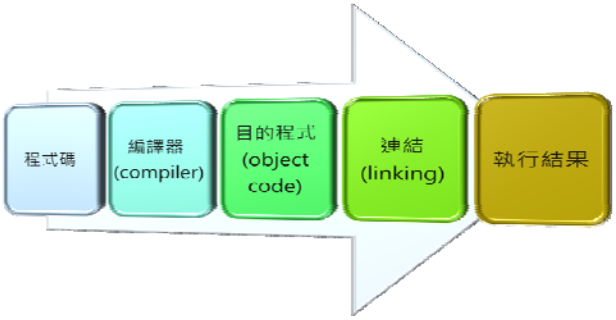
- 直譯式程式語言：
  此種語言，系統可以將原始程式的指令逐一的翻譯並執行，不需要經過編譯，如BASIC、LOGO、HTML等。它的特點是修改程式及除錯，較為簡單容易



7

依使用模式劃分

- 編譯式程式語言 ：
  撰寫的原始程式，需要經過編譯器編譯之後，輸出為電腦中直接執行的目的程式，如C、C++、PASCAL...等。他的特點是再次使用只要執行目的程式，無須重新在編譯其原始程式



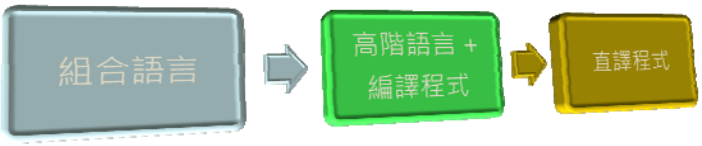
8

依應用劃分

- 一般用途：
  指用於一般範圍的通用程式語言，如BASIC、PASCAL、C...等
- 特殊用途：
  應用於特殊領域的程式語言，例如應用於商業的COBOL，工程的FORTRAN，網路的HTML或人工智慧領域的程式語言如LISP...等

9

執行速度



10

程式語言的用途

- C/C++ ：
  - C 語言最早的標準是K&R，在1989 年ANSI 制定標準C 語言後，則稱之為ANSI-C，之後在1999年參考C++語言的語法而作了少許更新，稱為C99
- 在1980 年代晚期，Bjarne Stroustrup 和其它實驗室的同仁替C語言新增物件導向的功能，稱為C++，C++已經成為目前Windows作業系統各種應用程式主要的開發語言。

11

程式語言的用途

- LISP：
  - Lisp主要是設計於提供給研究人工智慧(AI)的使用者開發軟體的語言。Lisp是在1950年代末期由當時研究AI的先驅者John McCarthy所研發，基本上Lisp裡頭所使用的資料結構大多是由一連串元素所組成的List
- Fortran：
  - Fortran是全世界第一個誕生的高階程式語言，由此可知電腦最早是使用在科學研究上，所以Fortran對於一些數值的計算支援的功能比較多，一般理工科系，像是土木工程系、電機工程系等，都有一門程式語言課，就是在專門教授Fortran

12

程式語言的用途

- Java：
  - JAVA是由美國昇陽電腦公司（Sun Microsystems, Inc.）所發展出的第一個能在國際網際網路（Internet）上面具有「硬體/軟體中立性」互動能力的程式語言 。Java 的語法上，和 C 語言非常相似，其物件導向的特性，也與 C++ 相仿。Java 本身和 C 與 C++ 的相似程式

13

高階程式語言轉變成執行檔的歷程

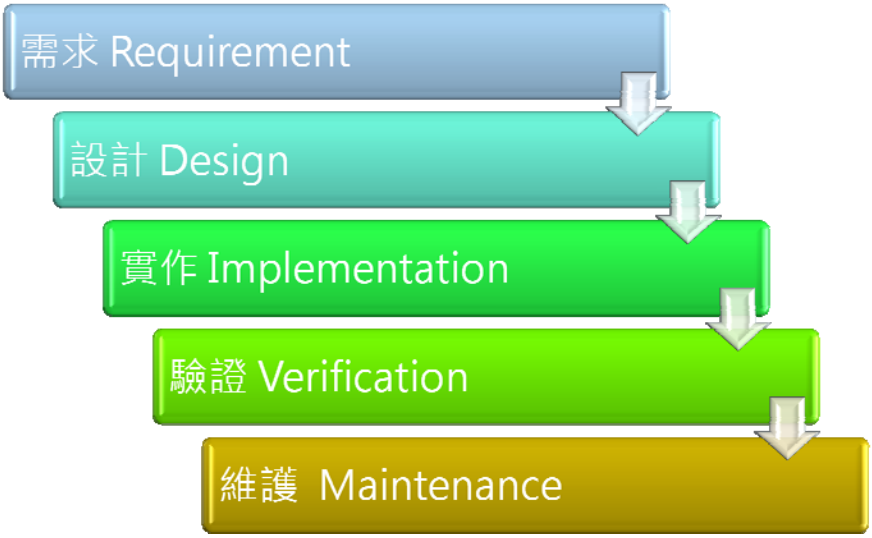
14

程式設計的過程

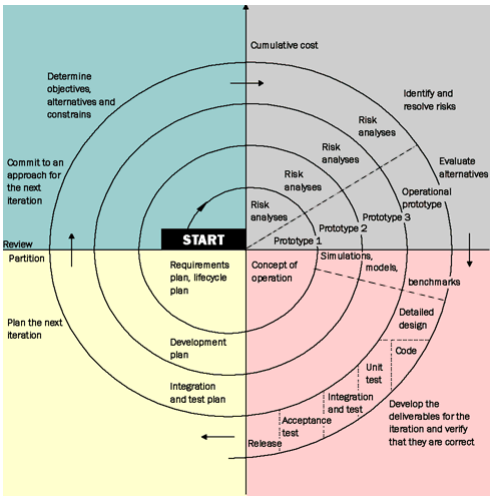
- 了解問題
- 資料結構規劃
- 演算法設計
- 撰寫程式
- 測試程式

15

瀑布式設計模型



螺旋式設計模型



17

了解問題

- 到一個問題需要電腦幫我們解決時，首先必須要了解我們會用到什麼樣的輸入資料(Input)
- 我們需要的是什麼樣的輸出結果 (Output)
- 最後要思考從這些輸入到輸出結果過程中要怎麼樣進行轉換的處理

18

資料結構的設定

- 資料結構只是一種工具，它主要是提供你一些資料的儲存方式及如何儲存資料使你更容易解決問題
- 挑選哪一種資料結構時，可以用以下的原則來思考 ：
  1. 資料儲存方式及其優缺點
  2. 這種資料儲存方式有那些對應的處理方法或是需常用的某種方法要那種儲存，方式較適合

19

演算法設計

- 好的演算法必須滿足下列五個條件 ：
  1. 有限性:要在有限個步驟內解決問題
  2. 明確性:每個步驟都必須清楚地表達出來
  3. 輸入資料:包含零個或一個以上的輸入資料
  4. 輸出資料:至少產生一個輸出
  5. 有效性:每一個步驟如果交由設計師以紙和筆來執行，必須在有限的時間內完成

20

撰寫程式

- 在撰寫程式時，請特別注意變數的命名，盡量少用一些無意義的字母或是符號去命名
- 在程式的撰寫風格上，請盡量簡潔，不要一行裡塞滿了許多行的程式，另外也可以使用縮排的方法，讓程式碼層次分明容易閱讀
- 特別注意再撰寫程式過程中要幫中程式碼寫上註解

21

測試程式

- 當我們在進行測試時，可以先列出一份測試計畫。這個計畫可以列出所有可以讓程式正常運作的步驟，然後我們在依照這些步驟一步一步的下去測試
- 在測試過程中找到問題時，要怎麼處理 ：
  1. 如果在這一個步驟測試中發生問題，我們可以在重新以同樣的步驟操作程式，讓錯誤再次發生，這麼一來我們大約就可以知道問題出現在哪裡了
  2. 如果這個程式之前有經過變動，然後發生問題，那麼我們可能就要進行反覆的測試，用不同的操作步驟找出示否有不正常運作的地方

22