

程式人



用十分鐘瞭解陳鍾誠的

《課程、教材與學習方法》

（程式設計、nand2tetris、科學計算、人工智慧）

陳鍾誠

2016 年 2 月 16 日

話說

- 我在念書的時候
- 好像沒有老師告訴我們過
他的教育想法

也很少老師告訴我們

- 為甚麼要上那門課！

於是

- 我們就像一張白紙那樣
- 走進了教室

學期末的時候

- 通常會變成一張
- 上面有很多問號的紙

然後

- 學期就結束了

現在

- 我也當了老師

為了避免學生和我一樣

- 變成很多問號的白紙

所以在上課的時候

- 我通常會告訴他們
- 那門課是做甚麼用的！

雖然如此

- 能夠真正學得好的學生
還是為數很少。

所以我想

- 或許我應該告訴他們，
我對教育的想法。

或許這樣

- 學生們會比較容易理解

我為甚麼那樣上課

為甚麼教那些內容。

現在

- 就請花上十分鐘的時間
讓我說明一下
我為甚麼教那些課程
還有隱藏在背後的那些想法。

在 2003 年的時候

- 我來到《金門技術學院資訊管理系》教書

當時的我

- 從來沒有當過老師

所以

- 我也是當了老師之後
才開始學習怎麼教書的！

在金門技術學院資訊管理系

- 我教過
 - 程式設計
 - 作業系統
 - 系統分析

等等課程

甚至

- 還教過《經濟學》

雖然

- 我是學《資訊科學與工程》的
- 但我確實教過《經濟學》...

XD

因為當年

- 系上只有三個老師
- 而且我大學時確實修過
《經濟學》

不過

- 教《經濟學》也是我自願的
- 沒有人逼我要教這一科

後來、我們改成資工系

- 於是《經濟學》就不見了
- 所以我再也沒機會教經濟學了

於是我教的課就開始以

- 程式設計為主，像是：
 - 視窗程式 C#
 - 系統程式 C
 - 手機程式 C# (smart phone) , Java(android)
 - 遊戲程式設計 C# (XNA)
 - 網路程式設計 (C#, node.js)

然後偶爾

- 我也會教一些
非程式類的課程

像是

- 微積分
- 機率統計
- 離散數學

還有

- 人工智慧
- 計算語言學

甚至、我還教過

- 動畫設計 (Blender)

後來、在我學了 verilog 之後

- 也開始教《計算機結構》
- 甚至也教過《數位邏輯》

反正系上老師不多

- 課都教不完

所以想教甚麼

通常就可以教甚麼

不像很多大學校

- 一個系有五十個老師

每門課都有人占據

想教還得先搶位子

所以我每學期

- 通常都會教一兩門新課
以便讓自己有些挑戰性
順便學點新東西！

當然、我的學生們

- 就成了我的白老鼠 …

奇怪的是

- 我的新課常常教得比舊課好

我想是因為

- 第一次教，總是戰戰兢兢
 - 而且東西才剛學會
- 絕對不會教得太深！

所以

- 我才會每學期都想教一兩門
新課程！

而且

- 教一門課的時候

我常常習慣自己寫教材

於是、這些年來

- 我寫了不少教材與電子書，像是：
 - 視窗程式設計 網路程式設計
 - 人工智慧 計算語言學
 - 機率統計 微積分
 - 計算機結構 系統程式
 - 計算理論 動畫設計

而且

- 我有個習慣
- 就是每門課程都要有《實作》
特別是《程式實作》

所以

- 上面的那些課程都採用了一些對應的《程式語言》與《開發工具》。

於是就得到下列課程地圖

- 視窗程式設計 (C#) 網路程式設計 (C#)
- 人工智慧 (JavaScript) 計算語言學 (C)
- 機率統計 (R) 微積分 (C)
- 計算機結構 (Verilog) 系統程式 (C)
- 動畫設計 (Blender)

2013 年開始

- 我越來越喜歡

node. js+javascript

這個搭配組合

所以我決定接下

- 大一整年的《程式設計》課程
- 並且用 node.js+javascript 為主軸
- 然後在下學期時教一點 C 語言
以便銜接後續的系統程式課程

然後

- 為了避免每一門課都要重新學語言
- 所以我的課程都盡量改用
node.js+javascript 來實作。

後來我決定把

- 《機率統計》也用 javascript 實作。
- 然後納入《線性代數、微積分、工程數學、數值分析》等課程的內容實作。
- 形成一們涵蓋《全部資工領域數學內容》的超級課程，那就是《科學計算》。

這時、一定有人會說

- 《全部資工領域數學內容》應該還有
離散數學阿！
- 那《離散數學》難道也放在《科學計算》課程裏嗎？

當然不是

- 因為我們埋下了一個伏筆
- 把《離散數學的程式實作》留給了《人工智慧》課程
- 像是《邏輯推論、圖論》等等。

這樣我就能用

- 《科學計算》和《人工智慧》
- 完整的涵蓋《資訊領域》的數學主題，把《數學和程式》融合為一體。

不過

- 上述的《科學計算》和《人工智慧》是三年級的課程。
- 這樣我們講完 1, 3 年級，卻漏掉了 2 年級。

那我打算教二年級的同學甚麼呢？

答案是

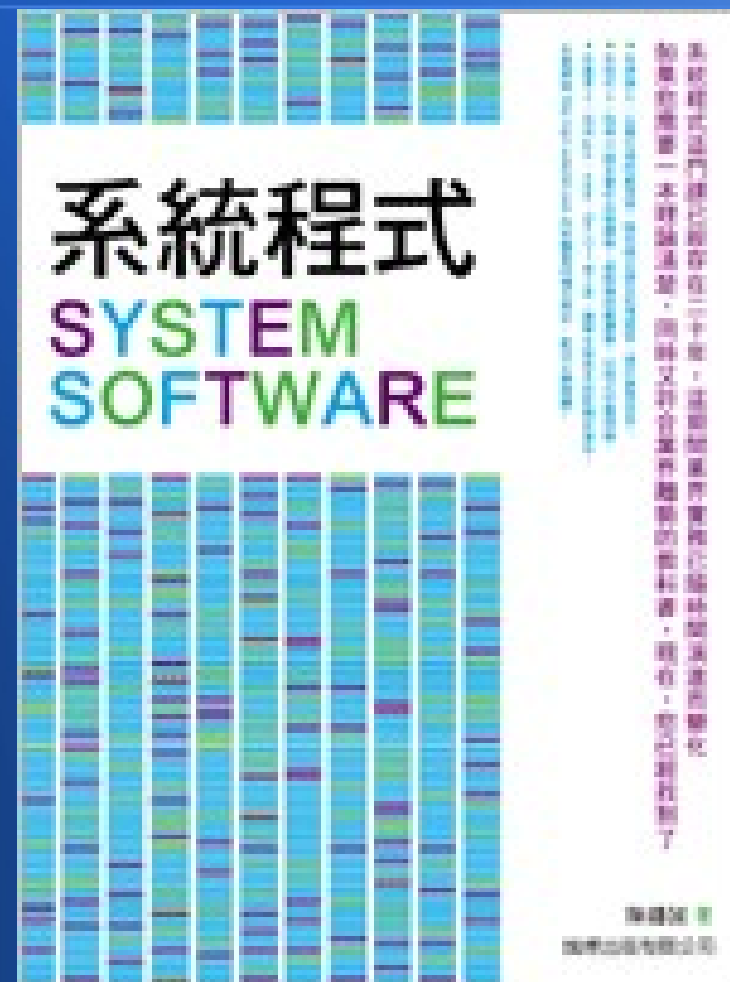
- 從硬體到軟體，完整設計一整台電腦的能力。
- 也就是《計算機結構 + 系統程式》

原本

- 我是自己用 verilog 設計處理器，
然後自己用 C 與 javascript 寫編譯
器、組譯器、虛擬機等等系統軟體。
- 用自己編寫的教材上課。

我甚至寫了一本

- 由旗標出版的
《系統程式》



但是自從我上過

- Nand2tetris 這門網路課之後
- 我決定改以 nand2tetris 為主軸
- 用 nand2tetris 作為《計算機結構》和《系統程式》課程的核心

於是、就有了下列的課程地圖

| 分類 | 課程 | 語言與軟體 | 學期 | 註 | 相關主題 |
|----|---------|------------------------|----|---|------------------------------------|
| 程式 | 程式設計 | JavaScript + Node.js | 一上 | 必 | HTML , CSS , JavaScript , 前端技術 |
| | 程式設計-進階 | JavaScript + C 語言 | 一下 | 必 | Canvas , 伺服器端 , 資料結構+演算法 |
| 電腦 | 計算機結構 | JavaScript + Verilog | 二上 | 必 | nand2tetris (Part I) , CPU設計 |
| | 系統程式 | JavaScript + C 語言 | 二下 | 必 | nand2tetris (Part II) , 編譯器 , 系統軟體 |
| 進階 | 科學計算 | JavaScript + R 軟體 | 三上 | 選 | 微積分 , 機率統計 , 線性代數 , 數值分析 |
| | 人工智慧 | JavaScript + Node.js | 三下 | 選 | 離散數學 , 計算理論 , 自然語言 , 機器學習 |
| 自學 | 3D 動畫設計 | Blender+Krita+Natron | 一上 | 選 | 電影後製 |
| | C#程式設計 | VisualStudio + WinForm | 二上 | 選 | 視窗程式 , Thread , 網路程式 |

如果您仔細看這張課程地圖

- 會發現大致可以分成四段

大一的時候

- 我主要教《程式設計》
- 以 javascript 為主，
C 語言為輔

因為我認為

- 程式設計是《資工系》一定要會的核心能力
- javascript 可以銜接網站程式，而且應用越來越廣
- C 則可以銜接系統程式與作業系統等，是嵌入式的主力語言。
- 透過這兩個語言能同時涵蓋到最廣的面向，而且具有《動態語言 + 靜態語言》的組合優勢。

我希望

- 學生們能在大一的時候
打好《程式設計》的基礎。

然後在大二的時候

- 透過 nand2tetris 課程，學會自己設計一整台電腦的軟硬體
- 而這也就是為何大二的《計算機結構》與《系統程式》以 nand2tetris 為主的原因了。

接著在大三的時候

- 透過《科學計算》與《人工智慧》兩門課，讓學生能進入一些更困難，更研究性的領域。

在《科學計算》與《人工智慧》 兩門課中

- 我仍然採用 javascript 實作，
以便銜接大一所學會的程式語
言。

接著

- 我就功成身退了！

此時、一定有同學會問

- 那大四我們要修甚麼？

關於這點

- 其實我的想法是，盡量把大四的時間留給學生。

留給學生幹嘛呢？

留給學生

- 想就業的，開始全力學習就業所需的技能，像是
 - 想走網站的可以深入 node.js+ 前端技術
 - 想走系統的就可深入 C/Linux/... 等領域
 - 想走手機 App 的就可以深入 android 或 iOS
 - 想走產線控制的就直接到公司裏實習吧！

而對於那些想繼續念研究所的

- 就可以開始尋找想深入的領域和老師，朝著研究的方向前進。

因此

- 我選擇將大四的一整年留給學生，沒有安排課程。

現在

- 你應該知道我為甚麼開那些課了！
- 但如果你是上這些課的同學、你該怎麼上這些課程呢？

關於這點

- 讓我們先轉換一下焦點
- 容我說說我對教育的看法！

有些人可能覺得奇怪

- 為何學生需要瞭解老師的教育想法呢？

請聽我慢慢細說！

從我大學畢業之後

- 台灣的教育就一直在
《教改、教改、教改》

改來改去、改來又改去！

問題是、好像愈改

- 批評的聲浪就愈大！

對於中小學生而言

- 改用多元入學，而非單一的聯考，是教改的最明顯特色

但是對於大學而言

- 這幾十年則是不斷在《論文
評判老師和學校》的這件事
上進行強化！

但是改來改去的結果

- 教育似乎並沒有因此而多元化！
- 相反的，似乎更加一元化了！

中小學生

- 還是很在意成績！
- 而且補習班更多了，考試導向的教育並沒有因此消除！

大學老師

- 全都開始追逐論文、升等與國科會計畫，反而荒廢了教學，甚至把研究生當成出論文的工具。

我不想

- 跟著這股潮流走
- 跑去寫論文！

我只想

- 好好的寫想寫的教材
上想上的課程
做想做的程式！

所以

- 我現在做的東西，就算是研究，最後也只用投影片、電子書和網站呈現。
- 不會去投稿論文！

這或許

- 是對體制的一種反抗！

也就是

- 體制愈要求我這樣做
- 我就愈是不要那樣做

教育部要我向東走

- 我就偏偏要向西走！

大家都擠到論文領域

- 都去申請國科會計畫！

我就偏偏不寫論文

- 不申請任何計畫！

因為我知道

- 人多的地方，不只競爭者多，問題也很多！
- 那我為甚麼不走另一條路，雖然沒什麼資源，但是卻能自由發揮，又沒有人和你競爭。

這就是我的人生哲學

- 也是我從《經濟學》上得到的生命啟示

各位同學們應該都聽過

- 22K 的問題
- 還有大學學歷貶值的問題！

我認為

- 22k 的成因，除了學歷貶值的因素之外，其實還有大家的技能過度單一化，造成競爭紅海所導致的。

台灣之所以落入困境

- 我認為原因正是《單一化》

為甚麼呢？

請讓我們用猶太人來做對照組

猶太人到了一個地方

- 如果有人開雜貨店賺了錢、
就會有人去旁邊開水電行、
開加油站、最後大家都賺到
錢了 ...

但是台灣人

- 如果到一個地方開雜貨店賺了錢
- 其他人就會蜂擁而至，然後
 - 開第 2 家雜貨店、第 3 家雜貨店、.....
 - 第 100 家雜貨店 ...
- 最後大家通通賺不到錢，因為所有市場都是一片紅海，充滿台灣人的紅海！

更糟的是

- 我們的政府，往往也會制定那種
《鼓勵單一項目》的規定，而非
《鼓勵多元發展的政策》。

但問題也不完全在政府

- 因為就算《政策鼓勵多元》
最後大家都會朝著《看來最有利》的
方向前進
- 於是又回到了《所有人擠在一起》的
困境，要死大家一起死！

所以在教育界

- 明明大學學歷貶值很嚴重
大家都還是拼命上大學。

然後在產業界

- DRAM 產業很競爭，大家還是拼命蓋廠房。

所以、我希望大家不要

- 跟著人家一窩蜂的去擠！

你必須要有

- 自己的想法
- 自己的特色

你必須要懂得

- 自我學習！
- 團隊合作！

對於我的任何一門課

- 你都可以提出自己的學習方法，甚至告訴我該如何評量你！

例如在《程式設計》這門課

- 當我們程式設計教 javascript 的時候，你可以選擇學習其他語言。
- 重點並不是學哪種語言，而是要學會《程式設計》這件事情，而且最好能運用自如！

對於《計算機結構》這門課程

- 雖然我們採用以 nand2tetris 設計處理器為主的《習題導向式》學習方式
- 但您也可以採用傳統的教科書進行學習，並選擇不做我指定的習題。我可以改用口試或考試等方式評量你的學習狀況。

對於《系統程式》課程

- 您也可以自行選擇實作《組譯器、虛擬機、編譯器》的程式語言。
- 或者乾脆自己發明一個語言也行，甚至用那個語言來重新打造作業系統。

對於《科學計算》與《人工智慧》

- 你也可以自由地採用任何語言來實作。
- 或者選擇自己認為最適當的《教科書》或《線上課程》等等

甚至您也可以開學三周內

- 告訴我你決定要在家自學
- 那你也可以不用每次都來上課。
- 不過我會和你討論成績的評量方式，例如口試或考試等等。

如果你沒告知我要採用其他學習方式

- 那就請來上課和我們一同學習
- 我們會針對每一門課採用較適當的上課方式。

像是程式設計課程

- 我通常每堂課會講解 15 分鐘左右，然後就讓大家進行實作練習。
- 上課的同學請把握剛講解完後的時間立刻練習，我認為這樣的效果會比較好。

而對於 nand2tetris 的兩門課

- 《計算機結構》和《系統程式》
- 我會在一開學時把所有內容很快地講一遍，之後大部分的時間都採用《習題導向》的方式讓大家自學。

對於《科學計算》和《人工智慧》

- 我會用自己的電子書來教
- 並且每個主題都會搭配程式實作。

這些課程我們會讓各位分組

- 原則上以四個人為一組
- 但是您也可以選擇自己一個人一組。

課程告一段落時

- 我會讓大家進行討論，原則上分組成員應當在一起討論
- 但也可以去找其他組一起討論

但是討論時間結束後

- 我會請各組提出問題
- 如果某些組沒有問題，那就換我問該組問題。
- 這些上課情況會做為平時成績評量的依據。

以上課程我通常不會有期中期末考

- 但很可能會有平時測驗
- 也可能會隨機進行口試

但是如果您覺得

- 有更好或更適合您的評量方式，也可以提出來，我們也可以考慮用你的那種方式來評量你

我常常會嘗試不同的教學方法

- 有時效果好，有時效果不好
- 好的會留下，不好的就改變

也歡迎您提供想法給我

- 告訴我你認為怎樣教會比較好

不過適合 A 老師的教法

- 並不一定適合 B 老師

例如有些老師很會講笑話

但若讓另一個老師來講

- 那整個學習就會變成笑話

所以

- 教學的方法，還是得因人而異

雖然我已經教了 13 年

- 但是我的上課方式與風格
也還在摸索調整當中！

歡迎您

- 和我一同來《學習如何學習》！

這幾年的經驗下來

- 我發現最好的學習，就是想辦法教人！

很多課程

- 我之所以學會了，就是為了要教你們！

其實

- 我在大學的時候，也沒有搞懂很多課程。
- 像是《機率統計、系統分析、計算機結構》等等。

但是我必須開始教大家時

- 我就得努力學會並寫下來
- 然後反覆地思索，或者進行實驗，這樣就自然而然地學會了。

因此、如果你的同學或組員

- 有不懂的情況，你可以試著教他，這樣你會學得更好。
- 如果你們整組都不懂，那就問我吧！

你們每問一個問題

- 都會讓我更清楚你們的學習狀況
- 提問題對各位絕對有很大的幫助
(包含成績在內) !

記得

- 如果你們沒有問題
- 那就換我問你們問題了！

你準備好

- 來上我的課了嗎？

歡迎您

一起來學習下列課程

| 分類 | 課程 | 語言與軟體 | 學期 | 註 | 相關主題 |
|----|---------|------------------------|----|---|------------------------------------|
| 程式 | 程式設計 | JavaScript + Node.js | 一上 | 必 | HTML , CSS , JavaScript , 前端技術 |
| | 程式設計-進階 | JavaScript + C 語言 | 一下 | 必 | Canvas , 伺服器端 , 資料結構+演算法 |
| 電腦 | 計算機結構 | JavaScript + Verilog | 二上 | 必 | nand2tetris (Part I) , CPU設計 |
| | 系統程式 | JavaScript + C 語言 | 二下 | 必 | nand2tetris (Part II) , 編譯器 , 系統軟體 |
| 進階 | 科學計算 | JavaScript + R 軟體 | 三上 | 選 | 微積分 , 機率統計 , 線性代數 , 數值分析 |
| | 人工智慧 | JavaScript + Node.js | 三下 | 選 | 離散數學 , 計算理論 , 自然語言 , 機器學習 |
| 自學 | 3D 動畫設計 | Blender+Krita+Natron | 一上 | 選 | 電影後製 |
| | C#程式設計 | VisualStudio + WinForm | 二上 | 選 | 視窗程式 , Thread , 網路程式 |

歡迎來到

- 金門大學資工系！

和我們一起

- 一同學習、共同成長！

希望這四年

- 能夠讓您覺得

不虛此行！

那我們這些老師

- 就可以問心無愧了！

現在

- 就讓我們一起開始這段

學習的旅程吧！