

計畫編號：A92090026

教育部委辦學術機構研究計畫結案報告

技專校院資訊相關科系課程教學與產業需求
配合檢討之研究—以程式設計師為例

計畫主持人：鍾乾癸

E-mail 位址：cgchung@mail.cgu.edu.tw

FAX 號碼：03-2118025

地址：桃園縣龜山鄉文化一路 259 號

共同計畫主持人：陳信良

E-mail 位址：slchen@mail.wfc.edu.tw

FAX 號碼：05-2264087

地址：嘉義縣民雄鄉建國路二段 117 號

研究機構：吳鳳技術學院

中華民國九十三年四月

目 錄

壹、中英文摘要.....	4
貳、研究緣起.....	6
參、研究方法.....	8
肆、技職體系學生程度分析.....	12
一、教育背景.....	12
二、環境背景.....	13
三、學習動機/認知.....	14
四、現在的教學環境.....	15
伍、資管系畢業生就業狀況與資訊應用能力調查報告.....	17
5.1 資管系四技畢業生就業狀況與資訊應用能力調查結果...17	
5.2 資管系二技畢業生就業狀況與資訊應用能力調查結果...33	
陸、資訊管理系程式設計類課程與資訊軟體人才培訓課程之對應 分析.....	36
柒、核心科目課程.....	42
7.1 計算機概論課程.....	42
7.1.1 課程設計理念.....	42
7.1.2 課程大綱.....	51
7.2 資料結構與演算法.....	60
7.2.1 課程設計理念.....	60
7.2.2 課程大綱.....	62
7.3 資料庫系統.....	67
7.3.1 課程設計理念.....	67
7.3.2 課程大綱.....	70
7.4 軟體發展技術.....	74
7.4.1 課程設計理念.....	74
7.4.2 課程大綱.....	77
7.5 程式語言.....	81
7.5.1 課程設計理念.....	81
7.5.2 課程大綱.....	84
7.6 XML 語言.....	98
7.6.1 課程設計理念.....	98
7.6.2 課程大綱.....	101
捌、與勞委會乙級程式設計師技能檢定考試科目之比較.....	105
玖、結論與建議.....	114
9.1 結論.....	114

9.2 建議.....	116
拾壹、參考文獻.....	118
附錄一、座談會會議記錄.....	125
附錄二、歷次會議記錄.....	132

壹、中英文摘要

摘要

軟體產業的興衰與軟體工程師的質與量有密切關連，我國大學與技職院校資訊相關科系每年畢業人數近三萬人，惟軟體業者仍感新進工程師之人數不足，而技職院校的畢業生仍有甚多找不到適當的本行工作，可見我國資訊工程師之訓用失調。有鑒於此問題之嚴重，行政院科技顧問組召集軟體業者依據軟體工程師的工作性質之不同，將軟體人才分為十四類，進而訂定各類人才應具的核心知識，以作為日後軟體工程師能力鑑定之依據，藉以提升我國軟體工程師的質。

技職院校資訊相關科系畢業生進入軟體產業主要以從事軟體程式設計為主，本計畫配合行政院資訊軟體人才培訓提升效能策略規劃構想，以程式設計師應具核心知識為依據，分析技職院校資訊相關科系的課程，發現二者間有不少差距，此為我國資訊工程師訓用失調之主因；因此依據程式設計師應具核心知識制定技職院校資訊相關科系程式設計類核心課程，共制定「計算機概論」、「資料結構與演算法」、「資料庫系統」、「軟體發展技術」、「程式語言」及「XML 語言」等六門課程，每一門課程均詳細說明其教學目標、課程內容、詳細課程大綱及實習項目。此六門課程之制定是採滾動發展方式完成，首先共同擬定「計算機概論」之教學目標與課程內容，續由一研究同仁邀集該校同事制定詳細課程大綱及實習項目，將此初稿交由其它研究同仁邀集所屬學校同事複審其完整性與合適性，再由全體研究同仁依各校所提問題逐一檢討與修正，若有不適合處則編入其它適合課程；再陸續制定其它課程；全部課程制定完成後，除與與勞委會乙級程式設計師技能檢定考試科目一一比較外，且以座談會方式邀請各技職院校資訊相關科系教師代表共同檢討與修正。

此研究所制定之課程與詳細課程大綱可供各技職院校資訊相關科系作為開課之參考依據，以提升教學品質；若能獲得教育部補助，邀請教學各技職院校資訊相關科系同仁共同制定適用國內技職院校資訊相關科系之本土教材與數位學習補充教材，將獲得更大效益。

關鍵字：軟體人才分類、程式設計師、核心知識、核心課程、證照

Abstract

The quality and the quantity of software engineers relate to the prospect of our software industry. In our country, there are more than thirty thousand of bachelors graduated from universities and institutes of technology from the domain of information system related departments. However, the software industry still has the problem to recruit the qualified candidates. It is the evidence of mismatching between our software engineer education programs from the industrial requirements. Upon the awareness of such problem, Science and Technology Advisory Group of Executive Yuan had submitted the adviser panel from the software industry to classify 14 categories of software engineer specialists according to their tasks. To prompt the quality of software engineer qualification, the core-knowledge of each specialist has been identified and specified as the criteria of certification programs.

It is the primary goal to help the graduated students to enter software industry for software engineering related works in the information related program at the vocational education institutes. It is the target of this project to in line with the strategic plan for upgrading the effectiveness of information software education programs of Executive Yuan. We based on the requirements of programmer's core knowledge to analyze the curricula of information system departments in vocational institutes and discovered the gap in between. It is the main cause of mismatching in the supply and demand of software engineers. With such understanding, the center curricula are proposed as the core knowledge for the information system related department in vocational institutes to coach programmers. Those core subjects including: "Introduction to Computers", "Data Structure and Algorithm", "Database system", "Software Development Techniques", "Programming Languages", and "XML languages." The teaching plan of each subject has been detailed with its teaching goal, subject contents, detail syllabus, and practicing exercises.

These teaching plans are developed according to the listed order to extend the profession. The development processes of "Introduction to Computers" are initiated by one researcher to gather advice from his/her colleagues to build up the detail syllabus and course contents. The draft was then passed through this research team to collect the suggestions for the completion and appropriateness. After the suggestions collected, the discussion meetings were held to clarify each adjustment and review. Each adjustment will be placed according to the conclusion of the meetings. After the completion of the subject, the contents were compared with the level-two qualification programmer exams of Council of Labor Affairs for appropriateness. A discussion panel that invited the representatives of in information system related teachers from vocational institutes for the final verification and review.

The purpose of this study is to develop a detail contents and syllabus for the related departments of vocational institutes as the curricula design reference. Hopefully, it can serve the purpose to upgrade the qualification of the graduation students in software industry. It may be more effective when Ministry of Education can further sponsor the processes of developing supplemental curricula material form the task force from the teacher of information system related department in vocational institutes.

Keywords: Software specialist category, programmer, core knowledge, core curriculum, certification

貳、研究緣起

自我國採取「高等教育開發」政策後，各地紛紛設立大學及技術學院，我國高等教育學府已近一百五十所，大專聯考的錄取率也逐年上昇，有志升學的高中職畢業生幾乎均可上大學就讀。然而，根據經建會的調查，有眾多科系的畢業生畢業後並無法從事本行工作，資訊科系畢業生亦不例外。資訊產業是我國最大產業，各企業每年均需招收新進人員以維持永續經營，然資訊廠商常抱怨找不到可用人才，尤其是軟體人才。有鑒於此問題的嚴重性，行政院科技顧問組於九十一年初組團赴印度考察其軟體產業及教育，團員涵蓋產官學研代表。印度的高等教育學府分研究型與教學型，教學型大學之硬體設備並不如國內大學，但教學內容卻與產業技術發展密切配合，且鼓勵學生參與國際証照技能鑑定，學生畢業後迅速投入職場而發揮所長。

此考察團返國後迅速提出考察報告，並由行政院科技顧問組邀請產官學研代表共同訂定軟體人才培訓策略，首先依軟體從業人員工作性質的不同將軟體專業人才分為十四類職類，如圖 1.1 所示，各職類人才再依其角色、任務及工作內容而訂出所需技能及核心知識，並由產學代表審閱其合適性與完整性，歷經半年方完成此分類工作。而後，為落實勞委會的軟體人才第二專長訓練工作，

遂進行「程式設計師」的訓練課程大綱及技能鑑定內容之訂定，並於九十二年三月完成。

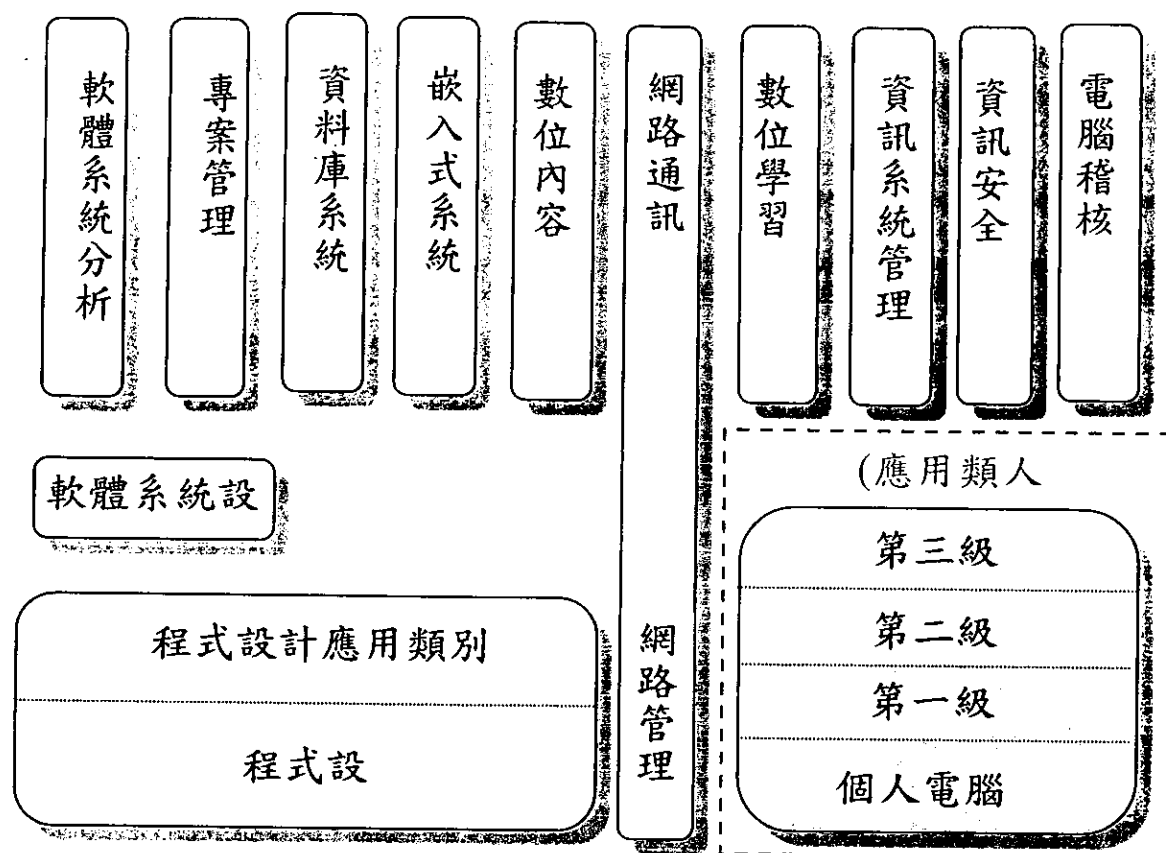


圖 1.1 資訊軟體人才分類

教育部技職司鑑於我國大學設有資工及資管系科者眾多，但學生程度差異眾多，實有必要依照行政院科技顧問組所訂「程式設計師」訓練課程大綱檢討資訊相關科系授課內容，進而擬定核心課程內容，作為日後教學的參考依據，以培育畢業生具有擔任「程式設計師」的核心技能，此即為本計畫的由來。

約廿年前，教育部曾對大學資訊工程學系及資訊科學系訂定必修課程名稱及學分數，而後遂歷經數次修訂，但變動不大，近

年來在「教育鬆綁」政策下已逐漸授權各大學自行訂定必修課程，唯各大學大都參考數所國立大學資訊科系之課程標準而制定其資訊科系之課程標準；然而一般大學的課程標準只制定必修及選修課程之名稱及學分數，並未細定其課程內容，因此在學生程度不一的情況下，同一門課程在不同學校的授課內容即有顯著差異，因而造成部份大學的學生之核心知識不足，進而影響其修其他進階課程知識不定的問題，甚至造成學生就業技能不足的問題。其次，教育部要求各大學提升師資陣容，各校教師擁有博士學位者日益增多，從提升學術研究水準的角度而言，此一政策至為正確；唯一般資訊科系的新進教師並未受過教學方法的訓練，且大學用書的作者均以其個人角度來編著書籍，因此不易以學生最易學習的角度來教導學生，再加上學生英文能力不足，遂易造成學生學習情緒低落，導至學習不足的現象。從就業的角度，學的多而不精，容易造成無法勝任工作的現象，而影響團隊工作進度，因此如何訓練學生擁有從事工作應有核心技能，是本研究的最重要的任務。

本報告之第參章介紹研究方法，第肆章分析技職體系學生程度；第伍章為畢業生就職情況、從事工作及所需技能之分析報告；第陸章為四所技職院校資訊課程之對應關係；第柒章為核心科目

課程之建議課程大綱，有「計算機概論」、「資料結構與演算法」、「資料庫系統」、「軟體發展技術」、「程式語言」、「XML 語言」等。第捌章為此課程大綱與行政院勞委會之乙級程式設計師技能鑑定考試內容之對應說明；第玖章為結論與建議。附錄一為九十三年一月十三日在長庚大學舉辦座談會之會議記錄；附錄二為歷次會議記錄。

參、研究方法

本研究的進行方式如下：

1. 了解目前學生程度低落的主要因素。
2. 調查畢業生就職情況、從事工作及所需技能，以作為制定課程之參考依據。
3. 依行政院科技顧問組所訂之「程式設計師」核心知識及訓練課程大綱檢討參與研究學校之課程符合程度。
4. 制定「程式設計師」所需核心知識之課程大綱，此項工作由一參與研究之同仁邀請其系內同仁共同討論擬定初稿後，再由其他大學之研究同仁共同討論。
5. 舉辦期中研討會，檢討課程大綱之實行性，並作修正。
6. 檢討所訂課程大綱內容與「程式設計師」技能鑑定科目的相關程度，並修正課程大綱內容。
7. 提出共同製作課程教材之建議方案。
8. 舉辦計畫研究成果說明會。

本計畫依上述方式執行尚稱順利，計畫成員每一至二週在台北行政院科技顧問組或參與研究同仁所屬學校開會，唯課程大綱擬定工作並非易事，其主要原因是研究同仁之看法差異甚大，因此先共同訂定「計算機概論」課程大綱後，才進行其他課程之擬

定，約花了近二個月才完成「計算機概論」課程大綱之訂定，而後才開始並行制定其他課程之課程大綱；此種方式雖影響工作進度，然卻拉近了大家看法而形成共識。

肆、技職體系學生程度分析

透過四校之教學座談會，收集各老師對技職體系學生程度分析提出之意見，經過交叉比對及去除該校自有問題後，歸納整理如下：

一. 教育背景

技職體系的教育訓練的主要目標是培育學生具有就業所需技能，然而，當老師在開始教學時，發現學生在程度上落差很大，使得老師在教學上遇到一些困難，其原因為：

1. 學生來源背景不同

技職體系學生來源管道比大學多，有高職畢業的學生，也有高中的學生，甚至有些學生在高中（職）所學的基本的知識與所考上的專業科系必不一致，使得學習遇到困難。

2. 課程內容不同

在現行的高職學校課程沒有一定的課程標準，內容有很大的差異，甚至有些課名相同，但上課的內容相差很大，使得學生所學的基本知識有很大的差異，進而造成老師在教學上的困擾。

3. 英文能力低落

技職體系學生的來源主要是高職學校，而高職學生普遍在

英文學習就比較不好，而在高職學校的英文課程訓練又不紮實，造成學生的英文能力低落，使得學生無法閱讀原文書。尤其對資訊科系學生，原文書閱讀能力更為重要。

二. 環境變遷

隨著時代的變遷，技職體系之教育也受到影響，其因素包括：

1. 大學數量急速增加

由於教育政策的改變，使得大學的數目不斷增加，造成所有高中（職）的學生都以考入大學做為第一要務，相對也造成技職體系學生素質低落或來源不足。

2. 資訊技術變化太快

近十年來，資訊技術的演進遠比過去四十年還快，一個程式語言可能在五到十年之內為另一個程式語言所取代，各類增進軟體發展的新技術及產品不斷問世，資訊老師需要不斷的進修學習，才不會與外界廠商的需求脫節，但校方設備卻無法跟進產業技術的腳步，訓練效益頗受影響。

3. 資訊業供需問題

學生從入學到畢業需要 2-4 年的時間，而資訊技術的變化又太快，使得當初因應需求訓練的學生，到了畢業後，可能因為技術改變而找不到適合工作。

三. 學習動機/認知

一般進入技職學校的學生，會因為不清楚技職學校教育的目的，而產生學習效果不佳的現象，主要現象包括：

1. 升學導向

目前技職教育的訓練目標究竟是就業導向，還是升學導向，每個人看法不一，不但使得老師在教學上有很大的差異性，有些老師只注重課程是否於學期內上完，而不考慮學生的吸收程度；有些學生只注重與研究所入學考試有關的科目，實務課程完全忽視，畢業後若考不上研究所，則毫無就業技能。

2. 對電腦的認知

部分學生認為讀資訊類科系就是玩電腦，並不清楚資訊類科系的學生畢業後應該具備哪些技能，所以當某些課跟學生興趣不合時，學生的學習意願會降低，進而造成學生在學習上的挫折。

3. 作業抄襲

現在的學生自我學習的能力很差，其原因可能是老師上課的方式無法為學生所接受，或是學生學習的意願低落，致

使對老師所出的作業練習，無法獨力完成，學生作業抄襲的情形非常嚴重。

4. 管理課程重於技術課程

對於資訊管理科系的學生來源，大多為商業類的學生。所以學生對管理的課程接受程度會重於資訊技術課程，使得管理課程的開課率大於資訊技術課程，進而使得資訊管理畢業學生的資訊技術不足以滿足外界的要求。

四. 現在的教學環境

目前各個技職體系為了學校的發展，都不斷的增加班級數或班級學生人數，而忽略教學環境的品質，其影響的結果有：

1. 大班

一般而言，班級上課的人數約在六十人左右，學生的個別差異很大，這對老師來說，很難注意到班上學生學習的成果。甚至，當學生上實驗課時，無法一人一機上課，使得學生在實作上的學習大打折扣。

2. 助教

在一般私立技職體系的學校裡，老師上課很少分配助教協助教學，這會造成老師花費許多的時間在改作業上，使得

老師不大願意出作業給學生作。

3. 師資來源

目前技職體系師資來源大多來自國內外大學的研究所畢業生，為了升等所要求的論文篇數，教學內容均以研究為導向，而非技職體系技術導向課程的訓練。

上述意見雖只是國內技職院校少數教師的意見，但這些意見反映高等教育開放過速的後遺症，其解決策略與方法非本研究範圍，故不深入討論。唯高中職的教育與技職院校的教學息息相關，希望教育當局能重視其關連性；而技職院校的教育目標、發展策略、及評鑑準則深深影響其教學，望能早日予以明確化，將有助於上述問題之解決。

伍、資管系畢業生就業狀況與資訊應用能力調查報告

調查畢業生就業狀況與目前正使用的資訊核心知識，一方面可了解畢業生在校所學知識在就業市場的實用性，另一方面也可作為規劃課程大綱之參考。本調查原欲對參與研究之四所技職院校資管系四技畢業生作調查，但畢業生人數有限，另委請中國技術學院對二技畢業生作調查，前者的問卷內容較豐富，後者則較簡；由於回卷人數有限（回收人，有效問卷人），研究小組僅對此作分析，但不列入本報告。

本章首先敘述資管系四技畢業生就業狀況與資訊應用能力調查結果，其次敘述資管系二技畢業生就業狀況與資訊應用能力調查結果。

5.1 資管系四技畢業生就業狀況與資訊應用能力調查結果

基本背景

1. 目前從事的工作（略）
2. 目前從事的工作屬性：
公家機關(7) 私人企業(含自營)(35) 個人工作室或自由業(1)
研究所進修中(7) 其他(1)
3. 目前從事的工作分類：
資訊或科技類(21) 一般產業與資訊相關部門(11)
一般產業與資訊無關部門(25) 其他(6)
4. 工作崗位上是否會用到電腦？ 是(50) 否(1)
5. 工作崗位上使用電腦的工作屬性：
系統開發與程式設計(11) 系統管理類(10)

- 行政與業務聯絡等一般應用(26) 其他(6)
6. 您的工作單位是否要求您或同仁取得相關專業證照？ 是(37) 否(11)
7. 整體而言，學校學的所有資訊科技課程對您在職場上的實用性，您覺得下面的主題是幫助較大的(可複選)
計算機概論(18) 電腦網路與通訊原理(42) 程式設計(各種)(26)
Internet(30) 資料庫相關課程(27) 影像處理(10) 軟體工程(5)
系統分析與設計(24) 演算法(7) 資料結構(10) 離散數學(0)
微積分/線性代數(4) 專業證照(21) 其他(3)
8. 整體而言，學校學的所有管理類的課程對您在職場上的實用性，您覺得下面的主題是幫助較大的(可複選)
管理學(31) 人力資源管理(10) 財務管理(11) 生產管理(2) 電子商務
(23) 會計學(11) 統計學(18) 經濟學(7) 物料管理(3) 作業研究(7)
其他(4)
9. 整體而言，學校學的所有通識類的課程對您在職場上的實用性，您覺得下面的主題是幫助較大的(可複選)
自然類(6) 婚姻類(5) 科學類(2) 情緒與心理類(35) 宗教類(0)
語文類(12) 其他(2)
10. 整個畢業專題的訓練過程，對您的幫助是什麼？(可複選)
系統分析(19) 系統開發能力(13) 團隊合作能力(19) 創意思考與解
決問題能力(23) 團隊合作的經驗(23) 文件撰寫(15) 其他(4)
11. 整體而言，您自己覺得在大學時對於資訊技術課程是一個什麼樣的學生？
非常用功(14) 普通(35) 不夠專心(4)
12. 整體而言，您自己覺得在大學時對於管理知識課程是一個什麼樣的學生？
非常用功(16) 普通(24) 不夠專心(2)
13. 整體而言，您自己覺得在大學時對於通識課程是一個什麼樣的學生？
非常用功(14) 普通(19) 不夠專心(9)
14. 整體而言，您覺得大學部畢業後就業應在學校培養那一類的能力(可複選)
資訊技術(28) 管理課程(20) 英文與語文能力(28) 通識課程(5)
其他(1)
15. 如果有機會您會願意從事資訊或科技相關行業嗎？ 是(38) 否(4)

A. 資訊基礎環境瞭解

- A-1. 請問您職場上電腦硬體環境包含(可複選)
PC(44) MAC(1) WORKSTATION(sum, hp, ibm, etc.)(17) 大型電腦(6) 超級電腦(0) 叢集電腦(2) 其他(3)
- A-2. 請問您職場上電腦作業環境包含(可複選) Windows(38)
Linux/UNIX(16) Mac OS(0) 其他(2)
- A-3. 每天用的最多的應用軟體類別為 (可複選)
E-mail(如 Outlook)(33) 文書處理(如 Word)(32)
數字處理(如 Excel)(16) 簡報作業(如 Powerpoint)(15)
繪圖處理(如 CorelDraw, Visio)(5) 簡易資料庫處理(如 Access)(8)
上網瀏覽器(IE(27) NS(0) 其他(請註明 1)) 其他(請註明 9)
- A-4. 電腦工作作業環境是否有連結網路 是(38) 否(2)
其網路環境是? Intranet(12) Internet(26) LAN(不對外)(4)
其他(請註明 1)
建置何種網路伺服器? Web server(29) DB server(13)
E-Mail server(15) Application server(7) 其他(請註明 1)
- A-5. 整體而言, 我在職場上利用電腦進行我的日常工作
這方面的能力
在學校就具備的(19) 工作中邊做邊學(19)

學校有開設這方面課程的基本知識, 你覺得 充足(6)
不足, 應要加強哪一部分:
資料庫應用設計規劃管理(20) 系統開發與專案管理(15)
文件撰寫(10) 硬體架設與維修(14) 資訊科技應用(7)
網路應用(15) 套裝軟體(3) 程式設計(17) 專案管理(9)
多層次應用軟體開發(4) 作業系統(12) 邏輯訓練(10)
人機介面(4) 資訊檢索(3) 產業環境及技術的分析(10)
了解人性需求問題並轉化為資訊問題(8) 網頁美化設計與製作(3)
動劃設計與製作(4) 遊戲設計與製作(0) 其他(請註明-)0

B. 程式開發相關問題

B-1. 請問您最近進行的程式開發為

內部行政支援與管理系統 (6) 企業內部研發專案 (2) 外部企業
委託專案 (3) 其他(請註明 0)

B-2. 您負責的是哪一部分(可複選)

系統分析

最常用的分析工具與方法

結構化分析法 (1) 物件導向式分析法+UML (2)

其他 (請註明 (0))

您在這方面的能力

在學校就具備的 (3) 工作中邊做邊學 (3)

學校有開設這方面課程的基本知識，您覺得 充足 (2)

不足 (3)，應要加強哪一部分(請註明)

功能模組程式設計

最常用的程式設計語言

Java C/C++ (2) VB Delphi (1) Cobol (2) SQL (0)

Prolog (0) Scripting language (ASP (4)、JSP (0)、PHP (5) 等)

其他(請註明) (1)

您在這方面的能力: 在學校就具備的 (3)

工作中邊做邊學 (4)

介面程式設計

最常用的介面設計語言或工具

Java (2) C/C++ (2) VB (2) Delphi (0)

html/css/dhtml (3) 其他(請註) (0)

您在這方面的能力 在學校就具備的 (3)

工作中邊做邊學 (4)

學校有開設這方面課程的基本知識，您覺得 充足 (1)

不足 (6)，應要加強哪一部分(請註明)

B-3. 在您的專案中，是否曾使用過兩種以上的程式語言？

Java (5) C/C++ (2) VB (2) Delphi (1)

Cobol (0) SQL (6) Prolog (0) ASP (7) PHP (4)

Perl (0) Python (0) JSP (3) html/css/dhtml (2)

其他(請註明) (0) (請繼續)

您在這方面的能力 在學校就具備的 (2)

工作中邊做邊學 (5)

學校有開設這方面課程的基本知識，您覺得 充足 (3) 不足 (7)，

應要加強哪一部分(請註明)

B-4. 就您在系統開發的經驗來看，系統開發與程式設計的技術中最常用到的技巧是(可複選)

● 資料結構形式(可複選)

串列(list) (1) 陣列(array) (4) 堆疊(stack) (4)
佇列(queue) (3) 雜湊表(hash table) (2) 樹(tree) (5)
圖型(graph) (2) 指標(pointer) (2) 函數(function) (7) 其他(請註明) (0)

您在這方面的能力 在學校就具備的 (2)

工作中邊做邊學 (5)

學校有開設這方面課程的基本知識，您覺得 充足 (2) 不足 (4)，應要加強哪一部分(請註明) (0)

● 演算法(若勾選此項請繼續回答下列題目)

演算法解題技巧(可複選)

分解法(divide and conquer) (1) 遞迴(recursion) (4)
線性規劃(linear programming) (1) 搜尋(search) (4) 排序(sort) (5)
最佳化(optimization) (1) 其他(請註明) (0)

您在這方面的能力 在學校就具備的 (2)

工作中邊做邊學 (5)

學校有開設這方面課程的基本知識，您覺得 充足 (2)

不足 (4)，應要加強哪一部分(請註明)

流程控制(For/While Loop If-Then-Else) (4)

輸出入及檔案處理 (4)

系統開發與元件製作(DLL, OCX, .so) (1)

● 資料庫存取控制(若勾選此項請繼續回答下列題目)

常用資料庫存取技術(可複選)

查詢(select) (5) 變動(insert、delete、update) (5)

資料庫設計(DDL) (2) 控制語言(DCL) (0)

預存程式(stored procedure) (1) 觸發程序(trigger) (1)

函數 (2) 檢視表 (1) 資料指標(cursor) (0)

其他(請註明) (0)

您在這方面的能力 在學校就具備的 (2)

工作中邊做邊學 (5)

學校有開設這方面課程的基本知識，您覺得 充足 (2)

不足 (4)，應要加強哪一部分(請註明)

● 資料庫管理(若勾選此項請繼續回答下列題目)

常用資料庫管理技術(可複選)

使用者權限管理 (3) 備份與還原 (3) 自動化管理 (1)

複寫 (1) 伺服器的監視 (1) 資料轉換 (4)
資料倉儲 (2) 分析服務(OLAP, Data Mining) (1)
Web Service 服務設定 (3) Notification 服務設定 (1)
其他(請註明) (0)

您在這方面的能力 在學校就具備的 (2) 工作中邊做邊學 (3)

學校有開設這方面課程的基本知識，您覺得 充足 (1) 不足 (3)，
應要加強哪一部分(請註明)

● 資料與檔案文件交換(若勾選此項請繼續回答下列題目) 0

常用資料與檔案文件交換技術(可複選)

XML (0) 自行寫 parser (0) 利用檔案進行 I/O (0)

利用資料庫進行 I/O (0) 其他(請註明) (3)

您在這方面的能力 在學校就具備的 (2)

工作中邊做邊學 (3)

學校有開設這方面課程的基本知識，您覺得 充足 (1)

不足 (3)，應要加強哪一部分(請註明)

● 物件導向設計(若勾選此項請繼續回答下列題目)

常用物件導向設計技術(可複選)

繼承(inheritance) (2) 多模(polymorphism) (0)

封裝(encapsulation) (1) 抽象化(abstraction) (0)

例外處理(exception handling) (2) 執行緒(thread) (1)

其他(請註明) (0)

您在這方面的能力 在學校就具備的 (1)

工作中邊做邊學 (1)

學校有開設這方面課程的基本知識，您覺得 充足 (1)

不足 (1)，應要加強哪一部分(請註明)

● 網頁美化設計與製作常用網頁設計工具或軟體

常用網頁設計工具或軟體(可複選)

MS FrontPage (3) DreamWeaver (4) HTML (4)

VBScript (3) JavaScript (3) PhotoImpact (4)

Flash(ActionScript) 4 其他(請註明) (0)

您在這方面的能力 在學校就具備的 (1)

工作中邊做邊學 (4)

學校有開設這方面課程的基本知識，您覺得 充足 (2)

不足 (3)，應要加強哪一部分(請註明)

B-5. 就您在系統開發的經驗，是否會有需要與資料庫進行連結

● 使用哪一種資料庫系統?(可複選)

Access (1) SQL server (6) Oracle (1) DB2 (0)

Informix (1) FoxPro (0) MySQL (0)

其他(請註明) (0)

● 連結資料庫的語言或工具(可複選)

Java (1) C/C++ (1) VB (1) Delphi (1)

JSP (2) ASP (3) PHP (3) 其他(請註) (0)

● 連結資料庫的介面(可複選):

ODBC (6) JDBC (2) 自行開發 (1)

其他(請註明) (0)

您在這方面的能力 在學校就具備的 (3) 工作中邊做邊學 (4)

學校有開設這方面課程的基本知識, 您覺得 充足 (2) 不足 (5),

應要加強哪一部分(請註明)

B-6. 就您在系統開發的經驗, 是否會用需要與網路進行連結

● 連結的的網路環境是?(可複選)

Intranet (3) Internet (5) LAN(不對外) (0)

其他(請註明) (0)

連結範圍(可複選)

Application layer (3) TCP/IP layer (4)

Socket layer (0) 其他(請註明) (0)

● 連結網路應用的語言或工具(可複選)

Java (1) C/C++ (2) VB (0) Delphi (0) JSP

(0) ASP (3) PHP (2) 其他(請註明) (0)

您在這方面的能力 在學校就具備的 (3) 工作中邊做邊學 (4)

學校有開設這方面課程的基本知識, 您覺得 充足 (2) 不足 (5),

應要加強哪一部分(請註明)

B-7. 您所參與的專案開發系統, 是否考量資訊安全? 對於資訊安全重視的程度如何?

以帳號及密碼方式管理 (4) 使用金鑰 (1) 使用防火牆 (4)

SSL (0) 其他(請註明) (0)

● 發行數位簽章: 自行發行 (0) SSL (2)

其他(請註明) (0)

● 公司內部是否要求定期更新密碼: 是 (4) 否 (0) 不要求(1)

您在這方面的能力: 在學校就具備的 (2) 工作中邊做邊學 (4)

學校有開設這方面課程的基本知識, 您覺得 充足 (2) 不足 (4), 應要

加強哪一部分(請註明)

B-8. 您所參與的專案開發系統, 是否特別考量軟體品質? 對於軟體品質如何處理?

● 程式註解: 要求以統一格式撰寫 (5) 各自發揮 (2) 不要求 (0)

- **程式測試:** 公司自訂統一測試資料 (2) 由客戶規定測試資料 (4)
不要求 (1) 版本控制與異動管理 (0)

您在這方面的能力: 在學校就具備的 (2) 工作中邊做邊學 (4)

學校有開設這方面課程的基本知識, 您覺得: 充足 (2) 不足 (4),
應要加哪一部分(請註明)

B-9. 整體而言, 我在職場上進行系統開發這方面的相關能力

在學校就具備的 (2) 工作中邊做邊學 (7)

學校有開設這方面課程的基本知識, 您覺得 充足 (1) 不足 (6),
應要加強哪一部分(請註明)

C. 系統管理相關問題

C-1. 請問你管理的系統為(可複選)

所有 MIS 系統 (7) 資料庫系統 (4) Mail Server (1)
硬體維修 (4) 內部網路 (6) 其他 (3)

C-2. 請問您是否取得系統管理相關證照? 有 (2) 沒有 (10)

C-3. 以系統管理面來說,您認為何種系統最易上手? Windows Base (12) Unix Base (0) 其他 (1)

C-4. 以系統管理面來說,您認為何種系統最易移轉資料?

Windows Base (12) Unix Base (0) 其他 (1)

C-5. 以系統管理面來說,您認為何種系統最易實行帳號整合?

Windows Base (10) Unix Base (3) 其他 (0)

C-6. 以系統管理面來說,您認為何種系統最不安全?

Windows Base (12) Unix Base (0) 其他 (2)

C-7. 以系統管理面來說,您認為何種系統最容易進行系統開發?

Windows Base (9) Unix Base (3) 其他 (1)

C-8. 您是否遇過系統出現狀況的情形? 是 (10) 不是 (0)

您的系統為: Windows Base 環境 (10) Unix Base 環境 (3) 其他 (1)

最常出現的狀況是: 系統本身錯誤 (4)

執行的程式造成系統錯誤 (4) 網路造成的問題 (8)

硬體毀壞 (5) 被入侵 (5) 未知的錯誤 (5) 其他 (1)

當遇到此類問題,您都如何解決? 自行處理為主 (8)

請廠商解決 (4) 其他 (2)

這一類狀況是否常常發生? 是 (4) 不是 (8) 其他 (0)

如果常常發生是否考慮更換系統環境?

是 (1) 不是 (11) 其他 (0)

您在這方面能力: 在學校就具備的 (5) 工作中邊做邊學習 (10)

學校是否有開設這方面課程的基本知識是否充足?

有 (2) 沒有 (2)

在學校學習此方面課程後,是否對工作有幫助?

有 (3) 沒有 (0)

學校是否有開設這方面課程的基本知識? 沒有,則學校是否有開設這方面課程的必要呢? 有 (3) 沒有 (1)

您建議學校開設哪些課程呢(可複選)?

微軟作業系統應用及管理 (5) FreeBSD 系統應用及管理 (1)

Linux 系統應用及管理 (5) Sun 作業系統應用及管理 (1)

資料庫的操作管理 (5) 應用程式管理 (0)

使用者端管理 (3) 設備管理 (2) 其他 (0)

如果學校要開設相關系統管理課程，學生應該先具備何種技能與知識(可複選)?

資料庫 (2) 資料結構 (2) 網路概念 (3)

系統開發與元件製作 (1) 網路安全 (0)

專案管理 (1) 硬體維修 (1) 演算法 (0)

微積分 (0) 統計 (0) 作業系統 (4)

流程控制 (1) 功能模組程式設計 (1)

系統分析 (2) 介面程式設計 (0) 輸出入及檔案處理 (1)

其他 (0)

C-9. 您修習這類課程是否已認證作為依歸? 是 (4) 不是 (8)

其他 (0)

C-10. 您修習這類課程是否考慮未來作相關工作? 是 (11)

不是 (2) 其他 (0)

系統安全相關問題:

C-11. 請問您遇過哪一類型問題(可複選)? 入侵與後門 (8) DoS (3)

監聽與截取 (0) 其他 (4)

C-12. 請問您當遇到安全問題大部分造成什麼後果(可複選)?

系統被修改 (1) 密法被破解 (1) 被遠端控制 (1)

被當跳板 (6) 伺服器無法服務 (6) 其他 (2)

C-13. 當遇到安全性問題時候如何解決?

自行修補相關漏洞或安全性問題 (8) 請廠商解決 (2)

其他 (2)

C-14. 您認為安全問題的發生是來自於何種情況(可複選)?

系統本身的安全性問題 (9) 使用者設定不良 (3)

病毒影響 (6) 駭客攻擊 (6) 其他 (1)

C-15. 您認為何種系統的安全問題最多?

Windows Base (7) Unix Base (0)

都差不多，端看使用者的設定與管理 (2) 其他 (0)

C-16. 請問您最常用什麼方式阻擋這一類安全性問題(可複選)?

架設防火牆 (8) 安裝自動修補機制 (2)

關閉不必要服務 (5) 架設 honeypot (0)

增加稽核機制與原則 (4) 其他 (0)

C-17. 最常用的檢查系統安全的工具: nessus (2) MBSA (0)

ISS (0) checkrootkit (1) tripewire (0) 其他 (3)

C-18. 您在這方面的能力: 在學校就具備的 (2)

工作中邊做邊學 (7)

C-19. 學校是否有開設這方面課程的基本知識？有，則學校開設這方面課程的基本知識是否充足？ 有 (1) 沒有 (0)

在學校學習此方面課程後，是否對工作有幫助？ 有 (0) 沒有 (0)

學校是否有開設這方面課程的基本知識？沒有，則學校是否有開設這方面課程的必要呢？有 (5) 沒有 (0)

您建議學校開設哪些課程呢(可複選)？

微軟作業系統應用及管理 (7) FreeBSD 系統應用及管理 (3)

Linux 系統應用及管理 (7) Sun 作業系統應用及管理 (2)

資料庫的操作管理 (6) 應用程式管理 (1)

使用者端管理 (6) 設備管理 (4) 其他 (0)

D. 資訊管理問題

D-1. 您負責的是哪一部門(可複選)?

● 業務行政部門

業務行政部門注重的專長是 ?

商事法 (1) 統計應用 (2) 資訊管理 (9) 行銷管理 (6)
企劃製作 (5) 資訊系統績效評估 (1) 作業管理 (5)
其他(請註明) (1)

您在這方面的能力：在學校就具備的 (4) 工作中邊做邊學 (12)

學校有開設這方面課程的基本知識，您覺得? 充足 (7)

不足 (8)，應要加強哪一部分(請註明)

● 人事行政部門

人事行政部門注重的專長是 ?

一般行政 (5) 統計應用 (0) 財務管理 (3)
管理數學 (1) 人力資源管理 (0) 其他(請註明) (0)

您在這方面的能力：在學校就具備的 (3)

工作中邊做邊學 (4)

學校有開設這方面課程的基本知識，您覺得? 充足 (6)

不足 (1)，應要加強哪一部分(請註明)

● 其他部門 您在這方面的能力：在學校就具備的 (1)

工作中邊做邊學 (6)

學校有開設這方面課程的基本知識，您覺得? 充足 (2)

不足 (5)，應要加強哪一部分(請註明)

D-2. 在您負責的業務中，與下面的管理課程是否有關(可複選)

● 會計學

會計學應用(可複選)

初會 (7) 中會 (0) 管理會計 (1)
財務報表 (2) 工廠會計 (0) 財務會計 (2)
其他(請註明) (0)

您在這方面的能力：在學校就具備的 (4) 工作中邊做邊學 (3)

學校有開設這方面課程的基本知識，您覺得? 充足 (4) 不足 (3)，應要加強哪一部分(請註明)

● 經濟學

經濟學應用(可複選) 個體經濟學 (1) 總體經濟 (1)

其他(請註明) 0

您在這方面的能力：在學校就具備的 (2)

工作中邊做邊學 (0)

學校有開設這方面課程的基本知識，您覺得？充足（1）

不足（1），應要加強哪一部分(請註明)

● 行政管理

在管理層次上注重的是？

資訊系統績效評估（3） 統計應用（6）

網路金融風險管理（1） 財務管理（5） 企劃製作（9）

虛擬組織管理（2） 其他(請註明)（1）

您在這方面的能力： 在學校就具備的（4）

工作中邊做邊學（11）

學校有開設這方面課程的基本知識，您覺得？ 充足（6）

不足（10），應要加強哪一部分(請註明)

● 統計應用

統計應用應用(可複選) 質性分析（3） 量化分析（5）

個案分析（8） 其他(請註明)（0）

您在這方面的能力： 在學校就具備的（5） 工作中邊做邊學（6）

學校有開設這方面課程的基本知識，您覺得？ 充足（4）

不足（1），應要加強哪一部分(請註明)

● 行銷企劃

在行銷實務應用層次上注重的是？

人力資源管理（2） 國際營運管理（0）

新產品開發規劃（2） 行銷策略系統管理（2）

網路行銷與電子商務（3） 其他(請註明)（0）

您在這方面的能力： 在學校就具備的（2） 工作中邊做邊學（1）

學校有開設這方面課程的基本知識，您覺得？ 充足（2）

不足（1），應要加強哪一部分(請註明)

● 物流與通路應用

實務應用層次上注重的是？

零售管理實務管理（1） 國際物流管理（0）

連鎖企業管理策略（0） 通路策略與管理（0）

供應鏈管理（0） 其他(請註明)（0）

您在這方面的能力： 在學校就具備的（0） 工作中邊做邊學（2）

學校有開設這方面課程的基本知識，您覺得？ 充足（0）

不足（2），應要加強哪一部分(請註明)

● 運輸與倉儲

其策略管理層次上注重的是？

運輸與倉儲營運管理（1） 產品策略與管理（0）

倉儲與物流自動化（0） 其他(請註明)（0）

您在這方面的能力： 在學校就具備的（0） 工作中邊做邊學（1）

學校有開設這方面課程的基本知識，您覺得？ 充足 (0)

不足 (1)，應要加強哪一部分(請註明)

● 策略資訊系統概念

其策略資訊管理層次上注重的是？

資訊與多媒體系統應用 (0) 企業資源規劃 (1)

資訊網路應用 (1) 資訊倫理與法律 (0)

人工智慧系統 (0) 資訊系統整合規劃 (2)

資訊科技與管理 (1) 其他(請註明) (0)

您在這方面的能力： 在學校就具備的 (0)

工作中邊做邊學 (2)

學校有開設這方面課程的基本知識，您覺得？ 充足 (0)

不足 (2)，應要加強哪一部分(請註明)

● 資訊系統績效評估

常用資訊系統評估方法(可複選) 平衡計分法 (0)

財務報表數據 (0) 其他(請註明) 0

您在這方面的能力： 在學校就具備的 (0) 工作中邊做邊學 (0)

學校有開設這方面課程的基本知識，您覺得？ 充足 (0)

不足 (0)，應要加強哪一部分(請註明)

● 財務管理

財務管理應用(可複選)

財務報表分析 (1) 證交法 (0)

會計事務 (1) 財務資訊系統 (0) 其他(請註明) (0)

您在這方面的能力： 在學校就具備的 (1) 工作中邊做邊學 (1)

學校有開設這方面課程的基本知識，您覺得？ 充足 (1)

不足 (0)，應要加強哪一部分(請註明)

● 國際營運管理

國際營運管理(可複選)

國際企業貿易 (1) 國際經貿法律與實務 (0) 國際金融 (1)

其他(請註明) (0)

您在這方面的能力：在學校就具備的 (0) 工作中邊做邊學 (1)

學校有開設這方面課程的基本知識，您覺得？ 充足 (0)

不足 (1)，應要加強哪一部分(請註明)

● 網路行銷與電子商務

常見資訊系統績效評估方法(可複選)

網路行銷實務 (1) 電子商務 (3) 網路廣告 (1)

其他(請註明) (0)

您在這方面的能力： 在學校就具備的 (1)

工作中邊做邊學 (3)

學校有開設這方面課程的基本知識，您覺得？ 充足 (3)

不足 (1)，應要加強哪一部分(請註明)

網路金融風險管理 (0)

企劃制作 (2)

虛擬組織管理 (1)

其他(請註明) (1)

D-3. 整體而言，我在職場上運用管理智識進行我的日常工作

在這方面的能力： 在學校就具備的 (6) 工作中邊做邊學 (7)

學校有開設這方面課程的基本知識，您覺得？ 充足 (6)

不足，應要加強哪一部份(可複選)

國際物流管理 (6) 連鎖企業管理策略 (6)

網各行銷與電子商務 (4) 統計應用 (2) 財務管理 (5)

網路金融風險管理 (2) 資訊科技應用 (9)

資訊科技系統企劃制作 (8) 虛擬組織管理 (3)

產品策略與管理 (6) 倉儲與物流自動化人力資源管理 (8)

新產品開發規劃資訊與多媒體系統實務 (1)

企業資源規劃管理 (7) 資訊倫理與法律 (1)

資訊網路應用 (5) 人工智慧系統 (4)

資訊系統整合規劃 (7) 資訊科技與管理 (7)

其他(請註明) (0)

E. 證照相關問題

E-1. 請問您目前取得何種證照(可複選)?

MCSE (2) RHCE (0) CCNA (12) MCDDBA (0)
Oracle (0) OCP (0) MSCD.NET (0) SCJP (3)
TCSE (0) CISSP (0) SSCP (0) 其他(請註明) (7)

E-2. 證照取得時間點:

在學校時候就取得證照 (7) 因工作需要才去考取證照 (3)
公司或部門提供經費考取證照 (0) 其他(請說明) (2)

E-3. 您在這方面的能力: 在學校就具備的 (6) 工作中邊做邊學 (5)

E-4. 學校是否有開設這方面課程的基本知識? 有 (7) 沒有 (5)

其他(請說明) (0)

選擇「有」, 則學校開設這方面課程的基本知識是否充足? 有 (5)

沒有 (3) 補充

在學校學習此方面課程後, 是否對工作有幫助? 有 (4) 沒有 (2)

選擇「沒有」, 則學校是否有開設這方面課程的必要呢? 有 (5)

沒有 (2) 補充

您建議學校開設哪些課程呢(可複選)?

MCSE (3) RHCE (1) CCNA (3)
MCDDBA (1) Oracle O OCP (1) MSCD.NET (1)
SCJP (0) TCSE (0) CISSP (0) SSCP (2)
其他(請說明) (0)

如果學校要開設相關認證課程, 學生應該先具備何種技能與知識(可複選)?

資料庫 (7) 資料結構 (3) 網路觀念 (5)
系統開發與元件製作 (1) 網路安全 (5)
專案管理 (2) 硬體維修 (2) 演算法 (1)
微積分 (0) 統計 (2) 作業系統 (6)
流程控制 (1) 功能模組程式設計 (0) 系統分析 (1)
介面程式設計 (1) 輸出入及檔案處理 (2)
其他(請補充) (0)

您是否認為認證為未來時勢所趨? 是 (5) 不是 (0)

您是否認為認證能夠代表一個人的能力與實力? 是 (6)

不是 (9)

5.2 資管系二技畢業生就業狀況與資訊應用能力調查結果

[從事資訊相關工作者調查分析報告]

本問卷針對業界所需資訊技能進行調查，請您就以下問題回答：

1. 您前一個學歷(專科或高職)是否為本科系 是-14人 否-9人
2. 您目前是否從事資訊相關工作 是-11人
3. 您目前工作電腦的作業系統環境 Windows-11人 Linux-4人 UNIX-2人
其他-1人(請註明-Novell)
4. 您目前工作是否需要利用網路 是-14人
 - (a) 網路環境為 Intranet-5人 Internet-11人 LAN(不對外)-2人
 - (b) 建置何種網路伺服器 Web server-8人 DB server-5人 E-Mail server-5人 Application server-4人
 - (c) 網路應用 script 語言在工作上較有用的 JSP-3人 ASP-6人 PHP-3人
5. 您目前工作的性質為(系統維護、系統開發、系統分析、網路/通訊管理)
 - 系統維護

(a) 您公司舊有系統大多以何種程式語言開發

Java-3人 C++-4人 Visual Basic-1人

(b) 使用的資料庫系統為

□ 系統開發

(a) 您公司現在大多以何種程式語言開發新系統

Java-9人 C++-5人

(b) 使用的資料庫系統為

SQL server-5人 Oracle-6人 Informix-1人
MySQL-3人

□ 系統分析

(以上三種工作性質回答)

	非常重要	很重要	普通	不重要	非常不重要
(a) 您覺得程式撰寫慣例及註解	6	3	1	0	0
(b) 您覺得程式語言結構	6	4	1	0	0

(c)您覺得人機界面設計	4	4	0	0	1
(d)您覺得軟體測試	5	3	1	0	1
(e)您覺得軟體文件製作	4	3	3	0	1
(f)您覺得資料結構	4	3	2	1	0
(g)您覺得資料庫設計	2	6	2	0	0
(h)您覺得 UML	3	2	1	2	1
(i)您覺得熟悉一種以上的程式語言	5	3	1	0	0

網路/通訊管理

	非常重要	很重要	普通	不重要	非常不重要
(a)您覺得 XML	2	5	2	0	0
(b)您覺得 Web-based 技術	3	2	1	0	2
(c)您覺得網路安全概念	4	5	0	0	0

6.管理課程在職場上的應用:

(a)您覺得生產管理/資訊系統	3	2	5	0	1
(b)您覺得行銷管理/資訊系統	0	3	6	0	2
(c)您覺得財務管理/資訊系統	0	2	5	0	2
(d)您覺得人力資源管理/資訊系統	1	3	5	1	2
(e)您覺得知識管理/資訊系統	2	5	2	0	2
(f)您覺得電子商務/資訊系統	2	8	0	0	1

7.基本資料

① 畢業學制：二技 10 人 四技 1 人

② 性別：男 -11 人

③ 年齡：21-25 -3 人 26-30 -4 人 30 以上 -4 人

④ 工作年資：1 年以下-1 人 2-5 年-5 人 5 年以上-5 人

陸、資訊管理系程式設計類課程與資訊軟體人才培訓課程之對映分析

了解畢業生所學技術與知識後，進一步分析三校開課程現況及其行政院科技顧問組所訂程式設計師核心知識的符合情況，以作為制定課程大綱的依據。其對映關係如下表所示：

資訊管理系程式設計類課程與資訊軟體人才培訓課程對映表

知識項目	時數(小時)	知識內容	中國技術學院			吳鳳技術學院			南華大學						
			課程名稱	時數(小時)	必/選修	課程名稱	時數(小時)	必/選修	課程名稱	時數(小時)	必/選修				
軟體環境基礎知識	47	30 電腦基本概念	資訊概論	45	72	必	計算機概論	45	70	必	計算機概論	45	66	必	
		12 XML					選	XML		12					
		6 網路通訊概念	網路概論、企業網路通訊	15		必選	企業資料通訊	12		必	企業資料通訊	9		必	
		3 資訊安全概念	網路安全	6		選	資訊安全	6		選	資訊安全	6		選	
		6 物件導向概念	程式設計	6		必	程式設計	6		必	程式設計	6		必	
軟體開發基礎知識	33	30 軟體開發概論	系統分析與設計、軟體工程	40	42	必	系統分析與設計、軟體工程	40		必	系統分析與設計	45	51	必	
		3 UML	軟體工程	2		必	UML 物件導向系統分析與設計	3		選	軟體工程	6		選	
軟體品質	33	15 軟體品質基本觀念	軟體工程	14	32	必	軟體工程	12	73	選	軟體工程	12	30	選	
		12 軟體測試策略與技術		14				12				12			

		6	軟體文件	系統分析與設計、軟體工程	4		必		6				6					
資料結構與演算法	42	3	鏈結串列	資料結構	4	70	必	資料結構	6	89	必	資料結構	6	93	必			
		3	堆疊及佇列		4				6				6					
		6	樹及其運用		8				9				9					
		3	圖形		2				6				9					
		2	演算法效能評估	資料結構、演算法	6		必	資料結構、演算法	3		必選	資料結構、演算法	3		必選			
		5	排序處理	資料結構	4		必	資料結構	6		必	資料結構	6		必	資料結構	3	必
		2	搜尋處理		4				2				3					
		3	雜湊應用		4				3				3					
		3	遞迴應用	程式設計	4		必		3				3					
		12	解題方法	演算法	30		必	演算法	45		選	演算法	45		選			
資料庫基本運用	60	3	資料庫基礎觀念	資料庫管理系統	3	75	必	資料庫管理系統	3	93	必	資料庫管理系統	3	45	必			
		10	資料庫管理系統		6				12				12					
		20	資料庫基本操作		24				18				9					
		18	資料庫設計	資料庫管理系統、分散式資料庫	30		必選		15				21					

		9	資料庫進階操作	資料庫管理系統、分散式資料庫	12		必選	網路資料庫	45		選	網路資料庫	45	45	
程式語言	180	90	Java、C++、Visual Basic、Delphi 以上四種語言選擇兩種上課	程式設計 (VB)	90	180	必	程式設計 (Java)	96	144	必	計概 (二) VB	45	90	必
		90		物件導向程式設計 (Java)	90		選	視覺化程式設計 (VB)	48		選	程式設計 (Java)	45		必
	90	24	語言基礎與流程控制	程式設計 (含實習)	80	92	必	程式語言 (含實習)	96	108	必	程式設計	45	90	必
		24	陣列指標或函數等結構												
		24	語言其他項目												
		9	資料庫存取控制												
9	系統開發與元件製作	程式設計、系統分析與設計、軟體工程	12	必	程式語言、系統分析與設計、軟體工程	12	必 必選	計概 (二) 程式設計	45	必					
專題製作	45	1. 採分組進行方式 2. 以所學之程式語言共同撰寫系統 3. 指導人員規劃下列項目：	專題研究 分組人數 (4-6人)	90	必	系統開發專題	120	必	系統開發專題	96	必				

		<ul style="list-style-type: none"> ● 專題題目 ● 開發環境 ● 系統架構 ● 分工方式 <p>4. 專題製作應強調 下列項目：</p> <ul style="list-style-type: none"> ● 團隊合作 ● 溝通協調 ● 程式相關文 件之撰寫 									
--	--	---	--	--	--	--	--	--	--	--	--

我們將樹德科技大學、中國技術學院、吳鳳技術學院及南華大學等資訊相關科系之程式設計人員類課程與行政院規劃之資訊軟體人才培訓課程作對映表

後，發現有些應該必備的核心課程有學校並沒有列，或有些核心科目有開課，但列在選修課程而非必修課程，即使是相同的必修課，其教授給學生的基本技能也不一。其原因為：

1. 目前各校教學內容有些並不合乎外界廠商所需人才應有之技能。
2. 課程名稱與課程內容並不一致，另外，課程間的垂直整合與橫向整合也不一致。
3. 老師授課方式以教課書為主，並沒有依學生能力來教學，以致於學生畢業後無法所學用於就學上。此現象可由中國技術學院、樹德科技大學所做的校友問卷調查中分析出來。

因此本計畫所規劃之課程內容時，擬以學生畢業時所需之技能為導向，並依現有學生程度為導向分配上課時數。目前各技職體系學校之資訊人才的基礎培訓課程項目並不一致，故本計劃試圖找出程式設計資訊人才必備之基礎專業科目，以利技職體系學校做教育上之訓練。本計劃試圖找出各核心課程及重點課程內容之項目，並規劃授課內容之順序，而制定的方法是以學生學習的角度來看，使學生能由淺入深循序漸進學習。在授課內容上，我們試圖找課程應教授的核心知識及學生應有的程式技巧訓練，並由學生程度為考量制定授課時數。所設計的授課內容可供其他學校做為參考，並希望能共同檢討以找出最適當的

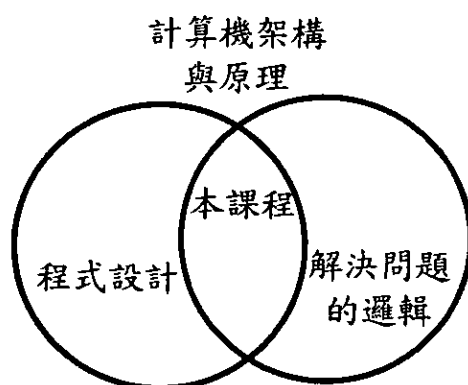
授課內容，且隨時將國內資訊產業所需之常用技術知識融入課程中，讓學生畢業後能儘快進入就業市場。

柒、核心科目課程

7.1 計算機概論課程

計算機概論課程是大一入門課程，主要介紹計算機運作之基本觀念與基礎知識，並訓練學生能運用所學程式語言撰寫程式以解決一些簡單問題，若程式撰寫的基本功夫沒練好，將影響學生學習其他課程之學習興趣，故可依學生程度而調整上課時數，且在課堂檢討學生常犯錯誤，使其不害怕寫程式。本章首先敘述課程設計理念，進而敘述課程大綱。

7.1.1 課程設計理念



教育目標：

1. 訓練學生瞭解計算機的基本運作原理與如何設計程式以解決問題。
2. 訓練學生熟悉兩種程式語言(包含一種傳統的循序式語言與物件導向式程式語言)的結構及使用，及運用物件導向觀念設計程式。
3. 訓練學生具備程式偵錯與測試能力。

課程內容：

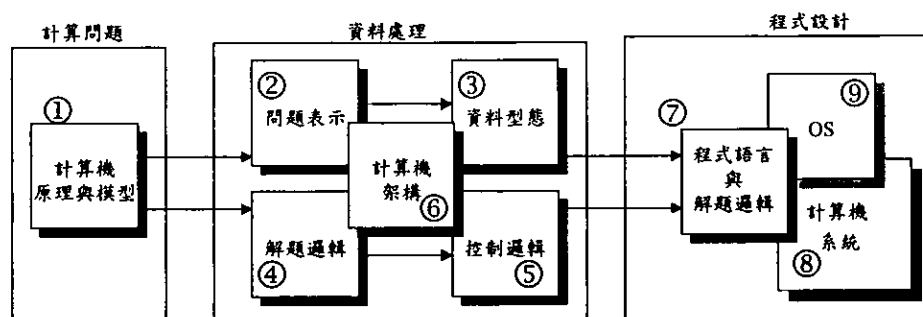
1. 電腦基本概念：硬體基本概念，作業系統，重要系統架構演進、多工、多緒、多處理器概念，執行環境概念，電腦網路。
2. 程式語言結構：Names, Scopes 及 Bindings，控制流程，資料型態，副程序及 Control Abstraction，建構可執行之程式。
3. 物件導向觀念：軟體元件、類別與物件、屬性、操作、方法、訊息、封裝、繼承。
4. 應用程式設計能力訓練。

設計構想與進行方式：

1. 課程的進行由淺顯易懂的計算機原理，配合實務操作，並經由程式語言的引導，讓學員能從解決問題的程式語言中，交互驗證計算機的運作。希望微觀(micro view)的從計算機資料處理，到宏觀(macro view)的problem-solving的過程，來介紹計算機、軟體程式、人與計算的關係。
2. 課程分成理論授課與實務操作兩部分，必須搭配一起進行，同時給予學生適度的作業練習與程式撰寫，以驗證所學
3. 適度介紹程式偵錯(debugging)的技巧與管理，並需要每週至少一小時與學生一起分析解決程式設計的錯誤與困難，並於適當時間實施期中期末測驗，以驗證學習成效。

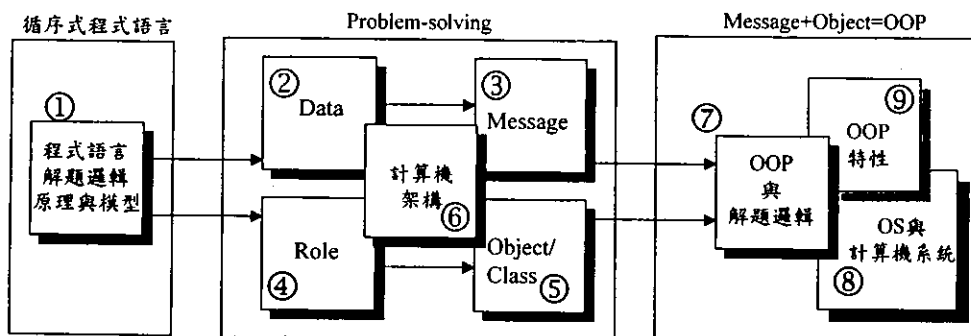
課程分成下列兩部分，分於上下學期實施：

- PART-I: (1) 從人類解決問題的需求出發，強調「計算(computing)」為解決問題的基本元素，並藉此介紹計算機的各部功能。(2) 解決問題的資料處理方式介紹問題的表示，並在計算機架構下對應資料型態。(3)以資料處理的解題邏輯，介紹計算機的處理方式(如內儲程式計算機架構)，並介紹計算機的控制邏輯功能。(4)介紹(2)及(3)的同時，需同步以程式語言的資料表示方式與控制邏輯加以對照式驗證，依照資料處理由小而大，由簡而繁的順序介紹程式如何進行資料的處理。(5)然後，由程式設計的角度看計算機四大元件的運作原理，讓學生知道在處理資料的過程中，程式如何在計算機中被處理(檔案，OS，網路等)。使學員能夠以一種循序式程式語言自行撰寫簡單的程式。概念如下圖。



- PART-II: (1) 從 PART-I 中介紹的第一種循序式程式語言的限制與問題解決 (problem solving & modeling)的角度，來(2)介紹解決問題不是以程式為思考的核心，而是以角色(role)與訊息(message)溝通的方式來進行。(3)由解決問題時的問題描述與解題邏輯對應至物件與訊息，在計算機系統的架構下，逐步導出物件導向式程式語言的基本原理。(4)課程引用同一個例子貫穿整個物件導向式程式設計的介绍：由物件(object)概念逐漸介紹到類別(class)，經由實例說明繼承(inheritance)及資料封裝(encapsulation)等物件式程式語言的重要特色。(5)以 OOP 的特色由簡而繁說明簡單物件與物件間，至複雜物

件間的繼承與交互合作的關係。同時，適度介紹其在計算機中運作方式與循序式程式的異同。概念如下圖。



分述如下

- PART-I: 從人類解決問題的需求出發，介紹計算機的沿革，強調「計算 (computing)」為解決問題的基本元素，並藉此介紹計算機的各部功能。然後，由程式設計的角度看計算機四大元件的運作原理。從簡單的數字計算，到資料龐大而複雜的問題，引導學員逐步瞭解循序式程式語言（如 VB, C 等）的發展與特色，更重要的是讓學員知道在處理資料的過程中，程式如何在計算機中被處理（檔案，OS，網路等）。使學員能夠以一種循序式程式語言自行撰寫簡單的程式。課程設計共分為分成 14 個單元，共 80 小時，每週上課 5 小時：

Unit 1: 序論

- 簡介人類對計算機的需求並介紹計算機擅長於處理大量、重複計算的工作。介紹電腦的誕生及演變過程，並說明為何各種問題需要轉換成計算問題。各種電腦應用系統之硬體、軟體、分類（超級電腦、迷你電腦、個人電腦、PC/Notebook/PDA）之簡介。同時需搭配動手認識電腦的實習，目的在於引發學生對計算機的興趣，並給予梗概式的認識。

Unit 2: 從人 problem-solving 的需求看計算機

- 點出計算 (computing) 與人 problem-solving 的關係。介紹各種不同資料形式與其運算操作的種類，並進一步說明操作型態與計算機器的設計需求。

Unit 3: 電腦/計算機系統

- 說明電腦/計算機系統由硬體、軟體、系統、人所組成。計算機四大功能運算、儲存、輸出、輸入的功能與之間的關係。同時以此概念式的方塊圖說明計算問題如何被處理。

Unit 4: 計算問題的表示方式

- 簡介表示問題的數字系統與數碼表示方式，說明 digit/bit/byte 與各種數字系統及轉換方式。介紹數字以外的問題表示方式，如符號系統表示，聲音與影像的表示。強調問題必須表示成數字型態的問題(呼應計算機四大功能& the black box)才能被計算機處理。

Unit 5: 計算機『運算』概念

- 延續前一單元說明資料如何被計算機處理的需求、概念、與演進。開始說明何謂『程式』與程式的基本元素(程式基本方塊)，並以范紐曼(Von Neumann)『內儲程式』概念計算機作為說明的基礎。同時搭配一種程式語言作為對照式的驗證與練習，給予基本計算數字型態的資料處理指令的練習，讓學生對「計算程式」有基本認識。

Unit 6: 計算機『儲存資料』概念

- 說明在 Unit 4 所表示出來的資料如何儲存在計算機中。說明計算過程對儲存體的需求與概念，包含資料儲存與程式的儲存。資料儲存部分介紹各種 Data types 在記憶體的位置與表示方式；程式的儲存部分介紹程式在記憶體中的位置與讀取到執行的過程。同時程式語言的練習，將各種不同資料型態表示、處理並儲存在程式中，作為對照式的驗證與練習，讓學生對「資料儲存」、「資料處理」與程式間的關係有基本認識。

Unit 7: 計算機的『控制單元、輸入、輸出』

- 介紹計算機的控制單元、輸入、輸出等功能。同時藉由控制的概念說明處理資料時，程式如何進行輸入輸出的動作。程式部分則介紹簡單的程式指令，布林代數運算及 IF...THEN...ELSE 的程式控制流程，並以此進行 I/O 的練習。

Unit 8: 計算機資料處理—I(資料處理流程)

- 介紹計算機進行資料處理的流程，強調資料處理是計算機的主要功能，並說明資料處理的型態與 problem-solving 的類型間的關係。藉此介紹進階的資料表示方式如 Array/Queue/Stack 等類型與原。再者，說明此種資料結構的適用時機與 problem-solving 對應各種控制結構 control structures(For, While, Do, Switch, etc.)。旨在讓學生瞭解計算機程式處理資料時對資料與程式控制間的關係。

Unit 9: 計算機資料處理—II(模組化程式設計)

- 延續 Unit 8 的資料處理觀念，說明將經常用到的資料處理步驟進行模組化的原因、優點與種類。介紹模組化程式設計的觀念，變數或參數的使用範圍與在記憶體中的關係。讓學生將前的程式加以模組化，並呼叫系統提供的現有的函數或副程式，藉由程式的練習，比較模組化程式的優點與特色。

Unit 10: 整合性描述

- 描述電腦的五大單元如何共同完成(程式的)計算工作，需對應至

程式與資料在五大單元的關連性，計算機提供的 BIOS+ROM+RAM 等硬體支援的提供。藉此讓學生從頭思索 Unit-1~9 所述計算機、程式、資料間的互動關係，並進行較大型的練習問題。

Unit 11: 資料處理—III (大量資料的處理)

- 延續 Unit 8 的資料處理觀念，說明計算機處理大量資料的時機與方式。藉著檔案系統的介紹讓學生瞭解檔案類型、檔案變換、檔案操作（建立、新增、刪除、修改等）與程式的關係。

Unit 12: 作業系統概念

- 介紹程式設計師應瞭解的作業系統概念與所提供的服務與資源。從對計算機資源的概念談程式對 OS 服務的需求，如 IO, file systems 等等。從程式的生命週期（開啟、執行、等待、保護）介紹單一程式與 OS 的關係，同時回顧計算機的五大功能中經由 OS 保護或控制的部分。

Unit 13: 作業系統與程式的互動

- 進一步從介紹程式設計師應瞭解的多工，多緒，多處理器的概念與此種環境下作業系統概念與所提供的服務與資源。並由此介紹重要系統架構演進(如中介軟體，組譯器與編譯器，遠端呼叫,主從式架構計算環境與各種 OS 的種類與適用時機)
- 開始讓學生進行整合式的程式練習。

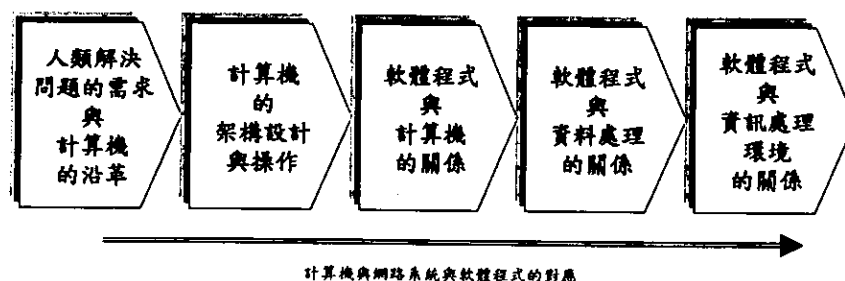
Unit 14: 電腦網路與網際網路

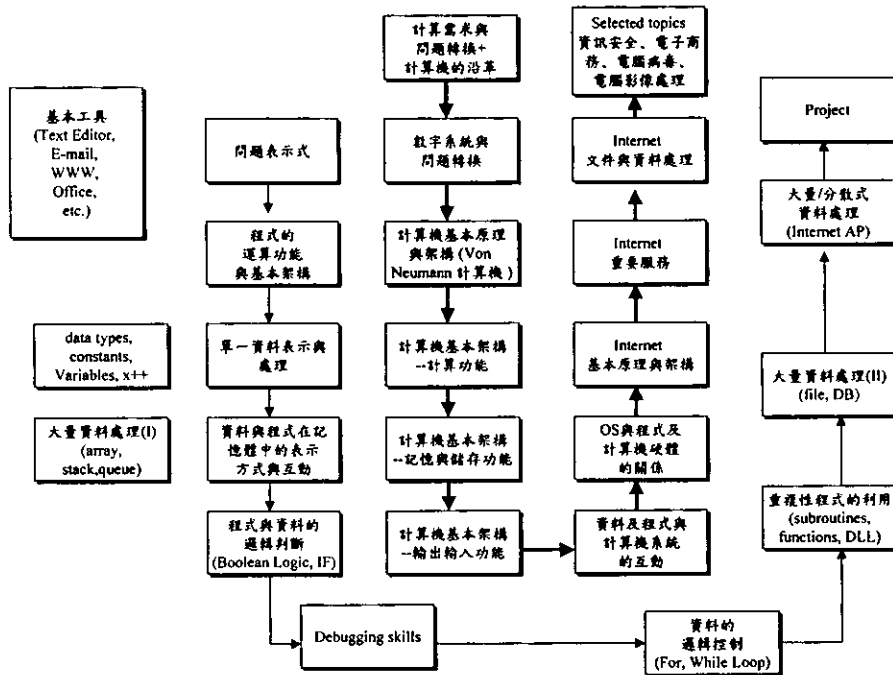
- 從「資源共享」的角度談對網路的需求與概念。經由共同約定式的通訊協定到標準化約定的通訊協定談 protocol 與網路類型 (LAN/MAN/WAN/Wireless)、網際網路的沿革、Internet 的發展歷史及台灣 Internet 的沿革、IP/router、Domain name、標記語言(markup language)如 HTML 與 XML 的概念與各種網際網路的應用與重要服務。實習部分則讓學生練習以現有軟體連上 Internet，並實際架設製作網頁(站)，並利用目前所學的程式技術(Java Applet, VB Script, Java Script)結合現有程式模組或函式庫撰寫簡單的 Internet 應用小程序式。

Unit 15: 資訊安全與電腦應用的未來趨勢

- 介紹資訊安全與電腦應用的未來趨勢。

各單元進行的順序的關係如下圖所示：





- PART-II: 從 PART-I 中介紹的第一種循序式程式語言的限制與問題解決 (problem solving & modeling) 的角度，來介紹解決問題不是以程式為思考的核心，而是以角色(role)與訊息(message)溝通的方式來進行。逐步導出物件導向式程式語言的基本原理。課程引用同一個例子貫穿整個物件導向式程式設計的介紹：比較分析其利用循序式程式語言解題時的限制，由物件(object)概念逐漸介紹到類別(class)，經由實例說明繼承(inheritance)及資料封裝(encapsulation)等物件式程式語言的重要特色。同時，適度介紹其在計算機中運作方式與循序式程式的異同。本課程僅介紹基本而重要的 OOP 技術，進階部分則在「軟體系統發展」課程中補充介紹。課程設計共分為分成 14 個單元，共 80 小時，每週上課 5 小時(上課學生必須先修 PART-I 課程)：

Unit 1: 以傳統程式語言發展軟體系統的限制

- 說明傳統程式語言發展軟體系統的限制，以導出物件導向式程式語言的需求。需以一個由傳統程式語言發展的軟體程式為例，說明其在設計、開發、維護、上的缺點與限制。可以讓學生用以前開發的程式，分析說明相對於本單元的限制。

Unit 2: Objects + Messages=OOP

- 介紹角色扮演(role-playing)式的問題解決策略，說明解決問題的角色與其所具有的屬性 (attribute)，對應 Data structure+algorithm=program 的觀念，說明 Objects + Messages= OOP 的重要性，而物件(object)對應角色是解決問題的基本單元。用一個較生活化的實例說明本單元的內容的重要觀念。

Unit 3: From Role Model/Object to CLASS

- 從物件的多元性與從屬關係導出類別(class)的觀念，並以實例說明

類別的定義與組成元素。以 Unit-2 的實例說明物件與類別的關係。

Unit 4: Problem Solving with Objects and Classes

- 說明以 OOP 方式解決問題時的步驟，並對應 Unit-1 的傳統程式語言發展軟體系統的限制，以說明 OOP 的優點。

Unit 5: Programming with Objects

- 開始說明以 OBJECT 為程式設計的基本單元，與物件導向程式的組成元素。說明如何建置一個 OOP 的程式，以 Unit-2 的實例說明如何在程式中定義並使用一個物件。再者，說明物件封裝屬性與繼承（完全相同屬性）的基本觀念。

Unit 6: Inheritance I—the relationship between Object and Class

- 介紹「繼承」的觀念與處理相同/差異的 properties，只談到屬性完全相同的屬性（垂直繼承+不衝突的屬性），以 Unit-2 的實例說明如何在程式中定義並使用物件的繼承關係。

Unit 7: Summary –I

- 對應 Unit-1 的傳統程式語言發展軟體系統的限制，以說明 OOP 的優點。

Unit 8: Life-cycle of Objects

- 說明軟體物件的生命週期與資料範圍(scope)，用以說明物件將資料隱藏起來的處理方式，與其在計算機系統如何被處理（物件起始化、執行、結束、多執行緒等重要觀念）。以 Unit-2 的實例說明程式中的物件的生命週期。

Unit 9: Use Package – I & Integration I

- 使用系統提供的軟體物件進行程式的輸出入(Window-based GUI, file I/O, etc.)。以 Unit-2 的實例說明如何在程式中加入 GUI 的物件。同時可以讓學生以至目前為止的 OOP 技術，設計一個較大型的作業。

Unit 10: Inter-Access of Properties among Objects

- 談物件屬性的公開與私密性 (private/public)，說明物件之間橫向與縱向的屬性呼叫與對應的方法(methods)，以 Unit-2 的實例說明程式中的物件如何使用父系物件的屬性，與其他類別的屬性。

Unit 11: Inheritance II—the relationship between Object and Class

- 說明屬性繼承發生衝突時的處理方式，介紹 subclass/superclass 的定義與使用，其相關的程式技巧。以 Unit-2 的實例說明程式中的物件增加與父系物件不同或衝突的屬性。

Unit 12: Abstract Class v.s. Normal Class

- 介紹概念式物件的意義與使用時機。說明介面物件與多重繼承的觀念。以 Unit-2 的實例說明程式中的物件如何使用概念式物件。

Unit 13: Use Package II --GUI & Container

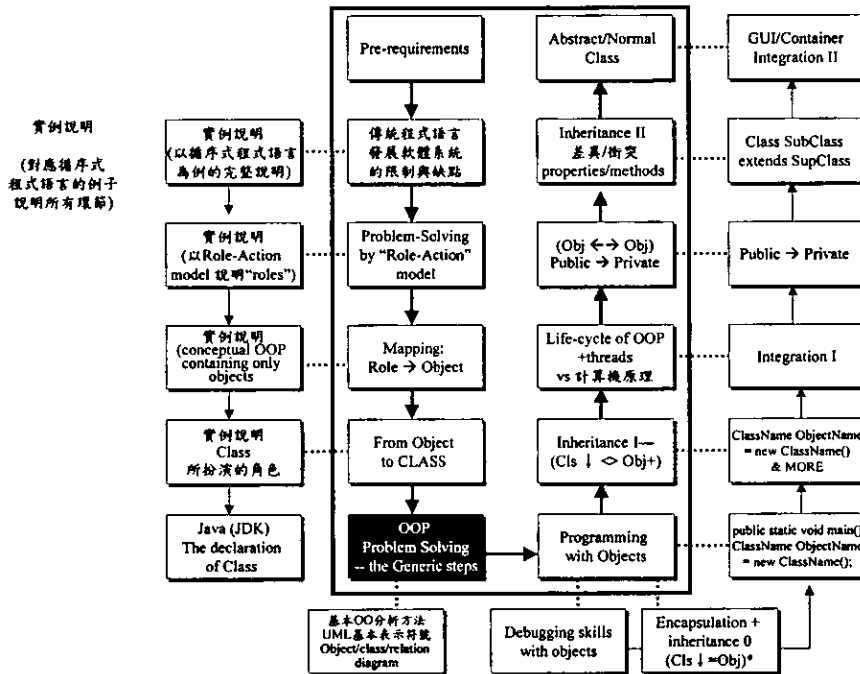
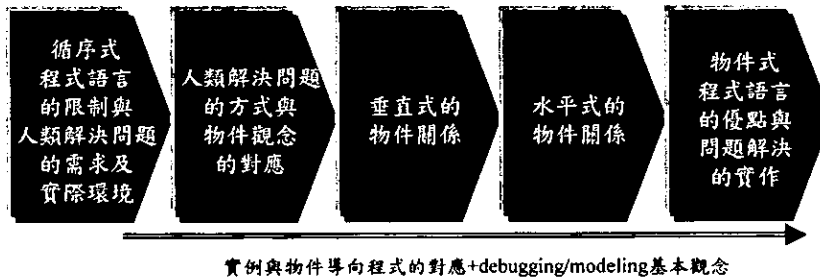
- 使用更多系統提供的軟體物件(.io, .util, .net)。以 Unit-2 的實例說明如何在程式中加入 GUI 與網路功能的物件。說明 event delegation model, container 等觀念同時可以讓學生以至目前為止的 OOP 技

術，設計一個大型的 project。

Unit 14: Summary II

- 對應 Unit-1 的傳統程式語言發展軟體系統的限制，以說明 OOP 的優點。

各單元進行的順序的關係如下圖所示：



7.1.2 課程大綱

計算機概論（一）課程大綱

Core Knowledge	授課時數 (Hr)	Programming Skills+LAB	實習時數 (Hr)
<p>1. 序論</p> <ul style="list-style-type: none"> ■ 對計算機的需求—人、資料(大量、重複計算) ■ 電腦的誕生及演變過程 <ul style="list-style-type: none"> ◆ 齒輪與碼表的模型 ◆ 速度(Hz)與散熱問題 ■ 問題的轉換—(各種問題→計算問題) ■ 硬體、軟體、電腦的分類 <ul style="list-style-type: none"> ◆ 超級電腦、迷你電腦、個人電腦、PC/Notebook/PDA ■ 各種電腦應用系統 ■ 簡介與其他 	3	<ul style="list-style-type: none"> ● 基本工具認識安裝(e.g., text editor) ● LAB: 各種應用軟體 ● LAB: 各種電腦外觀、種類、型態、廠牌... ● LAB: PC DIY** 	3
<p>2. 從人 problem-solving 的需求看計算機</p> <ul style="list-style-type: none"> ■ 點出計算(computing)與人 problem-solving 的關係 ■ The needs of representation and operations of data on machines ■ The requirements of such a "machine" 	1	<ul style="list-style-type: none"> ● LAB:各種軟體程式 <ul style="list-style-type: none"> ■ 系統程式 ■ 應用程式 ■ BIOS/firmware 	
<p>3. 電腦/計算機系統--硬體、軟體、系統、人</p> <ul style="list-style-type: none"> ■ 計算機四大功能 ■ 運算、儲存、輸出、輸入(只談外觀、種類、型態、廠牌...) ■ A Black Box Model – Computer ■ 問題的轉換—(各種問題→計算問題) 	2	<ul style="list-style-type: none"> ● LAB: Windows 小算盤 ● LAB: UltrEditor: 各種資料型態的內部表示法 ● LAB: UltrEditor: 程式執行檔的內部表示法 	
<p>4. 數字系統與數碼表示方式</p> <ul style="list-style-type: none"> ■ digit/bit/byte + base 2/8/16/32 ■ 數字系統轉換 ■ 符號系統表示 ■ 聲音與影像的表示 ■ 計算機運算模型---問題必須表示成數字型態的問題 <p>(呼應計算機四大功能& the black box)</p>	4	<ul style="list-style-type: none"> ● Homework Assignment: 	

<p>5. 計算機『運算』概念</p> <ul style="list-style-type: none"> ■ 需求與概念 ■ 計算機的操作-->開關 -->1000110-->程式(不談語言演進) ■ 何謂『程式』? ◆ 程式的基本元素(程式基本方塊) ■ 計算機運算模型--『內儲程式』概念計算機 <ul style="list-style-type: none"> ◆ 需求 ◆ 范紐曼(Von Neumann)『內儲程式』概念計算機 ◆ 何謂「程式」 ◆ 『內儲程式』概念計算機架構與運算原理 ◆ 此架構下的各種工作 ■ CPU/register (program counter) 	3	<ul style="list-style-type: none"> ● Programming-00 (建議用 VB/Java) <ul style="list-style-type: none"> ■ program 基本結構 ■ commands; statements ■ sequential execution ■ compiler ■ Ex: simple commands, e.g., printout 3+5 (numeric +-*÷-only) (暫不談 control commands) ● Homework Assignment: 	3
<p>6. 計算機『儲存』概念</p> <ul style="list-style-type: none"> ■ 需求與概念 ■ 資料儲存 <ul style="list-style-type: none"> ◆ 沒有記憶體體的計算機 CPU 如何運作 ◆ 記憶體—physical CKT/memory space/logic meaning ◆ Data types in memory ■ 程式儲存 <ul style="list-style-type: none"> ◆ 程式在記憶體中的位置 ◆ 讀取程式 ■ 程式執行—interaction among CPU/program/memory/data ■ CPU/register/memory 	2	<ul style="list-style-type: none"> ● Programming-01 <ul style="list-style-type: none"> ■ More commands; statements ■ data types ■ data/constants/variables/program in memory space ■ the use/call of data in programs ■ Ex: simple data operation, e.g., printout 3+5, "Red"+"Apple", "a"→"A" (暫不談 control commands) ● Debugging Skills in Programming with dumping memory for viewing constants and variables--I (配合記憶體與程式執行的觀念) ● Homework Assignment: 	3
<p>7. 計算機『控制單元、輸入、輸出』概念</p> <ul style="list-style-type: none"> ■ 需求與概念 ■ 程式如何輸入輸出資料 <ul style="list-style-type: none"> ◆ 控制單元 ◆ 輸入—執行的步驟 ◆ 輸出—執行的步驟 ■ 資料處理—0(資料的表示與輸出入處理) 	2	<ul style="list-style-type: none"> ● Programming-02 <ul style="list-style-type: none"> ■ Simple calculation & variable assignment ($y=x+1$; $x=x+1$) ■ 布林代數運算元與運算子 ■ 布林代數運算 in VB ($>=<$ IF...THEN...ELSE) ■ Simple I/O control (暫不談其他 control commands) ● Debugging Skills in Programming with dumping memory for viewing constants and variables+check points--II (配合記憶體與程式執行的觀念) ● Homework Assignment: 	6
<p>8. 資料處理—I(資料處理結構與流程)</p>	6	<ul style="list-style-type: none"> ● Programming-03:大量資料的處理 	3

<ul style="list-style-type: none"> ● 資料處理是計算機的主要功能 ● 資料處理的型態 v.s. problem-solving <ul style="list-style-type: none"> ■ 處理結構性資料的方式 <ul style="list-style-type: none"> ◆ Array/Queue/Stack ◆ 類型與原理 ◆ 適用時機+比較 		I <ul style="list-style-type: none"> ■ Array/Queue/Stack 類型與原理 ● Debugging Skills in Programming with array/queue/stack ● Homework Assignments 	
9. 資料處理—I(資料處理結構與流程) <ul style="list-style-type: none"> ■ problem-solving 對應各種 control structures ■ 程式的 control 結構 <ul style="list-style-type: none"> ◆ control structure 的種類 ◆ control structure 的應用時機 ■ control structure 的原理 	6	● Programming-03:大量資料的處理 I <ul style="list-style-type: none"> ■ 程式的 control 結構 <ul style="list-style-type: none"> ■ control structure 的種類 ■ control structure 的應用時機 ■ control structure 的原理 ■ 以 do loop 處理資料 <ul style="list-style-type: none"> ◆ 在 Memory 如何動作 ◆ For/While in do loop ■ 整合性練習 ● Debugging Skills in Programming with control structures ● Homework Assignment: 	3
10. 資料處理—II(模組化程式設計) <ul style="list-style-type: none"> ■ what/when 模組化程式設計 ■ why no in a single program? ■ subroutine 的觀念 ■ subroutine 的種類 ■ call-by-value/reference/address ■ modulized programming ■ scope of program/variable ■ 主從程式間的關係 (對應 program counter/memory space) 	6	● Programming-04:大量資料的處理 II <ul style="list-style-type: none"> ■ Procedure/function 的類型 ■ 將前面的程式碼加以模組化 # ● Debugging Skills in Programming with modulized structures ● Homework Assignment: 	2
11. 整合性描述 <ul style="list-style-type: none"> ■ 電腦的五大單元如何完成(程式的)計算工作? ■ 對應至 code/data/extra/stack segment 觀念 ■ BIOS+ROM+RAM, etc.: relationships and block diagrams ■ 對應 unit-3 的觀念 	1		0
12. 資料處理—III(大量資料的處理) <ul style="list-style-type: none"> ■ what/when 大量資料的處理 ■ why no in a single program with many constants and variables? ■ 檔案類型 ■ File Handle/File System 	2	● Programming-05:大量資料的處理 III <ul style="list-style-type: none"> ■ IO by Files ■ File handle (creation/deletion/modification) ■ Data to/from files ■ File 的轉換 by programs ● Debugging Skills in Programming with files ● Homework Assignment: 	2

<p>13. 作業系統概念</p> <ul style="list-style-type: none"> ■ 需求與概念 ■ computational resources (分配與控制) ■ 再談何謂「程式」與 program counter ■ services provided by OS <ul style="list-style-type: none"> ◆ IO, file systems, etc. ■ 單一 CPU 系統與程式 ■ creation/execution/protection/waiting/termination of a program ■ 單一程式與 OS 的關係 <ul style="list-style-type: none"> ◆ program 的生命週期(對應 unit-9 modularized programs) ◆ flow-char/control sequence ◆ 程式執行環境設定 ■ 系統軟體 	2		2
<p>14. 作業系統與程式的互動</p> <ul style="list-style-type: none"> ■ 多工, 多緒, 多處理器概念與程式間的關係 ■ Program vs OS (多重程式與多工多處理器 OS 的關係, program 的生命週期) ■ 重要系統架構演進 <ul style="list-style-type: none"> ◆ 中介軟體, 組譯器與編譯器 vs 作業系統 ◆ 遠端呼叫 ◆ 主從式架構計算環境與程式執行 ◆ OS 的種類與適用時機 (Windows/Unix/Linux/Mac OS, etc) 	2	<ul style="list-style-type: none"> ● LAB: Windows (task manager/ MyComputer/ server/ monitor) ● Programming-06 <ul style="list-style-type: none"> ■ Use services provided by OS (e.g., printing, file systems, sound, video, etc.) ■ Windows API/DLL ● Homework Assignment: 整合型練習—using array/stack/queue + files + sub-routines + OS services to take care different size of data ● Integrated project (given by some existing algorithms) 	
<p>15. 電腦網路與網際網路</p> <ul style="list-style-type: none"> ■ 需求與概念 ■ 網路的興起--網路的需求+資源共享 ■ Protocol 概述—共同約定式→標準化約定 ■ 網路類型--LAN/MAN/WAN/Wireless ■ 網際網路的沿革 & Internet 的發展歷史 <ul style="list-style-type: none"> ■ ARPAnet, NSFNET, 台灣 INTERNET 的沿革, TANet, TANet2 ■ Internet 上的身分識別 <ul style="list-style-type: none"> ◆ IP/router, Domain name 	6	<ul style="list-style-type: none"> ● LAB: Access to the Internet ● LAB: 網路的架設實務 <ul style="list-style-type: none"> ■ File server/Printer Server ■ 網路線/網路卡 ■ HUB/Modem/ADSL ■ 在 Windows 的相關設定 ● LAB: <ul style="list-style-type: none"> ■ 建立 Homepage ■ 作業系統環境之建立與設定 ■ 區域網路與網際網路設定 ■ 伺服器管理 ● Homework Assignment: <ul style="list-style-type: none"> ■ Programming-07: Access Internet with Java Applet (Java Script/VB Script) ■ Programming-08: connecting 	4

<ul style="list-style-type: none"> ■ Internet 上的文件 <ul style="list-style-type: none"> ◆ The concept of markup language ◆ HTTP/HTML ◆ XML—concept & applications ■ Internet 上的重要服務與伺服器管理與重要功能介紹 <ul style="list-style-type: none"> ◆ firewall/telnet/ftp/SSH/RPC/email/news/proxy/search engine/ICQ/E-mail/ftp/BBS/Plugins/其他 		<p>to the Internet (optional:: Connect to Internet or Intranet, Access to file server/printer server (using VB/Java components))</p>	
<p>16. 資訊安全與電腦應用的未來趨勢</p> <ul style="list-style-type: none"> ■ 資訊安全 ■ 電腦應用的未來趨勢 	1		0
<p>時數建議—四學分五小時 (16 週共 80 小時)</p>	49		31

計算機概論（二）課程大綱

Core Knowledge	授課時數 (Hr)	Programming Skills+LAB	實習時數(Hr)
<p>0. Pre-requirements</p> <ul style="list-style-type: none"> ● Procedure-based language 基本觀念與實作 ● Program running on a computer (CPU, Memory, I/O, OS) ● Von Neumann's Computer Architecture ● 程式語言--review <ul style="list-style-type: none"> ■ 機器語言/組合語言/高階語言 ■ 4GL/自然語言 ■ Compiler/Interpreter ■ Programming Language – Definition, Usage, Declaration 	0	<ul style="list-style-type: none"> ● Step-by-Step statement; ● Constant & Variable ● Data type ● Data type transformation 未來可對應到 <ul style="list-style-type: none"> ◆ data type →(Class) ◆ variable →(Object) ◆ DTT→(Polymorphism)* ● Statement (Std-in and Std-Out) ● Control Structures <ul style="list-style-type: none"> ◆ If else ◆ Do-loop (For/While) ● Array/Queue/Stack ● Procedures/Functions (by reference and by value) ● File (FILESTREAM) ● Java environment* <ul style="list-style-type: none"> ■ JDK、javac、java(virtual machine)、path environment setup、Java Code Conventions 	0
<p>1. 以傳統程式語言發展軟體系統的限制</p> <ul style="list-style-type: none"> ● One Big Main File ● Main File + Functions/Procedures <ul style="list-style-type: none"> ● Reusable codes ● Complete and complex ALGORITHM design ● Cooperative development ● Data Structure + algorithm = Program ● 缺點 <ul style="list-style-type: none"> ● Program management ● Debugging ● Reusability ● Development Hour/Man ● Hard to design complete ALGORITHM ● The needs of programming languages 	3	<p>以實例說明之</p> <p>Ex.</p> <ul style="list-style-type: none"> ● A simple Lotto/Workflow program implemented by traditional languages. (需搭配完整的程式實例) ● Homework Assignment: 	2
<p>2. Objects + Messages=OOP</p> <ul style="list-style-type: none"> ● Problem-solving by roles, not only by algorithms ● Modeling the roles by objects 	6	<p>以人類的問題解決方式與環境說明之</p> <ul style="list-style-type: none"> ● 分析 Lotto/Workflow problem (不需談程式)+其中的 role/object 	3

<ul style="list-style-type: none"> ● Role playing by “attributes” ● The mission of roles = handling data ● Role actions(intra-/inter-roles) = program ● The Role Model = Object ● Problem-solving by OOP = ● 只有 Object 是不夠的, An object is an instance of Class 		<ul style="list-style-type: none"> ● the role model of object in problem solving ● A “conceptual” OOP containing only objects 	
<p>3. From Role Model/Object to CLASS</p> <ul style="list-style-type: none"> ■ Various objects ■ Class, why? ■ The definition of Class ■ The component of Class <ul style="list-style-type: none"> ■ Data members (properties) ■ Member functions (methods) 	6	<p>I: 以人類的思考邏輯說明之</p> <ul style="list-style-type: none"> ● 分析 Lotto/Workflow program 中的 Class (不需談程式) <p>II: In Java (JDK)</p> <ul style="list-style-type: none"> ● The declaration of Class ● Lotto/Workflow Class in Java 	3
<p>4. Problem Solving with Objects and Classes</p> <ul style="list-style-type: none"> ■ Generic steps ■ 對應 Unit-1 Class/Object 的優點是什麼? 	2	<ul style="list-style-type: none"> ● 問題轉換 ● Lotto/Workflow problem→OOP 的步驟 (不需談程式) 	1
<p>5. Programming with Objects</p> <ul style="list-style-type: none"> ■ The definition of Application ■ The declaration of Application ■ The definition of Object ■ The construction of Object ■ Simple program in object/class ■ Encapsulation + property inheritance 0 <ul style="list-style-type: none"> ● Object 封裝屬性, 方法的特性 ● 基本繼承的觀念 	6	<p>以傳統程式語言的觀點說明對 inheritance 的需求</p> <p>I: 以人類的思考邏輯說明之</p> <ul style="list-style-type: none"> ● 分析 Lotto/Workflow program 中的 Class/Object (不需談程式) <p>II: In Java (JDK)</p> <ul style="list-style-type: none"> ● The entrance of program execution <ul style="list-style-type: none"> ● public static void main() ● ClassName ObjectName= new ClassName(); ● Ex: Handling Lotto/Workflow Objects in Java ● Debugging Skills in Programming with objects --I (配合記憶體與程式執行的觀念) ● Homework Assignment: 	2
<p>6. Inheritance I—the relationship between Object and Class</p> <ul style="list-style-type: none"> ■ What/Why inheritance ■ 繼承什麼東西--definition ■ 處理相同/差異的 properties ■ 只談到屬性完全相同的屬性垂直繼承+不衝突的屬性 	5	<p>I.以人類的思考邏輯說明之</p> <p>分析 Lotto/Workflow program 中 Class/Object 的 inheritance 關係 (不需談程式)</p> <p>II: In Java (JDK)</p> <p>Object/Class Inheritance in the Lotto/Workflow problems using Java</p> <p>Debugging Skills in Programming with object inheritance --I (配合記憶體與程式執行的觀念)</p> <p>Homework Assignment:</p>	2
<p>7. Summary I -- OOP 對應傳統循序式</p>	1		0

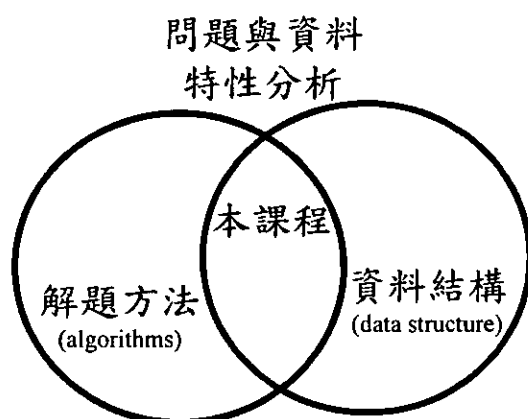
語言的優勢			
<p>8. Life-cycle of Objects</p> <ul style="list-style-type: none"> ● The Life-Cycle & Life-Scope of Object ● Data driven/Event driven <ul style="list-style-type: none"> ■ The invoking of objects by data/events ■ Initialization, Survival, Finalization ● The scope of objects <ul style="list-style-type: none"> ■ Data hiding ■ Access control of properties ■ Access control of methods ● The control of objects in a view of OS <ul style="list-style-type: none"> ■ Objects in memory spaces ■ Data/message passing among objects ● Threads (the needs and basic conceptual model) for managing objects in OS ● 對應 Von Neumann 計算機架構 ● OOP 的程式架構 (對應 unit I-12) 	6	<p>I.以計算機的運作角度說明之 Data/Even driven in the Lotto/Workflow problems(不需談程式) 說明 class/object/property 在計算機的四大元件+OS 的概念模型 (不需談程式)</p> <p>II: In Java (JDK)</p> <ul style="list-style-type: none"> ■ Constructor ■ this&this() ■ Destructor ■ Exercises ■ Debugging Skills in Programming with object (配合記憶體與程式執行的觀念) ■ Homework Assignment: 	2
<p>9. Use Package – I</p> <ul style="list-style-type: none"> ● File Access using Java Packages <ul style="list-style-type: none"> ■ java.io 	3	<ul style="list-style-type: none"> ■ File objects, FileReader, FileWriter ■ GUI version “Hello, World” ■ 將 Lotto/Workflow 的問題資料儲存起來 <p>Integration I</p> <ul style="list-style-type: none"> ■ 完整的 JAVA 作業/project ■ Learning Management & Summarization of Bugs 	2
<p>10. Inter-Access of Properties among Objects</p> <ul style="list-style-type: none"> ● Public → Private ● Private/Public properties <ul style="list-style-type: none"> ■ Access “Properties” ■ Private/Public--definitions ■ The access of Object’s properties ■ The access of Object’s methods 	6	<p>I. 以人類的思考邏輯說明之</p> <ul style="list-style-type: none"> ■ 分析 Lotto/Workflow program 中 Class/Object 之間(垂直+水平) private/public 屬性關係(不需談程式) <p>II: In Java (JDK)</p> <ul style="list-style-type: none"> ■ (public v.s private, not including protected) ■ setProperties & getProperties 	1
<p>11. Inheritance II—the relationship between Object and Class</p> <ul style="list-style-type: none"> ● 處理差異/衝突的 properties/methods ● The definition of subclass and superclass ● The declaration of subclass ● The component of Subclass 	6	<p>I. 以人類的思考邏輯說明之</p> <ul style="list-style-type: none"> ■ 分析 Lotto/Workflow program 中 Class/Object 的 inheritance 關係 (不需談程式) <p>II: In Java (JDK)</p> <ul style="list-style-type: none"> ■ Class SubClassName extends SupClassName {...} 	1

<ul style="list-style-type: none"> ■ Data members(properties) ■ Member functions (methods) ■ The inherited data members ■ The inherited member functions 		<ul style="list-style-type: none"> ■ Inheritance in the Lotto/Workflow problems using Java ■ Exercises 	
12. Abstract Class v.s Normal Class <ul style="list-style-type: none"> ● Definition ● Interface ● Multiple Inheritance 	3	I. 以人類的思考邏輯說明之 分析 Lotto/Workflow program 中需要 Abstract Class 的時機 (不需談程式) II: In Java (JDK) Shape(abstract class) v.s. Triangle, Circle, Square (concrete class)	1
13. Use Package II --GUI & Container <ul style="list-style-type: none"> ● Using Java Packages <ul style="list-style-type: none"> ■ java.io, java.util, java.net, etc. ■ Swing I ● Event delegation model ● GUI Components <ul style="list-style-type: none"> ■ Containers, etc. ■ Swing II 	3	GUI/Internet version of the Lotto/Workflow problems	1
14. Summary II -- OOP 對應傳統循序式語言的優勢	2	Integration II <ul style="list-style-type: none"> ■ 完整的 JAVA 作業/project ■ Learning Management & Summarization of Bugs 	1
時數建議—每週 5 小時 (共 80 小時)	58		22

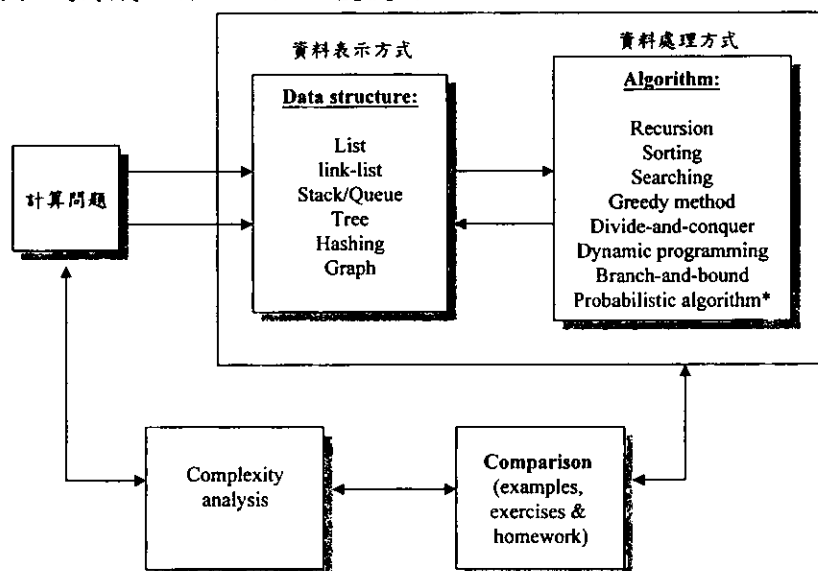
7.2 資料結構與演算法

7.2.1 課程設計理念

「資料結構與演算法」課程主要訓練學生如何運用適確的資料結構與解題策略來解決問題，是程式設計人員必備的核心知識與技術。程式除了需能解決問題外，當資料量龐大時，其執行的效能更是設計考量的重點，程式效能與選用之資料表示方式及演算法有密切關係。本課程的目標，在使學員充分了解各種常用的資料結構及演算法，使其正確的應用在程式設計上以發展出高品質的程式。



本課程的設計著重實際應用，避免艱深的問題解析。課程中儘量使用與生活相關的範例解說，配合程式實作與比較，使學員能活用資料結構與演算法。實作練習中並以物件導向的概念，要求學員將程式寫成元件，以供日後使用。



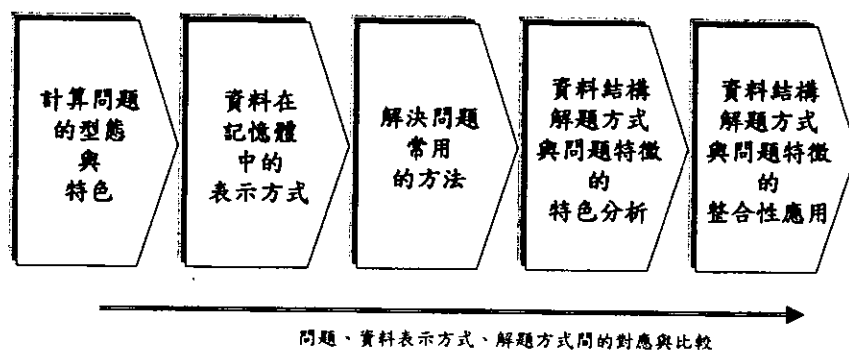
教育目標：

本課程的目標，在使學員充分了解各種常用的資料結構及演算法，且能將其正確的應用在程式設計上以發展出高品質的程式。

設計構想與進行方式：

本課程的設計著重實際應用，避免艱深的問題解析。課程中儘量使用與生活相關的範例解說，配合程式實作與比較，使學員能活用資料結構與演算法。實作練習中並以物件導向的概念，要求學員將程式寫成元件，以供日後使用。在課程進行中，前後單元有關的題材，儘量用同樣的範例說明，以供對照比較，使學員對整個課程融會貫通。

各單元進行的順序的關係如下圖所示：



課程內容：

本課程的內容涵蓋鏈結串列(Linked List)、堆疊(Stack)、佇列(Queue)、遞迴(Recursive)、樹狀結構(Tree)、資料排序(Sorting)、搜尋(Search)、圖形(Graph)、和 5 種基本的演算法。

Topic 1：課程開始時，以解決問題的觀點，說明資料有那些型態、需要如何處理，並以矩陣相乘的範例讓學員體會資料結構和資料處理的關係。

Topic 2：從陣列在應用上的限制開始，導入鏈結串列的需求，並比較兩者的優缺點及應用場合。

Topic 3、Topic 4：堆疊與佇列亦為密切相關的資料結構，也需要比較其個別的應用時機。

Topic 5：介紹遞迴的程式技巧，以供接續的各單元應用。

Topic 6：第六單元樹狀結構從一般樹的缺點帶入二元樹的主題，再從二元樹可能遇到的問題引入各種特殊樹；此單元很多內容都深具實用價值，故安排較長的時間，以便充分的探討。

Topic 7：資料排序與搜尋，先介紹一些常用的排序方法，使學者了解各種不同排序方法的效能和適用場合。排序的目的是為了方便搜尋，因此接著介紹各種排序對應的搜尋演算法。

Topic 8：探討圖形。在介紹了圖形的定義與表示法等基本觀念後，以生活上的最短路徑問題說明其應用。

Topic 9：介紹解題的 5 種演算法，屬於進階的應用。

7.2.2 課程大綱

Core Knowledge	Programming skills/LAB	時數
<p>1. 資料結構概論</p> <ul style="list-style-type: none"> ■ 資料與資訊 ■ 資料型態 <ul style="list-style-type: none"> ◆ 基本資料型態 - 字元、整數、浮點數 ◆ 延伸資料型態 - 陣列、鏈結串列、堆疊、佇列、類別 ■ 資料的操作 - 建立、取用、新增、刪除 ■ 資料結構 + 演算法 = 程式 ■ 評估程式的準則 <ul style="list-style-type: none"> ◆ 執行速度快 ◆ 佔用記憶體空間少 ■ 資料結構的意義 ■ 演算法的意義 ■ 陣列的意義(複習) ■ 以矩陣相乘運算為例，說明原始資料的類型及採用不同的資料結構，對程式執行速度與記憶體使用的影響。 ■ 空間複雜度的意義 ■ 時間複雜度的意義及 Big-O 表示法 ■ 當資料量很大時，時間複雜度的各種等級的關係 <p>$\log n < n < n \log n < n^2 < n^3 < 2^n$</p>	<ul style="list-style-type: none"> ● 以陣列實作兩個 100×100 的矩陣相乘運算，使用兩種類型的矩陣: 1、所有元素都不為 0，及 2、大部份元素都為 0(稀疏矩陣)。 	3
<p>2. 鏈結串列</p> <ul style="list-style-type: none"> ■ 陣列結構在新增與刪除資料時的問題 ■ 鏈結串列的意義 <ul style="list-style-type: none"> ◆ 有序的串列 ◆ 使用動態配置的分散記憶體空間 ■ 鏈結串列的運算 - 走訪、連結、節點的插入、節點的刪除、結構的反轉 ■ 陣列與鏈結串列的比較 	<ul style="list-style-type: none"> ● 利用鏈結串列實作兩多項式相乘。就多項式缺項的多寡，比較使用鏈結串列實作與使用陣列實作的優劣。 ● 以程式實作鏈結串列的各種運算。 	3

<ul style="list-style-type: none"> ◆ 資料的新增與刪除 ◆ 記憶體空間的大小 ◆ 資料的存取 ■ 鏈結串列的延伸 <ul style="list-style-type: none"> ◆ 環狀單向鏈結串列 ◆ 雙向鏈結串列 ◆ 環狀雙單向鏈結串列 ■ 		
<p>3. 堆疊</p> <ul style="list-style-type: none"> ■ 堆疊的定義 ■ 堆疊的基本運算 ■ 堆疊的實作 <ul style="list-style-type: none"> ◆ 用陣列結構 ◆ 用鏈結串列 ■ 堆疊的應用 <ul style="list-style-type: none"> ◆ 副程式的呼叫與返回 ◆ 運算式的轉換與求值 ■ 應注意的問題 - 資料是否超過堆疊最大容量 	<ul style="list-style-type: none"> ● 使用堆疊計算中序運算式的值。 	3
<p>4. 佇列</p> <ul style="list-style-type: none"> ■ 佇列的定義 ■ 佇列的基本運算 ■ 線性佇列與環狀佇列的實作 <ul style="list-style-type: none"> ◆ 用陣列結構 ◆ 用鏈結串列 ■ 優先佇列 ■ 雙佇列 ■ 佇列的應用 - 買票問題 	<ul style="list-style-type: none"> ● 實作排隊買票問題 	3
<p>5. 遞迴 (Recursive)</p> <ul style="list-style-type: none"> ■ 遞迴的意義 ■ 比較遞迴與迴圈的優缺點，以實作階乘 (Factorial) 運算為例。 <ul style="list-style-type: none"> ◆ 執行速度 ◆ 使用記憶體空間 	<ul style="list-style-type: none"> ● 分別以遞迴與迴圈實作階乘 (Factorial) 運算。 	3
<p>6. 樹狀結構</p> <ul style="list-style-type: none"> ■ 樹的定義 	<ul style="list-style-type: none"> ● 實作二元樹的走訪。 ● 實作 AVL 樹。 	12

<ul style="list-style-type: none"> ■ 樹的表示法 - 陣列表示法、鏈結串列表示法 ■ 一般樹的缺點 - 節點的分支數差異大時，浪費儲存空間。 ■ 二元樹的基本性質 ■ 二元樹的儲存方式 ■ 二元樹的建立 ■ 二元樹的走訪 - 前序、中序、後序走訪 ■ 引線二元樹 ■ 二元搜尋樹 ■ 二元樹的應用 - 運算式處理 ■ 二元樹的缺點 - 可能產生歪斜樹 ■ AVL 樹 ■ m 元搜尋樹及 B 樹 ■ Huffman 樹 		
<p>7. 資料排序與搜尋</p> <ul style="list-style-type: none"> ■ 排序的定義 ■ 內部排序法 <ul style="list-style-type: none"> ◆ 氣泡排序法 ◆ 選擇排序法 ◆ 二元樹排序法 ◆ 堆積排序法 ◆ 快速排序法 ■ 外部排序法 <ul style="list-style-type: none"> ◆ 合併排序法 ■ 各種排序法的比較 ■ 搜尋的定義 ■ 在循序結構上的搜尋 <ul style="list-style-type: none"> ◆ 循序搜尋法 ◆ 二分搜尋法 ◆ 內插搜尋法 ■ 索引結構的搜尋 <ul style="list-style-type: none"> ◆ 直接索引 ◆ 二元搜尋樹的索引 ◆ B 樹的索引 	<ul style="list-style-type: none"> ● 實作下列一種排序。可採分組方式，各組實作不同的排序。 <ul style="list-style-type: none"> ◆ 氣泡排序 ◆ 選擇排序 ◆ 二元樹排序 ◆ 堆積排序 ◆ 快速排序 ◆ 合併排序 ● 實作下列一種搜尋。可採分組方式，各組實作不同的搜尋，但使用相同的資料。例如：資料為某班學生的成績，考慮分別使用學生姓名、學號、座號為 key 的情況。 <ul style="list-style-type: none"> ◆ 循序搜尋 ◆ 二分搜尋 ◆ 內插搜尋 ◆ 雜湊法 	<p>10</p>

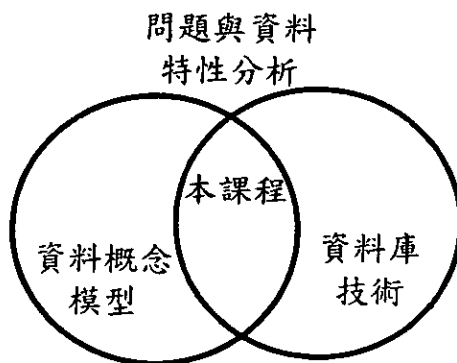
<ul style="list-style-type: none"> ■ 雜湊法 (Hashing) 		
<p>8. 圖形</p> <ul style="list-style-type: none"> ■ 圖形的定義 ■ 圖形的表示法 ■ 表示圖形的資料結構 <ul style="list-style-type: none"> ◆ 鄰接矩陣(adjacency matrix) ◆ 鄰接串列(adjacency list) ◆ 加權圖的鄰接矩陣 ■ 圖形的走訪 <ul style="list-style-type: none"> ◆ 廣度優先走訪(Breadth First Search,BFS) ◆ 深度優先走訪(Depth First Search,DFS) ■ 擴張樹 (Spanning Tree) ■ 最短路徑 (Shortest Path) ■ 拓樸排序 (Topological Sorting) ■ 關鍵路徑 (Critical Path) 	<ul style="list-style-type: none"> ● 實作 Dijkstra 演算法找最短路徑。 ● 實作拓樸排序。 	6
<p>9. 解題方法</p> <ul style="list-style-type: none"> ■ 貪婪演算法 (Greedy Algorithms) <ul style="list-style-type: none"> ◆ 貪婪演算法的精神 ◆ 貪婪演算法實例一：Minimum Spanning Trees (Kruskal and Prim Algorithms) ◆ 貪婪演算法實例二：Huffman 樹 ■ 分裂征服演算法 (Divide-and-Conquer Algorithms) <ul style="list-style-type: none"> ◆ 分裂征服演算法的精神 ◆ 分裂征服演算法實例一：二分搜尋法 ◆ 分裂征服演算法實例二：快速排序法 ■ 動態規劃演算法 (Dynamic Programming) <ul style="list-style-type: none"> ◆ 動態規劃演算法的精神 ◆ 動態規劃演算法實例一：Knapsack Problem 	<ul style="list-style-type: none"> ● 實作 Game tree 	20

<ul style="list-style-type: none"> ◆ 動態規劃演算法實例二： Matrix-Chain Problem ■ 分支設限演算法 (Branch-and Bound) <ul style="list-style-type: none"> ◆ 分支設限演算法的精神 ◆ 分支設限演算法實例一：Game Trees ◆ 分支設限演算法實例二： 3-Coloring Problem ■ 機率演算法 (Probabilistic Algorithms) <ul style="list-style-type: none"> ◆ 機率演算法的精神 ◆ 機率演算法實例一：Randomized Quicksort ◆ 機率演算法實例二：Random Search 		
	合計	63

7.3 資料庫系統

7.3.1 課程設計理念

前言：介紹資料庫相關的技術，並著重資料庫程式設計師在資料庫軟體設計過程中所扮演的角色應注意的事項。



教育目標：

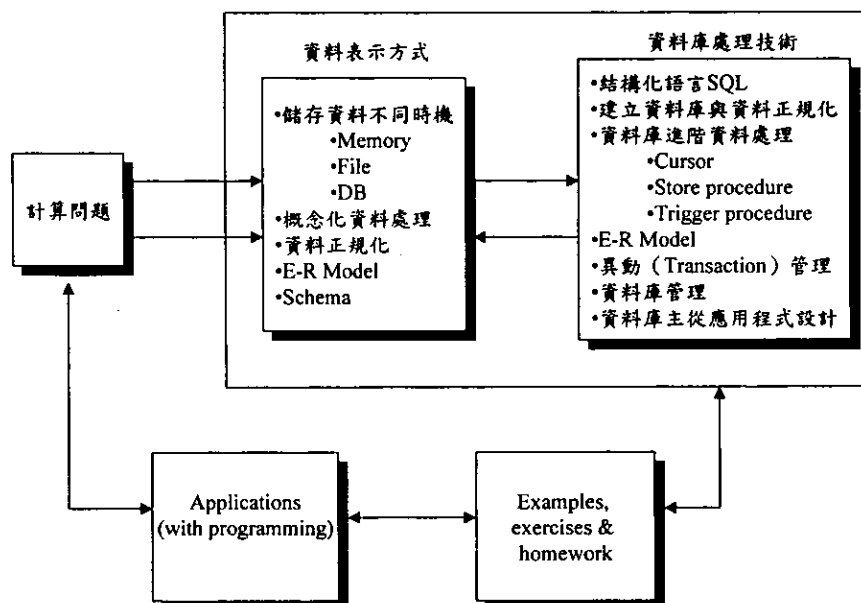
1. 訓練學生使用資料庫、建立資料庫及撰寫綱要 (Schema)。
2. 學生瞭解使用資料庫時機，並設計應用程式使用資料庫。

課程內容：

- (1) 資料庫系統基本觀念
- (2) 結構化語言 SQL
- (3) 建立資料庫與資料正規化
- (4) 資料庫進階資料處理——Cursor、Store procedure、Trigger procedure
- (5) 介紹使用 E-R Model
- (6) 異動 (Transaction) 管理
- (7) 資料庫管理
- (8) 資料庫主從應用程式設計

設計構想與進行方式：

「資料庫系統」是當程式設計師應該瞭解的系統；但是對系統使用方法的誤解，將使程式未受其利先受其害。由於資料庫系統的普及，甚至有免費的資料庫系統可以使用，參與的人員也越來越多，使得資料庫系統的程式開發變成是一個複雜的問題。因此如何有效地利用資料庫的技術管理資料，使資料處理更為效率，變成為一個重要的課題。



本課程強調的是程式設計師在資料庫系統應用程式開發過程中，應具備的知識與技能。希望學員完成此課程後，能夠在設計資料庫應用程式時，可以有效地利用資料庫的資料處理技巧，以最有效的方式完成設計程式的任務，解決問題。課程的基本精神在於讓學生從實作中驗證所學的技术，並從傳統的資料處理方式講起，一步一步向異質性資料庫的方向教學，希望本課程著重學生在實務上的應用，而不是在資料庫理論的教學。

本課程分成 12 個主題，共 58 小時：

Topic 1: 導論 (4 hr)

首先介紹資料儲存及處理模式的演進，進而使學生瞭解檔案系統與資料庫系統的差異，並說明在什麼情況下使用資料庫系統會比使用檔案系統來的方便。同時介紹資料庫系統架構，它各層 (Layer) 之間的關係。

Topic 2: 資料庫系統的操作環境介紹 (3hr)

介紹資料模式，主要說明放在關聯式資料庫系統上，使學生瞭解資料間的關連，進而介紹關聯表綱要及綱要與關聯表的關係。接下來說明主鍵、外來鍵在資料表中的重要性，並初略介紹資料庫的使用語法。

Topic 3: 結構化語言 SQL (I) —查詢部分(9hr)

從本單元開始詳細介紹 SQL 語法，這單元是以查詢部分。為了讓學生能充分瞭解 SQL 語法，我們將著重學生實作的部分。

Topic 4: 結構化語言 SQL (II) —異動部分(3hr)

介紹 SQL 語法中有關異動的部分。

Topic 5: 建立資料庫與資料表 (6hr)

介紹建立資料庫及資料表 SQL 語法，並說明建立資料表的規劃流程、如何蒐集資料並轉換成欄位、資料的完整性、及資料庫的正規化，同時介紹資料字典及系統目錄。

Topic 6: 關聯式資料庫之進階資料處理 (12hr)

介紹關聯式資料庫之進階資料處理技巧——Cursor、Store Procedure、Trigger Procedure、User Defined Function 等。

Topic 7: E-R Model (3hr)

介紹 E-R Model 的觀念、圖形符號及 E-R Model 與關聯式資料表的關係。

Topic 8: 檢視表 (View) (3hr)

介紹檢視表的用途，使用檢視表的優點。同時說明如何建立、修改檢視表，在檢視表上的資料處理方式。

Topic 9: 異動管理 (Transaction) 管理 (3hr)

介紹異動管理的目標及四個重要特性，當錯誤產生後，其回復處理的機制。最後介紹異動管理可能造成資料鎖定及鎖定死結問題。

Topic 10: 資料庫系統之管理 (2hr)

介紹資料庫系統之使用者的種類及各使用者在資料庫系統上的主要工作。之後介紹資料庫管理系統所提供資料管理之功能，及善用索引 (index) 增加資料庫系統查詢效率之方法。

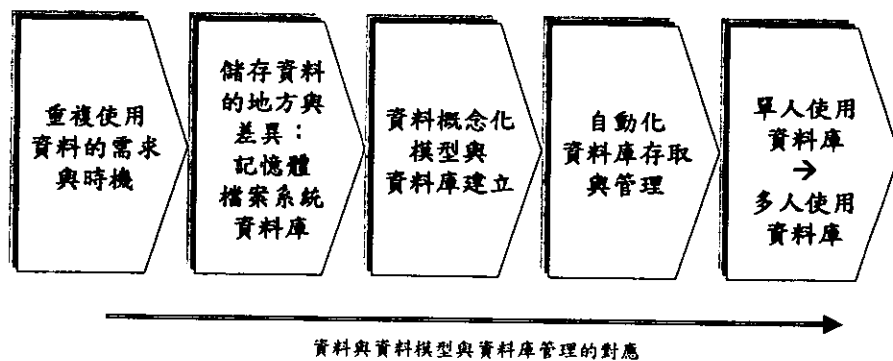
Topic 11: 發展主從式資料庫系統應用程式 (6hr)

介紹主從式系統架構及主從式系統中資料庫系統如何被使用，在使用前如何設定環境變數，如何以 ODBC (JDBC) 連結資料庫，並以實作驗證所學。

Topic 12: 其他資料庫系統介紹 (3hr)

介紹資料模式及其他資料庫系統，如：分散式資料庫系統、物件導向資料庫系統。另外教育學生使其瞭解使用資料庫的時機及注意事項。

各單元進行的順序的關係如下圖所示：



7.3.2 課程大綱

Core Knowledge	Programming Skills	時數 (小時)
<p>1. 導論 (3 小時)</p> <ul style="list-style-type: none"> ● 資料儲存及處理模式的演進 <ul style="list-style-type: none"> ◆ 第一階段——紙本檔案人工處理方式 ◆ 第二階段——電腦化循序檔案處理方式 ◆ 第三階段——電腦化直接存取式檔案處理方式 ◆ 第四階段——以記錄為處理單元的資料庫系統 ● 檔案處理問題 <ul style="list-style-type: none"> ◆ 一人多檔處理 ◆ 多人多檔處理 ● 資料庫管理系統的需求與功能 ● 資料庫系統架構 <ul style="list-style-type: none"> ◆ 外部層 (External Level) ◆ 概念層 (Conceptual Level) ◆ 內部層 (Internal Level) ◆ 各層之間的映對 (Mapping) 	<ul style="list-style-type: none"> ● 設計一實作，讓學生瞭解在檔案系統中處理資料的不方便性： <ul style="list-style-type: none"> ◆ 一人多檔案同時處理資料 ◆ 多人多檔案同時處理資料 設計多個成績檔案，由學生練習設計程式，以瞭解使用檔案系統的問題。 ● 設計一實作，讓學生瞭解資料放在檔案、資料庫、或記憶體中的優缺點。 ● MS SQL Server 的安裝 ● MS SQL Server 的系統目錄 	4
<p>2. 資料庫系統的操作環境介紹</p> <ul style="list-style-type: none"> ● 資料模式簡介 ● 關聯表綱要簡介 ● 關聯表的內容 <ul style="list-style-type: none"> ◆ 認識關聯、Primary Key 與 Foreign Key ● 基本命令指令 	<ul style="list-style-type: none"> ● 讓學生瞭解在的概念： <ol style="list-style-type: none"> 1. 利用的資料檔案，設計一個程式，以學號為 key 搜尋學生的各科成績。 2. 設計一個程式，以學號為 key，可修改加退選功能。 	3
<p>3. 結構化語言 SQL (I) ——查詢部分</p> <ul style="list-style-type: none"> ● 認識結構化語言 SQL ● SQL 語言的查詢用法 <ul style="list-style-type: none"> ◆ select 敘述的基本結構 ◆ select 子句 ◆ from 子句 ◆ Order by 子句 ◆ 聚合函數 ◆ Group by 子句 ◆ Having 子句 	<ul style="list-style-type: none"> ● 啟動 MS SQL Server ● 設定環境 ● 將成績資料轉換成資料表後，可讓學生做以下的實作： <ol style="list-style-type: none"> 1. 搜尋 2. 排名次 3. 算學期成績 4. 找出男、女生成績最好者 <p>使學生瞭解結構化語言 SQL 中查詢部分的用法</p> 	9

<ul style="list-style-type: none"> ◆ 巢狀式查詢 ◆ 聯集 (Union) 		
<p>4. 結構化語言 SQL (II) — 異動部分</p> <ul style="list-style-type: none"> ● SQL 語言的異動用法 <ul style="list-style-type: none"> ◆ Insert 敘述 ◆ Update 敘述 ◆ Delete 敘述 ◆ Select into 敘述 	<ul style="list-style-type: none"> ● 延續上個資料庫，請學生實作下列動作： <ol style="list-style-type: none"> 1. 修改成績 2. 加/退選學生名單 3. 學生調班 <p>使學生瞭解結構化語言 SQL 中異動部分的用法</p>	3
<p>5. 建立資料庫與資料表 (6 小時)</p> <ul style="list-style-type: none"> ● 資料定義語言 <ul style="list-style-type: none"> ◆ Create DataBase 敘述 ◆ Alter DataBase 敘述 ◆ Create Table 敘述 ◆ Alter Table 敘述 ◆ Drop Table 敘述 ◆ Truncate Table 敘述 ● 規劃關聯式資料庫 <ul style="list-style-type: none"> ◆ 簡易的規劃流程 ◆ 蒐集資料並轉換成欄位 ◆ 資料的完整性 ◆ 資料表關聯的種類 ◆ 資料庫的正規化 <ul style="list-style-type: none"> ➢ 第一正規化 ➢ 第二正規化 ➢ 第三正規化 ◆ 資料字典/系統目錄 	<ul style="list-style-type: none"> ● 請學生設計圖書館借書系統所需要的資料庫系統： <p>使學生瞭解規劃關聯式資料庫及資料庫與資料表的產生。</p>	6
<p>6. 關聯式資料庫之進階資料處理</p> <ul style="list-style-type: none"> ● Cursor <ul style="list-style-type: none"> ◆ Cursor 簡介 ◆ Cursor 的宣告、開啟、關閉、與移除 ◆ 透過 Cursor 修改或刪除資料 ◆ 使用 Cursor 的技巧 ● Store procedure <ul style="list-style-type: none"> ◆ Store procedure 簡介 ◆ Store procedure 的建立、使用、與修改 ◆ 設計 Store procedure 的技巧 ● Trigger procedure <ul style="list-style-type: none"> ◆ Trigger procedure 簡介 ◆ Trigger procedure 的建立與 	<ul style="list-style-type: none"> ● 使用權限的授與、取回 ● 預儲程序與觸發程序的設計：在圖書館借書系統的資料庫中建立 Cursor、Store procedure、Trigger procedure、自訂函數。 	12

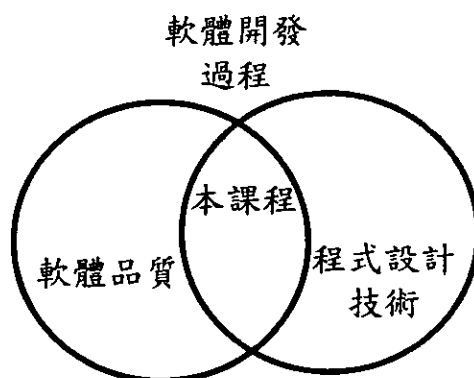
<ul style="list-style-type: none"> 修改 ◆ 設計 Trigger procedure 的技巧 ● 自訂函數 <ul style="list-style-type: none"> ◆ 自訂函數的特色 ◆ 自訂函數的建立、使用、與修改 ◆ 自訂函數的使用技巧 		
<p>7. E-R Model</p> <ul style="list-style-type: none"> ● E-R Model 的觀念 ● E-R Diagram ● E-R Model 與關聯式資料表的關係 	<ul style="list-style-type: none"> ● 畫出之前所設計之圖書館借書系統資料庫的 E-R Model。 	3
<p>8. 檢視表 (View)</p> <ul style="list-style-type: none"> ● 檢視表 (View) 簡介 ● 檢視表 (View) 上的資料處理 ● 檢視表 (View) 的優點 ● 建立檢視表 (View) <ul style="list-style-type: none"> ◆ 檢視表的用途 ◆ 檢視表的建立、修改 	<ul style="list-style-type: none"> ● 利用之前所設計之圖書館借書系統資料庫建立暫時需要的檢視表 (view)，如： <ol style="list-style-type: none"> 1. 借書未還的檢視表 2. 出借書籍中所有原文書所組成的檢視表 	3
<p>9. 異動 (Transaction) 的管理</p> <ul style="list-style-type: none"> ● 異動管理的目標 ● 異動的四個特性 ● 回復 (Failure Recovery) 處理 ● 異動的並行控制 ● 資料鎖定及鎖定死結問題 	<ul style="list-style-type: none"> ● 在圖書館借書系統資料庫中，請學生實作下列動作： <ol style="list-style-type: none"> 1. 借書。 2. 還書。 3. 查詢。 並在這些動作中加入異動管理，利用 SQL 語法來模擬多人同時使用資料表時異動的管理 	3
<p>10. 資料庫系統之管理</p> <ul style="list-style-type: none"> ● 資料庫系統之使用者 <ul style="list-style-type: none"> ◆ End-User ◆ Application programs ◆ DataBase Administrator ● 資料庫管理系統的基本功能 <ul style="list-style-type: none"> ◆ 資料定義 ◆ 資料處理 ◆ 資料安全 ◆ 資料備份 ● 善用索引 (index) 加快查詢效率 <ul style="list-style-type: none"> ◆ 索引介紹 ◆ 索引的種類 ◆ 索引的處理 ◆ 建立索引的注意事項 	<ul style="list-style-type: none"> ● 使用權限的授與、取回 ● 資料庫系統的備份與還原 	3
<p>11. 發展主從式資料庫系統應用程式</p>	<ul style="list-style-type: none"> ● 設計一個主從式的應用程式，他需 	6

<ul style="list-style-type: none"> ● 主從式架構簡介 ● 存取各種資料庫的準備工作與觀念 ● 環境設定 ● 以 ODBC (JDBC) 連結資料庫 ● ODBC (JDBC) API 介紹及其使用 	<p>用利用 ODBC (JDBC) 連結資料庫，而應用程式可分為：</p> <ol style="list-style-type: none"> 1. standalone。 2. Web application。 3. MS Access 	
<p>12. 其他資料庫系統介紹</p> <ul style="list-style-type: none"> ● 資料模式介紹 <ul style="list-style-type: none"> ◆ 階層式資料模式 ◆ 網路式資料模式 ◆ 關聯式資料模式 ● 分散式資料庫系統簡介 ● 物件導向資料庫系統簡介 ● 使用資料庫時機及注意事項 	<ul style="list-style-type: none"> ● 	3
<p>Total:</p>		58

7.4 軟體發展技術

7.4.1 課程設計理念

前言：今日市面的應用軟體功能強大，數十至數百萬行的程式比比皆是，需告團隊合作方可在指定期間內完成，因此有必要訓練學生熟悉軟體文件標準及寫作技巧、軟體測試與除錯技術等等，以培育其製作高品質的大型軟體模組能力。。



教育目標：

1. 使學生了解軟體生命週期、及軟體工程之由來、目的及範圍。
2. 使學生瞭解軟體品質的意義與指標，與提升軟體品質的一些技術。
3. 訓練學生熟悉程式偵錯與測試技術，及運用進階的(OOP)程式開發技術開發複雜軟體。
4. 訓練學生使用 UML 製作軟體文件的能力。
5. 訓練學生瞭解軟體發展各階段工作內容與軟體製作階段之所需技能。

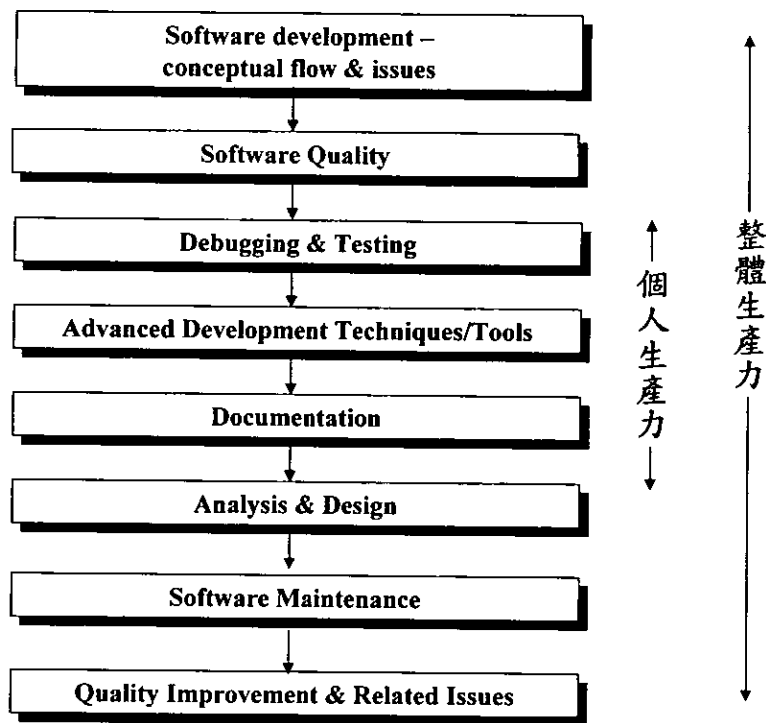
課程內容：

- (1) 軟體品質基本觀念與軟體開發相關程序
- (2) 軟體工程概論
- (3) 軟體測試策略與技術
- (4) 軟體文件種類與製作
- (5) UML
- (6) 軟體分析，設計維護及其他相關議題

設計構想與進行方式：

本課程強調的是程式設計師在整個軟體系統開發過程中，所扮演的角色及其相對應具備的知識與技能。希望學員完成此課程後，能夠在設計程式時，可以兼顧軟體品質，以最有效的方式完成設計程式的任務，解決問題。希望微觀(micro view)的從單一程式設計師設計單一程式的經驗與軟體品質的要求出發，進而涵蓋宏觀(macro view)的整個軟體系統的開發與品質，來強調程式設計師、軟體品質、軟體開發技術與工具的關係。課程的基本精神在於讓

學生在做中學(learning by doing)。課程進行時以學生以前設計開發的程式為對照經驗，用以驗證各單元的內容。並適度加入專題或較大型的程式系統開發，以強化學習成效。



本課程分成 8 個單元，共 48 小時：

Unit 1: Software development – conceptual flow & issues (3 hr)

- 簡介軟體系統開發過程，從單一程式設計師設計單一程式到多人參與的大型開發案可成會遭遇的問題，引導學生瞭解軟體工程的需求與核心問題，並從中瞭解程式設計師的定位。

Unit 2: Software Quality (3 hr)

- 介紹軟體品質的定義及需求，並介紹影響軟體品質不良的可能環節。同時請學生用本單元的經驗分析以前自己開發的程式的軟體品質。

Unit 3: Debugging & Testing (9 hr)

- 介紹經由 debugging 與 testing 的方式來強化軟體品質的過程與技巧。介紹各種可以提升程式設計師個人生產力的 debugging 技巧與工具，及軟體進行 testing 時的各種要求與標準。同時請學生用本單元的技巧與準則分析以前自己開發的程式的軟體品質。

Unit 4: Advanced Development Techniques/Tools (9 hr)

- 介紹數種可以提升程式設計師個人生產力的程式設計技巧與工具，例如新的技術標準(如 C#, .Net, OOP, XP 等)。強調 OOP 的進階技術，以銜接以前程式設計課程。同時請學生用本單元的技巧與準則進行幾個相關的練習。

Unit 5: Documentation (12 hr)

- 介紹文件撰寫在軟體開發工程中扮演的重要性，並特別強調與程式設計師在開

發程式的過程中有關的文件製作。利用 UML 此一工具，讓學生用本單元的技巧與準則進行以前自己開發的程式的文件撰寫。

Unit 6: Analysis & Design (6 hr)

- 簡介軟體系統的設計與分析過程。利用 UML 讓學生學會系統分析與設計相關的工作產出 (如 use cases, sequence diagram, collaboration diagram, etc.)。

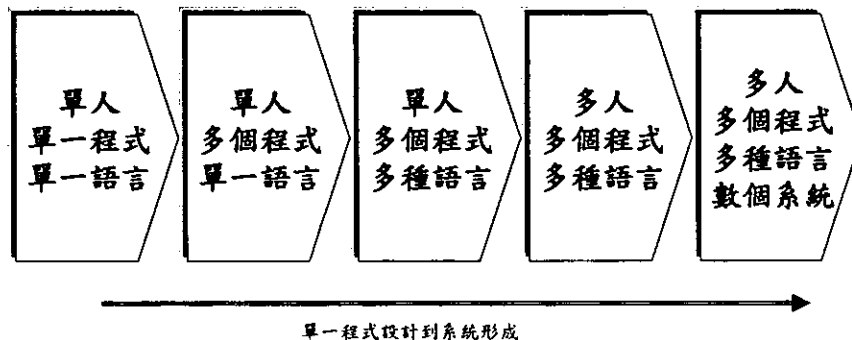
Unit 7: Software Maintenance (3 hr)

- 介紹軟體系統(或程式)完成後如何維護，內容涵蓋的是與程式設計師在開發時應該注意的相關知識與技巧。介紹軟體維護的工作種類與內容及管理程序等，如何將軟體設計的易於安裝、維護與更新，並介紹例如 CVS 此類的軟體版本控制方式。同時請學生用本單元的技巧與準則調整以前自己開發的程式的安裝與管理方式。

Unit 8: 軟體品質改善相關議題 (3 hr)

- 由軟體系統開發的不同階層 (個人、工作小組、組織) 的觀點簡介軟體開發程序與軟體品質改善相關議題，如 PSP、TSP、CMMI 等。

各單元進行的順序的關係如下圖所示：



7.4.2 課程大綱

Core Knowledge	Programming Skill Correspondence	Hours
<p>1. Software development – conceptual flow & issues</p> <ul style="list-style-type: none"> ■ Concept & Definition ■ Software crisis (not on schedule, meet budget and requirements) ■ A programmer with one (big) program → a team of programmers with one big and complicated software system <ul style="list-style-type: none"> ◆ The need for SOP (standard operating procedure) for software development ◆ The needs of Documentation, process, management (of people and software) Why? ■ Life-cycle of software development <ul style="list-style-type: none"> ◆ Requirement phase, Specification phase, Design phase, implementation phase, test & integration phase, maintenance phase, retirement phase ◆ The tasks performed in each phase ◆ The requirement and possible errors in each phase ■ Software engineering: <ul style="list-style-type: none"> ◆ Why? ◆ definitions, scope ◆ the needs ◆ objective: method→process→tool ■ The role of programmers in software engineering ■ How/Why can SE improve the productivity of programmers and the quality of software? 		3
<p>2. The concept of software quality (SQ)</p> <ul style="list-style-type: none"> ■ Concept, Definition & examples ■ How to justify & evaluate the quality of software <ul style="list-style-type: none"> ◆ Several indicators ◆ By one programmer v.s. by a team of programmers ◆ The flow/process that effect SQ ◆ The factors that effect SQ in the development of software by programmers ■ SQA(Software Quality Assurance) and V&V <ul style="list-style-type: none"> ■ SQA and V&V—methods and evaluation 	<ul style="list-style-type: none"> ■ Define and evaluate the quality of the programs that you have ever done. 	3

<ul style="list-style-type: none"> ■ For programmer: bug-free; debugging & testing ■ For software: meet the requirements <ul style="list-style-type: none"> ◆ Code review vs SQA: why & how ◆ Documentation vs SQA: why & what & how 		
<p>3. Debugging & Testing</p> <ul style="list-style-type: none"> ■ Debugging: Concept, Definition & examples <ul style="list-style-type: none"> ◆ 除錯技術：Bugs Management 概念、除錯方法、除錯工具之觀念與使用 ◆ Bug finding/allocation ◆ Bug management ◆ Bug fixing <p>----- (3hr)</p> ■ Testing: <ul style="list-style-type: none"> ◆ 軟體測試功能 ◆ 軟體測試種類：功能性測試、模組化測試、安全性測試，整合測試 (Graph-based 測試, etc.)、等價劃分、邊界值分析、比較測試、GUI 測試，主從系統測試、即時系統測試 ◆ 軟體測試策略：白箱測試、黑箱測試 ◆ 軟體測試步驟：測試計畫之撰寫、測試計畫之執行，單元測試、整合測試、驗證測試、系統測試 <p>----- (4hr)</p> ■ Test cases <ul style="list-style-type: none"> ◆ Generation & Preparation of test cases: how and where ◆ Correctness/Completeness/soundness ◆ 測試文件：測試計畫文件、測試案例 <p>----- (3hr)</p> 	<ul style="list-style-type: none"> ■ According to your programming experience <ul style="list-style-type: none"> ◆ Summarize the types, reasons, and solutions of bugs that you ever met ◆ List the debugging techniques that you ever did ◆ Analyze the effectiveness of new debugging techniques when being applied to your programs ◆ Test the programs that you have ever developed 	9
<p>4. Advanced Development Techniques/tools</p> <ul style="list-style-type: none"> ■ 模組化程式設計 ■ 程式設計程序、模組分割、元件化程式設計 ■ New (latest) technologies in programming/system development <ul style="list-style-type: none"> ◆ OOP、XP、C#、.net、etc. <p>----- (2hr)</p> ■ Advanced techniques in OOP <ul style="list-style-type: none"> ◆ 物件導向的進階技術：more about 類別、物件與屬性、套件與介面、關聯、繼承與聚集、多形與連結 ◆ Polymorphism <p>----- (2hr)</p> ◆ Method Overriding/Overloading <p>----- (2hr)</p> 	1-2 Homework Assignments	9

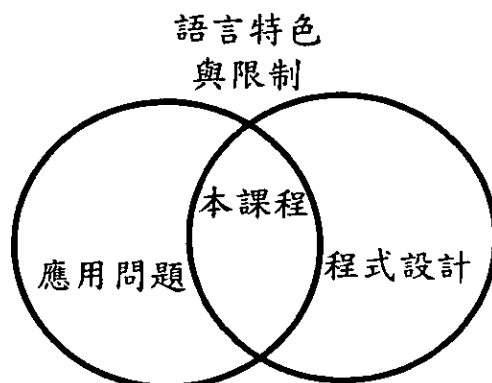
<ul style="list-style-type: none"> ◆ Java Beam ----- (1hr) ◆ N-tier applications with Java ----- (2hr) 		
<p>5. Documentation (especially the ones related to programming by one or many programmers)</p> <ul style="list-style-type: none"> ■ The needs, concept & definition, types of documentation ■ Documentation for development, management, and usage & deployment of software → 強調與 programmer 有關的 ■ 虛擬碼、Coding Convention ■ 軟體文件種類、軟體文件格式 <p>----- (3hr)</p> <ul style="list-style-type: none"> ■ UML-based documentation : <ul style="list-style-type: none"> ◆ Concept, Definition & examples ◆ Basic notations in UML <p>----- (6hr)</p> <ul style="list-style-type: none"> ◆ Write your own and refer to others ◆ 使用案例圖、類別圖與物件圖、順序圖與合作圖、狀態圖與活動圖、元件圖與佈署圖、界面、合作、擴充機制、框架與樣式 <p>----- (3hr)</p> <ul style="list-style-type: none"> ■ ISO/IEC 12207, 15504 文件規範簡介 	<ul style="list-style-type: none"> ■ Denoting the programs that you have ever developed in UML (all documents related to programmer) ■ Read the documents of your partner ■ Documentation for your programs <p>實作：Rational Rose</p> <p>UML for existing software</p>	12
<p>6. Analysis & Design</p> <ul style="list-style-type: none"> ■ Concept, Definition & examples ■ Analysis: 軟體系統開發人員應具備之軟體系統分析相關議題 <ul style="list-style-type: none"> ■ 軟體分析的工作內容 ■ 需求分析、領域分析 ■ UML for software analysis ■ 物件導向分析方法:情節為主之分析方法、使用案例之用法、使用案例規格、類別圖之推導, sequence diagram, collaboration diagram 之推導 ■ Design: 軟體系統開發人員應具備之軟體系統設計相關議題 <ul style="list-style-type: none"> ■ 軟體設計的工作內容--架構設計、元件設計、細部設計 ■ UML for software development (設計規格書:設計規格書項目、設計規格書撰寫) ■ Corresponding to Unit-5: The needs of documentation (architecture, components, details) 	<p>Sequence Diagram Collaboration Diagram</p> <p>UML for requirements of the outsides</p>	6
<p>7. Maintenance of software</p> <ul style="list-style-type: none"> ■ Concept, Definition & examples ■ 軟體維護 	<ul style="list-style-type: none"> ■ According to your programming experience ◆ Maintenance 	3

<ul style="list-style-type: none"> ◆ 種類與步驟 ◆ 軟體可維護性、軟體維護程序、軟體維護評量、軟體維護計劃、軟體維護管理、軟體維護記錄 ■ 軟體組態管理 (configuration management) : <ul style="list-style-type: none"> ◆ 軟體組態程序管理、軟體組態識別、軟體組態控制、軟體組態稽核、軟體組態報告、軟體交付管理 ■ CVS(Current Versions System) 	<ul style="list-style-type: none"> ◆ Configuration management ◆ Upgrading & version control 	
<p>8. 軟體品質改善相關議題</p> <ul style="list-style-type: none"> ■ Concept, Definition & examples ■ 階層式軟體品質改善 <ul style="list-style-type: none"> ■ A person→group/team→organization: different requirements and methods ■ Personal: PSP(Personal Software Process) ■ Grouped: TSP(Team Software Process) ■ Organizational; CMMI(Capability Maturity Model Integration) ■ 軟體工程的未來趨勢 		3
<p>Total: 3*16=48</p>		48

7.5 程式語言

7.5.1 課程設計理念

前言：學生已熟悉數種程式語言，且可運用這些語言設計程式，唯欲善用程式語言設計優質程式，實有必要了解程式語言的由來、類別及各自的弱點，進一步可自我學習新的程式語言，比較此新語言與其他語言之差異，使能依問題應用領域之不同而選擇適當語言。



教育目標：

1. 使學生了解程式語言之類別及應用範圍
2. 使學生了解各類程式語言演進的緣由，特性，及弱點
3. 訓練學生具有自我學習新程式語言的能力

課程內容：

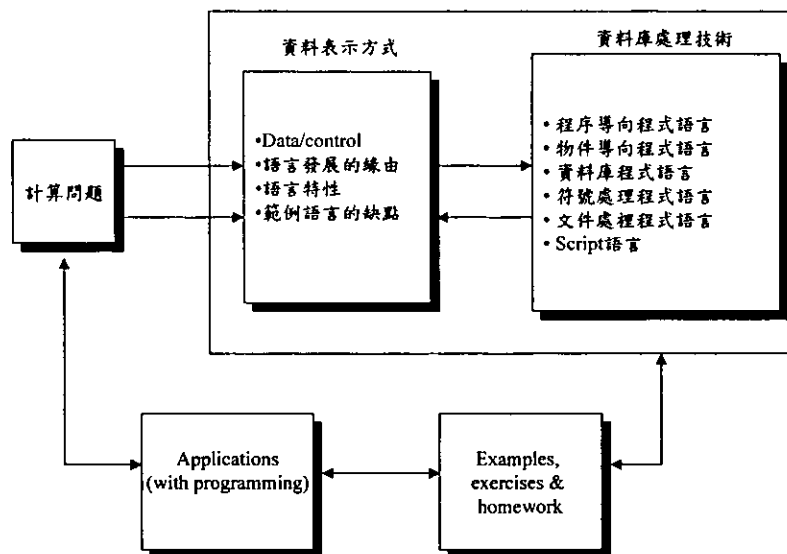
1. 程序導向程式語言
2. 物件導向程式語言
3. 資料庫程式語言
4. 符號處理程式語言
5. 文件處理程式語言
6. Script語言

設計構想與進行方式：

課程的進行依語言特性分成：程序導向程式語言；物件導向程式語言；資料庫程式語言；符號處理程式語言；文件處理程式語言；Script語言。每一類語言依其發展之時間由教師理論授課，而每一語言課程授課的流程：

1. 先強調語言發展的緣由：目的讓學生了解影響該程式語言演進之主要因素，如應用之影響、程式結構之影響、計算機架構之影響、程式撰寫的方便性之影響、軟體設計模式之影響。
2. 語言特性：目的讓學生了解該程式語言的一些重要性質。
3. 舉一範例：以實例說明，讓學生榮一體會與了解。

4. 語言的缺點：說明該程式語言之缺點及其使用注意事項。



本課程共分成 7 個主題，共 48 小時，各主題說明如下：

單元一、程式語言之目的，分類與演進(3hr)

1. 導論
2. 程式語言的目的
3. 程式語言的類別、性質與主要應用
4. 程序導向程式語言
5. 物件導向程式語言
6. 資料庫程式語言
7. 符號處理程式語言
8. 文件處理程式語言
9. Script語言

單元二、程序導向語言(Procedure-Oriented Language) (16hr)

1. 背景—機器語言與組合語言
2. 機器語言
3. Von Neumann Machine
4. 組合語言
5. FORTRAN
6. COBOL
7. BASIC
8. PASCAL
9. C
10. Ada
11. VB

單元三、物件導向語言 (5hr)

1. small talk
2. C++
3. Jave

單元四、資料庫語言 (5hr)

1. SQL, SQL1
2. SQL2
3. SQL3

單元五、符號處理程式語言 (4hr)

1. LISP
2. Prolog

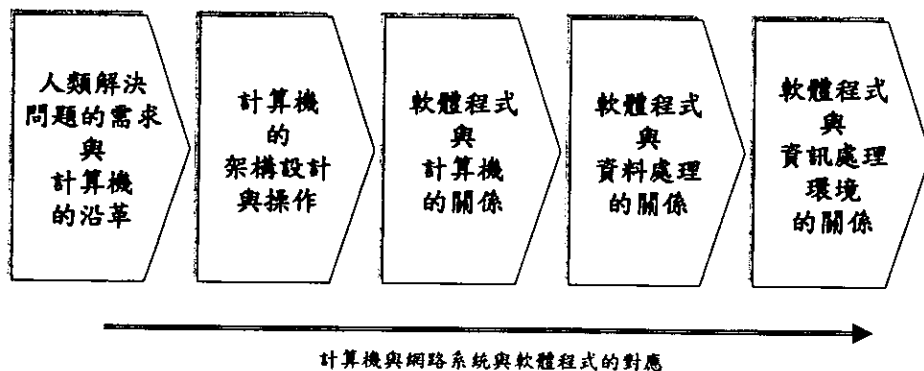
單元六、文件處理程式語言 (7hr)

1. Tex
2. Postscript
3. Latex
4. PDF

單元七、Script語言 (8hr)

1. Perl
2. Tcl/tk
3. Python
4. Ruby
5. PHP

各單元進行的順序的關係如下圖所示：



7.5.2 課程大綱

Core Knowledge	Programming skills/LAB	時數
<p>單元一、程式語言之目的，分類與演進</p> <p>2. 導論</p> <ul style="list-style-type: none"> ■ 程式語言的目的 ■ 程式語言的類別、性質與主要應用 <ul style="list-style-type: none"> ◆ 程序導向程式語言 ◆ 物件導向程式語言 ◆ 資料庫程式語言 ◆ 符號處理程式語言 ◆ 文件處理程式語言 ◆ Script 語言 ■ 語言的執行 <ul style="list-style-type: none"> ◆ compiler ◆ assembler <p>3. 影響程式語言演進之主要因素</p> <ul style="list-style-type: none"> ■ 應用 ■ 程式結構 ■ 計算機架構 ■ 程式撰寫的方便性 ■ 軟體設計模式 <p>4. 程式語言的發展史</p>		3
<p>單元二、程序導向語言(Procedure-Oriented Language)</p> <p>1. 背景—機器語言與組合語言(1956)</p> <ul style="list-style-type: none"> ■ 機器語言(1 hr) ■ Von Neumann Machine ■ 動機 <ul style="list-style-type: none"> ◆ ■ 語言特性 <ul style="list-style-type: none"> ◆ 	<ul style="list-style-type: none"> ● 用 JAVA 語言來模擬 FORTRAN 的 Static storage ● 用 JAVA 語言來模擬 COBOL 的報表輸出 ● 用 JAVA 語言來模擬 PASCAL 的 link list ● 用 JAVA 語言來模擬 C 的 low level control ● 用 JAVA 語言來模擬 Ada 的 	16

<ul style="list-style-type: none"> ■ 範例 ■ weakness <ul style="list-style-type: none"> ◆ <p>2. 組合語言(1956)</p> <ul style="list-style-type: none"> ■ 動機 <ul style="list-style-type: none"> ◆ 提高程式可讀性、Relocatable Program Concept ■ 語言特性 <ul style="list-style-type: none"> ◆ Operation Code、Register 名稱用符號代替 ◆ 變數名稱取代位址 ◆ 巨集指令(Macro) ■ 範例 ■ weakness <ul style="list-style-type: none"> ◆ Relocatable Loader 之導入 — 程式起始位置之變更 <p>3. FORTRAN(1957)</p> <ul style="list-style-type: none"> ■ 動機 <ul style="list-style-type: none"> ◆ 近似人常用的自然語言— 第一個高階語言 ◆ Formula Translation ■ 特性 <ul style="list-style-type: none"> ◆ Declaration statements, execution statements ◆ Simple variables, array variables ◆ Assignment statements, control statements, I/O statements, subroutine ◆ Statements, FORMAT statements ◆ Static storage ◆ Automatic type conversion ◆ Library ◆ Comment ■ 範例 ■ Weakness <ul style="list-style-type: none"> ◆ Static storage ◆ No recursive feature ◆ FORMAT statements 	<p>concurrent</p>	
--	-------------------	--

- ◆ Difficult to handle input/output

4. COBOL (1959)

■ 動機

- ◆ Business data processing
- ◆ Report, numeric
- ◆ Fortran 無法處理 string, file

■ 特性

- ◆ 程式看起來很像是英文的口語
- ◆ manipulate input/output file
- ◆ divides programs into: identification, environment, data, procedure divisions.
- ◆ 提供 records 的資料結構

■ 範例

■ Weakness

- ◆ 無結構化

5. BASIC(1964)

■ 動機

- ◆ Students easy to learning
- ◆ Fortran, COBOL 變數一定要宣告
- ◆ Fortran, COBOL 語言 compiler 如有錯很難除錯

■ 特性

- ◆ Var. cannot be declared
- ◆ Interpreter
- ◆ Friendly error messages
- ◆ GoSub...Return
- ◆ Respond fast for small programs

■ 範例

■ Weakness

- ◆ 無副程式參數傳遞的觀念
- ◆ Interpreter (slow: 每次執行須重新編譯)

6. PASCAL (1970)

■ 動機

- ◆ Fortran, COBOL, BASIC 語言無結構化概念
- ◆ 之前的語言無 dynamic (pointer), record 的資料結構

■ 特性

- ◆ Variables must be defined
- ◆ 提供 dynamic (pointer)的資料結構
- ◆ well-structured
- ◆ 提供 Sets, files, records 的資料結構

■ 範例

■ Weakness

- ◆ One pass compiler--procedures 要依順序寫 (被呼叫要寫在前)

7. C (1971)

■ 動機

- ◆ 之前的語言採 one pass compiler
- ◆ A general-purpose programming language
- ◆ For UNIX
- ◆ 之前的語言少提供 library routines

■ 特性

- ◆ access to low level hardware
- ◆ library routines
- ◆ UNIX system
- ◆ weak type check

■ 範例

■ Weakness

- ◆ flexible but dangerous

8. Ada (1970)

■ 動機

- ◆ 美國防部所制定
- ◆ 之前的語言無 concurrent 的能力
- ◆ 之前的語言無結構化概念
- ◆ 支援 embedded and real-time systems

■ 特性

- ◆ concurrent (parallel)

<ul style="list-style-type: none"> ◆ semaphores ◆ bounded buffer ◆ structured ◆ run-time checking ■ 範例 ■ Weakness <p>9. VB (1991)</p> <ul style="list-style-type: none"> ■ 動機 <ul style="list-style-type: none"> ◆ 之前的語言處理視窗介面難 ◆ 之前的語言與資料庫之間難連結 ■ 特性 <ul style="list-style-type: none"> ◆ rapid development of GUI for Windows ◆ complex database object library ◆ network-savvy ◆ multithreaded ■ 範例 ■ Weakness <ul style="list-style-type: none"> ◆ 程式碼大 		
<p>單元三、— 物件導向語言</p> <p>1. Small talk (1969)</p> <ul style="list-style-type: none"> ■ 動機 <ul style="list-style-type: none"> ◆ object-oriented ◆ 跨平台 ■ 特性 <ul style="list-style-type: none"> ◆ first OOP ◆ Everything is an object ◆ Everything is available for modification ◆ Types are dynamic ◆ Bytecodes ◆ Multithreaded ◆ Concurrency ◆ 繼承 (不支援多重繼承) ■ 範例 	<ul style="list-style-type: none"> ● Java Applet 	<p>5</p>

■ Weakness

- ◆ garbage collection 不完整
- ◆ Bytecodes--執行速度慢

2. C++ (1983)

■ 動機

- ◆ C + O.O.

■ 特性

- ◆ Compatible with C
- ◆ 多重繼承的觀念
- ◆ Strict compile-time check
- ◆ Template
- ◆ Exception handling

■ 範例

■ Weakness

- ◆ Single-threaded
- ◆ 無 garbage collection
- ◆ 多重繼承的觀念-- 一個物件的兩個父物件
可能有衝突

3. JAVA (1995)

■ 動機

- ◆ independent of the host platform
- ◆ for networking
- ◆ securely

■ 特性

- ◆ portable
- ◆ bytecodes
- ◆ 完整的 garbage collection
- ◆ network-savvy
- ◆ multithreaded
- ◆ Class library

■ 範例

■ Weakness

- ◆ Bytecodes--執行速度慢

- SQL(1970), SQL1(1989)
 - 動機
 - ◆ 針對關聯式資料庫發展的語言
 - 特性
 - ◆ Data definition in SQL-- table definition, schema definition and update, user defined domain, relational constraints, relational catalogues
 - ◆ SQL queries -- selection, insertion, deletion, update
 - 範例
 - Weakness
 - ◆ 缺少 Changing security settings
 - ◆ syntax rules, create queries 較不方便

- SQL2(1992)
 - 動機
 - ◆ Enhance SOL1
 - 特性
 - ◆ new reserved words, wildcard
 - ◆ GRANT, REVOKE
 - ◆ DISTINCT
 - ◆ LIMIT TO
 - 範例
 - Weakness
 - ◆ 無法處理 Multimedia

- SQL3(1999)
 - 動機
 - ◆ applications 的需求改變--multimedia
 - 特性
 - ◆ for objects-oriented
 - ◆ for data-mining
 - ◆ for spatial-data

<ul style="list-style-type: none"> ◆ for temporal-data ◆ for on line analytical ◆ for data-warehousing ◆ for multimedia-data ◆ 將程式區分成兩部 – 1. core specification: for RDBMS, 2. option specialized packages: for applications ■ 範例 ■ Weakness 		
<p>單元五、—符號處理程式語言</p> <ul style="list-style-type: none"> ■ LISP(1958) <ul style="list-style-type: none"> ■ 動機 <ul style="list-style-type: none"> ◆ 人工智慧計算的需求 ■ 特性 <ul style="list-style-type: none"> ◆ rule-based ◆ lambda-calculus & functional computation ◆ list as data and program ◆ dynamic data typing (late binding) ◆ symbolic computation ◆ focused on problem solving strategies ◆ suitable for AI applications (lower level) ◆ fast prototyping ◆ stack-based interpretation ■ 範例 ■ Weakness <ul style="list-style-type: none"> ◆ some data is not suitable to be described as lists or functions ◆ procedure-like computation ◆ still too –lower for AI applications ◆ slow interpretation (compared with C) ◆ large memory consumption ■ Prolog(1970) <ul style="list-style-type: none"> ■ 動機 	●	4

<ul style="list-style-type: none"> ◆ 單一一種程式計算方式 (logic interpretation)：以 logic clauses 處理所有運算並表示所有資料型態 ■ 特性 <ul style="list-style-type: none"> ◆ data=program; logic clauses as data and program ◆ dynamic data typing (late binding) ◆ symbolic computation ◆ focused on problem solving strategies ◆ suitable for AI applications (higher level) ◆ fast prototyping ◆ Interpretation ■ 範例 ■ Weakness <ul style="list-style-type: none"> ◆ slow interpretation (compared with C) ◆ huge memory consumption 		
<p>單元六、—文件處裡程式語言</p> <ul style="list-style-type: none"> ■ Tex(1970) <ul style="list-style-type: none"> ■ 動機 <ul style="list-style-type: none"> ◆ 製作排版文件，處理數學公式與各種符號。 ■ 特性 <ul style="list-style-type: none"> ◆ 制訂排版指令 ◆ 以寫程式方式進行排版與文件製作(文件=程式) ◆ 編譯式語言 ◆ 分離文件製作與輸出 (tex, dvi, ps) tex→dvi→ps→... ◆ 可跨平台 ■ 範例 ■ Weakness <ul style="list-style-type: none"> ◆ 不容易記憶各種排版指令 ◆ 需要進行編譯(compilation) ◆ 需安裝字型與相關函式庫 		7

■ Postscript(1976)

■ 動機

- ◆ 於各種終端機顯示文件，由各種印表機輸出文件。增加文件的可攜性

■ 特性

- ◆ 制訂螢幕輸出與印表機輸出指令
- ◆ 分離文件製作與輸出
- ◆ 提供各種跨硬體的文件可攜性
- ◆ interpretation
- ◆ 提供多種轉換成 PS 格式的工具

■ 範例

■ Weakness

- ◆ 文件檔案龐大
- ◆ 需安裝字型與相關函式庫

■ Latex(1985)

■ 動機

- ◆ tex 指令太多，不容易使用。LaTeX 於 tex 中加入巨集，並提供各種可重複套用的版形。

■ 特性

- ◆ 同 tex，但加入巨集，並提供各種可重複套用的版形
- ◆ 使用者需記憶的指令較少

■ 範例

■ Weakness

- ◆ (仍然) 不容易記憶各種排版指令
- ◆ 需要進行編譯(compilation)
- ◆ 需安裝字型與相關函式庫

■ PDF

■ 動機

- ◆ PS 型態文件檔案太大，不利於流通於網路上。PDF 提升網路流通的可攜性文件。

■ 特性

- ◆ 同 PS

<ul style="list-style-type: none"> ◆ 文件較精簡，SIZE 較小 ◆ 可嵌入 browser ◆ 提供多種轉換成 PDF 格式的工具 ■ 範例 ■ Weakness <ul style="list-style-type: none"> ◆ 需安裝字型與 drivers 		
<p>單元七、—Script 語言</p> <p>1. Perl(1987)</p> <ul style="list-style-type: none"> ■ 動機 <ul style="list-style-type: none"> ◆ 對問題中的文字字串進行快速、複雜的文字處理。Perl 是目前 Script Language 中最熱門的語言，由於一開始結合了 C，awk，sed，sh，and BASIC，等語言的特性，因此在 Script Language 引起十分廣泛的回響，特別是在 Unix-like 的作業系統上，傳統的網管與系統管理的工作，八成幾乎主要使用 Perl 語言來完成，另外 Perl 社群亦建立 Comprehensive Perl Archive Network (CPAN).來統一管理各個模組，因此 Perl 成為支援性最好的 Script Language. 然而 Perl 亦繼承了 sed/awk 的負擔，整個語言的設計十分不易閱讀，程式設計師，往往一個月後，無法看懂之前所寫的程式，換言之，程式的可擴充性相對較不易。 ■ 特性 <ul style="list-style-type: none"> ◆ regular expression ◆ Perl takes features from other languages，such as C，awk，sed，sh，and BASIC，among others. ◆ Perl's database integration interface (DBI) supports third-party databases including Oracle，Sybase，Postgres，MySQL and others. ◆ Perl works with HTML，XML，and other mark-up languages. 	●	8

- ◆ Perl supports both procedural and object-oriented programming.
- ◆ Perl interfaces with external C/C++ libraries through XS or SWIG.

■ 範例

■ Weakness

- ◆ 執行效率較差
- ◆ Readable is much poor!

2. Tcl/tk(1988)

■ 動機

- ◆ 對電腦各種（周邊）設備進行處理，給予程式設計師以單一程式語言處理所有電腦相關設備。

■ 特性

- ◆ EDA/CAD: Tcl has become the defacto standard in EDA and CAD applications.
- ◆ Test Automation: Tcl is the de facto standard for automated testing.
- ◆ Dynamic Web Content: Some of the industry's most popular high-traffic web sites are Tcl-powered.
- ◆ Network and System Management: Tcl provides a platform for network and system management applications with its rapid GUI development, easy extensibility and cross-platform support.

■ 範例

■ Weakness

- ◆ The main advantage of Tcl is that *all* its variables are strings
- ◆ The main advantage of Tcl is that it has no syntax.

3. Python(1991)

■ 動機

- ◆ Python 係針對 Perl 的程式碼不易閱讀與不易擴充問題，重新精簡了程式的元素，例如利用縮排來取代程式區塊，去除資料型的概念，並將物件導向觀念引進，因此相對於 Perl 來說，程式碼顯得易懂且易於擴充。以下就 Python 在幾個特色表現優異之處，一般來說 Python 的 Advanced data structures (> Tcl, Perl)、Python 的 Powerful data-parallel arrays (> Tcl)、Readability and modularity (> Perl)、High-level (> C, C++, Fortran) Platform independence (> C++, Java)。

■ 特性

- ◆ Interpreted, High level, Object-Oriented
- ◆ Flexible and Extensible
- ◆ Introspection, self-documenting
- ◆ Platform Independent
- ◆ Open Source
- ◆ Rapidly gaining acceptance

■ 範例

■ Weakness

- ◆ The main disadvantage of python (apart from not enough people knowing about it) is that it is interpreted from bytecode and therefore executes slower than C++.

4. Ruby(1993)

■ 動機

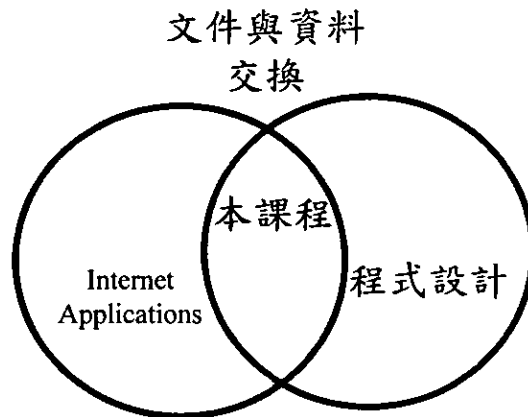
- ◆ Ruby(紅寶石)比起其他的 script language 來說，最大的特色來自於這個語言設計時特別強調完整的物件導向概念，除與 Perl 一樣，可用來做系統管理工具，另外也有正規表示式與強大的檔案和字串處理能力，但是確沒有像 Perl 一般不易閱讀的語法，比起 python 來說更 OO，更有彈性，因為 Ruby 全都是 OO。

<ul style="list-style-type: none"> ■ 特性 <ul style="list-style-type: none"> ◆ simpler syntax ◆ redefinable methods. ◆ Complete, full, pure object oriented language: OOL. ◆ OO in Ruby is carefully designed to be both complete and open for improvements. ◆ single inheritance only , *on purpose*. ◆ true closures. Not just unnamed function , but with present variable bindings. ◆ Ruby features a true mark-and-sweep garbage collector. ◆ needs no variable declarations. ◆ OS independent threading. ◆ highly portable: ■ 範例 ■ Weakness <p>5. PHP(1995)</p> <ul style="list-style-type: none"> ■ 動機 <ul style="list-style-type: none"> ◆ C-like syntax, suitable for web . ■ 特性 <ul style="list-style-type: none"> ◆ interpretation ◆ 跨平台 ◆ fast prototyping ◆ connect to DB ◆ widely adpoted ■ 範例 ■ Weakness <ul style="list-style-type: none"> ◆ 需安裝相關函式庫 ◆ 執行效率較差 		
	合計	48

7.6 XML 語言

7.6.1 課程設計理念

前言：隨著網際網路的普及，人們已習慣將資訊利用標記語言來呈現，但 HTML 標記本身有使用上的限制，因此全球資訊網路協會（W3C）統一網路科技的標準，並制訂 XML（Extensible Markup Language）延伸標記語言。XML 文件之「結構化」與「資訊內容導向化」二特色，特別是用在電子資料的傳遞、交換與分享，有助於原始資料的回溯、處理，避免不必要的重複，因而有效地提升其效率。另外由於 XML 具有「可延伸」和「跨平台」的特性，可促進各個不同的應用領域發展出各自的文件標準，以利資訊交換。本課程將介紹 XML 的應用。

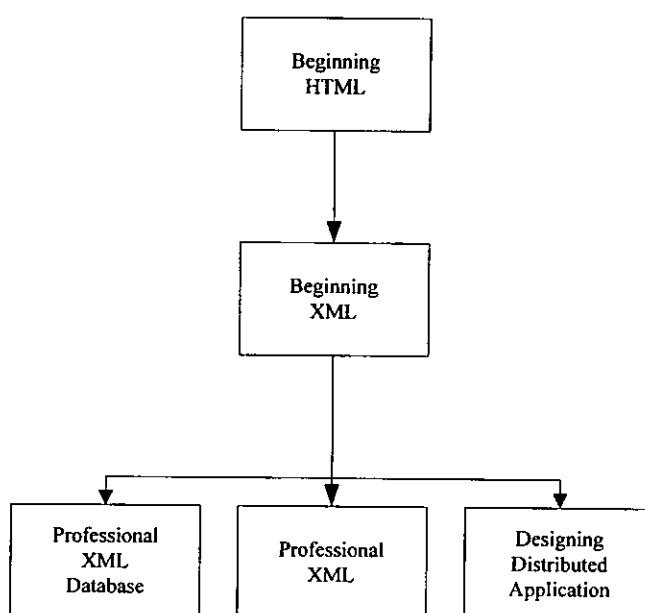


教育目標：本課程的重點在於 XML 的應用發展，包括在客戶端處理 XML、在伺服器端處理 XML，以及使用 XML 來處理資料等。

課程內容：

- (1) 網際網路 (internet) 及全球資訊網 (WWW) 簡介
- (2) 超文件標記語言 (HTML) 介紹
- (3) 串接樣式表 (Cascading Style Sheet -- CSS) 介紹
- (4) 以 XML 建立標記
- (5) 驗證 XML 文件
- (6) 可擴充樣式表語言轉換 (XSLT)
- (7) 客製標記語言 (Custom Mark Language)
- (8) 文件物件模組 (Document Object Model, DOM)
- (9) XML 的簡易應用程式介面 (SAX)
- (10) XML 與資料庫
- (11) Web-Based 系統
- (12) 簡易物件存取協定 (SOAP)
- (13) B2B 與微軟 BizTalk 伺服器

設計構想與進行方式：超文件標記語言 (Hyper-Text Markup Language) 是處理網際網路資料之語言，但其有本身上的限制。這讓 W3C 想要設計一個有「可延伸」和「跨平台」特性的標記語言——XML。因為 XML 優異的特性，使得 XML 的應用不斷的增加。為了讓學生盡快瞭解 XML 的語法及其應用，我們將本課程分成三個部分。第一部份介紹什麼是標記語言。在這裡我們將從 HTML 開始，使學生先瞭解標記語言。第二部分介紹 XML 語言及如何驗證 XML 文件，如何將 XML 文件顯示在網頁上。第三部分介紹 XML 應用的範圍，如何將 XML 檔案當作資料庫使用到在分散式系統上 XML 的應用。本課程共五十二個小時，適合在一學期開課。如果因為課程時數的限制，且在其他課程已經講述 HTML 的概念，則可從 Topic3 開始到 Topic6 約十二個小時。第三部分是選擇性的，各老師可選擇重要部分教授及延長教授的時間。整個課程的架構如下圖所示：



本課程分成 12 個主題，共 52 小時：

Topic 1: 導論 (2hr)

首先介紹網際網路的發展及全球資訊網路的組織。之後介紹 Markup Language 的由來及其興起的原因，他與傳統的 program language 之間有何不同。此外，為了讓學生有所瞭解，我們也介紹標準通用標記語言 (SGML)。

Topic 2: 標記語言初步的介紹 (4hr)

這部分先介紹 HTML 語言，讓學生對於什麼是標記語言有初步的認識。因為 HTML 是網頁最常用的語言，因此我們以此為入門的語言，並介紹 HTML 與 Browser 的關係及 Browser 的運作模式。為了讓學生對標記語言使用有所瞭解，我們將介紹 HTML 中一些常用的標記。

Topic 3: 樣式表介紹 (2hr)

標記語言是用來儲存資料用的，對於資料的呈現則需要樣式表。這部分將介紹 HTML

的樣式表及其語法，讓學生瞭解如何修改 HTML 資料的呈現。

Topic 4:以 XML 建立標記 (2hr)

當學生對標記語言初步的認識後，我們在這部分將介紹 XML 標記語言，他與 HTML 之間的差異，及它的語法。

Topic 5:驗證 XML 文件 (4hr)

這部分我們將介紹驗證 XML 的目的，well-formed 及 validated XML 之間的差異，及用來驗證 XML 文件的語法 (DTD 及 Schema)。

Topic 6:可擴充樣式表語言轉換 (XSLT) (3hr)

介紹如何將 XML 轉換成其他格式的語言及轉換的目的。

Topic 7:客製標記語言 (Custom Mark Language) (3hr)

介紹目前常用的一些標記語言如：WML、BML、ebXML。

Topic 8:文件物件模組 (Document Object Model, DOM) (3hr)

介紹什麼是 DOM，為何需要 DOM，他與 XML 文件結構的關係，及如何利用程式及 DOM 來撰寫應用程式。

Topic 9:XML 的簡易應用程式介面 (SAX) (3hr)

介紹 XML 文件在程式應用的另一種方式的使用。說明為什麼要 SAX 的方式及 SAX 應用程式範例。

Topic 10:XML 與資料庫 (9hr)

介紹 XML 與資料庫的相似性，如果將 XML 文件當成是一個簡易的資料庫，我們又如何使用，及目前這方面的應用。

Topic 11:Web-Based 系統 (6hr)

介紹目前正流行的 Web-Based 系統，內容有網路的協定、Web 上資料庫的驅動、遠端程序呼叫、分散式物件及其系統，最後介紹企業間 Web-Based 應用程式。

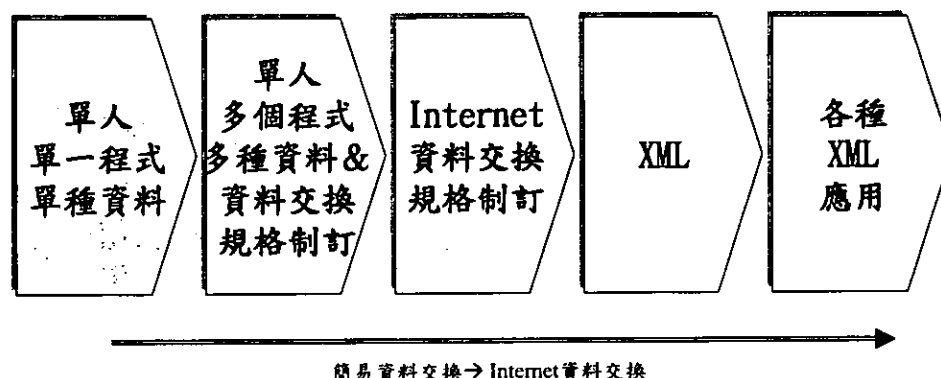
Topic 12:簡易物件存取協定 (SOAP) (3hr)

介紹標記語言在分散式系統上的應用，及簡易物件存取協定，並用一個實例說明之。

Topic 13:B2B 與微軟 BizTalk 伺服器 (5hr)

這部分介紹標記語言在 B2B 商務上的使用。先介紹 B2B 商務的需求，在介紹目前 Microsoft 的 BizTalk 伺服器及其範例。

各單元進行的順序的關係如下圖所示：



7.6.2 課程大綱

Core Knowledge	Programming Skills	時數 (小時)
<p>2. 網際網路 (internet) 及全球資訊網 (WWW) 簡介</p> <ul style="list-style-type: none"> ● 網際網路的歷史 ● 全球資訊網的歷史及全球資訊網協會 (WWW Consortium) ● Markup Language 的由來 <ul style="list-style-type: none"> ■ 由 program language 到 markup language 的變化 ● 標準通用標記語言 (SGML) 的歷史 ● Markup Language 興起的理由及未來的趨勢 		2
<p>12. 超文件標記語言 (HTML) 介紹</p> <ul style="list-style-type: none"> ● Browser 與 HTML 的關係 ● Browser 如何呈現 HTML 文件 <ul style="list-style-type: none"> ■ HTML parser ● 標記語言 ● 編輯 HTML <ul style="list-style-type: none"> ■ 元件 ■ 檔頭 ■ 連結 ■ 圖像 ■ 特殊的字元 ■ 列單 <ul style="list-style-type: none"> ◆ 排序 ◆ 未排序 ■ 表格 ■ 表單 ■ 框架 (frameset) 設定 ■ <meta>標籤 	<ul style="list-style-type: none"> ● 設計一個入口網站的首頁 本實作需含有： <ul style="list-style-type: none"> ■ 連結 ■ 圖像 ■ 列單 <ul style="list-style-type: none"> ◆ 排序 ◆ 未排序 ■ 表格 ■ 表單 ■ 框架 (frameset) 設定 	4
<p>13. 串接樣式表 (Cascading Style Sheet -- CSS)</p> <ul style="list-style-type: none"> ● CSS 用途介紹 ● 內嵌樣式 (InLine Styles) ● 外部樣式表連結 ● 元件內建立樣式表 ● 樣式 <ul style="list-style-type: none"> ◆ Positioning Element ◆ Background 	<ul style="list-style-type: none"> ● 將上一個作業的內容做樣式上的修改 	2

<ul style="list-style-type: none"> ◆ Dimensions ◆ Text Flow and Box Model ● 使用者樣式表 		
<p>14. 以 XML 建立標記</p> <ul style="list-style-type: none"> ● HTML 的不足 <ul style="list-style-type: none"> ■ 標記已經事先定義好而無法自行定義新的標記 ● 介紹 XML 標記 ● parser 及 Well-formed 的 XML 文件 ● XML 所用的字元 ● 標記 ● CDATA 區塊 ● XML 命名空間 	<ul style="list-style-type: none"> ● 將一份一般會議記錄轉換成 XML 的檔案。 	2
<p>15. 驗證 XML 文件</p> <ul style="list-style-type: none"> ● 驗證 XML 文件的目標 <ul style="list-style-type: none"> ◆ Strong type check 的介紹 ● well-formed 及 validated XML 文件的差別 ● 文件類型宣告 (DTD) <ul style="list-style-type: none"> ◆ element type declaration ◆ attribute type declaration ◆ 條件區 ● Schema <ul style="list-style-type: none"> ◆ Schema 與 DTD 的差異 ◆ Microsoft XML 與 W3C XML ◆ 描述元素 ◆ 描述屬性 ◆ 資料類型 	<ul style="list-style-type: none"> ● 設計上一個作業的 DTD 檔，作為資料交換的依據。 	4
<p>16. 可擴充樣式表語言轉換 (XSLT)</p> <ul style="list-style-type: none"> ● XSLT 的目的 ● 什麼是轉換 ● 讀取資料：XPath ● 轉換資料：XSLT <ul style="list-style-type: none"> ■ 樣版 ■ 建立元素和屬性 ■ 重複與排序 ■ 條件處理 ■ 節點複製 ■ 結合樣式表 ■ 變數 ● XML 轉換到 XML 範例 	<ul style="list-style-type: none"> ● 將之前的會議記錄之 XML 檔轉換成利用 HTML 檔 	3
<p>17. 客製標記語言 (Custom Markup Language)</p>		3

<ul style="list-style-type: none"> ● 無線標記語言 (WML) ● Java 元件標記語言 (BML) ● 電子商務標記語言 (ebXML) 		
<p>18. 文件物件模組 (Document Object Model, DOM)</p> <ul style="list-style-type: none"> ● DOM 應用的需求 ● DOM 的介紹 ● XML 文件的結構 ● DOM 的使用 ● DOM 與 XML 並用的應用程式範例 	<ul style="list-style-type: none"> ● 設計一程式顯示之前會議記錄之 XML 檔案內容。 	3
<p>19. XML 的簡易應用程式介面 (SAX)</p> <ul style="list-style-type: none"> ● SAX 的應用需求 ● 載入 Reader <ul style="list-style-type: none"> ■ Content Handlers ■ Error Handlers ■ DTD Handlers ● 驗證 ● 範例 	<ul style="list-style-type: none"> ● 利用 SAX 將之前會議記錄之 XML 檔內容以樹狀方式呈現。 	3
<p>20. XML 與資料庫</p> <ul style="list-style-type: none"> ● 資料模型化 <ul style="list-style-type: none"> ■ 設計 XML 文件 ■ 撰寫 Schema ● XML 資料連結 <ul style="list-style-type: none"> ■ 使用 Castor 內建的 introspection ■ 使用 Castor 對映檔 ■ 使用 Castor 的來源產生器 ● 查詢 XML <ul style="list-style-type: none"> ■ W3C XML Query 語言 ■ XQuery 與 XSLT 比較 ● 個案研究 	<ul style="list-style-type: none"> ● 設計一個簡易 XML 資料庫檔，之後用 W3C XML Query 語言來實作 	10
<p>21. Web-Based 系統</p> <ul style="list-style-type: none"> ● 應用程式之協定 ● Web 工程之各項原則 ● Web 端之資料庫驅動 ● 遠端程序呼叫 ● 輕量級分散式物件 ● 分散式物件系統 ● 企業間 Web-Based 應用程式 	<ul style="list-style-type: none"> ● 設計一 Web-based 應用程式 	6
<p>22. 簡易物件存取協定 (SOAP)</p>	<ul style="list-style-type: none"> ● 設計一 SOAP 應用程式 	5

<ul style="list-style-type: none"> ● 介紹 SOAP ● SOAP 結合到傳送協定 ● 透過 HTTP 的同步 SOAP 實例 		
23. B2B 與微軟 BizTalk 伺服器 <ul style="list-style-type: none"> ● B2B 商務的需求 ● 解決方案 ● 微軟 BizTalk 伺服器 ● 一個 B2B BizTalk 範例 		5
Total:		52

捌、與勞委會乙級程式設計師技能檢定考試科目之比較

本計畫為培訓資訊軟體人才、提升資訊軟體人才能力，規劃資訊相關科系課程為研究目標，規劃技專院校資訊相關系科教學課程之課程架構、核心課程、及課程大綱，並結合資訊能力鑑定與專業證照考試，建構畢業學生專業能力指標，以評鑑學生資訊專業教育的水準。所以特將本研究小組規劃的核心課程內容，與職訓局電腦軟體設計技能檢定規劃的核心課程，做以下之比較，如下表所示本研究小組規劃的核心課程內容，涵蓋絕大部分職訓局規劃的內容，由此顯見，將來所培養的技專院校學生，能具備證照鑑定所需的能力。

I. 電腦系統

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
1.作業系統	a. 電腦硬體基本輸入、處理、輸出及儲存	是	計概：第 1,13 節
	b. 作業系統	是	計概：第 13 節
	c. Windows、Linux	是	計概：第 13 節
	d. 多工、多緒、多處理器概念	是	計概：第 23 節
	e. 各種伺服器管理及其他重要功能介紹	是	計概：advance
	f. 系統環境建立與設定	是	計概：advance
	g. 主從式架構	是	計概：advance
	h. 遠端呼叫	是	計概：第 14 節
	i. 重要系統架構演進	是	計概：advance
2.系統程式	a.系統軟體	是	計概：第 0 節
	b. 中介軟體(Middleware)	是	計概：第 0 節
	c. 組譯器	是	程式語言：第一單元
	d. 編譯器	是	程式語言：第一單元
3.資料模式	資料模型之設計	是	計概：第 17 節
4.資料庫管理與設計	a.各類檔案結構與應用	是	資料庫：第 1 節
	b.資料庫系統概觀	是	資料庫：第 1 節
	c.表格(Table)及視界(view)的建立與管理	是	資料庫：第 5,8 節
	d.資料庫物件	是	資料庫：第 11,12 節
	e.資料字典/系統目錄	是	資料庫：第 5 節

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
	f. 權限控制	是	資料庫：第 10 節
	g. 預儲程式及觸發程式	是	資料庫：第 6 節
	h. 交易處理	是	資料庫：第 9 節
5. ER Model	正規化及實體-關係模式	是	資料庫：第 5 節
6. SQL	a. 基本 SQL 命令	是	資料庫：第 3,4 節
	b. SQL 命令運用	是	資料庫：第 3,4 節
	c. 程序化 SQL	是	資料庫：第 6 節
7. 資料通訊	a. 網路通信及網際網路背景及歷史	是	計概：第 14 節
	b. 網路架構	是	計概：第 14 節
	c. 網路及通信協定、網路多媒體系統、分散式計算、移動式與無線式計算	是	計概：第 14 節
	d. 網路通訊 (資料傳送之方式與設備、傳輸媒體與網路拓撲、通訊協定與標準)	是	計概：第 14 節
	e. Client-Server Computing Applications	是	計概：第 14 節
	f. Web-Based Systems (應用程式程之協定、Web 工程之各項原則、資料庫驅動之 Web 端、遠端程序呼叫、輕量級分散式物件、分散式物件系統、企業間 Web-Based 應用程式)	是	XML: 第 11 節
	g. XML (標記與核心概念、連接資源、版面配置、文件模型、轉換)	是	XML：第 4,5,6,8 節
8. 電腦區域網路	a. 區域網路設定	是	計概：第 14 節
	b. 區域網路作業系統環境之建立與設定	是	計概：第 14 節

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
9. 電腦網路管理	a.作業系統環境之建立與設定	是	計概：第 14 節 (homework)
	b.區域網路與網際網路設定	是	計概：第 14 節 (homework)
	c.伺服器管理	是	計概：第 14 節 (homework)

II. 程式設計

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
1. 資料結構及演算法	a. 鏈結串列	是	資料結構：第 2 節
	b. 堆疊及佇列	是	資料結構：第 3,4 節
	c. Tree 及其運用	是	資料結構：第 6 節
	d. 圖形	是	資料結構：第 8 節
	e. 演算法效能評估	是	資料結構：第 1 節
	f. 排序	是	資料結構：第 7 節
	g. 搜尋	是	資料結構：第 7 節
	h. 雜湊	是	資料結構：第 7 節
	i. 遞迴	是	資料結構：第 5 節
	j. 解題方法	是	資料結構：第 9 節
2. 結構化分析與設計	軟體生命週期及開發模型	是	軟體發展技術：第 1,4,5,6 節
3. 結構化程式設計	軟體生命週期及開發模型	是	軟體發展技術：第 1,4,5,6 節
4. C/C++ 程式設計	a. 語言基礎	是	計概：第 5,6,7 節
	b. 流程控制	是	計概：第 0,7,9,16 節
	c. 函數	是	計概：第 10 節
	d. 指標與陣列	是	程式語言：JAVA
	e. 類別與物件	是	計概：第 17,18,19 節
	f. 運算子超載	是	程式語言：JAVA
	g. 輸出入資料流及檔案	是	計概：第 12,24 節
	h. 例外處理	是	計概：第 26 節
	i. 樣板	是	程式語言：C++

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
	j.名稱空間	是	程式語言：C++
	k.標準樣版程式庫	是	程式語言：C++
5.JAVA 語言 程式設計	a. Java 執行環境	是	計概：第 5 節
	b.基本資料型態，運算子與指定運算	是	計概：第 5,6,7 節
	c.宣告、存取控制及型態轉換	是	計概：第 6,7 節
	d.流程控制	是	計概：第 0,7,9,16 節
	e.陣列處理	是	計概：第 8 節
	f.類別與物件	是	計概：第 16, 17, 18, 19, 20, 21 節
	g.例外處理	是	計概：第 26 節
	h.多工執行緒	是	計概：第 17,23 節
	i.輸入/輸出處理	是	計概：第 12,24 節
	j.套件介紹	是	計概：第 24 節
	k.圖形介面(GUI)程式	是	計概：第 28 節
l. Java Applet	是	程式語言：Java (homework)	
6.程式語言	a. 程式語言目的，種類及演進	是	程式語言
	b. 程式語言結構		
	(1) 命名、範圍及繫結	是	程式語言

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
	Abstraction (5) 建構可執行之程式 (6) 資料抽象化及物件導向 (7) 並行	是	程式語言
7.物件導向程式設計及相關知識	a.物件導向觀念 (1) 軟體元件、類別與物件 (2) 屬性、操作及方法 (3) 訊息 (4) 封裝、繼承及多形 b. 人機界面設計	是	計概：第 17,18,20,21 節

III. 系統發展

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
1.系統發展文件	a. 軟體文件		
	(1) 虛擬碼	是	軟體發展技術：第 5 節
	(2) 撰寫規則	是	軟體發展技術：第 5 節
	(3) 軟體文件種類、軟體文件格式	是	軟體發展技術：第 5 節
2.系統測試	a. 軟體測試策略與技術		
	(1) 軟體測試策略：單元測試、整合測試、驗證測試、系統測試	是	軟體發展技術：第 3 節
	(2) 軟體測試技術：程式碼審查，白箱測試、黑箱測試、安全性測試	是	軟體發展技術：第 3 節
	(3) 測試文件：測試計畫文件、測試案例	是	軟體發展技術：第 3 節
	(4) 除錯技術：錯誤管理、除錯方法、除錯工具	是	軟體發展技術：第 3 節
3.軟體工程	a. 軟體生命週期及開發模型	是	軟體發展技術：第 1 節
	b. 軟體品質基本觀念		
	(1) 軟體相關程序	是	軟體發展技術：第 8 節
	(2) CMMI 簡介		
	(3) PSP 簡介		
(4) TSP 簡介			
4.專案管理	軟體生命週期及開發模型	是	軟體發展技術：第 1 節
5 物件導向技術	a. UML		
	(1) UML 架構	是	軟體發展技術：第 5 節
	(2) 類別及物件	是	軟體發展技術：第 5 節

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
	(3) Activity Diagram , Process 、 Thread & Active Class	是	軟體發展技術：第 5 節
	(4) Use Case	是	軟體發展技術：第 6 節
	(5) Data Modeling	是	軟體發展技術：第 4 節
	(6) Event , State	是	軟體發展技術：第 5 節
	(7) Abstract Class	是	軟體發展技術：第 5 節
	(8) Component & Component Diagram , Mode & Deployment Diagram	是	軟體發展技術：第 5 節
	(9) Template Classes , Collaboration , Patterns 、 Mechanisms & Frameworks	是	軟體發展技術：第 5 節
	(b) UML 應用	是	軟體發展技術：第 7 節

IV. 資訊安全

職訓局規劃的相關知識	職訓局規劃的核心課程	本研究規劃的核心課程是否涵蓋	本研究規劃的章節
1. 資訊安全與管理	a. 資訊安全威脅的種類	是	計概：第 15 節
	b. 資訊安全的標準與規範	是	計概：第 15 節
	c. 認證與權限控制概念	是	計概：第 15 節
	d. 資料加密的應用	是	計概：第 15 節
	e. 網路安全	是	計概：第 15 節
2. 電腦病毒防治	a. 電腦病毒原理	是	計概：第 15 節
	b. 電腦系統不當使用之災害	是	計概：第 15 節

玖、結論與建議

9.1 結論

大學工程教育的目的是培植社會所需之工程設計與研究高級人材，大學需密切注意產業的技術發展趨勢與環境變化而調整授課內容及實習與實驗設備，使畢業生進入社會可迅速融入企業而發展所學。資訊電子產業是我國最大產業，我國資訊硬體產業在世界市場已佔有重要地位，但軟體產業的發展卻不如預期，其最主要因素是人才；大學資訊相關科系是培養軟體工程師的主要搖籃，目前我國設有資訊相關科系的大學及技職院校之數量眾多，每年畢業的學生人數也不少，但為何軟體產業仍不發達呢？從這次研究中可發現，我們未依畢業生就業所需的核心知識的重要性及優先序而給予紮實的訓練，及未對學生的素質與背景因材施教。

資訊相關科系畢業生剛進入社會而擔任軟體相關工作，大部份的工作是設計與維護程式，即擔任程式設計師的角色；因此本研究從軟體程式設計師應具的核心知識為本，規劃及擬訂應教授的課程內容及其訓練方式，進而規劃出「計算機概論」、「資料結構與演算法」、「資料庫系統」、「軟體開發技術」、「XML 語言」與「程式語言」等六門核心課程的詳細課程大綱；換言之，即由程式設計師的工作內容而訂出應具備的核心知識，由這些核心知識再進一步分析與整合而導出訓練課程應教那些子項，再由訓練課程的子項導列出詳細課程大綱，藉用此種展開方式所得出的六門課程詳細大綱，在此研究過程中，我們共同研

討廿次，每門課大約經過四次以上的共同討論，若加上每門課負責的教授在其學校內與同仁的討論，討論次數至少有八次以上，工作相當艱巨；討論過程中才了解「要訓練學生達到預定的目標」實不是一件容易的事，研究成員相互間學習了不少教學心得。

在座談會中，我們也討論是否馬上實施此課程大綱，但要自編教材，與會之各校教師均認為不易達到預期效果，一致共認應先共同製作教材後才推廣，因此本計畫之預定工作項目「課程大綱之實施建議及績效評估」，均在研究成員認為「實施有困難」之情形下不再討論，因此本報告也未列入此部分之研究心得。

其次，在制定課程大綱的過程，我們也發現每一本大學用書都是作者依其個人教學目標而編定，內容、範例與習題均與學生的背景有密切關係，我國學生背景與外國不儘相同，無法一致適用；當我們將此心得在九十三年二月十三日的座談會提出，參與的各校教師均有相同看法，因此共同編訂適用本國技職院校教學的教材確有其必要性。

本研究之工作雖然辛苦，但只能算是大學教學改進的一小步，本研究的成果在教育部技職司主辦的期中與期末檢討會中，獲得與會審查委員之肯定及寶貴修正意見，是我們的最大收穫，希望今後在教育部的主導下，有更多教授同仁參與大學教學改進工程工作，共同提升我國技職院校資訊相關科系的教學水準，為我國培育更多優秀的軟體人才。

9.2 建議

1、 對資訊相關科系之建議

程式設計是循序漸進的，本研究所提出的課程大綱，雖包括習題建議，但未考慮程式的行數，我們建議在大一「計算機概論」課程中，應以訓練學生熟悉程式指令為主要的，因此程式行數儘量以不超過一百行為原則，初期撰寫一、二十行的程式，以提升學生興趣，再逐步擴充行數。大二之「資料結構與演算法」課程，各習題之程式行數由一百行左右而漸進到三百行左右，每一習題儘量在二週內完成，且提供測試數據 (test data) 供學生自我測試。大三之「軟體發展技術」課程則用五百行左右的程式習題來訓練學生作好程式及其文件。目前各技職院校均有一年之「專題實驗」課程，可以設計每一學生需要完成三千行左右的程式來規劃，且以二至三人為一組，一方面可訓練他們分工合作的工作方式，另一方面可由五至八千行的程式來訓練學生作好軟體文件，藉此循序漸進方式訓練學生程式設計能力，程式偵錯技術、程式設計文件製作此能力。

2、 對教育部之建議

本研究是技職院校資訊相關科系教學改進工程的初始工作，還需繼續編寫教材，方能發揮效責，教材編寫若由一人來作，實不易在短期內搜集足夠資料而撰寫完成，若能以專案方式，邀集數位教授來共同編撰，先以投影片方式編寫及試教，進一步再編寫教本、相關習題及程式

練習，接著推廣數校試教，二年內應可完成上述課程教材而廣泛使用，從面提升我國資訊相關科系之教學水準。教育部每年補助大學的經費，大都用於硬題建設，盼望能多注重教材、教學精進的工作，方能進一步提升技職院校資訊相關科系的就業率，發揮教育成本效益。

此報告所提出的課程大綱，是本計畫成員的研究心得，我們是以「拋磚引玉」的心情來提出此課程大綱，希望能獲得其他教授提出更好的教學內容建議，期望教育部技職司每學期能以研討會方式來修正此大綱，進而使我們的學生能以更有效的方式，來學習程式設計應具有的核心知識，帶動學生設計程式之興趣，進而樂於投入軟體產業的行列，如此提升我國軟體產業的國際競爭力就指日可待了。

壹拾、文獻參考

1. Chyan-Goei Chung, Ta-Yu Tseng, Ming-ji Wu, and Hwi-Min Kuo, **The Strategy to enhance the effectiveness of software engineer training program in Taiwan**, International Conference on Engineering Education and Research, Olomouc and Bouzov Castle, Czech Republic, June 27-30, 2004.
2. 技專校院資訊相關系科課程教學與產業需求配合檢討之研究。
3. [Abelson85] Harold Abelson and Gerald Jay Sussman with Julie Sussman. *Structure and Interpretation of Computer Programs*. Cambridge, MA: MIT Press, 1985.
4. [ABET2000] Accreditation Board for Engineering and Technology. Accreditation policy and procedure manual. Baltimore, MD: ABET, Inc., November 2000.
<http://www.abet.org/images/policies.pdf>.
5. [ACM65] ACM Curriculum Committee on Computer Science. An undergraduate program in computer science—preliminary recommendations. *Communications of the ACM*, 8(9):543-552, September 1965.
6. [ACM68] ACM Curriculum Committee on Computer Science. Curriculum '68: Recommendations for the undergraduate program in computer science. *Communications of the ACM*, 11(3):151-197, March 1968.
7. [ACM78] ACM Curriculum Committee on Computer Science. Curriculum '78: Recommendations for the undergraduate program in computer science. *Communications of the ACM*, 22(3):147-166, March 1979.
8. [ACM99] ACM Two-Year College Education Committee. Guidelines for associatedegree and certificate programs to support computing in a networked environment. New York: The Association for Computing Machinery, September 1999.
9. [ACM2001] Association for Computing Machinery. ACM code of ethics and professional conduct. New York: The Association for Computing Machinery, May 2001. <http://www.acm.org/constitution/code.html>.
10. [AP2000] Advanced Placement Program. Introduction of Java in 2003-2004. The College Board, December 20, 2000. <http://www.collegeboard.org/ap/computer-science>.

11. [BCS89a] British Computer Society and The Institution of Electrical Engineers. Undergraduate curricula for software engineers. London, June 1989.
12. [BCS89b] British Computer Society and The Institution of Electrical Engineers. Software in safety-related systems. London, October 1989.
13. [Beidler85] John Beidler, Richard Austing, and Lillian Cassel. Computing programs in small colleges. *Communications of the ACM*, 28(6):605-611, June 1985.
14. [Bennett86] W. Bennett. A position paper on guidelines for electrical and computer engineering education. *IEEE Transactions in Education*, E-29(3):175-177, August 1986.
15. [Bott91] Frank Bott, Allison Coleman, Jack Eaton, and Diane Rowland. Professional issues in software engineering. London: Pitman, 1991.
16. [Carnegie92] Carnegie Commission on Science, Technology, and Government. Enabling the future: Linking science and technology to societal goals. New York: Carnegie Commission, September 1992.
17. [COSINE67] COSINE Committee. Computer science in electrical engineering. Washington, DC: Commission on Engineering Education, September 1967.
18. [CSAB86] Computing Sciences Accreditation Board. Defining the computing sciences professions. October 1986. http://www.csab.org/comp_sci_profession.html.
19. [CSAB2000] Computing Sciences Accreditation Board. Criteria for accrediting programs in computer science in the United States. Version 1.0, January 2000. http://www.csab.org/criteria2k_v10.html.
20. [CSTB94] Computing Science and Telecommunications Board. Realizing the information future. Washington DC: National Academy Press, 1994.
21. [CSTB99] Computing Science and Telecommunications Board. Being fluent with information technology. Washington DC: National Academy Press, 1999.
22. [Curtis83] Kent K. Curtis. Computer manpower: Is there a crisis? Washington DC: National Science Foundation, 1983. <http://www.acm.org/sigcse/papers/curtis83/>.

23. [Davis97] Gordon B. Davis, John T. Gorgone, J. Daniel Couger, David L. Feinstein, and Herbert E. Longnecker, Jr. IS'97 model curriculum and guidelines for undergraduate degree programs in information systems. Association of Information Technology Professionals, 1997. <http://webfoot.csom.umn.edu/faculty/gdavis/curcomre.pdf>.
24. [Denning89] Peter J. Denning, Douglas E. Comer, David Gries, Michael C. Mulder, Allen B. Tucker, A. Joe Turner, and Paul R. Young. Computing as a discipline. *Communications of the ACM*, 32(1):9-23, January 1989.
25. [Denning98] Peter J. Denning. Computing the profession. *Educom Review*, November 1998.
26. [Denning99] Peter J. Denning. Our seed corn is growing in the commons. *Information Impacts Magazine*, March 1999. http://www.cisp.org/imp/march_99/denning/03_99denning.htm.
27. [EAB83] Educational Activities Board. The 1983 model program in computer science and engineering. Technical Report 932, Computer Society of the IEEE, December 1983.
28. [EAB86] Educational Activities Board. Design education in computer science and engineering. Technical Report 971, Computer Society of the IEEE, October 1986.
27. [EC77] Education Committee of the IEEE Computer Society. A curriculum in computer science and engineering. Publication EHO119-8, Computer Society of the IEEE, January 1977.
28. [Gibbs86] Norman E. Gibbs and Allen B. Tucker. Model curriculum for a liberal arts degree in computer science. *Communications of the ACM*, 29(3):202-210, March 1986.
29. [Gorgone2000] John T. Gorgone, Paul Gray, David L. Feinstein, George M. Kasper, Jerry N. Luftman, Edward A. Stohr, Joseph S. Valacich, and Rolf T. Wigand. MSIS 2000: Model curriculum and guidelines for graduate degree programs in information systems. Association for Computing Machinery and Association for Information Systems, January 2000. <http://cis.bentley.edu/ISA/pages/documents/msis2000jan00.pdf>.
30. [IEEE2001] Institute for Electrical and Electronic Engineers. IEEE code of ethics.

Piscataway, NJ: IEEE, May 2001. http://www.ieee.org/about/what_is/code.html.

31. [Kelemen99] Charles F. Kelemen (editor), Owen Astrachan, Doug Baldwin, Kim Bruce, Peter Henderson, Dale Skrien, Allen Tucker, and Charles Ban Loan. Computer Science Report to the CUPM Curriculum Foundations Workshop in Physics and Computer Science. Report from a workshop at Bowdoin College, October 28-31, 1999.
32. [Koffman84] Elliot P. Koffman, Philip L. Miller, and Caroline E. Wardle. Recommended curriculum for CS1: 1984 a report of the ACM curriculum task force for CS1. *Communications of the ACM*, 27(10):998-1001, October 1984.
33. [Koffman85] Elliot P. Koffman, David Stemple, and Caroline E. Wardle. Recommended curriculum for CS2, 1984: A report of the ACM curriculum task force for CS2. *Communications of the ACM*, 28(8):815-818, August 1985.
34. [Lee98] Edward A. Lee and David G. Messerschmitt. Engineering and education for the future. *IEEE Computer*, 77-85, January 1998.
35. [Lidtke99] Doris K. Lidtke, Gordon E. Stokes, Jimmie Haines, and Michael C. Mulder. ISCC '99: An information systems-centric curriculum '99, July 1999. <http://www.iscc.unomaha.edu>.
36. [Martin96] C. Dianne Martin, Chuck Huff, Donald Gotterbarn, Keith Miller. Implementing a tenth strand in the CS curriculum. *Communications of the ACM*, 39(12):75-84, December 1996.
37. [Mulder75] Michael C. Mulder. Model curricula for four-year computer science and engineering programs: Bridging the tar pit. *Computer*, 8(12):28-33, December 1975.
38. [Mulder84] Michael C. Mulder and John Dalphin. Computer science program requirements and accreditation—an interim report of the ACM/IEEE Computer Society joint task force. *Communications of the ACM*, 27(4):330-335, April 1984.
39. [Mulder98] Fred Mulder and Tom van Weert. Informatics in higher education: Views on informatics and noninformatics curricula. *Proceedings of the IFIP/WG3.2 Working Conference on Informatics (computer science) as a discipline and in other disciplines: What is in common?* London: Chapman and Hall, 1998.

40. [Myers98] J. Paul Myers, Jr. and Henry M. Walker. The state of academic hiring in computer science: An interim review. *SIGCSE Bulletin*, 30(4):32a-35a, December 1998.
41. [NACE2001] National Association of Colleges and Employers. Job outlook '01 (online version). <http://www.jobweb.com>
42. [Neumann95] Peter G. Neumann. Computer related risks. New York: ACM Press, 1995.
43. [NSF96] National Science Foundation Advisory Committee. Shaping the future: New expectations for undergraduate education in science, mathematics, engineering, and technology. Washington DC: National Science Foundation, 1996.
44. [NTIA99] National Telecommunications and Information Administration. Falling through the Net: Defining the digital divide. Washington, DC: Department of Commerce, November 1999.
45. [Nunamaker82] Jay F. Nunamaker, Jr., J. Daniel Couger, Gordon B. Davis. Information systems curriculum recommendations for the 80s: Undergraduate and graduate programs. *Communications of the ACM*, 25(11):781-805, November 1982.
46. [OTA88] Office of Technology Assessment. Educating scientists and engineers: Grade school to grad school. OTA-SET-377. Washington, DC: U.S. Government Printing Office, June 1988.
47. [Paulk95] Mark Paulk, Bill Curtis, Mary Beth Chrissis, and Charles Weber. The capability maturity model: Guidelines for improving the software process. Reading, MA: Addison-Wesley, 1995.
48. [QAA2000] Quality Assurance Agency for Higher Education. A report on benchmark levels for computing. Gloucester, England: Southgate House, 2000.
49. [Ralston80] Anthony Ralston and Mary Shaw. Curriculum '78—Is computer science really that unmathematical. *Communications of the ACM* (23)2:67-70, February 1980.
50. [Roberts95] Eric Roberts, John Lilly, and Bryan Rollins. Using undergraduates as teaching assistants in introductory programming courses: An update on the Stanford experience. *SIGCSE Bulletin* (27)1:48-52, March 1995.

51. [Roberts99] Eric Roberts. Conserving the seed corn: Reflections on the academic hiring crisis. SIGCSE Bulletin (31)4:4-9, December 1999.
52. [SAC67] President's Science Advisory Commission. Computers in higher education. Washington DC: The White House, February 1967.
53. [SEEPP98] IEEE-CS/ACM Joint Task Force on Software Engineering Ethics and Professional Practices (SEEPP). Software engineering code of ethics and professional practice (Version 5.2). <http://www.acm.org/serving/se/code.htm>.
54. [Shaw85] Mary Shaw. The Carnegie-Mellon curriculum for undergraduate computer science. New York: Springer-Verlag, 1985.
55. [Shaw91] Mary Shaw and James E Tomayko. Models for undergraduate courses in software engineering. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, January 1991.
56. [Shaw92] Mary Shaw. We can teach software better. Computing Research News 4(4):2-12, September 1992.
57. [SIGCHI92] Special Interest Group on Computer-Human Interaction. ACM SIGCHI Curricula for Human-Computer Interaction. New York: Association for Computing Machinery, 1992.
58. [SWEBOK01] Software Engineering Coordinating Committee. Guide to the Software Engineering Body of Knowledge (SWEBOK). Stone Man Version 0.95. A Project of the IEEE Computer Society, May 2001. <http://www.swebok.org/stoneman/version095.html/>.
59. [Tucker91] Allen B. Tucker, Bruce H. Barnes, Robert M. Aiken, Keith Barker, Kim B. Bruce, J. Thomas Cain, Susan E. Conry, Gerald L. Engel, Richard G. Epstein, Doris K. Lidtke, Michael C. Mulder, Jean B. Rogers, Eugene H. Spafford, and A. Joe Turner. Computing Curricula '91. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, 1991.
60. [Walker96] Henry M. Walker and G. Michael Schneider. A revised model curriculum for a liberal arts degree in computer science. Communications of the ACM, 39(12):85-95, December 1996.
61. [Zadeh68] Lofti A. Zadeh. Computer science as a discipline. Journal of Engineering

Education, 58(8):913-916, April 1968.

附錄一、座談會會議記錄

時間：民國九十三年一月十三日（星期二）

地點：長庚大學工學大樓六樓國際會議廳

出席人員：

萬能技術學院	資工系	黃春福	副教授
高苑技術學院	資管系	李若愚	副教授
嶺東技術學院	資訊科技系	洪瑞鍾	系主任
嘉義私立大同技術學院	資訊管理科	朱晏瑩	講師
弘光科技大學	資管系	許昌齡	系主任
中華醫事技術學院	資管系	羅逸文	講師
遠東技術學院	資管系	蔡文彬	系主任
修平技術學院	資管系	謝志明	系主任
建國技術學院	電子工程系	邱謙松	助理教授
元培科學技術學院	資管系	張世鵬	講師
龍華科技大學	資訊管理系	李之中	講師
醒吾技術學院	資管系	劉家驊	系主任
大華技術學院	資管系	林永修	講師
嶺東技術學院	資管系	黃國豪	系主任
長庚技術學院	資管系	丁毓華	副教授
康寧專校	資管系	李仁鐘	系主任
育達商業技術學院	資管系	林耀仁	助理教授
建國技術學院	資管系	魏能城	系主任
台北護理學院	資管系	李炯三	系主任
南亞技術學院	資管系	文德蘭	
台南女子技術學院	資管系	劉育釗	

技專校院資訊相關系科課程教學與產業需求配合檢討 之研究講習研討會議簽到表

時間：民國九十三年一月十三日（星期二）

地點：長庚大學工學大樓六樓國際會議廳

學校名稱	系所職稱	姓名	簽名
萬能技術學院	資工系副教授	黃春福	黃春福
高苑技術學院	資管系副教授	李若愚	李若愚
嶺東技術學院	資訊系系主任	洪瑞鍾	洪瑞鍾
大同技術學院	資訊科講師	朱晏瑩	朱晏瑩
萬能技術學院	電子系講師	林文彥	
黎明技術學院	教學資源中心主任	陳林宏	
弘光科技大學	資管系系主任	許昌齡	許昌齡
中華醫事技術學院	資管系講師	羅逸文	羅逸文
遠東技術學院	資管系系主任	蔡文彬	蔡文彬
修平技術學院	資管系系主任	謝志明	謝志明
建國技術學院	電子系助理教授	邱謙松	邱謙松
元培科學技術學院	資管系講師	張世鵬	張世鵬
龍華科技大學	資管系講師	李之中	李之中
醒吾技術學院	資管系系主任	劉家驊	劉家驊
蘭陽技術學院	資管系講師	吳季偉	
大華技術學院	資管系講師	林永修	林永修
嶺東技術學院	資管系系主任	黃國豪	黃國豪
長庚技術學院	資管系副教授	丁毓華	丁毓華
康寧專校	資管系系主任	李仁鐘	李仁鐘
育達商業技術學院	資管系助理教授	林耀仁	林耀仁
澎湖技術學院	資管系系主任	黃國光	
建國技術學院	資管系副教授	魏凱城	魏凱城

業

台北護照學院 資管系主任

李仁鐘

南亞技術學院 資管系主任

李仁鐘

台南... 資管系主任

李仁鐘

計算機概論課程的意見

1. 計算機概論時數分配似乎不夠，一般來說學校計概的課程，每週時數為三小時，但本計畫卻安排五小時，這會不會造成老師上課時，無法教完。另外，是不是要分優先順序，讓老師知道哪一部份要先教，以作為上課內容調整時的依據。

A：我們在設計這些課程時，是假設學生應該有某種程度的瞭解，如果沒有，是不是就應該在學校的課程上做些調整。如果每週三小時還不夠，就應該調整為五小時，而不是一昧地想把所有的知識在一定的時間交給學生而忽略其學習效果。教育部在做評鑑時，發現有些學校把一些高級課程（如：AI、影像處理等）放入課程中。如果學生基礎知識都不夠好，這些課程適不適合學生就值得商榷了。這課程（四學分，五小時）是本組在整體課程考量下設計出來，內容有建議的習題讓學生在學習過程中可以充分運用所學。因為這些內容已經是最精簡了，所以沒有分優先順序。
2. 計概課程太偏向資訊相關科系。現在的軟體開發已有模組型態的出現，是不是還需要教這麼基礎的知識？另外，此課程內容並沒有著重資管學生需要的文書處理，也沒有電腦硬體介紹的部分，而且各校都將計算機概論做為學校的特色，而不應該以這課程為主。

A：此「計算機概論」的名稱只是暫訂的，因為一時找不到適當的名稱。但本課程設計是以訓練學生為程式設計師為出發點，如果各校覺得需要加入其他特色，可以以本課程的規劃為主，再加入其他的部分。另外，如果學生只會用模組方式來設計程式，不見得對程式的內容有所瞭解，因此程式的基礎訓練要比會用模組設計程式來的重要。
3. 在嶺東技術學院，計概是全校必修的課程，其內容差異很大，有些科系是教 Office；有些科系是教 XP。資管系為了要有所區別，特別將課名改為「計算機導論」。而本計畫所規劃的計概課程的學分數太多，這對學生的重修會產生問題，也會對其他課程產生排擠效應。另外，資管學生的素質不一，在課程內容安排上會產生困難，尤其是 Java 語言，學生的接受度較差。

A：像 Office 這種電腦的知識是一般大學生都要有的，並非只限於學資訊的學生，而我們應該訂出什麼是資訊學生應該有的 Core Knowledge。本組是以目前企業需要資訊人才來定義 Core Knowledge。至於課程內容是以由簡到難，循序漸進的方式設計，這些都是 Core Knowledge。至於各校要如何調整，請自行決定。資訊管理或管理資訊都需要資訊的 Core Knowledge，資管的學生目前是沒有就業的問題，但未來的發展將會遇到問題，其原因就是對資訊的 Core Knowledge 不夠瞭解。以現在印度及中國大陸資訊相關人員的能力來看，台灣資管學生的未來會有很大的隱憂。
4. 樹德技術學院資管系的專題課程設計得很好，學生能力並不輸給資工系學生的能力，不知道是如何達到的？

A：我們學校成立了資訊學院，內有電算中心、資工系、資管系三部。因此在課程上，我們

設計了 IT 的必修課程，其中 JAVA 程式設計課程就需要六學期。另外，系上注重動手做，本系的專題是三學分三學期。所有的學生不管是做何種專題題目都要設計一個系統出來。此外，為了讓學生能注重專題，系上提供適當的實習空間可以讓學生有討論的空間。

5. 目前一般私立技職院校的資管學生人數都非常多，與其他科系相較之下，資管教學環境及教具常有一些不足問題，另外教師也需要時間備課，不知道研究小組是不是可以建議教育部提供教學環境協助。

A：這部分本組會寫入報告書中。

資料結構與演算法課程的意見

1. 資料結構與演算法的時數只有 63 小時，在時數上似乎有所不足，尤其是對技職體系的學生來說。

A：我們同意你的看法，要如何教這門課會是一個重要的課題，但我們在定義這課程時，是以教授基本概論為主，而不是以考試為導向。時數上的不足我們會再考量。另外，我們也需要一個機制來評量教學及教材內容。

2. 以嶺東技術學院的技職學生程度來說，需要一年的時間才能完成資料結構的課程，而演算法為選修課。本計畫的課程安排在我們學校實施會有問題。

A：課程的安排上要考慮學生的學習能力，而不是注重教授課程內容。有些資料結構課本的內容太重視細節，對技職的學生來說，似乎不適合，因為學生只要瞭解幾種資料結構的重要方法，而不是要去準備考研究所。

3. 資料結構的課程設計跟早上的計概課程相比，有點粗糙，課程內容的順序考慮的不是很周到，有些內容會用到其他還沒教到的章節，而解題技巧所用的時數又太多，會造成學生在全盤瞭解的情況下就要學解題技巧。另外，學生可能並不知道背景知識為何？跟其他知識有何關係？且有些知識內容要到大三才能上完。課程中沒有物件導向的習題，但有物件導向的授課內容，學生這時可能並不瞭解什麼是物件導向。

A：TREE、STACK、QUEEN 以物件導向元件的方式來教，應該注重於學生在觀念上的瞭解，而不是元件的問題。有些資料結構的問題（如：檔案處理）已經在計概中教授，所以我們在課程上已經考慮它們之間的關連性。在 Graph 的部分，時間上似乎少了，我們會考慮增加。我們在課程設計上，以職業上需求來設計，觀念瞭解最重要，細節就不需要太多的琢磨。

4. 課程設計上是不是考慮到如何將軟體應用系統對應到物件導向的範例中，例如：資料庫系統。

A：在課程上我們會考慮範例的設計，力求學生瞭解及應用，而時間上，我們會 再做調整。

資料庫課程的意見

1. 本校的資料庫課程分為理論及實務兩部分，各占 12 小時，不知道放這麼多時間是不是合理？是不是會排擠到其他課程？

A：我們在課程上安排也是朝向這方向走。在你們學校，這種的時間安排是足夠了。目前瞭解 DBA 的人太少了，正需要這樣的訓練。

2. 本校的資料庫是以 MSSQL 為主，但以本計畫所提課程在時間規劃上有點少，在我們學校裡有些單元是另外開課來上。

A：這樣的安排可能太多了。另外，學校要找出自己的特色，而不是每一種資料庫都教，大家可以坐下來商量，是不是有些學校教 Oracle，有些學校教 MSSQL。

3. 到目前為止，我們都只談 How to do，而不談 What to do、Why to do。應該教學生「為什麼要這麼做」，例如 MIS 的教科書內容就都不一樣，且學生很難瞭解。是不是由劇本（系統）導向來教學，這樣我們就可以將需要的知識交給學生，讓學生瞭解現有系統是如何設計，會使用到哪些技術。專題的好處就是讓學生學習如何發現問題，解決問題。

A：教書是種藝術，如何將課程教好是經驗的累積。你所提劇本的觀念是對的，吳教授就是以此方法來設計計概課程。我們希望其他的課程也能以此方法來設計。

XML 課程的意見

1. 在企業中，資料就是資產；資料就是智慧財產。如何讓資料得以保存及傳遞就變的非常重要，XML 的使用是一個趨勢。很高興看到本計畫能將最新的 IT 技術放入課程中。

A：XML 課程是必需的。電腦科技變化太快，將來還會有新的技術出現，如何將新的資訊技術加入課程中，並將課程教好，將會是一大挑戰。我認為數位學習很重要。目前這部分各國都在自行發展，但如果台灣發展的速度太慢，而由外國人進來發展，則台灣可能失去競爭力。數位學習設計中，舉例很重要，要能跟日常生活結合，學習效果才會好。預計到了 2010 年，網路大學會非常普遍。

軟體發展課程的意見

1. 在系統分析課程中，我們通常在前半部會教結構化系統分析，後半部才教物件導向系統分析及 UML，這樣會不會造成學生學習上的困擾。

A：學生沒有在外面工作的經驗，所以很難瞭解課程的內容。最好是能跟外面的企業合作，以現成的例子來作系統分析。在大學裡要教物件導向系統分析是很難的，而 UML 只是工具，它只能做到部分的分析。如何讓學生瞭解系統的架構才是重點，但這也很困難。

目前業界最需求是程式文件、品質文件的製作，這也是本課程的重點。

程式語言課程的意見

1. 本計畫當初是否以產業需求來設計課程，是不是有事先做過分析來分類課程內容，如多媒體，網路程式設計，其程式語言課程就有所不同。

A：行政院經濟部有做產業人才需求分類，並沒有課程分類。本計畫認為核心知識很重要，因此著重程式語言的核心知識，至於課程延伸出去的分類可作為各校用來顯示特色之用。本計畫也做已畢業學生就業問卷，就是希望能找出目前產業真正需要的知識，但因為訪問的對象只有本計畫的四所學校，樣本太少，希望其他學校也能協助我們做問卷調查，讓我們更了解外面的需求。

產業的意見（台塑）

課程的安排跟產業的需求差不多，但使用工具不一樣，因此，我們會依不同領域來找人，再做在職訓練，訓練的內容跟課程差不多，但以實用為主。訓練時間為一個半月，之後才能上線。而且是由小系統維護做起、然後中系統、再到大系統。在訓練的過程中，我們發現知識的管理非常重要，但這也是技職學生最缺乏，尤其是當資訊（資料）越來越大時，如何解決這些問題就更顯重要。這計畫內容很合乎產業的需求，如果學校能再跟產業合作就更好了，例如 ERP 的 demo 可以讓學生了解產業的運作。在長庚大學裡，學生是需要實習，如此才可以將學校所學用在工作上。另外，這研究也讓我們瞭解未來課程的趨勢。

綜合討論的意見

1. 程式設計課程，上課人數限制一般都是五十人，最合適的人數應為二十五人，但這又與學校成本相衝突。我們是不是可以建議學校，課程以程式設計為主軸，將大三、大四的部分課程取消，以加強基礎知識的訓練。

A：這部分因為各學校狀況不同，我們很難有一個共同的方法可以解決，但我們可以建議降低老師的時數及增加助教的方式來強化上課的品質。

2. 技職體系的學生缺乏實作經驗，所以應該多利用實例讓學生了解課程內容，尤其是系統分析課程。

A：系統分析課程應著重在系統架構上介紹，但這是不是應該到研究所才教，因為技職體系學生的經驗可能還不足以教授系統架構，所以我們在系統分析課程上作了一些調整。

3. 資料庫課程的內容是不是需再加強，因為要在這麼短的時間內將所有資料庫課程教完，似乎不太可能，尤其是對 SA 的 Domain 知識瞭解，否則很難將資料庫處理好。

A：資料庫的課程不好設計，一般書籍所教的都是講資料庫的架構，並沒有將「為什麼要使用資料庫？什麼情況下使用資料庫是有效的？什麼情況下是不好的？」說清楚。因此在課

程安排上，我們從資料的有效儲存使用再到資料庫的架構。如果需要更進一步的課程教授，是不是再額外開課教授。

4. 技職學校的授課應該注重實作，例如程式設計課應有實習課程安排，最好能利用業界建教合作，將一些實例放入到教學中。另外，在作評鑑時，是不是可以將學校建教合作的成果當作是一個評分的重點。

A：實習很重要，但學校在什麼時候安排實習，也很重要。一般來說，大三的暑假是比較適合。教育部目前很難要求各個學校都要實行，而且企業要有認知，因為學生實習對企業來說，是減分的效果，而非加分。

5. 這課程規劃是由資訊相關系科課程教學與產業需求來設計，不過我有幾點建議：一、課程細分的空間上，是不是還有其他課程需要加入。二、產業的需求，在這些課程裡，是不是已經滿足。三、有沒有後續的計畫。四、感覺上教育部比較不重視技職的學生。

A：我們希望還有後續計畫，讓大家能一起來設計公用教材，同時教育部能重點補助這方面的計畫，這樣國內的學生才能與外國競爭。

附錄二、 歷次會議記錄

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第一次會議開會通知

- 一、日期：九十二年四月三日[星期四] 下午 2:00-4:30
- 二、地點：行政院科技顧問組（台北市和平東路二段科技大樓五樓）
- 三、主席：鍾乾癸教授
- 四、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 五、記錄：
- 六、開會主題及題案討論
 - 1、本計畫執行目的、目標項目簡介，研究計畫分工及說明
 - 2、資訊軟體人才培訓品質提升計畫---程式設計人員類說明
 - 3、資訊軟體人才培訓課程規劃說明
 - 4、分析國內技專院校資訊相關系科課程內容及教學實施情形，檢討畢業生就業與業界需求人才問題，以研訂改進課程教學策略，以符合業界用人需求。
 - 5、依據資訊軟體人才分類之各知識體系，規劃課程系統架構，以課程系統架構為基礎，設計學程課程。
- 七、準備資料
 - 1、技專校院資訊相關系科課程教學與產業需求配合檢討之研究計畫
 - 2、資訊軟體人才培訓品質提升計畫---程式設計人員類簡報
 - 3、資訊軟體人才培訓課程規劃書
 - 4、運用印度軟體人才提升我國軟體競爭力簡報
- 八、簽到：

鍾乾癸

吳志宏 吳光閔 趙國連

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第一次會議開會會議紀錄

- 一、日期：九十二年四月三日[星期四] 下午 2:00-4:30
- 二、地點：行政院科技顧問組（台北市和平東路二段科技大樓五樓）
- 三、主席：鍾乾癸教授
- 四、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 五、記錄：
- 六、開會主題：
 - 1、本計畫執行目的、目標項目簡介，研究計畫分工及說明
 - 2、資訊軟體人才培訓品質提升計畫---程式設計人員類說明
 - 3、資訊軟體人才培訓課程規劃說明
- 七、主席報告：
 - 1、技專校院資訊相關系科課程教學與產業需求配合檢討之研究計畫
 - 2、資訊軟體人才培訓品質提升計畫---程式設計人員類簡報
 - 3、資訊軟體人才培訓課程規劃書
- 八、題案討論

題案一：本計畫執行目的、目標項目簡介，研究計畫分工及說明案說明：

 - 1、技專校院資訊相關系科課程教學與產業需求配合檢討之研究計畫，如附件一
 - 2、資訊軟體人才培訓品質提升計畫---程式設計人員類，如附件二
 - 3、資訊軟體人才培訓課程規劃說明，如附件三

決議：

 - 1、依技專校院資訊相關系科課程教學與產業需求配合檢討之研究計畫執行本計畫
 - 2、配合行政院資訊軟體人才培訓提升效能策略規劃構想，以資訊軟體人才分類為基礎，依各類資訊軟體人才所需知識體系為範圍，規劃資訊相關科系課程。
 - 3、依據資訊軟體人才分類之各知識體系，規劃課程系統架構，以課程系統架構為基礎，設計學程課程。
 - 4、以吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系為本計畫之主要研究對象
 - 5、分析國內技專院校資訊相關系科教學課程與行政院資訊軟體人才培訓課程規劃之差異
 - 6、以問卷方式調查資管系畢業生就業情形及業界對資訊人才專業之需求狀況
- 九、下次會議事項：
 - 1、第二次會議訂在四月十二日（星期六）中午 12:00-4:00 地點：高雄中山大學
 - 2、各專案研究員請收集該學系之課程與行政院規劃之資訊軟體人才培訓課程對映表以便分析討論國內技專院校資訊相關系科教學課程與行政院資訊軟體人才培訓課程規劃之差異問題

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第二次會議開會通知及簽到表

- 一、日期：四月十七日（星期四）
- 二、時間：上午 09:30-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閩（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、分析討論國內技專院校資訊相關系科教學課程與行政院資訊軟體人才培訓課程規劃之差異問題
 - 2、討論資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷
- 八、準備資料
 - 1、各專案研究員收集吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系之課程與行政院規劃之資訊軟體人才培訓課程對映表
 - 2、各專案研究員提供資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷項目
- 九、簽到：

鍾乾癸

陳信良

張正源

吳志宏

吳光閩

丁國順

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第二次會議開會會議紀錄

- 一、日期：四月十七日（星期四）
- 二、時間：上午 09:30-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、各專案研究員請收集該學系之課程與行政院規劃之資訊軟體人才培訓課程對映表以便分析討論國內技專院校資訊相關系科教學課程與行政院資訊軟體人才培訓課程規劃之差異問題
 - 2、資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷
- 八、主席報告：
 - 1、請各專案研究員介紹各校與培訓課程之對映表。
 - 2、請各專案研究員提供各學校畢業校友之就業分析。
- 九、題案討論
 - 1、題案一：各專案研究員提供資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷項目
 - (一)、說明：爲了要瞭解資管系畢業學生就業情形，是否將受訪對象擴及至進修專校及學院的學生。
 - (二)、決議：
 - (1). 這次研究的主要對象是四技的學生，如果能從進修專校及學院的學生中得取資料也不錯，但此資料只能做爲參考，並不能代表研究的母群體。
 - (2). 爲了使研究的結果更精確，我們要設計一份問卷，在請各學校拿回做調查。
 - 2、題案二：各專案研究員收集吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系之課程與行政院規劃之資訊軟體人才培訓課程對映表
 - (一)、說明：爲了要瞭解各學校之資管系的課程與行政院規劃之資訊軟體人才培訓課程之差異，請各學校能提供資料說明之間的差異。
 - (二)、決議：
 - (1). 由於各學校說明文件之格式不一致，因此將設計統一之表格，再由各學校填寫。
 - (2). 在表格中請填寫相關的課程及上課的時數。
- 十、下次會議事項：
 - 1、請中國技院資管系張主任先設計畢業生就業情形及業界對資訊人才專業之需求狀況

調查問卷的草稿，之後再由各專案研究員共同研究討論修改。

- 2、請吳鳳技院資管系陳主任先設計資管系之課程與行政院規劃之資訊軟體人才培訓課程對映表，再交由各專案研究員填寫，以求統一。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第三次會議開會通知及簽到表

- 一、日期：四月二十四日（星期四）
- 二、時間：上午 09:30-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、分析討論國內技專院校資訊相關系科教學課程與行政院資訊軟體人才培訓課程規劃之差異問題
 - 2、討論資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷
- 八、準備資料
 - 1、各專案研究員收集吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系之課程與行政院規劃之資訊軟體人才培訓課程對映表
 - 2、各專案研究員提供資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷項目
- 九、簽到：

鍾乾癸

吳志宏 吳光閔 趙國連
丁國順

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第三次會議開會會議紀錄

- 一、日期：日期：四月二十四日（星期四）
- 二、時間：上午 09:30-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、分析討論國內技專院校資訊相關系科教學課程與行政院資訊軟體人才培訓課程規劃之差異問題
 - 2、討論資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷
- 八、主席報告：
 - 1、請各專案研究員介紹各校與培訓課程之對映表。
 - 2、請各專案研究員提供各學校畢業校友之就業分析。
- 九、題案討論
 - 1、題案一：樹德科大資管系之畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷設計討論
 - (一)、說明：爲了要瞭解資管系畢業學生就業情形，樹德科大資管系吳主任設計了畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷。
 - (二)、決議：
 - (1). 這次設計的內容不錯，但是有些調查的內容需要做修改。如果能將問題事先作個流程上的設計，是不是可以將我們需要的訊息從調查中獲得。
 - (2). 在問題的設計上，是不是可以更斟酌，不要讓受訪者不知如何回答。
 - 2、題案二：中國技院資管系之進修部畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷設計討論
 - (一)、說明：爲了要瞭解資管系進修部在學或畢業學生就業情形，樹德科大資管系吳主任設計了進修部在學及畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷。
 - (二)、決議：
 - (1). 調查的內容能否像樹德科大設計的方式一樣，先將將問題事先作個流程上的設計。
 - (2). 在進修部就學的學生，有許多是非本系出生，因此在做調查時，應以職場上以從事資訊行業的人爲主。
 - 3、題案三：各專案研究員收集吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系之課程與行政院規劃之資訊軟體人才培訓課程對應表

(一)、說明：爲了要瞭解各學校之資管系的課程與行政院規劃之資訊軟體人才培訓課程之差異，請吳鳳技院資管系陳主任設計的對應表，並由各學校填寫。

(二)、決議：

(1). 各學校都有寫對應表，但資料還是有缺少。是不是能請各學校將不足的部分補齊，之後再來討論資訊人才專業課程的需求。

十、下次會議事項：

- 1、請中國技院資管系張主任先設計進修部畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷的草稿，之後再由各專案研究員共同研究討論修改。
- 2、請樹德科大資管系吳主任先設計畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷的草稿，之後再由各專案研究員共同研究討論修改。
- 3、請各學校將資管系之課程與行政院規劃之資訊軟體人才培訓課程對映表填寫完成，之後再由各專案研究員共同研究討論

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第四次會議開會通知及簽到表

- 一、日期：六月二十六日（星期四）
- 二、時間：上午 10:00-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷設計問題
 - 2、討論吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系之課程與行政院規劃之資訊軟體人才培訓課程對映表
- 八、準備資料
 - 1、各專案研究員收集吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系之課程與行政院規劃之資訊軟體人才培訓課程對映表
 - 2、各專案研究員提供資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷項目
- 九、簽到：

張正源
丁國順
吳志宏
鍾乾癸

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第四次會議開會會議紀錄

- 一、日期：日期：六月二十六日（星期四）
- 二、時間：上午 10:00-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論資管系畢業生就業情形及業界對資訊人才專業之需求狀況調查問卷設計問題
 - 2、討論吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系之課程與行政院規劃之資訊軟體人才培訓課程對映表
- 八、主席報告：
 - 1、問卷的內容，樹德科大已經設計好了，請中國技院能依此修改進修學院的問卷。將所有問卷能皆下來的時間裡在幾所學校中對學生做問卷。
 - 2、皆下來的時間，我們也將開始設計課程內容，因此，我們先以計算機概論為例，各學校先自行設計內容，下次開會時，大家再來討論，希望能設計出統一的規格。
 - 3、由於技職體系學生的程度一般比大學學生的程度要差，因此，我們找出原因並用於課程設計上。
- 九、題案討論
 - 1、題案一：問卷的內容已經設計好了，請中國技院能依此修改進修學院的問卷，並將所有問卷在幾所學校中對學生做問卷。
 - (一)、說明：爲了取足夠的樣本，我們需要其他學校的幫忙來做問卷。
 - (二)、決議：請各個學校開始做問卷，並請樹德科大吳主任跟南台科大及昆山科大聯絡做問卷，以增加問卷的樣本數。
 - 2、題案二：由於技職體系學生的程度一般比大學學生的程度要差，因此，我們找出原因並用於課程設計上。
 - (一)、說明：爲了能真正設計適合技職體系學生的課程，我們必須瞭解技職體系學生程度低落的原因。
 - (二)、決議：請各個學校先在學校中開會討論這議題，再將這議題的結果於下次會議中討論。
 - 3、題案三：我們將開始設計課程內容，因此，我們先以計算機概論為例，各學校先自行設計內容，下次開會時，大家再來討論。
 - (一)、說明：爲了能真正設計適合技職體系學生的統一課程。
 - (二)、決議：請各個學校先自行設計內容，再將其結果於下次會議中討論。
- 十、下次會議事項：
 - 1、討論資管系學生程度低落及教學問題

2、討論吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之計算機概
論課程教學大綱設計

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第五次會議開會通知及簽到表

- 一、日期：七月十日（星期四）
- 二、時間：上午 8:30-10:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論資管系學生程度低落及教學問題
 - 2、討論吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之計算機概論課程教學大綱設計
- 八、準備資料
 - 1、各專案研究員收集吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之資管系之課程研討會心得及探討
 - 2、各專案研究員提供資管系之計算機概論課程教學大綱設計內容
- 九、簽到：

鍾乾癸

張正源

陳信良

丁國順

吳志宏（謝文川代）

吳光閔

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第五次會議開會會議紀錄

一、日期：日期：七月十日（星期四）

二、時間：上午 8:30-10:00

三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）

四、主席：鍾乾癸教授

五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）

六、記錄：丁國順（吳鳳技院資管系助理教授）

七、開會主題：

1、討論資管系學生程度低落及教學問題

2、討論吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之計算機概論課程教學大綱設計

八、主席報告：

1、資管系學生程度低落及教學問題很多，是不是請各位提出各校所得的結果，並將問題彙整（如附件）。

2、這次大家所提出教學大綱內容都格式不一致，是不是現在我們現在就先討論出一個版本，之後請各位依照這版本，將「計算機概論」教學大綱提出後再討論。

九、題案討論

1、題案一：因為教學大綱格式不統一，是不是現在就討論出結果出來。

（一）、說明：因為各校提出的教學大綱格式都不一致。爲了求一致，請決定最後的版本。

（二）、決議：依吳鳳技院所提的格式來填寫。

2、題案二：是不是將「計算機概論」教學大綱格式統一，並彙整，下次開會時，大家再來討論。

（一）、說明：因為各校提出的教學大綱格式都不一致，無法確實討論出結果，因此請各個學校能依新表格填寫，再討論。

（二）、決議：請各位負責的老師先將自己學校的教學大綱依我們討論的格式填寫，再由吳鳳技院負責彙整，於下次開會時討論。

十、下次會議事項：討論吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之計算機概論課程教學大綱設計

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第六次會議開會通知及簽到表

- 一、日期：七月二十四日（星期四）
- 二、時間：上午 10:00-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之計算機概論課程教學大綱設計
- 八、準備資料
 - 1、各專案研究員提供資管系之計算機概論課程教學大綱設計內容，事前交吳鳳技院丁老師彙整，於當天開會時提出討論。
- 九、簽到：

陳信良、張正源
丁國順 吳志宏
鍾乾癸

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第六次會議開會會議紀錄

- 一、日期：七月二十四日（星期四）
- 二、時間：上午 10:00-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論吳鳳技院、南華大學、樹德科大、中國技院、明新科大等五所學校之計算機概論課程教學大綱設計
- 八、主席報告：
 - 1、各個學校的課程教學大綱設計都有自己的一套方法及原則，且格式也不一樣。爲了要使本教學大綱能合乎技職學生的需求，我們應該從下面的方向來思考計算機概論課程教學大綱的內容：
 - (一)、從電腦內部到電腦外部的介紹。
 - (二)、爲了讓學生能瞭解電腦運作的方式，在教學的內容上，應該多給實例。
 - (三)、程式設計跟計概課程的內容很接近，因此將程式設計放入計概課程的第二部分。
- 九、題案討論
 - 1、題案一：請吳鳳技院及樹德科大設計之計算機概論課程教學大綱，之後我們再來討論。
 - (一)、說明：各個學校的課程教學大綱設計都有自己的一套方法及想法，爲了能加速計畫的進行，我們先請吳鳳技院及樹德科大先行參考其他學校的資料後，再設計計算機概論課程教學大綱，
 - (二)、決議：請吳鳳技院及樹德科大設計計算機概論課程教學大綱，之後我們再來討論。
- 十、下次會議事項：
 - 1、討論吳鳳技院及樹德科大設計之計算機概論課程教學大綱設計

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第七次會議開會通知及簽到表

- 一、日期：八月七日（星期四）
- 二、時間：上午 10:00-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論吳鳳技院及樹德科大設計之計算機概論課程教學大綱設計
- 八、準備資料
 - 1、請吳鳳技院及樹德科大準備設計之計算機概論課程教學大綱設計內容，於當天開會時提出討論。
- 九、簽到：

張正源

鍾乾癸

吳光閔

吳志宏

丁國順

/

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第七次會議開會會議紀錄

- 一、日期：日期：八月七日（星期四）
- 二、時間：上午 10:00-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論吳鳳技院及樹德科大設計之計算機概論課程教學大綱設計
- 八、主席報告：
 - 1、兩校所提的計算機概論課程教學大綱設計，內容上大致一樣，不過樹德科大設計的順序比較適合本計畫的想法。吳教授在這方面做的不錯，因此我們請吳教授繼續幫我們設計課程教學大綱。
 - 2、資訊相關科系學生在四年學習生活中，應該學習至少兩種語言，因此在計概課程中應該加入其中一種程式語言。如果時數不夠用，我們可以分成兩個部分來教，前半部份著重於計概的基礎知識，後半部則是程式設計的部分。在課程設計上，應該前後連貫，由簡單到複雜，並能給予學生實作，使學生瞭解上課內容的應用。
- 九、題案討論
 - 1、題案一：請樹德科大設計計算機概論課程教學大綱，並於下次會議討論。
 - (一)、說明：這是我們第一個課程大綱的設計，所以我們先全力設計此課程，待大家都瞭解課程設計的方法後，我們才能以分工的方式設計其他的課程。
 - (二)、決議：請樹德科大設計計算機概論課程教學大綱。
- 十、下次會議事項：

討論樹德科大設計之計算機概論課程教學大綱設計。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第八次會議開會通知及簽到表

- 一、日期：八月二十一日（星期四）
- 二、時間：上午 10:00-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論樹德科大設計之計算機概論課程教學大綱設計
- 八、準備資料
 - 1、請樹德科大準備設計之計算機概論課程教學大綱設計內容，於當天開會時提出討論。
- 九、簽到：

鍾乾癸

吳志宏

丁國順

張正源

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第八次會議開會會議紀錄

一、日期：日期：八月二十一日（星期四）

二、時間：上午 10:00-12:00

三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）

四、主席：鍾乾癸教授

五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）

六、記錄：丁國順（吳鳳技院資管系助理教授）

七、開會主題：

1、討論樹德科大設計之計算機概論課程教學大綱設計

八、主席報告：

1、計概課程教學大綱設計上，沒有什麼問題。但在實作的內容上及時數上需要再研究。

2、我們花了比較多的時間在計概課程教學大綱設計上，就是希望經過大家的討論後能找出設計課程的最好方法。之後，我們可以依照這些方式，大家分工將其他課程設計出來。

九、題案討論

1、題案一：請中國技院負責資料結構課程及吳鳳技院負責資料庫課程教學大綱設計

（一）、說明：我們花了比較多的時間在計概課程教學大綱設計上，爲了加快計畫執行的速度，請中國技院負責資料結構課程課程教學大綱設計，吳鳳技院負責資料庫課程教學大綱設計。

（二）、決議：請兩校各負責此課程設計。

十、下次會議事項：

討論樹德科大設計之計算機概論課程教學大綱、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第九次會議開會通知及簽到表

- 一、日期：九月四日（星期四）
- 二、時間：中午 12:00-14:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論樹德科大設計之計算機概論課程教學大綱、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。
- 八、準備資料
 - 1、請上述學校準備設計之課程教學大綱設計內容，於當天開會時提出討論。
- 九、簽到：

陳信良 鍾乾癸

吳光閔 張正源

丁國順 吳志宏

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第九次會議開會會議紀錄

- 一、日期：日期：九月四日（星期四）
- 二、時間：中午 12:00-14:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論樹德科大設計之計算機概論課程教學大綱、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。
- 八、主席報告：
 - 1、樹德科大之計算機概論課程教學大綱內容設計非常好，這部分可以結束。
 - 2、中國技院之資料結構論課程教學大綱內容設計需要再修改，尤其是實作部分並沒有設計出來，另外對於內容的順序及時數上有做修改之必要。
 - 3、吳鳳技院之資料庫課程教學大綱內容設計太偏重資料庫理論，這對技職體系學生來說，比較不適用。另外，對於為何使用資料庫及時機並沒有設計。這部分應該加入。
- 九、題案討論
 - 1、題案一：請中國技院及吳鳳技院繼續負責相關負責課程教學大綱設計
 - (一)、說明：由於兩校設計的教學大綱內容不盡理想，所以請兩校再設計，下次會議提出。
 - (二)、決議：請兩校設計後，下次會議提出。
- 十、下次會議事項：
 - 1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十次會議開會通知及簽到表

- 一、日期：九月九日（星期二）
- 二、時間：上午 9:30-12:00
- 三、地點：台北市徐州路五號十八樓（中央聯合辦公大樓第二會議室）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
研商技專校院資訊管理系定位與課程教學相關事宜。
- 八、簽到：

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十次會議開會會議紀錄

- 一、日期：日期：九月九日（星期二）
- 二、時間：上午 09:30-12:00
- 三、地點：台北市徐州路五號十八樓（中央聯合辦公大樓第二會議室）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
研商技專校院資訊管理系定位與課程教學相關事宜
- 八、主席報告：
略。
- 九、下次會議事項：
 - 1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十一次會議開會通知及簽到表

- 一、日期：九月十八日（星期四）
- 二、時間：中午 9:30-11:30
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。
- 八、準備資料
 - 1、請上述學校準備設計之課程教學大綱設計內容，於當天開會時提出討論。
- 九、簽到：

鍾乾癸

吳志宏
丁國順

張正源

吳水淇

09-26987311
chienju.chang@msa.hinet.net

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十一次會議開會會議紀錄

一、日期：日期：九月十八日（星期四）

二、時間：中午 9:30-11:30

三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）

四、主席：鍾乾癸教授

五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管系主任）、張正源（中國技院資管系主任）

六、記錄：丁國順（吳鳳技院資管系助理教授）

七、開會主題：

1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。

八、主席報告：

1、吳鳳技院之資料庫課程教學大綱內容的順序需做調整，尤其是課程的連貫性。應該利用前面教過的內容來說明上課的主題，學生比較容易接受。此外對於檔案系統與資料庫的差異部分還是沒有設計出來，同時也要設計實作，讓學生瞭解它們之間的差異。

2、中國技院之資料結構教學大綱內容中，演算法的部分太多也太深了。資料結構是需要提到演算法，但應該著重於實用性，有些部分是研究所考試用，不該放入課程中。另外，我們在計概課程中有教物件導向的觀念，是不是在這裡可以將資料結構以物件導向的方式呈現。

九、題案討論

1、題案一：請中國技院及吳鳳技院繼續負責相關課程教學大綱設計

（一）、說明：兩校設計課程大綱仍有部分需做修改。

（二）、決議：請兩校設計後，下次會議提出。

2、題案二：到學校內與授課老師舉行討論

（一）、說明：到目前為止，我們已經設計出三門課程的教學大綱，對於教學大綱內容是否合於學生能力，需與老師舉行討論，因此下次會議希望能在學校內舉行。

（二）、決議：到樹德科技大學資訊管理系開會，並請吳教授幫忙。

十、下次會議事項：

中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十二次會議開會通知及簽到表

- 一、日期：十月四日（星期六）
- 二、時間：上午 8:30-12:00
- 三、地點：高雄縣燕巢鄉橫山村橫山路 59 號（樹德科技大學 資訊管理系）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。
- 八、準備資料
 - 1、請上述學校準備設計之課程教學大綱設計內容，於當天開會時提出討論。
- 九、簽到：

鍾乾癸

丁國順

吳光閔

吳志宏

張正源

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十二次會議開會會議紀錄

- 一、日期：十月四日（星期六）
- 二、時間：上午 8:30-12:00
- 三、地點：高雄縣燕巢鄉橫山村橫山路 59 號（樹德科技大學 資訊管理系）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。
- 八、主席報告：
 - 1、中國技院之資料結構課程教學大綱設計在課程順序上需再做調整，另外在時數上需考慮實作部分，似乎不夠。
 - 2、吳鳳技院之資料庫課程教學大綱設計在順序上也需要做調整，此外，應該著重資料庫應用的 domain 知識，不要放太多理論的內容。
- 九、題案討論
 - 1、題案一：請中國技院及吳鳳技院繼續負責相關負責課程教學大綱設計
 - (一)、說明：兩校設計課程大綱仍有部分需做修改。
 - (二)、決議：請兩校設計後，下次會議提出。
 - 2、題案二：討論期中報告內容
 - (一)、說明：本計畫已經進行了將近四個月，需要交給技職司一份期中報告。
 - (二)、決議：下次會議在交通大學召開，請吳鳳技術學院準備內容。
- 十、下次會議事項：
 - 1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。
 - 2、討論期中報告內容。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十三次會議開會通知及簽到表

- 一、日期：十月十日（星期五）
- 二、時間：上午 9:30-12:00
- 三、地點：新竹市大學路 1001 號工三館四樓（國立交通大學 資訊工程系）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。
 - 2、討論期中報告內容。
- 八、準備資料
 - 1、請上述學校準備設計之課程教學大綱設計內容，於當天開會時提出討論。
 - 2、請各學校準備期中報告資料，於當天開會時提出討論。
- 九、簽到：

鍾乾癸

吳志宏 丁國順

吳光閔

張正源

陳信良

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十三次會議開會會議紀錄

- 一、日期：十月十日（星期五）
- 二、時間：上午 9:30-12:00
- 三、地點：新竹市大學路 1001 號工三館四樓（國立交通大學 資訊管理系）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、中國技院之資料結構與吳鳳技院資料庫等課程教學大綱設計。
 - 2、討論期中報告內容。
- 八、主席報告：
 - 1、中國技院之資料結構課程教學大綱設計大致上已經可以，需做細部的調整。
 - 2、吳鳳技院之資料庫課程教學大綱設計大致上也可以，但在時數上需做調整以合乎當初規劃的時數。
 - 3、期中報告內容應將目前設計的課程教學大綱放入。
- 九、題案討論
 - 1、題案一：設計尚未討論課程之教學大綱設計
 - (一)、說明：目前我們尚有「軟體開發」、「程式語言」、與「XML 語言」等課程尚未設計，所以各校負責。
 - (二)、決議：樹德科大負責「軟體開發」課程，南華大學負責「程式語言」，吳鳳技院負責「XML 語言」。
- 十、下次會議事項：
 - 1、討論尚未討論課程之教學大綱設計。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十四次會議開會通知及簽到表

- 一、日期：十月三十日（星期四）
- 二、時間：上午 9:00-11:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論尚未討論課程之教學大綱設計。
- 八、準備資料
 - 1、準備需要設計之課程名稱，於當天開會時提出討論。
- 九、簽到：

鍾乾癸

丁國順

吳光閔

張正源

吳志宏

陳信良

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十四次會議開會會議紀錄

- 一、日期：十月三十日（星期四）
- 二、時間：上午 9:00-11:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論尚未討論課程之教學大綱設計。
- 八、主席報告：
 - 1、XML 課程教學大綱設計中，因當初分配的時數並不多，所以是不是把內容分為三個部分，其中基礎知識在分配的時數中教完，其他的部分就由學校自行決定選擇。
 - 2、軟體發展課程教學大綱設計中，應有物件導向的部分，所以本課程應在計概課程之後，另外，本課程應著重於目前企業所需要的程式文件、品質文件製作上。
 - 3、程式語言課程教學大綱設計上，是不是能以功能演進的方式來介紹。請樹德科大的吳教授也能幫忙協助，看看如何設計較好。
- 九、題案討論
 - 1、題案一：繼續 XML 課程、程式語言課程之教學大綱設計
 - (一)、說明：XML 課程、程式語言課程之教學大綱仍有修改必要。
 - (二)、決議：請吳鳳技院、南華大學繼續設計。
- 十、下次會議事項：
 - 1、討論 XML 課程、程式語言課程之教學大綱設計。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十五次會議開會通知及簽到表

- 一、日期：十一月八日（星期六）
- 二、時間：上午 10:00-12:00
- 三、地點：高雄縣燕巢鄉橫山村橫山路 59 號（樹德科技大學 資訊管理系）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論 XML 課程、程式語言課程之教學大綱設計。
- 八、準備資料
 - 1、準備 XML 課程、程式語言課程之教學大綱，於當天開會時提出討論。
- 九、簽到：

丁國順

吳光閔

鍾乾癸

吳志宏

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十五次會議開會會議紀錄

- 一、日期：十一月八日（星期六）
- 二、時間：上午 10:00-12:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論 XML 課程、程式語言課程之教學大綱設計。
- 八、主席報告：
 - 1、XML 課程的內容設計是沒有問題，但各單元之間的銜接部分需要再修改。應當讓學生瞭解前一單元的觀念之後，才能介紹下一個單元的內容。實作的設計也很重要。另外，可選擇的部分要考慮整個整體課程的先後順序，如果有其他課程無法編入，又與 XML 有關的內容，是不是可以放進來。
 - 2、程式語言課程內容，應當以介紹各種語言的優缺點為主，讓學生瞭解某一種語言的發明是因為之前的語言無法解決某些問題而誕生，這樣可以增加學生對程式語言在系統開發上的使用有所瞭解。
 - 3、對於之前在教育部所開期中報告會議中，與會人員所提之問題及意見，如果需要修改課程內容，請各負責人將課程內容作修改。
- 九、題案討論
 - 1、題案一：舉辦一天到兩天 workshop，希望能廣徵其他學校老師對於我們課程設計的意見。
 - (一)、說明：本計畫所設計的課程大致都已經完成，爲了能廣徵其他學校老師對於我們課程設計的意見，是不是可以舉辦一個 workshop。
 - (二)、決議：由吳鳳技術學院負責規劃，於下次會議提出。
- 十、下次會議事項：
 - 1、討論 XML 及程式語言課程之教學大綱設計。
 - 2、討論期中報告修改的項目。
 - 3、討論 workshop 舉行及工作分配。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十六次會議開會通知及簽到表

- 一、日期：十二月四日（星期四）
- 二、時間：上午 8:30-10:30
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論 XML 及程式語言課程之教學大綱設計。
 - 2、討論期中報告修改的項目。
 - 3、討論 workshop 舉行及工作分配。
- 八、準備資料
 - 1、準備需要設計之課程名稱，於當天開會時提出討論。
- 九、簽到：

鍾乾癸

丁國順 陳信良

張正源 吳志宏 吳光閔

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十六次會議開會會議紀錄

- 一、日期：十二月四日（星期四）
- 二、時間：上午 8:30-10:30
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論 XML 及程式語言課程之教學大綱設計。
 - 2、討論期中報告修改的項目。
 - 3、討論 workshop 舉行及工作分配。
- 八、主席報告：
 - 1、XML 語言課程之教學大綱的設計可以算完成了。
 - 2、程式語言課程之教學大綱內容，以語言的特性來分類是可行的，如果能加上時間上的分類就更好了。至於實作的設計，這可能有困難。
 - 3、workshop 的舉行地點以北部比較方便，而時間最好選擇寒假或期末考週。
- 九、題案討論
 無
- 十、下次會議事項：
 - 1、討論程式語言課程之教學大綱設計。
 - 2、討論 workshop 舉行及工作分配。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十七次會議開會通知及簽到表

- 一、日期：十二月十八日（星期四）
- 二、時間：上午 8:30-10:30
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論程式語言課程之教學大綱設計。
 - 2、討論 workshop 舉行及工作分配。
- 八、準備資料
 - 1、準備需要設計之課程名稱，於當天開會時提出討論。
- 九、簽到：

吳光閔 張正源
丁國順
吳志宏
鍾乾癸

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十七次會議開會會議紀錄

- 一、日期：十二月十八日（星期四）
- 二、時間：上午 8:30-10:30
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論程式語言課程之教學大綱設計。
 - 2、討論 workshop 舉行及工作分配。
- 八、主席報告：
 - 1、本計畫將接近尾聲，為了將使本計畫所研究的成果更能切合實際，我們將舉辦 workshop，邀請各校相關科系系主任、老師參與。這部分請吳鳳技術學院與教育部技職司聯絡，是不是由教育部技職司發出通知，邀請各校相關科系系主任、老師參加。
 - 2、程式語言課程之教學大綱設計內容可以算完成了。
 - 3、對於本計畫整體課程間的順序，是不是可以用圖形表示。
- 九、題案討論
 - 1、題案一：設計本計畫整體課程間的架構圖
 - (一)、說明：為了讓別人瞭解本計畫整體課程間的關係，應設計一個先後關係架構圖。
 - (二)、決議：由吳鳳技術學院負責。
 - 2、題案二：討論 workshop 舉行及工作分配
 - (一)、說明：本計畫將接近尾聲，為了將使本計畫所研究的成果更能切合實際，我們將舉辦 workshop，邀請各校相關科系系主任、老師參加。
 - (二)、決議：
 - (1)、本會議將在長庚大學舉行。
 - (2)、舉辦時間為九十三年一月十三日。
 - (3)、開會時間為一天。
- 十、下次會議事項：

討論 workshop 準備的情況。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十九次會議開會通知及簽到表

- 一、日期：三月五日（星期五）
- 二、時間：上午 9:00-11:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論本計畫結案及工作分配。
- 八、準備資料
 - 1、準備相關資料，於當天開會時提出討論。
- 九、簽到：

陳信良
丁國順
吳光閔
吳志宏
鍾乾癸

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第十九次會議開會會議紀錄

- 一、日期：三月五日（星期五）
- 二、時間：上午 9:00-11:00
- 三、地點：台北市和平東路二段科技大樓五樓（行政院科技顧問組）
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論本計畫結案及工作分配。
- 八、主席報告：
 - 1、對於上次 Workshop 會議上，有些主任提出了一些意見，請各課程設計負責人能依其意見在課程上做部分調整。
 - 2、爲了瞭解本課程大綱設計的內容是否適當，是不是可以將「行政院勞委會乙級程式設計師技能檢定考試科目」內容與本計畫課程內容做比較，以瞭解本計畫所規劃的內容是否合乎外界需要。
 - 3、本計畫要接近尾聲，我們應將所有研究的成果收集做成結案報告，並設計結案報告投影片。
- 九、題案討論
 - 1、題案一：設計「行政院勞委會乙級程式設計師技能檢定考試科目之對照表」
 - (一)、說明：爲了瞭解本課程大綱設計的內容是否適當，有沒有需要再修改。
 - (二)、決議：請南華大學負責設計
 - 2、題案二：設計結案報告投影片
 - (一)、說明：爲了本計畫結案報告用。
 - (二)、決議：請樹德科大負責設計
- 十、下次會議事項：
 - 1、討論本計畫結案工作及文件整理。

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第二十次會議開會通知及簽到表

- 一、日期：三月二十六日（星期五）至三月二十七日（星期六）
- 二、時間：三月二十六日下午 2:00 至三月二十七日下午 6:00
- 三、地點：阿里山賓館
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題及題案討論
 - 1、討論本計畫結案工作及文件整理。
- 八、準備資料
 - 1、準備相關資料，於當天開會時提出討論。
- 九、簽到：

鍾乾癸

吳志宏

陳信良

丁國順

吳光閔

技專校院資訊相關系科課程教學與產業需求配合檢討之研究會議

第二十次會議開會會議紀錄

- 一、日期：三月二十六日（星期五）至三月二十七日（星期六）
- 二、時間：三月二十六日下午 6:00 至三月二十七日 下午 6:00
- 三、地點：阿里山賓館
- 四、主席：鍾乾癸教授
- 五、出席：陳信良（吳鳳技院資管系主任）、吳光閔（南華大學資管系主任）、吳志宏（樹德科大資管教授）、張正源（中國技院資管系主任）
- 六、記錄：丁國順（吳鳳技院資管系助理教授）
- 七、開會主題：
 - 1、討論本計畫結案工作及文件整理。
- 八、主席報告：
 - 1、南華大學所提供的「與勞委會乙級程式設計師技能檢定考試科目之對照表」設計的不錯，相對的也讓我們知道檢定考試科目內容與本計畫課程設計內容的差異，對於檢定內容有的課程內容而本計畫課程沒有的部分，是不是請各負責的人在相關的課程上再做修改。
 - 2、樹德科大所設計的期末報告投影片在內容上沒有什麼問題，只有小部分要修改。另外，是不是設計一份精簡版的投影片，以備報告時間較少時使用。
 - 3、請吳鳳技院將本計畫規劃之課程教學大綱、課程設計理念及相關文件收集整理，做成本計畫之結案報告。
- 九、題案討論
無
- 十、下次會議事項：