

檔 號：  
保存年限：

## 國立臺灣大學 函

地址：10617 臺北市羅斯福路4  
段1號

聯絡人：黃芝鈞

電話：(02)33664888#305

傳真：02-2362-8167

電子郵件

：alison.tien@csie.ntu.edu.tw

受文者：國立暨南國際大學

發文日期：中華民國102年12月16日

發文字號：校電資字第1020100510號

速別：普通件

密等及解密條件或保密期限：

附件：軟體工程聯盟推廣到校協同教學申請須知-2013-12 version (1020100510-0-0.doc，共1個電子檔案)

主旨：本校資訊工程系執行教育部資訊及科技教育司資訊軟體人才培育計畫，於每學期辦理「軟體工程」到校協同教學，今將辦理持續推廣，供欲申請之學校於103年1月10日前提出申請，請查照。

說明：

- 一、「軟體工程」到校協同教學每學年提供38門課程供大專院校申請；目前於上學期完成29門協同教學課程，今將辦理持續推廣，開放9門課程於下學期進行，供欲申請之學校於103年1月10日前備妥申請表暨開課課程大綱提出申請。
- 二、申請辦法請詳閱附件、至<http://www.sec.org.tw/> 下載申請表或<http://www.itsa.org.tw> 作相關資訊連結。
- 三、請於申請期限內將電子檔寄至e-mail box: alison.tien@csie.ntu.edu.tw，信件標題請註明「協同教學持續推廣申請案-申請學校之名稱」。
- 四、申請截止後將召開審查小組，依所提課程大綱與申請課程模組之關聯性與適切性作評分，取排名前9個課程模組申請案。審查結果將於103年1月31日以前以e-mail回覆。
- 五、通過之協同教學請於103年3月8日以前完成，以利後續計畫相關費用核銷與結案。



正本：公私立大專校院

副本：教育部資訊及科技教育司、成功大學資工系數位生活科技研發中心（劉育均）

、資訊工程學系、資訊工程學系教授李允中

102/12/17

09:48:55

## 校長楊泮池

擬辦：

- 一、本案擬掃瞄 Email 通識教育中心及各系所卓參，並請各教學單位轉知所屬專兼任教師如需申請台大資工系「軟體工程」到校協同教學，請依本文之說明辦理，並於 103 年 1 月 10 日前提出申請。
- 二、上傳文件公告系統週知。
- 三、文陳閱後存查。

組員雷意蘭

資訊工程學系  
課務組組長 陳小芬  
1/18

秘書室 宋守中  
專門委員

教授兼  
教務長 江大樹  
1/18

教授兼  
主任秘書 孫同文  
1/17

資訊工程學系  
教授 蘇玉龍  
102/12/17

教育部資訊及科技教育司資訊軟體人才培育計畫-「軟體工程推廣」

到校協同教學申請須知

軟體工程聯盟召集人：李允中 教授

國立臺灣大學資訊工程系

聯絡人：黃芝鈞([alisontien@csie.ntu.edu.tw](mailto:alisontien@csie.ntu.edu.tw), 0233664888 ext. 305)



## 教育部資訊及科技教育司資訊軟體人才培育計畫-「軟體工程推廣」

### 到校協同教學申請須知

#### 一、緣由

為有效提升國內軟體工程教育的素質，增加臺灣軟體工業的產值，教育部資科司於民國九十一年七月委託國立中央大學規劃國內軟體工程課程改進計畫，並據此更新計畫於民國九十三年四月成立軟體工程聯盟，期使國內各大專院校的軟體工程教學資源得以有效的整合，互相交流，以提高軟體工程教育的質與量，培育產業界所需的相關人才。過去幾年軟體工程教育在軟體工程聯盟推動下，臺灣軟體工程教育的普及率有顯著的成長。

近年來軟體工程聯盟的推廣重心從軟體工程教育「量」的提升轉換到「質」的深化。為此，軟體工程聯盟提供軟體工程到校教學服務，協同講授軟體相關課程的教師將根據課程需求在課程中融入軟體工程重要的元素，讓學生可以及早體驗軟體開發過程中軟體品質與團隊合作等重要性以及實務工具的使用。

#### 二、申請說明

1. 各大專院校教師提出申請，每一系所限申請 1 次每次 1~3 個軟體工程課程模組，每一課程模組授課時數 6 小時。
2. 請參考課程模組教學大綱(附件四)於申請表(附件一)上勾選符合課程之軟工課程模組，將依據所提上課時間搭配到校教學教師時間後擇出 1~3 個課程模組。
3. 提出申請時須檢附開課課程與申請之軟工課程模組整合規劃(附件二)。
4. 申請軟體工程到校輔導教學之班級規模不得少於 20 人，開課系級不得為在職專班。
5. 到校教學教師鐘點費、差旅費及教材印製費由教育部補助。
6. 到校協同授課期間開課教師須全程參與。

### 三、配合事項

#### 1. 教室環境

- (1) 前三個小時的理論課程，以教室為佳。
- (2) 後三個小時的實務課程，需在電腦教室上課，電腦需具有上網之功能，上課前需把相關軟體安裝完成。且應避免授課地點為到校協同教學教師無法走至學員面前之教室。
- (3) 教室內需備有投影機、筆記型電腦。(視各授課教師需求而定)

#### 2. 課前準備

- (1) 於開始授課前二個星期將學員人數、年級、校區地圖、教室地圖 email 給承辦單位。

#### 3. 學員-每一次上課

- (1) 爭取發言，參與討論。
- (2) 實際操作實務課程中所使用的軟體。
- (3) 填寫課程問卷調查(請參見附件三)。
- (4) 寫下修習本課程後的收穫心得。

#### 4. 主場教師-學期前準備

- (1) 到校協同教學如何融入原課程中(包含課程整合、設備支援與評量方式等)。
- (2) 軟體工程課程模組與原課程之關聯性。

#### 5. 主場教師-每一次上課

- (1) 完成到校協同教學前所有準備工作
- (2) 引導學員聽課和參與討論。
- (3) 協同學員在實務課程中操作所需的軟體工具。
- (4) 寫下到校協同教學對原課程的助益。

## 6. 授課時間表

進行項目	時間(分)	累計時間(分)	負責人	備註
引言	5	5	主場教師	一般教室授課
理論教學(一)	45	50	協同教學教師	
休息	10	60		
理論教學(二)	50	110	協同教學教師	
休息	10	120		
理論教學(三)	30	150	協同教學教師	
檢討、Q&A	20	170	主場教師	
引言	5	5	主場教師	電腦教室授課
實務教學(一)	45	50	協同教學教師	
休息	10	60		
實務教學(二)	50	110	協同教學教師	
休息	10	120		
實務教學(三)	30	150	協同教學教師	
檢討、Q&A	20	170	主場教師	

## 四、申請方式

1. 申請時間：103 年 1 月 10 日(五)前提出申請。
2. 請於申請期限內將電子檔寄至 E-mail：alison.tien@csie.ntu.edu.tw，信件標題請註明申請軟體工程協同教學。
3. 報名截止後將召開審查小組，依據所提課程大綱與申請課程模組之關聯性與適切性評分，取排名前 9 個課程模組申請案。
4. 審核結果將於 103 年 1 月 31 日(五)前以 E-mail 回覆。

## 五、聯絡資訊

教育部資科司軟體工程聯盟

聯絡人：黃芝鈞

TEL：(02)3366-4888 #305 FAX：(02)2362-8167

E-mail：alisontien@csie.ntu.edu.tw

教育部資科司軟體工程聯盟：<http://www.sec.org.tw>

教育部資訊及科技教育司資訊軟體人才培育計畫-「軟體工程推廣」  
到校協同教學申請表

學校名稱			
課程名稱			
開課系級	系所		年級
開課時間	星期_____ : _____ ~ _____ : _____	修課人數	
開課教師		職稱	
E-mail			
聯絡電話	( )	傳真	( )
聯絡人姓名		職稱	
E-mail			
聯絡電話	( )	傳真	( )
聯絡地址	□□□□□		
協同教學 可上課日期	_____年_____月_____日、_____年_____月_____日、_____年_____月_____日 _____年_____月_____日、_____年_____月_____日、_____年_____月_____日 授課日期請至少填寫三個日期(從 103/2/1 至 103/3/7)，並請填寫周六的日期。(實際上課時間日後請開課教師與種子教師協調)		
勾選	軟體工程課程模組名稱	審核結果(承辦單位填寫)	
<input type="checkbox"/>	「專案管理」實務與工具課程		
<input type="checkbox"/>	「Code Review」實務與工具課程		
<input type="checkbox"/>	「軟體測試」實務與工具課程		
<input type="checkbox"/>	「Bug Tracking」實務與工具課程		
<input type="checkbox"/>	「資料結構」軟體工程輔助課程		
<input type="checkbox"/>	「物件導向程式設計」軟體工程輔助課程		

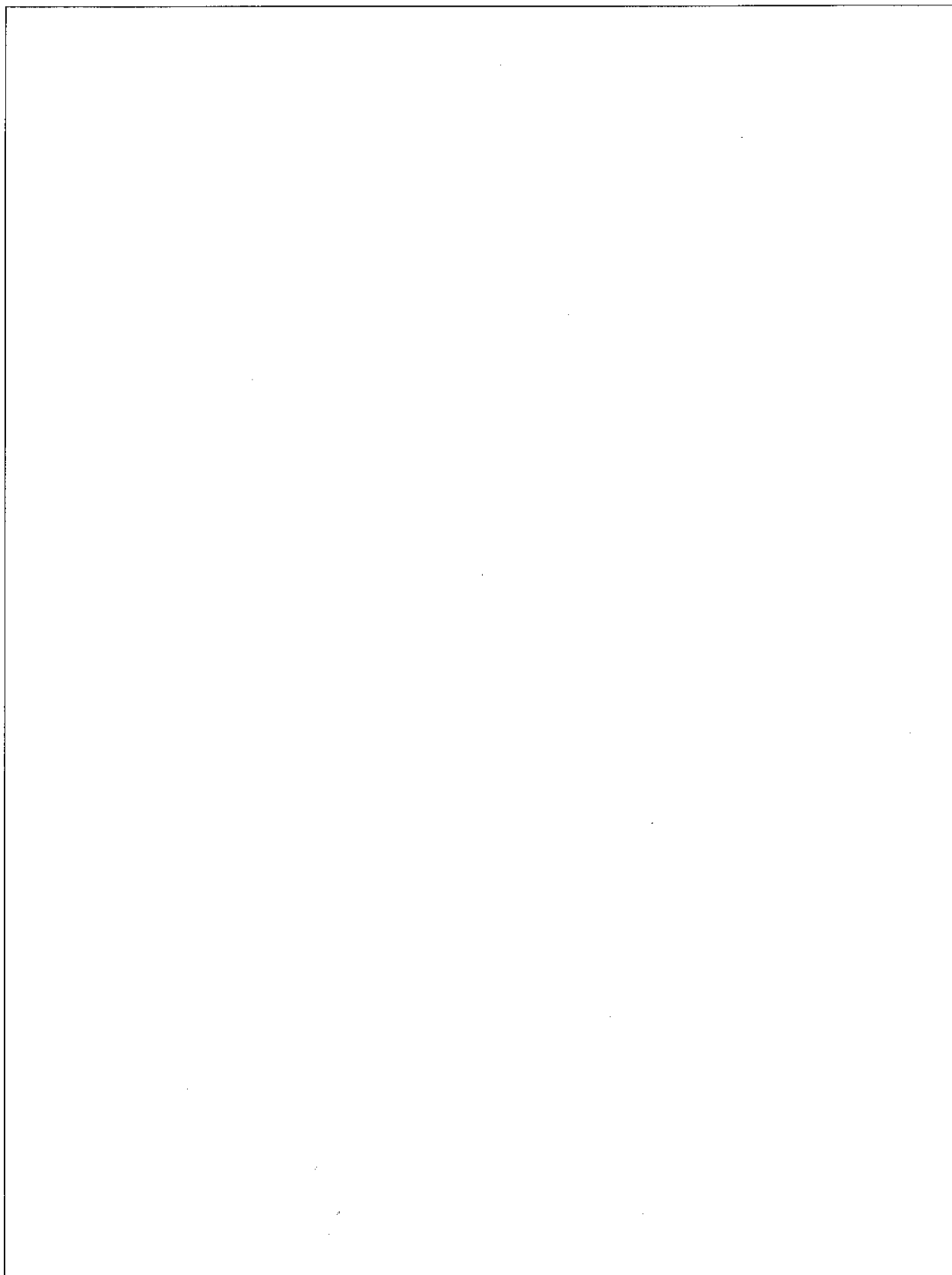
申請教師：\_\_\_\_\_ 開課系所主管：\_\_\_\_\_

\_\_\_\_\_年\_\_\_\_\_月\_\_\_\_\_日



3. 課程大綱：







課後問卷分析

課程名稱：

上課地點：

講 師：

日 期：

分類	項目	給分	改善建議事項
講師部份	專業知識		
	表達能力		
	學習氣氛/ 節奏掌控		
	問題回答		
課程部份	教材內容		
	進行方式		
	時數安排		
	學習期望		
行政	訓練場地		
	座位安排		
課程滿意度最高分為5分,最低分為1分			
5：非常滿意,4：滿意,3：尚可,2：不滿意,1：非常不滿意			

## 「專案管理」實務與工具課程

### 主題

「專案管理」實務與工具課程

### 教學目標

許多軟體組織在即時交付高品質的軟體系統時，往往遭遇許多困難。大部分的組織或團隊成員都相當有才華，而且他們幾乎全部都知道系統開發有了問題。這些問題的解決事實上在世界上數以千計的軟體組織中已經發現、詮釋、測試過，答案通常是直接而且容易落實的。然而，對於不了解專案管理的人卻不是那麼直覺易懂，因而很難導入正確的方案。本課程期望能將這些解決方案介紹給學員，幫助他們將這些方案與其組織整合，以解決軟體開發遭遇的相關問題。

本教學目標主要為結合專案管理的方法論與產業界的實務經驗和開放原始碼之專案管理工具以進行專案管理實務與工具教學。此包含專案管理方法、作業、工具介紹與單元規劃範例等，在課程中教導專案管理的實務經驗及開放原始碼工具的運用。藉此，學生可學習到專案管理的知識，同時藉由作業、練習與工具使用學習並認識專案管理的理論與實務操作，讓學生能掌握專案分工、專案預估、專案計畫、與專案監控等能力。

### 授課教師

「專案管理」種子教師團隊

主持人：葉道明教授/國立高雄師範大學軟體工程學系

### 教材

(1) 「專案管理」實務與工具教材與投影片

### 課程大綱

專案管理實務與工具(Project Management Practices and Tools)

Chapter 1: 專案管理概念(Project Management Concept)

1.1 專案定義(Project Definition)

1.2 專案管理生命週期(Project Management Life Cycle)

1.3 發展專案計畫(Project Plan Development)

Chapter 2: 專案範圍(Project Scope)

2.1 專案初始(Project Initiation)

2.2 工作分解結構(The Work Breakdown Structure)

Chapter 3: 專案預估(Project Estimation)

3.1 軟體大小預估(Size Estimation)

3.2 工時預估(Duration Estimation)

3.3 經費預估(Cost Estimation)

Chapter 4: 專案時間排程(Project Scheduling)

4.1 甘特圖(Gantt Chart)

4.2 專案網路圖(Project Network Diagram)

4.3 要徑分析法(Critical Path Method)

Chapter 5: 風險管理(Risk Management)

5.1 風險分析(Risk Analysis)

5.2 風險控管(Risk Control)

Chapter 6: 專案監控(Project Monitoring and Control)

6.1 專案監督(Project Monitoring)

6.2 專案控制(Project Control)

Chapter 7: 專案管理工具與應用(Project Management Tool and Application)

7.1 專案管理工具介紹(Open Source Project Management Tools)

7.2 應用專案管理工具(Applying Project Management Tools)

# 「Code Review」實務與工具課程

## 主題

「Code Review」實務與工具課程

## 教學目標

品質是軟體是否合乎其規格且能滿足使用者需求的關鍵因素，因此如何有效提升軟體的品質一直是開發人員努力的目標。而根據研究指出，缺陷(defect)在軟體開發過程中停留越久，排除該錯誤的成本越高，在測試階段修正錯誤的成本將近是在需求階段排除這些錯誤的 15~40 倍，因此若能即早發現和修正軟體的缺陷將可大幅減少提升軟體品質所需的成本。

本課程的目標主要是介紹各種形式同儕審查的方法與實務，以及程式碼審查相關的自由軟體工具，並設計程式碼審查的個案實習，以培育學員具備同儕審查的知識與能力，能善用同儕審查的工具，進行程式碼(或其他工作產出)的審查，以有效改善軟體的品質。透過此次教學，讓教授程式設計或需要實作程式之課程的教師，在不影響原授課內容下有機會納入程式碼審查的概念，讓學生熟悉程式碼審查的實務與工具，訓練學生具有程式碼審查的能力，以改善軟體的品質。

## 授課教師

「Code Review」種子教師團隊

主持人：劉建宏助理教授/國立臺北科技大學資訊工程學系

## 教材

(2) 「Code Review」實務與工具教材與投影片

## 課程大綱

### Chapter 1: Introduction to Peer Reviews

- 1.1 The Cost of Quality
- 1.2 Overviews of Peer Reviews
- 1.3 Reviews and Team Culture
- 1.4 Peer Reviews Formality Spectrum
  - 1.4.1 Software Inspections
  - 1.4.2 Team Review
  - 1.4.3 Walkthrough
  - 1.4.4 Pair Programming

- 1.4.5 Peer Deskcheck
- 1.4.6 Ad Hoc Review
- 1.5 Guiding Principles for Peer Reviews

## Chapter 2: Software Inspection

- 2.1 The Inspection Process
- 2.2 Planning the Inspection
- 2.3 Preparing the Inspection
- 2.4 The Inspection Meeting
- 2.5 Rework and Follow-Up

## Chapter 3: Inspection Data Collection and Analysis

- 3.1 Why Collect Inspection Data?
- 3.2 Inspection Metrics
- 3.3 Inspection Data Analysis
- 3.4 Measuring the Impact of Inspections

## Chapter 4: Implementing Peer Review Programs

- 4.1 The Peer Review Organization
- 4.2 Peer Review Process Assets
- 4.3 Peer Review Training
- 4.4 Guidelines and Practices for Peer Reviews

## Chapter 5: Open Source Tools for Peer Reviews

- 5.1 Jupiter
- 5.2 Rietveld
- 5.3 CodeStriker
- 5.4 Splint
- 5.5 PMD

## Chapter 6: Class Workshop on Code Review

- 6.1 Introduction to the case study
- 6.2 Responsibilities
- 6.3 Input
- 6.4 Entry Criteria
- 6.5 Procedures
- 6.6 Exit Criteria
- 6.7 Output
- 6.8 Data Collection
- 6.9 Supporting Tools and Inspection Forms
- 6.10 Code Review Workshop

## 「軟體測試」實務與工具課程

### 主題

「軟體測試」實務與工具課程

### 教學目標

目前大專院校資工、電機、通訊、資管等資通訊相關系所，甚至機械、財務等其他領域系所，藉由軟體系統的開發，以驗證其發展的理論方法。學生透過資訊化方式呈現重要的研究成果，甚至進行商業化經營。在軟體系統開發過程中，可能因為人為疏失、溝通不良而造成規格不符、設計錯誤，進而造成程式撰寫疏漏、錯誤等問題，如此將嚴重影響軟體的品質，不僅延誤系統開發時程，更導致軟體開發成本的增加。

為了能找出各種因素造成的軟體錯誤、瑕疵與損害，需要經由軟體驗證與確認 (Verification and Validation) 以保證軟體系統的品質。亦即使用軟體測試方法以系統化的步驟檢驗軟體系統，發現潛藏的錯誤，以期提升軟體的品質。本課程將闡述軟體測試技術的基本概念，以規劃、設計與執行等工程步驟來介紹軟體測試的相關知識，使學員在學習軟體測試的技術知識時，能夠同時獲得工程化思維方式的訓練。此外，也會探討有效的軟體測試作業程序，以提升軟體開發專案的品質能力；並透過開放原始碼工具的介紹與實作範例的練習，讓學員實際學習到軟體測試的方法與如何應用測試工具進行軟體測試，以完整結合測試的理論與實務。

### 授課教師

「軟體測試」種子教師團隊

主持人：郭忠義副教授/國立臺北科技大學資訊工程學系

### 教材

「軟體測試」實務與工具教材與投影片

### 課程大綱

- (1) 軟體測試的基本觀念：測試目標、測試 V 模型。
- (2) 軟體測試的實務作業程序：測試規劃、準備、執行、監控。
- (3) 軟體測試文件：軟體測試計畫書、測試案例、測試記錄與報告。
- (4) 測試案例設計之技術：白箱測試技術、黑箱測試技術。
- (5) 各種軟體測試的類別：單元測試、整合測試、系統與驗收測試。

軟體測試實務案例：JUnit、Jmeter 工具介紹，以及實物案例。

# 「Bug Tracking」實務與工具課程

## 主題

「Bug Tracking」實務與工具課程

## 教學目標

在資通訊相關系所，如資工、電機、通訊、資管等，甚至其他領域的系所，軟體開發是這些系所學生未來的主要出路。然而在軟體發展過程中，可能因為程式人員疏失、設計錯誤，等等諸多的原因造成程式無法正常的執行。這些錯誤一般俗稱為臭蟲(bug)。研究顯示，程式人員花費在解決臭蟲的時間可能達開發流程三分之一的時間，甚至更多。從臭蟲的發現到最終臭蟲的獲得解決，管理與監控這些流程的技術稱之為 bug tracking。

Bug tracking 在實務上都藉由專業的系統來進行，這類系統也是軟體開發必備的軟體工程輔助工具之一。廣義來說 Bug Tracking 系統也是一種 Issue tracking 系統。理論上，大部分的 Issue Tracking 系統都可以用來做為 bug tracking。近年來拜開放原始碼以及 web 技術的進展之賜，許多 bug tracking 系統都已經可以輕鬆的使用，取得與安裝。不過，學生在學校的求學階段當中，普遍缺乏使用這類工具的經驗與知識。另外，工具是死的，如何正確地使用它來確實獲得軟體工程上的好處，也需要完整的教育與訓練。

本課程將闡述 bug tracking 的基本概念，使學員在學習軟體工程的技術知識時，能夠同時獲得工程化思維方式的訓練。此外，我們也會探討必要的 bug tracking 作業程序，以提升軟體開發專案過程中的流程管理；並透過開放原始碼工具的介紹與實作範例的練習，讓學員實際學習到 bug tracking 的實務應用，以完整結合測試的理論與實務。

## 授課教師

「Bug Tracking」種子教師團隊

主持人：鄭永斌副教授/國立中央大學資訊工程學系

## 教材

「Bug Tracking」實務與工具教材與投影片

## 課程大綱

- (1) Bug tracking 的基本觀念
  - 軟體測試基本概念介紹
  - 軟體錯誤的種類
  - Bug tracking 的目標
  - Bug 的生命週期
- (2) Bug tracking 的實務作業程序
  - 如何建立一個 Bug

- Bug 的分級
- Bug 的追蹤與管理
- Internal/public bug report
- 使用 Bug tracking 的重要準則

(3) Bug 的統計與軟體品質

- Bug report
- Bug 趨勢資料
- Bug density

# 「資料結構」軟體工程輔助課程

## 主題

- 「資料結構」軟體工程輔助課程

## 教學目標

資料結構課程在於資通訊相關系所，如資工、電機、通訊、資管等，不只是普遍的課程，更是一門重要的課程。大學部的學生，通常在修過程式語言的課程之後，就被安排修習資料結構。課程內容設計通常注重於，學生對不同問題所需的資料設計與演算法的瞭解能力。如果有配合程式的作業，就能讓學生學習進階的程式開發能力。因此，對於大學軟體人才培育，這是一門關鍵的課程。可是，以往學生沒有軟體工程訓練，導致程式開發沒有頭緒，通常以土法煉鋼模式將程式寫出，無法有效的提高程式開發能力。如能讓學生學習軟體開發所需的分析、設計、測試等能力，將能影響學生往後軟體系統開發的技術，以及對軟體工程價值的認同。

本課程的目標主要是使教授資料結構課程的教師，在授課內容中有機會納入軟體工程概念，以及進階程式創作的機會。藉此，學生不只是學習到資料結構的知識，同時藉由實做學習並認識軟體工程的工作程序與理念，讓學生能掌握軟體系統分析與設計、測試、版本管理、分工等能力。

## 授課教師

「資料結構」種子教師團隊

主持人：劉立頌教授/國立中正大學電機工程學系

## 教材

- 「資料結構」軟體工程輔助課程教材與投影片

## 課程大綱

課程規劃以模組與單元的方式呈現

模組	單元	內容
軟體工程觀念		
	分析與設計	問題瞭解與需求分析，演算法於設計中的呈現
	測試	程式的驗收
	版本管理	單獨程式的整理與團隊開發內容的

		管理
	個人能力管理	Time-log 為主的能力分析
軟體工程工具		
	測試工具	JUnit、JUnit4
	版本管理工具	SVN
作業範例		
	四種個別作業	Lists, stacks, queues, trees, sorting 相關作業

# 「物件導向程式設計」軟體工程輔助課程

## 主題

「物件導向程式設計」軟體工程輔助課程

## 教學目標

隨著各種 3C 產品(如智慧型手機、i-PAD 等)與網路技術的快速演進，各種資通訊應用有如雨後春筍，不斷推陳出新，而這些應用的開發與其開發平台本身都是以「物件導向程式」為基礎的，因此，物件導向程式設計已成為資通訊相關系所(如資工、電機、通訊、資管等)學生所必須具備的基本能力。目前，國內大學部的課程規劃，通常在修過程式語言(例如 C 語言)課程之後，就安排學生修習物件導向程式設計(使用 C++、C#、Java 等語言)。

物件導向程式設計課程內容通常著重於物件導向的基本觀念(如類別、封裝、繼承、多型等)，配合程式的作業，培養學生撰寫物件導向程式的能力。但是，隨著應用軟體的規模越來越大，學生往往缺乏軟體工程訓練，導致程式開發時沒有適當的規劃，僅以土法煉鋼模式寫出程式，無法發揮物件導向程式的特質，以至於寫出來的軟體缺乏彈性，無法修改、無法維護，甚至牽一髮而動全身，往往為了修改一點點功能，就必須整個重寫。因此，如果能讓學生學習正確的物件導向模型方法，在程式中引進良好的設計，將影響學生軟體系統開發的能力，以及對軟體工程價值的認同。

開發應用軟體時，若能採用適當的「設計樣式(design pattern)」做為架構規劃的基礎，將能大幅地提升其彈性與可維護性，本計畫擬將設計樣式做為物件導向程式設計課程的輔助教材，藉由作業與範例提供優質的軟體開發訓練。授課教師不需修改物件導向程式設計課程的內容，而是納入本計畫的產出，讓學生能在大學初期對架構的設計(而不僅僅是程式設計)有一些認識，做為往後正式修習軟體工程課程時的準備。

## 授課教師

「物件導向程式設計」種子教師團隊

主持人：陳偉凱副教授/國立臺北科技大學資訊工程學系

## 教材

「物件導向程式設計」軟體工程輔助課程教材與投影片

## 課程大綱

課程規劃以模組與單元的方式呈現：

模組	單元	內容
設計樣式觀念	變更點(variation point)	瞭解應用軟體中的變更點，也就是不穩定的地方，容易變更的地方，程式設計時應該對變更點做適當的規劃，以減少變更所引起的衝擊。
	設計樣式範例	設計樣式簡介，並詳細介紹一個設計樣式(如 Strategy、State)做為範例，以及使用此設計樣式能克服哪些變更所引起的問題。
	UML	介紹 UML 類別圖(class diagram)的用法，並使用 UML 類別圖說明設計樣式範例。
作業範例	兩個作業	將設計樣式應用在 sorting，利用物件導向程式多型的能力，解決變更所引起的問題。