

教育部資通訊軟體創新人才推升計畫-「軟體品質推廣」
軟體工程翻轉式協同實務演練教學申請須知

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：錢怡君 小姐 E-mail: joycecheng@ntu.edu.tw 電話: 0233664888 ext. 305

教育部資通訊軟體創新人才計畫-「軟體品質推廣」

軟體工程翻轉式協同實務演練教學申請辦法

一、緣由

為有效提昇國內軟體工程教育的素質，增加台灣軟體工業的產值，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，台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，而過去幾年軟體工程聯盟提供的到校協同教學服務深受到各大專院校的肯定，本年度亦將持續推廣，但為使課程更容易重複應用以利推廣，此次將採線上課程搭配翻轉式教學的方式來推廣，讓學生於課前預習，在課程中則以實務演練為主，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 1~2 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)，並於申請表(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 1~2 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，並需配合填寫課後問卷。

三、配合事項

1. 教室環境

- (1) 課程進行採實務演練方式進行，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老師無法走至學員面前之教室。
- (2) 教室內需備有投影機、筆記型電腦。(視各授課老師需求而定)

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。
- (2) 協同教師會先將線上課程網址給予主場老師，請主場老師協助學生於協同教學課程前先對課程進行影片及講義的閱讀，完成課前的預習指引。

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. 主場老師-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. 主場老師-每一次上課

- (1) 完成到校協同教學前所有準備工作
- (2) 引導學員於課前練習並於課堂上進行實務演練並參與討論。
- (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. 翻轉式協同實務演練教學上課方式

課程主場老師應與協同教學老師配合，進行課前的預習指引，協同教學老師到校後將進行習題的演練。

- A. 課前：學生於課前必須先對課程進行影片及講義的閱讀，在透過簡單的問題回答（選擇或是非題）了解自己對問題的瞭解程度。學生在正式上課前應準備 1-2 個問題。
- B. 課中：協同教學老師上課時會先用 5-10 分鐘講解單元重點，接著讓學生提問。提問活動後開始進行習題演練或問題討論，讓學生能夠應用基本的知識進行討論。可能會採取的做法如下：
- a. 分組討論，2-5 同學為一組進行討論，並上臺報告。
 - b. 上機實作，針對某些程式問題或工具進行實際操作。
 - c. 辯論，針對某一主題進行辯論討論。
- C. 課後：開課單位需配合填寫課後問卷，以了解學生對於翻轉式教學的效果。

7. 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	主場老師	電腦教室授課
單元重點講解 (一)	10	15	協同教學老師	
學生演練	35	50		
休息	10	60		
單元重點講解 (二)	10	70	協同教學老師	
學生演練	40	110		
休息	10	120		
單元重點講解 (三)	10	130	協同教學老師	
學生演練	20	150		
檢討、Q&A	20	170	主場老師	
引言	5	5	主場老師	
單元重點講解(四)	10	15	協同教學老師	

學生演練	35	50		
休息	10	60		
單元重點講解(五)	10	70	協同教學老師	
學生演練	40	110		
休息	10	120		
單元重點講解(六)	10	130	協同教學老師	
學生演練	20	150		
檢討、Q&A	20	170	主場老師	

四、申請方式

1. 申請時間：105 年 8 月 30 日(二)前提出申請。
2. 請於申請期限內將電子檔寄至 E-mail：joycecheng@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會議，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。本年度總計將提供 40 個課次，每門課接受 10 門以上之課程申請。
4. 審核結果將於 105 年 9 月 12 日(一)前以 E-mail 回覆。

五、聯絡資訊

國立臺灣大學資訊工程學系

聯絡人：錢怡君 小姐

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

E-mail：joycecheng@ntu.edu.tw

軟體工程聯盟網站：<http://www.sec.org.tw>

教育部資通訊軟體創新人才計畫-「軟體品質推廣」
軟體工程翻轉式協同實務演練教學申請表

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 105/10/1 至 105/12/31），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)	
☐	「軟體設計」課程		
☐	「軟體測試」課程		
☐	「安全軟體設計」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	線上課程		
	教材內容		
	進行方式 (翻轉式教學方式)		
	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			

「軟體設計」課程 2016 年規劃 (第二年)

主題

「軟體設計」課程 2016 年規劃(第二年)

教學目標

軟體設計是軟體工程相當重要的一環，有良好的軟體設計才能有良好的分工、良好的軟體品質。本課程透過翻轉式教學方式建立同學良好的物件導向軟體設計觀念，並能運用軟體設計原則、樣式與流程，建立具品質的軟體系統。本年度課程介紹基本的物件導向設計基本觀念與物件導向的設計原則，學生學習此二課程主題後，日後將可繼續深入學習各類物件導向設計樣式，與物件導向軟體重構方法。

學生應具備的基本能力

學生應具有物件導向程式設計(OOP)基本概念

授課教師

「軟體設計」種子教師團隊

教材

- (1) 「軟體樣式 I」教學影片
- (2) 「軟體樣式 I」自我練習題庫
- (3) 「軟體樣式 I」教學投影片
- (4) 「軟體樣式 I」實習教材

課程大綱

1. 設計樣式介紹 (影片預習)
 - (1) 軟體設計原則
 - (2) 軟體設計
 - (3) 設計樣式介紹
 - (4) 設計樣式分類
 - (5) UML 類別模組介紹與操作 (實機練習 2h)
2. 策略設計樣式 (Strategy design pattern) (影片預習)
 - (1) 策略設計樣式的動機與目的
 - (2) 策略設計樣式的結構
 - (3) 策略設計樣式的應用

- (4) 策略設計樣式的樣板程式碼
- (5) 範例與練習 (實機練習 1h)
- 3. 觀察者設計樣式 (Observer design pattern) (影片預習)
 - (1) 觀察者設計樣式的動機與目的
 - (2) 觀察者設計樣式的結構
 - (3) 觀察者設計樣式的應用
 - (4) 觀察者設計樣式的樣板程式碼
 - (5) 範例與練習 (實機練習 1h)
- 4. 工廠方法設計樣式 (Factory Method design pattern) (影片預習)
 - (1) 工廠方法設計樣式的動機與目的
 - (2) 工廠方法設計樣式的結構
 - (3) 工廠方法設計樣式的應用
 - (4) 工廠方法設計樣式的樣板程式碼
 - (5) 範例與練習 (實機練習 1h)
- 5. 抽象工廠設計樣式 (Abstract factory design pattern) (影片預習)
 - (1) 抽象工廠設計樣式的動機與目的
 - (2) 抽象工廠設計樣式的結構
 - (3) 抽象工廠設計樣式的應用
 - (4) 抽象工廠設計樣式的樣板程式碼
 - (5) 範例與練習 (實機練習 1h)
- 6. 適配器設計樣式 (Adaptor design pattern) (影片預習)
 - (1) 抽象工廠設計樣式的動機與目的
 - (2) 抽象工廠設計樣式的結構
 - (3) 抽象工廠設計樣式的應用
 - (4) 抽象工廠設計樣式的樣板程式碼
 - (5) 範例與練習 (Homework)

「軟體測試」 2016 年規劃 (第二年)

主題

「軟體測試」課程 2016 年規劃(第二年)

教學目標

資訊軟體系統在開發過程中，可能因為人為疏失、溝通不良而造成規格不符、設計錯誤，進而引發程式撰寫疏漏錯誤等問題，如此將嚴重影響軟體的品質，不僅增加資訊系統開發成本與時程，更導致資訊系統的錯誤與失效。為確保軟體系統品質符合使用者的需求，軟體必須透過各種類型的測試，以偵測和移除軟體可能的錯誤。本年度課程介紹軟體技術中的單元測試、如何運用白箱和黑箱測試技術來設計單元測試案例、以及利用測試覆蓋率來評估測試是否足夠。本課程將讓同學針對程式設計課程中所撰寫的程式碼，練習使用白箱和黑箱測試技術設計測試案例，並透過單元測試工具自動化執行測試，以及使用和測試覆蓋率工具查看所設計的測試案例對程式碼的覆蓋程度，結合測試的理論與實務，訓練學員有效的單元測試技巧，以提升軟體測試的能力。

學生應具備的基本能力

學生應具備基礎軟體程式撰寫能力 C/C++/Java/C#其中一種。

授課教師

「軟體測試」種子教師團隊

教材

- (1) 「軟體測試」教學影片。
- (2) 「軟體測試」自我練習題庫。
- (3) 「軟體測試」教學投影片。
- (4) 「軟體測試」實習教材。

課程大綱

1. 軟體測試的基本觀念：軟體測試目標、軟體測試 V 模型、軟體測試種類。
2. 單元測試方法：單元測試技巧與 JUnit 工具運用。
3. 測試案例設計方法：白箱測試技術、黑箱測試技術。
4. 測試覆蓋率：程式碼測試覆蓋率與 Emma 工具運用。

「安全軟體設計」課程 2016 年規劃 (第一年)

主題

「安全軟體設計」課程 2016 年規劃(第一年)

教學目標

由於現今軟體已佔據人們日常生活極為重要的一部分，安全的軟體設計將是未來軟體開發的一大議題與挑戰，透過於開發時期導入安全的軟體設計概念與技術將是改善軟體安全的重要手段，也是軟體工程重要的一環。本課程透過翻轉式教學方式建立同學良好的軟體安全設計觀念，並能運用軟體安全設計原則、程式碼與流程，建立具安全特性的軟體系統。本年度課程介紹基本的軟體安全程式設計概念，並以 Java 以及 C 兩種程式語言作為範例，學習如何進行安全程式撰寫物，學生學習此二課程主題後，日後將可繼續深入學習其他相關安全程式議題與方法。

學生應具備的基本能力

學生應具有物件導向程式設計(OOP)基本概念。

授課教師

「安全軟體設計」種子教師團隊

教材

- (1) 「安全軟體設計」教學影片
- (2) 「安全軟體設計」自我練習題庫
- (3) 「安全軟體設計」教學投影片
- (4) 「安全軟體設計」實習教材

課程大綱

1. 軟體安全程式設計概論
 - (1) Introduction to secure programming
 - (2) MITRE common weakness enumeration (CWE)
 - (3) SEI CERT coding standard
2. C 安全程式設計
 - (1) Introduction to buffer overflow
 - (2) Stack overflow
 - (3) Array index error

- (4) Format string bugs
- (5) Prevention of buffer overflow
- 3. Java 安全程式設計
 - (1) Never hard code sensitive information
 - (2) Do not use public static nonfinal fields
 - (3) Do not use a null in a case where an object is required
 - (4) Do not ignore values returned by methods
 - (5) Prevent SQL injection
 - (6) Normalize strings before validating them
 - (7) Limit accessibility of fields