

國立臺灣大學課程綱要格式

課程資訊			
課程名稱	軟體開發方法		
課程編號	725 U3340	班次	學分數 3
全/半年	半年	必/選修	選修
授課教師	蔡益坤	開課系所	資管所
上課時間	星期四 上午 9:10--12:10	上課地點	管院一館 101
備註	授課教師研究室：管院二館 1108；電話：3366-1189		
課程網頁	http://im.ntu.edu.tw/~tsay/courses/sdm/		
課程大綱			
為確保您的權利，請尊重智慧財產權及不得非法影印			
課程目標	使學生熟悉軟體開發實務上常用的一些方法及工具，並藉此接觸正規軟體驗證的基礎知識，以為未來的軟體開發事業做好準備。		
課程概述	本課程引介一些能提升學生開發正確、高品質軟體能力的理論、實務方法及工具。我們將以工程師的角度來思考，因此主要課程重點是軟體開發流程中的技術層面。在大致了解課程主題及正規邏輯與程式正確性的觀念後，我們將仔細研習 UML、設計樣式以及基本的正規軟體驗證的方法與工具。		
關鍵字	設計樣式、程式正確性、軟體驗證、UML		
課程要求	先修課程：物件導向程式設計及離散數學。 本課程包括一次期中考試、數次作業、及一個專題計畫。在專題計畫部分，我們預定將同學們每八到十人分成一組；我們期待同學們發揮良好的團隊精神。		
Office Hours	星期三下午 1:30--2:30 或另行約定		
指定閱讀	Class Notes and Selected Readings (available on the course Web site)		
參考書目	<ol style="list-style-type: none"> 1. <i>Logic for Computer Science</i>, J.H. Gallier, Harper & Row Publishers, 1985. 2. <i>The UML Resource Page</i>: http://www.uml.org/, OMG. 3. <i>The Unified Modeling Language User Guide, 2nd Edition</i>, G. Booch, I. Jacobson, and J. Rumbaugh, Addison-Wesley, 2005. 4. <i>Design Patterns: Elements of Reusable Object-Oriented Software</i>, E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Addison-Wesley, 1995. 5. <i>The SPIN Model Checker: Primer and Reference Manual</i>, G. J. Holzman, Addison-Wesley, 2003. 		

	<p>6. <i>Spin - Formal Verification Page</i>: http://spinroot.com/.</p> <p>7. <i>Temporal Verification of Reactive Systems: Safety</i>, Z. Manna and A. Pnueli, Springer-Verlag, 1995.</p> <p>8. <i>The Formal Methods Page</i>: http://vl.fmnet.info/, J. Bowen.</p>																
評量方式	<table border="1"> <thead> <tr> <th>No.</th> <th>項目</th> <th>百分比</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>期中考</td> <td>40%</td> <td></td> </tr> <tr> <td>2.</td> <td>作業</td> <td>20%</td> <td></td> </tr> <tr> <td>3.</td> <td>專題</td> <td>40%</td> <td></td> </tr> </tbody> </table>	No.	項目	百分比	說明	1.	期中考	40%		2.	作業	20%		3.	專題	40%	
No.	項目	百分比	說明														
1.	期中考	40%															
2.	作業	20%															
3.	專題	40%															
週次	單元主題																
第 1 週	Introduction: overview of software requirements, development process, design methods, and testing/verification tools																
第 2 週	Formal Logic: propositional logic, first-order logic																
第 3 週	Program Correctness: axiomatic semantics of programs, partial and total correctness																
第 4 週	UML - I: introduction, basics of modeling, overview of the UML, structural modeling																
第 5 週	UML - I: behavioral modeling, architectural modeling																
第 6 週	Design Patterns: introduction, creational patterns																
第 7 週	Design Patterns: structural patterns																
第 8 週	Design Patterns: behavioral patterns																
第 9 週	Design Patterns: architecture review, design document wrap-up, advanced patterns, refactoring																
第 10 週	Invited Industrial Talk																
第 11 週	UML - II: advanced behavioral modeling																
第 12 週	UML - II: Object Constraint Language																
第 13 週	Program Verification Tools: JML, Spec#																
第 14 週	Midterm																
第 15 週	Formal Modeling Tools: Alloy																
第 16 週	Software Model Checking: linear-time model checking																
第 17 週	Software Model Checking: the Spin model checker																
第 18 週	Term Project Presentations																