


國立臺灣大學資訊工程學系教學大綱表

					適用學期	96 下
課程名稱	中文：物件導向程式設計				授課教師	陳俊良
	英文：Object-Oriented Software Design				課 號	902 10730
學 分 數	3	必/選修	必修	開課年級	大一	班 次 01, 02 班
課程關係	 <pre> graph LR A[計算機程式] --> B[物件導向程式設計] B --> C[資訊系統原理] B --> D[軟體設計模式] </pre>					
課程概述	<p>如何成為一個程式設計師 (programmer)? 第一、要懂語言 (language); 第二、要懂程式 (program)。語言指的是語法、功能等等。比如, C 語言 if 敘述的條件要用小刮號刮起來, 就是語法。C 語言的 * 和 & 有很多效果, 就是功能。程式指的是程式設計師用語言表達出來的軟體。如何成為一個好的程式設計師? 除了上述之外, 還要會構思軟體。</p> <p>早先的程式語言以及軟體構思方法大概都是程序導向 (procedural-oriented) 的, 比如 C 語言。後期才有物件導向 (object-oriented) 的概念。比較新的語言, 比如 Java 和 C#, 都是歸屬於物件導向。過去的經驗證明物件導向的觀念與作法易於開發維護大型軟體。</p> <p>物件導向的概念並不是說非得用物件導向語言來表達。假如我們喜歡, 是用物件導向的概念來構思一個軟體系統, 然後用程序導向的語言實作出來, 只不過將會很累。那麼物件導向語言有何特點? 其特點為, 當我們利用物件導向語言來撰寫程式時, 容易表達出一些大家認同的物件導向大概念, 比如抽象 (abstraction)、封裝 (encapsulation)、繼承 (inheritance)、多型 (polymorphism) 等。當我們利用物件導向語言來表達程式的意圖時, 因為物件導向語言的文法等規範, 會驅使我們寫出隱含物件導向概念的軟體。</p> <p>本課程目的在於讓修課同學理解物件導向的概念, 熟悉物件導向語言, 以建立將來撰寫大型物件導向程式的基礎。</p>					
課程目標	<p>本課程的目標在於讓修課同學：</p> <ul style="list-style-type: none"> ● 理解物件導向的概念, 比如抽象、封裝、繼承、多型等。 ● 熟悉一個物件導向語言的語法和功能。本課程以 Java 語言為主, C# 語言為輔。 ● 有能力利用物件導向語言表達出物件導向的概念。比如利用 Java 語言的關鍵字 extends 表達出繼承的概念。 ● 有能力遵循物件導向概念構思一個小規模的程式, 進而有能力利用物件導向語言實作該程式。 ● 有能力翻閱物件導向語言的 API, 進而有能力引用 API 來協助物件導向程式的開發。 					
課 程 大 綱						
單 元 主 題	內 容 綱 要				預 估 週 數	備 註
Introduction	<ul style="list-style-type: none"> ● HelloWorld.java ● execution of a Java program ● language vs. program 				1	

		<ul style="list-style-type: none"> ● procedural-oriented vs. object-oriented 		
Status of object		<ul style="list-style-type: none"> ● concept of abstraction ● class vs. instance ● static variable ● instance variable ● instance creation expression ● reference data type 	1.5	
Behavior of object		<ul style="list-style-type: none"> ● local variable ● static method ● instance method ● constructor ● scope of class member ● method overloading 	1.5	
Object-oriented program		<ul style="list-style-type: none"> ● program vs. process/thread ● main method ● package & import statements ● frequently used API's 	1	
Encapsulation		<ul style="list-style-type: none"> ● concept of encapsulation ● access modifier ● setter & getter ● C# property & indexer 	1	
Inheritance		<ul style="list-style-type: none"> ● concept of inheritance ● variable hiding ● static method hiding ● instance method overriding 	2	
Polymorphism		<ul style="list-style-type: none"> ● polymorphism of object ● polymorphism of reference variable ● type checking mechanism ● virtual method invocation 	2	
Interface		<ul style="list-style-type: none"> ● interface declaration & implementation ● interface type variable ● why interface ● multiple interface inheritance 	1	
Generics		<ul style="list-style-type: none"> ● concept of generics ● generic class ● generic method 	1	
Exception		<ul style="list-style-type: none"> ● concept of exception ● issue of exception ● exception handling 	1	
Multi-thread		<ul style="list-style-type: none"> ● thread creation ● competitive synchronization ● cooperative synchronization 	1	
教學 要點 概述	教材編選	教科書：無		
		參考書：K. Sierra and B. Bates, <i>Head First Java</i> , 2nd Ed., O'Reilly, 2005. K. Arnold, J. Gosling and D. Holmes, <i>The Java Programming Language</i> , 4th Ed., Addison-Wesley, 2006.		

	課程網頁： https://ceiba.ntu.edu.tw/962oop
	<input checked="" type="checkbox"/> 自行製作 <input type="checkbox"/> 教科書商提供 <input type="checkbox"/> 其它：
教學方法	<input checked="" type="checkbox"/> 投影片 <input type="checkbox"/> 板書 <input type="checkbox"/> 實習/實驗 <input checked="" type="checkbox"/> 其它：程式實作
評量方式	<input checked="" type="checkbox"/> 考試 <input checked="" type="checkbox"/> 作業 <input type="checkbox"/> 報告 <input type="checkbox"/> 其它：
教學資源	<input checked="" type="checkbox"/> 單槍投影機 <input type="checkbox"/> 一般投影機 <input type="checkbox"/> 白板 <input type="checkbox"/> 黑板 <input checked="" type="checkbox"/> 一般教室 <input type="checkbox"/> 實驗室 <input type="checkbox"/> 其它：
其他事項	