

國立雲林科技大學
資訊管理系碩士班

碩士論文

一個支援多語言的程式設計學習環境
A Programming Learning Environment for
Multiple Languages

研究生： 林彥宏

指導教授： 董少桓 博士

中華民國九十八年六月

一個支援多語言的程式設計學習環境

A Programming Learning Environment for
Multiple Languages

研究生：林彥宏

指導教授：董少桓博士

國立雲林科技大學

資訊管理系碩士班

碩士論文

A Thesis

Submitted to

Institute of Information Management

National Yunlin University of Science & Technology

in Partial Fulfillment of the Requirements

for the Degree of

Master of Business Administration

in

Information Management

June 2009

Doulin, Yunlin, Taiwan, Republic of China

中華民國九十八年六月

一個支援多語言的程式設計學習環境

學生：林彥宏

指導教授：董少桓 博士

國立雲林科技大學資訊管理系碩士班

摘 要

軟體技術日新月異，需要培訓更多資訊人才，才能滿足科技發展的需求。然而，程式設計的教學並不容易，需要讓學生有大量練習的機會，但是傳統的學習環境並不容易提供，因為次數不多的作業及測驗，就足夠讓教師及學生忙碌，想要將教材的每個範例，甚至教材以外的題目動手實際練習，幾乎是不可能達到的要求。程式設計的教學需要更有效的方法，讓學生進入「從做中學」的學習環境，透過實際完成程式碼的撰寫，才能真正瞭解教材提供的學習內容。

建構一個有效的學習環境，需要教學平台的功能與教材製作機制的搭配。本研究的目的是改善程式設計學習環境，設計一個支援多種程式語言的教學平台，讓多數的程式語言，都能藉由這個平台進行教學。此外，教材的設計需要有效率，並且容易針對各種不同需求，讓教材的作者自行定義、擴充，因此也需要一個新的教材架構與製作方法。

本研究所完成的成果，是一個學習環境的實作，包含網頁架構的教學平台，方便使用者用瀏覽器進行操作，以及能夠滿足教學需求的整合性教材開發環境，讓作者在程式設計專用的編輯器中，以有效率的方式開發程式設計教材，並提供學生使用相同的編輯器進行練習，學習的過程由系統予以記錄，並彙整成為教師授課的參考資訊。這個學習環境已經在許多學校的課程中實際使用，以協助教師解決教學實務中所遇到的問題。

關鍵字：程式測試、數位教材、程式語言教學、程式語言學習環境

A Programming Learning Environment for Multiple Languages

Student: Yan-Hong Lin

Advisor: Dr. Shao-Huan Tung

Institute of Information Management
National Yunlin University of Science & Technology

ABSTRACT

Software technology changes very quickly, industries require training more IT professionals for the needs of scientific and technological development. However, the design and implementation of programming learning environments is not easy. Such systems need to provide students more opportunities to practice. Inadequate homework and exam can easily make the learning curve unacceptable for both teachers and students. As a result, it is almost impossible to teach each subject by practicing exercises. In order to learn effectively, programming learning environments must provide tools to guide students to learn through the "learning by doing" approach. That is actually writing runnable code to learn how to program.

An effective learning environment requires two functionalities: producing exercises and practicing exercises. The purpose of this study is to implement a programming learning environment that supports multiple programming languages so that most of the programming languages can be taught by this platform. In order to improve the efficiency of producing exercises, a new structure and editing method for teaching materials need to be implemented.

The result of this study is an implementation of programming learning environment that supports web-based practicing as well as an integrated development environment for producing programming exercises. This environment provides authors with a suitable program editor for creating exercises in an efficient way. This editor is also used to allow students to write program code. The coding process is recorded by the system and is provided for teaching as needed. This learning environment has been used in many universities to help teachers teach programming courses.

Keywords: Program Testing, Digital Material, Programming Language Tutoring, Programming Learning Environment

誌 謝

由衷感謝辛苦指導我完成研究論文的董少桓老師，以及感謝擔任口試的許中川、林迺衛、施學琦老師。因為有老師們的指教與幫忙，改進研究的缺失，才能讓本論文更加完整。

特別感謝董少桓老師的指導，在大學時因董老師的程式語言課程，讓我有機會更深入瞭解程式語言的奧妙。也在因緣際會之下，董老師擔任我大學專題、研究所論文的指導教授，因此有更多時間能夠完成一項研究，有更多的改進與檢討，得以累積多一點的成果。董老師除了指導我的學業，在品格教育及思想啟發，也給我許多幫助，讓我有更多的信心面對困難與挑戰。

感謝我的家人，使我衣食無缺，能夠專心課業學習，並支持我選擇的人生方向。感謝許多陪我度過這幾年的朋友，因為有你們的情誼及扶持，讓我的求學生活更加快樂圓滿。

目 錄

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
表目錄	vii
圖目錄	viii
一、緒論	1
1.1 研究背景	1
1.2 研究動機	2
1.2.1 支援多種程式語言	2
1.2.2 內容產生機制	3
1.3 研究目的	4
1.4 研究限制	4
二、文獻探討	6
2.1 Programming Learning Environment	6
2.1.1 Intelligent Tutoring System	6
2.1.2 Standalone Learning Environment	7
2.1.3 Web-based Learning Environment	7
2.2 Learning Objects	8
2.2.1 SCORM	8
2.2.2 XML 的二進位資料編碼	9
2.3 執行程序的自動化	10
2.3.1 Make	10
2.3.2 Ant	11
2.4 程式測試工具	11
2.4.1 Unit Testing	12
2.4.2 Web Testing	12
2.4.3 GUI Testing	12
2.5 整合開發工具	13
三、系統設計	14

3.1 教學環境設計	14
3.2 教學平台架構設計	15
3.3 教材格式設計	17
3.4 教材開發流程設計	20
3.5 測試案例設計	21
四、系統實作	23
4.1 資料庫與網站功能實作	23
4.2 教材格式實作	24
4.3 編輯器實作	29
五、教材案例實作	31
5.1 Java 程式設計教學	31
5.1.1 軟體安裝與環境設定	31
5.1.2 教學案例描述	31
5.1.3 教材實作	32
5.1.4 測試案例	35
5.2 PHP 網頁程式設計教學	37
5.2.1 軟體安裝與環境設定	37
5.2.2 教學案例描述	38
5.2.3 教材實作	38
5.2.4 測試案例	40
5.3 C#視窗介面程式設計	40
5.3.1 軟體安裝與環境設定	41
5.3.2 教學案例描述	41
5.3.3 教材實作	42
5.3.4 測試案例	44
六、使用狀況	46
6.1 程式語言教學課程	46
6.2 程式設計能力測驗	47
七、結論與未來研究	48
7.1 結論	48
7.2 未來研究	48
參考文獻	50

附錄一、JAVA 教材設計附檔	52
GCD.html	52
GCD.java	52
GCD.part	54
GCDTest.java	56
GCDTest.xml	57
附錄二、PHP 教材設計附檔	59
form1_doc.html	59
form1.html	59
form1.php	60
form1_test.pl	61
附錄三、C#教材設計附檔	65
ex01.html	65
ex01.build	65
Program.cs	66
MainForm.cs	67
MainForm.Designer.cs	68
MainForm.Designer.part	75

表 目 錄

表 3-1 學習紀錄的主要資料類型.....	16
表 3-2 製作不同程式語言教材可能存在之差異.....	20
表 4-1 主要資料表用途說明.....	23
表 4-2 主要網站功能用途說明.....	24
表 4-3<project>標籤定義與說明.....	25
表 4-4<property>標籤定義與說明.....	25
表 4-5<file>標籤定義與說明.....	25
表 4-6<task>標籤定義與說明.....	26
表 4-7<command>標籤定義與說明.....	27
表 4-8 常用系統屬性說明.....	28
表 5-1Java 與 Abbot 相關系統環境變數設定.....	31
表 5-2 Java 教學案例的屬性設定.....	33
表 5-3 Java 教學案例的指令設定.....	33
表 5-4 測試 Java 教學案例的屬性設定.....	35
表 5-5 測試 Java 教學案例的指令設定.....	36
表 5-6 LighTPD 設定說明.....	37
表 5-7 PHP 教材的屬性設計.....	39
表 5-8 測試 PHP 教材的屬性設定.....	40
表 5-9 測試 PHP 教材的指令設定.....	40
表 5-10 C#教材的屬性設定.....	43
表 5-11 C#教材的指令設定.....	43
表 5-12 C#教材的指令設定.....	45
表 6-1 教學平台課程數量統計.....	47
表 6-2 測驗數量統計.....	47

圖 目 錄

圖 3-1 教學環境實體關係圖.....	14
圖 3-2 教學平台使用案例.....	15
圖 3-3 教材、學習紀錄與使用者角色之間的關係.....	17
圖 3-4 教材單元的 Project 組成元素	19
圖 3-5 教材開發流程圖.....	21
圖 4-1 教材格式 XML 範例資料.....	24
圖 4-2 原始版本的 jEdit 編輯器畫面	29
圖 4-3 啟動程式畫面	30
圖 4-4 學習環境的編輯器畫面	30
圖 5-1 Java 教學案例描述示意圖.....	32
圖 5-2 教材編輯器的 Task Editor 功能	34
圖 5-3 JAVA 教材的編輯器畫面.....	35
圖 5-4 Abbot Framework 的 Test Script Editor 畫面	36
圖 5-5 Java 教材編輯、使用與測試流程.....	36
圖 5-6 PHP 教學案例描述示意圖	38
圖 5-7 PHP 教材的編輯畫面	39
圖 5-8 PHP 網頁執行畫面	39
圖 5-9 PHP 教材編輯、使用與測試流程圖	40
圖 5-10 學習環境的編輯器畫面.....	42
圖 5-11 SharpDevelop 視窗編輯的畫面	42
圖 5-12 C#教材的編輯畫面	44
圖 5-13 C#教材編輯、使用與測試流程圖	45

一、緒論

傳統的程式語言教學，教師較難以掌握學生的學習狀況；利用網路化的程式語言教學平台，有助於簡化教學工作，並將學生的學習過程予以紀錄，教師藉由系統功能及資訊回饋，將有更多提昇教學品質的方法可以利用。除了教學系統提供的功能之外，數位化教材內容的質與量，對於教學平台的價值也有相當程度影響。有許多程式語言教學平台的設計，缺乏完善的教材製作方法，使得教材製作過程變得困難，成為一件不容易進行的工作，適用的程式語言種類也有所限制，教學過程經常與一般撰寫程式慣用的方法與流程離異。本研究旨在發展一個支援多程式語言的學習環境，包括一個教學平台的實作、數位化程式設計教材的架構與發展工具，能夠提供一致性的教材內容製作方法，以支援多種程式語言及自動化測試方式，並且讓教材的編輯與使用機制達到整合。

1.1 研究背景

隨著我國近年來產業的改變與成長，各行業在推行資訊化、網路化的過程中，企業紛紛招募資訊人才以滿足運用科技的需求，也帶動資訊技術相關行業的快速成長。根據資策會 2008~2010 年台灣產業科技人才供需調查，數據的估計顯示“新增供給無法滿足新增需求”，並預期資訊人才缺口的情況，仍可能在未來幾年持續發生。

大學教育提供資訊管理、資訊科學、資訊工程等科系培育資訊人才，每年有數量眾多的畢業生投入就業市場，卻無法解決供需失衡的問題，資策會的調查報告指出問題成因之一為「學校教育重理論，實務性不足，與業者需求有落差」。對於社會新鮮人的就業力，學校教育扮演著舉足輕重的角色，培訓基礎資訊能力的實務課程，若能夠在教學品質獲得提升，將有機會改善人才供給問題。

在企業每年徵聘的資訊人才中，程式設計相關的工作職缺佔有一定的比例。資策會的資訊服務產業調查發現，2008 年整體產業的新聘人員中，軟體程式設計師的需求最高，大約為 22 個百分點。企業對資訊技能需求的重視，則以資訊系統開發為主流，特別是程式設計、資料庫管理及系統分析與設計均超過 60 個百分點，其中扮演基礎能力的程式設計更高達 63%為三項之首(資策會，2008 台灣產業科技

人才供需調查)。

多數企業希望找到立即可用的人才，具有程式設計的實務經驗尤佳，對於缺乏工作經驗的畢業生，實務經驗的來源即為學校課程練習及專題實作。Google 全球副總裁暨大中華區總裁李開復，就曾經在文章中給予大學生建議，應徵程式設計工作的大學畢業生，最好能夠累積十萬行的程式設計經驗，理由為「實踐性的技術需要在實踐中提高」，計算機科學終究是實踐的學問，不動手永遠也學不會(李開復，給中國學生的第四封信)。在既有學校教育的程式設計的相關課程中，若教材提供質量足夠的練習，讓學生在修業獲得學分的同時，也能有更多機會動手設計程式、解決問題，就可以為日後從業奠定良好的基礎。

1.2 研究動機

過去有許多程式語言教學的研究，包括程式語言的教學設計、應用電腦輔助教學的方法，在程式語言、軟體開發方法論或學習理論的基礎上，可以應用在學習環境設計的研究，進而成為程式設計學習環境的實作。董少桓教授於 2005 年所提出的 PLWeb 程式設計學習網(林宗德、董少桓，2006)，是用來實際解決教學問題的系統實作，歷年來陸續發展出支援 Scheme、Java、JSP 等程式設計的教學平台，改變傳統授課方式，給予學生「learning by doing」的學習方式，同時也能獲得遠距教學的好處。

利用既有的程式語言教學理論及工具，建構適用於學校課程的學習環境，可以幫助學生有更多機會練習寫程式，快速累積程式設計的實作經驗。學習環境研究的價值，需要應用在實際的教學中才能得到驗證，並以解決實務問題作為學習環境實作的目標。並透過 PLWeb 所提供的基礎架構，建置理想學習環境的方向更加明確，將 PLWeb 應用於課程教學時，讓我們能夠發現、並且探討與解決以下的問題。

1.2.1 支援多種程式語言

過去許多程式語言學習系統，僅能夠支援單一或少數類型的程式語言，若需要支援新的程式語言，通常會受到系統架構的限制而無法達成，或者必須進行相當多的系統修改。這些系統難以由終端使用者自行擴充功能，在軟體技術日新月異的時代，並不容易滿足使用需求的轉變。教學者的需求差異甚多，包含使用不

同程式語言、不同編譯器、不同作業系統以及不同程式測試方法等，甚至教學內容的呈現方式也有所不同，若因系統限制而改變教學課程設計，難免有削足適履的疑慮。若能建構支援多種程式語言的學習系統，允許使用者依照各種不同需求，自行定義及擴充新功能，那麼將有機會解決上述問題。

以 TIOBE 程式語言需求排行為例，提供良好同步 (Concurrent)、容錯 (Fault-Torrent) 能力，適合用來開發分散式系統的程式語言 Erlang，在短短兩個月內排名上升了 15 個名次 (TIOBE, Programming Community Index)，在多核心處理器普及的現今，出現使用 Erlang 語言發展系統新需求，也帶起學習熱潮。在 Web 2.0 應用逐漸熱門的時候，許多企業選用 Ruby on Rails 作為加速開發的利器，或是利用 AJAX 技術改善現有的 Web 應用程式，也帶來相關的書籍與課程。

相當可惜的是，這些新的程式語言或軟體技術需求浮現時，也已經有不少教學資源與學習需求，但教學者很難找到合適的教學系統可以利用。因為軟體技術的變化是日新月異，多數的教學系統，功能發展的速度趕不上變化，也可能不會把這些新的需求列在發展計畫中。對於發展已有時日的教學平台，基礎功能較完備，有穩固的根基；但是對於新的需求變化，無法迅速改變、適應，以滿足既有的使用者或吸引新的使用者，就侷限系統的發展、延續及應用範圍。

1.2.2 內容產生機制

多數的程式設計教學系統，皆缺乏“有效”的教材編輯機制，使得建立的教材數量受限，也不容易做出品質好的教材。由於教材格式並無法滿足程式設計的需求，必須在諸多限制之下設計教材，例如只支援填空、選擇、是非題型的教學系統，僅能比對學生填入的答案是否與標準解答相符，無法讓學生實際執行完整的程式、觀察修改後程式碼的輸出。這樣的教學系統難以達到效用，與實際撰寫程式相比，學生獲得的經驗與學習成效有限。有些教學系統雖然提供教材編輯機制，並且支援讓學生實際撰寫程式的功能，也可以提供學生透過互動的方式，從實際撰寫程式與獲得回饋，達到比較好的學習成效；但教材的製作程序往往過於複雜，例如需要撰寫大量測試案例或進行繁瑣的編輯過程，教材內容的生產缺乏效率。

許多書籍所附的程式範例光碟，以及既有教材的作業及程式碼，都有程式碼可供教學使用，但並不容易在傳統的教學系統上加以利用，原因可能是教學系統

並不提供合適的教材格式，或是沒有適當的工具可以將這些資源進行轉換。一個好的內容產生機制，將有助於教材的量產與品質提昇，教學設計者能夠將焦點內容本身的設計，而降低其他因系統限制產生的阻力，教材的質與量能夠提昇，教學系統的價值才得以彰顯。

1.3 研究目的

首要能夠解決的問題，是建構一致性且支援多程式語言的內容產生方法。用來設計 A 程式語言教材內容的方法及流程，也能應用在 B 程式語言，並且必要的調整可以在教材中完成，而不必修改系統實作。在這個架構下，儘管不同程式語言有差異之處，也能夠提供一致的教材製作方式，使得在教學系統整體不改變的前提下，僅在教材的細節部份的進行調整。這樣的方法將使系統在未來支援新程式語言時，不必面臨需要改寫或擴充平台才能支援的限制。

第二項待解決的問題，是教材編輯與使用的整合機制。學習者使用教材的流程，需要與一般程式撰寫的習慣相仿，即提供教材專屬的整合開發環境，包含適合撰寫程式碼的編輯器，以及提供單鍵執行就能獲取結果的操作功能。作者進行教材編輯的過程，也可以透過相同的開發環境，使介面與流程在教材的編輯與使用時具有共通性，並在編輯模式提供更多相關輔助功能。這種整合機制改善教學系統的教材編輯與使用方法，使程式設計的教材製作過程，能夠更貼近一般程式撰寫的習慣。

1.4 研究限制

支援多種程式語言，需要考慮程式語言所適用的平台及開發工具。多數程式語言的實作，皆有提供命令列介面(command-line interface)工具，可以藉由指令的方式進行編譯、測試等操作，因此在教材定義中提供自訂工具指令的功能，就可以直接應用不同的程式語言工具達成教學需求。這種設計能夠滿足支援多數程式語言的需求，使教材可以透過自訂指令擴充功能，但同時也帶來一些限制。

有些程式語言的教學，並非訓練學習者對語法的認識或邏輯的觀念，而是著重圖形介面開發工具的使用操作。例如 Object Pascal 通常以 Delphi 開發工具進行教學，對學習者來說，除了瞭解程式邏輯與 Object Pascal 語法，還必須熟悉 Delphi 工具的使用方法，才能夠實際以 Delphi 開發軟體。其它如 Visual Basic、AutoCAD(AutoLISP)等程式設計的教學，也具有相同的情況。使教學環境能夠支援

這些開發工具，需要相當大的客製與維護成本，並不符合效益；也有些開發工具屬於封閉式的商業軟體，不易加入擴充功能。因此，依賴圖形介面工具的程式語言，並不在本研究的範圍之內。

另一種應用受限的情況，則是某種程式語言教學所建立的程式，無法或很難以自動化的方式進行測試，使得學習結果無法判定或產生回饋資訊。例如嵌入式裝置的軟體開發，需要將建立的程式傳輸到實體裝置或模擬器，才能運作執行、得到結果，並不容易在過程之中加入測試。在網頁內嵌的 **JavaScript** 程式碼，必須在瀏覽器上執行，其過程與結果又可能是網頁畫面的變化，即記憶體中的 **DOM** 內容狀態改變，外部測試程式難以涉入其中判斷結果。其它狀況還包含，測試程式可以被建立，但過程太複雜以致於不易製作教材。因此，若程式語言實作缺乏良好的測試工具，會造成本研究應用時的限制。

二、文獻探討

本章探討發展程式設計學習環境的基礎，從過去相關研究的理論及技術知古鑑今，分析既有的學習環境與程式發展工具，找出可以整合利用的特色，以期為學習環境的發展奠定穩固的基礎。2.1 節探討程式設計學習環境的相關發展，介紹 ITS 及不同類型的 PLE。數位教材格式的設計，參考自 2.2 節介紹 SCORM 標準及 XML 的二進位資料儲存方法。在 2.3 及 2.4 節分別介紹軟體開發常用的部署及測試工具，應用在教材設計中，可以簡化學生撰寫程式的流程，並提供練習結果的自動化測試。2.5 節則列舉整合開發工具應支援的功能，並介紹在本研究中所採用的 jEdit 編輯器。

2.1 Programming Learning Environment

程式設計學習環境(PLE, Programming Learning Environment)行之有年，是學習管理系統(LMS, Learning Management System)的特定領域應用。PLE 採用電腦作為程式語言學習的輔助工具，發揮電腦化資訊系統的優勢，減輕教學者的負擔，也可以為學習者提供更多互動、回饋功能。

2.1.1 Intelligent Tutoring System

智慧型教學系統(ITS, Intelligent Tutoring System) 實現「從做中學(learning by doing)」的學習理論，並依學習者的個別差異，給予不同的指導或回饋。實作 ITS 的技術有不少種類，許多研究將人工智慧、專家系統應用在 ITS，以提供擬真的教學情境。當程式設計的學習環境具有智慧，包含大量學習過程常見的問題癥結與解決方法，學習者遇到困難時，就可以適時獲得指導，進而減少挫折對學習的阻礙。

在 ITS 與學習者之間的「介面」，是影響學習流程能否成功的關鍵。若 ITS 的介面令人感到困惑或者對教學目標領域不支援，則教學成效將無法彰顯(Stephen Baudendistel, 1987)。從這個觀點可以推論，學習領域若為程式語言時，學習環境提供給使用者的介面相當重要，介面的設計需要直覺化、容易使用。不夠友善、操作困難的介面可能導致學習者的抗拒。介面的設計若為真實的程式設計環境，而非選擇、填空、問答等替代方式，對於教學目標領域的支援就得以強化。

2.1.2 Standalone Learning Environment

用於單機獨立運作的 PLE 軟體，不必透過網路連線就可以使用。此類型的 PLE 軟體，通常是專為特定程式語言設計，因此與學習領域更緊密關聯，對於程式語言的特色能夠提供更精緻的功能，使學習者能夠更深入追蹤、瞭解程式的運作，例如在撰寫一段類別的代碼後，可以透過視覺化的圖形操作介面，將類別實現成為看得到的物件，再與此物件的方法進行互動。

例如 BlueJ 的設計目的是 Java 程式語言教學，它提供視覺化的學習環境，可以建立類別關係圖，讓學習者對於物件導向的概念不必憑空想像，而是在畫面上拖曳出類別的圖形，建立類別之間的關聯，使用這些類別產生物件，並允許學習者與物件更直接的互動，在過程中瞭解 Java 程式的撰寫、測試與除錯(BlueJ, The interactive Java environment)。

除提供視覺化的學習環境，也可以「從遊戲中學習」。例如 Robocode 就是一款模擬坦克車對戰遊戲的教育軟體，參與的玩家必須自行撰寫 Java 程式碼，使坦克車擁有可以移動、閃躲及攻擊的邏輯處理，在達成遊戲目標的過程中，可以提升程式設計的技巧。Alice 則是提供 3D 環境的教育軟體，讓學習者著手建構一個 3D 虛擬的遊戲世界，讓程式邏輯可以用圖形介面的拖曳方式完成，並產生可以看得見的執行結果，使學習者透過寓教於樂的方式學習程式設計。

這些 PLE 軟體大多擁有豐富的功能，能在許多方面給予學習者助益。但都有相同的限制，即缺少教學者與學習者之間的互動、對於錯誤也較難給予回饋與指導。在課程中使用這些工具，教師並不容易掌握個別學生的學習狀況，通常只能透過作業、測驗來檢視學習成效，而不容易參與每位學生的學習過程並給予更多的指導。對學習者而言，缺少練習結果的紀錄，並不能夠清楚地知道自已的學習成就。

2.1.3 Web-based Learning Environment

CodeLab 為 Turing's Craft 公司的商業產品，以網頁介面提供程式設計課程的練習，支援的程式語言包括 Python、Java、C++、C 及其他語言。網站提供的練習題，由學生完成符合語意及語法的片段程式碼，並由系統自動偵測回答是否正確，教師也可以透過網站觀看學生的答題狀況。CodeLab 提供相當多的練習題，但是答案只是片段的程式碼，通常過於簡短，學生並無法練習撰寫完整程式碼。

CodeSaw 是 Liqwid Krystal 經營的網站，有建立及儲存完整程式碼的功能，並

提供虛擬的執行環境，無須在電腦安裝開發工具，就可以執程式碼並觀看結果。學生可以在 CodeSaw 觀看程式碼如何撰寫，以及獲得執行的結果，但無法自己動手練習，缺少從實作中學習的機會。

ELJT(E-Learning Java Trainer)的研究同樣以 Web-based 環境提供程式設計教學 (António Mendes, 2005)，支援 Java 程式語言，利用 Server 端虛擬環境編譯及執行學生或老師送出的程式碼，因此也無須在使用者的電腦加裝開發工具。學生在瀏覽器畫面中撰寫程式碼，在按下送出按鈕後，Server 端接收程式碼並編譯執行，並將結果與測試資料作比對，若不符合預期結果，則會依照老師設定的指示訊息回應給學生。

Web-based 的 PLE 功能往 Server 端發展，由網站供應所需要的功能，甚至包含程式碼編譯、執行的環境，也是在 Server 端的虛擬環境提供開發工具。這種設計讓使用者得到很高的便利性，不管在什麼樣的電腦操作環境，只要打開瀏覽器便可以使用。

2.2 Learning Objects

IEEE 對 Learning Object 的定義為「一個學習物件是被用來學習、教育或訓練的任何數位或非數位實體」(IEEE, 2002)。對於 e-Learning 的應用，製作良好的教學資源需要相當昂貴的成本，但是在一個學習環境製作的教材，很可能無法在另一個學習環境延續使用。要解決這項問題，需要制定共通的標準，在 2002 年 6 月，IEEE 同意一項 LOM(Learning Object Metadata)的標準，這項標準的前身則是由 IMS(Instructional Management System)發展。

2.2.1 SCORM

SCORM 全名為 Sharable Course Object Reference Model，由 ADL(Advanced Distributed Learning)所擬定，它整合數個組織對於 Web-based e-Learning 的“高階(high-level)”標準和規格。SCORM 讓學習內容可以重複使用，使用支援 SCORM 標準的 LMS(Learning Management System)，就可以共享已有的學習內容。SCORM 也支援對內容順序(content sequencing)的定義，使教學者能夠依照學習策略安排內容(ADL, What Is Scorm)。

符合 SCORM 標準的教材，可以滿足以下的高階需求。

- ♦ Accessibility(可獲取性)：學習者在任何時間地點可取得適當學習內容。

- ♦ Interoperability(可互通性)：學習內容在任何開發系統或學習平台皆可使用。
- ♦ Durability(耐久性)：當科技提升或改變時，不需要重新修改應用程式或教材。
- ♦ Reusability(可再使用性)：不同應用環境下，學習內容可重複使用。

(徐文杰、林沛傑，2003)

2.2.2 XML 的二進位資料編碼

數位化的學習內容，可以藉由網路傳輸讓使用更為便利，此過程需要具有可攜性的資料格式標準。由 W3C 制定的 XML 規範，可以成為數位內容的格式標準。在 W3C 建議的 XML Schema 規範中，定義兩種資料型態用來儲存二進位資料，分別是 hexBinary 及 base64Binary(W3C, XML Schema Part 2)：這兩種資料型態使用不同的編碼方式，將二進位資料轉換為不含特殊字元的字串，再加入 XML 內容中保存(王志弘，2004)。

hexBinary 使用 HEX 編碼，即 16 進位表示法。假設二進位資料的前 3 個位元組以 10 進位整數值表示為(99, 255, 156)，則 HEX 編碼結果為(0x63, 0xFF, 0x9C)，每一位元組的 16 進位可由固定雙字元 00~FF 表示之，因此在 XML 中可以保存為 <hexContent>63FF9C</hexContent>。

base64Binary 則以 Base64 演算法編碼，它每次以 8 位元取 3 個位元組(共 24 位元)，再重新以 6 位元一組的方式重組為 4 個位元組，由於 2 的 6 次方僅需要 64 種符號表示，常用可見字元的大寫字母[A-Z]、小寫字母[a-z]、數字[0-9]加起來就有 62 種符號，只需要再加上兩種可見字元例如[+, /]或[-, _]就有 64 種，因此能夠將二進位不可見位元的資料轉換為可見字元的符號來表示，電子郵件夾帶的圖片或附檔，通常就是以 Base64 編碼的 MIME 格式儲存。

教材資源若以 XML 格式儲存，經常需要進行編碼，除圖片、聲音等多媒體及可執行檔案之外，使用 DBCS(Double Byte Char System，雙位元組字元系統)的文字檔案，雖然內容為純文字，但其編碼可能是 BIG5、GB2312、UTF-8、UTF-16 等各種不同的類型，很可能與 XML 標示所使用的編碼不同。在不同編碼之間進行轉換，可能缺乏對應的字集，而造成轉碼結果失真，例如一個包含中文、日文、

韓文的 UTF-8 編碼文字，強制轉碼成 BIG5 的結果，將導致日文及韓文的資料遺失。為使學習環境朝向國際化(internationalization)發展，並考慮有些程式語言的實作不支援 Unicode，因此也需要將文字檔案進行 Base64 或 HEX 編碼，再保存至 XML 以不影響檔案的原貌。

2.3 執行程序的自動化

為使學習者能順利進行專注於學習目標的程式設計實際練習，在完成一段程式碼的編輯後，就能以學習環境提供的“執行”功能，獲得練習結果的回饋。“執行”的程序實際包含一連串繁瑣的步驟，例如「前置處理、檢查程式碼、編譯程式、執行程式、單元測試、擷取及比對輸出結果…等。」這些步驟若由學習者在每次練習時重複進行，不但花費時間也干擾學習，而不容易聚焦在學習目標的達成。

將“執行”所包含的程序，由教學設計者在教材中定義，除提供教材設計階段的自動化，也可以幫忙學習者減少不必要的負擔。許多軟體開發工具，被設計用來簡化繁瑣的流程，將維護程序自動化。將這些工具應用在教學中，可以減少教學不相關的問題干擾學習。

2.3.1 Make

實務上通常將大型程式分割為較小、可管理的片段來處理。這些小片段的處理方式有所不同，例如有些需要由巨集處理器執行、有些則由複雜的程式產生器處理，同時包含選項、定義及宣告的設定，也可能有測試等相關維護作業。對程式設計者而言，檔案之間的關聯、最近修改那個檔案及操作的先後次序等，都相當不容易記住，尤其是在長時間的編輯後。在大部分的案例中，描述這些程序的檔案並不會經常改變，因此一般的程式發展程序可以簡化為“*think --- edit --- make --- test ...*”，亦即輸入簡短的 `make` 指令，就可以取代繁瑣的維護程序(Stuart Feldman, 1979)。

`Make` 經過許多演變的過程，例如 `BSD make` 及 `GNU make` 等分支。而較為廣泛使用的版本，則是由 `GNU` 發行的 `make` 工具。藉由 `makefile` 檔案提供的資訊，可以控制執行檔及其他非原始碼檔案的產生。它可以讓使用者在建立(build)及安裝軟體時，不必瞭解細節如何完成。自動化偵測那些檔案曾經修改而需要更新，而

只執行變更相關的程序，不必重頭再執行一次。因為能夠與 Shell 指令結合，所以 make 並不限定任何特定的語言才能使用，也能夠完成許多其他用途的任務(Free Software Foundation, GNU Make)。對教材而言，若以 makefile 預先定義編譯及測試所需的指令，學習者就能用簡單的 make 指令，取得程式撰寫練習的結果。

雖然 make 不限定搭配的程式語言，但仍有其他類似的工具，針對不同程式語言的需求而設計，例如 Java 語言開發的 JMake，以及由微軟公司推出運用在 Windows 系統的 nmake。無論是那一種版本的 make，其基本的設計都是由 Makefile 描述檔定義指令步驟，固然有許多優點，但也有一些缺失，例如對文字格式的要求，多一個空白字元就可能造成影響，容易造成使用者困擾，以及需要藉由其他產生器工具的搭配，才能解決如跨平台需要動態產生相關設訂的問題。

2.3.2 Ant

為解決 Makefile 的缺失，Ant 的設定檔以 XML 為基礎，所以有容易解析、讀取的結構，並以任務(task)的定義取代 Shell 指令，且使用 Java 程式實作這些任務，因此能夠解決跨平台執行的問題。許多 Java 的開發工具支援 Ant，例如 NetBeans 整合 Ant 為專案提供設定配置，Eclipse 也內建 Ant 的執行功能。

除提供 Java 程式開發的編譯、執行等任務，Ant 也內建許多檔案系統操作指令，例如檔案的複製及刪除、資料夾的建立及壓縮等。加入一些由 Java 程式撰寫的擴充任務，還能夠完成版本控制(SVN 或 CVS)、網路操作(Mail、Telnet 及 FTP 等)，也能與 JUnit 的測試功能整合，為專案定義自動化測試的任務(Apache, Ant User Manual)。

2.4 程式測試工具

程式測試為學習環境提供第一道的檢驗程序，判斷學習者練習的結果是否與測試資料相符合。通過測試並不表示程式碼撰寫無誤，例如輸入的測試資料不足以檢驗出邏輯設計的錯誤，或是題目要求使用特定的語法、結構，程式碼並未依循指示，卻可以產生同樣的結果。儘管許多情況需要人工檢視才能判定程式碼的正確與否，但自動化的程式測試卻可以提供給學習者立即的回饋，也減輕教學者的負擔。

2.4.1 Unit Testing

eXtreme Programming(簡稱 XP)是由 Kent Beck 在 1995 年提出的軟體開發方法論，倡導測試導向開發方法(Test-Driven Development)，建議開發者在撰寫程式碼時，先建立測試程式，以減少錯誤的發生，並且在日後改寫程式碼時，也能以同樣的測試程式檢驗。測試的最小單位是單元測試(Unit Testing)，用來測試一個類別的方法或副程式(Kent Beck, 1999)。

JUnit 由 Kent Beck 與 Design Patterns 作者之一的 Erich Gamma 共同開發，是 Java 支援單元測試的框架實作，廣受 Java 開發者採用。由於 JUnit 只適用於 Java 程式開發，因此其他程式語言的開發環境，必須實作專用的單元測試框架，例如.NET 環境的 NUnit、C 語言的 CUnit、PHP 語言的 PHPUnit 等。在教材的練習題設計中，可以整合這些單元測試的工具，以針對不同程式語言提供較細緻的測試。

2.4.2 Web Testing

對於動態網頁程式設計，僅使用單元測試並不足夠，由使用者的操作與網頁程式產生互動，其功能是否能夠正確運作需要被檢驗，因此網頁測試(Web Testing)是一種功能測試(Functional Testing)。網頁測試程式通常模擬瀏覽器的行為，藉由 HTTP 協定進行 URL 的資料存取，將測試資料以表單或網址參數方式傳遞，取得網頁程式的回應再分析其內容。

網頁測試工具可以簡化測試工作的進行，例如允許開發者以 Script 撰寫測試流程，或是提供 Proxy 軟體側錄瀏覽器與網頁伺服器之間的傳輸資料，並自動產生測試案例。例如 Apache 開發的 JMeter 軟體，可以用於網頁功能測試、負載壓力測試等。

2.4.3 GUI Testing

圖形化介面的視窗程式，除單元測試外，也需要對介面的功能進行測試。這種測試的流程需要模擬滑鼠點擊、鍵盤輸入等事件，並能夠取得介面的狀態改變，以判斷程式的功能是否正確。這類的測試工具通常針對特定的開發環境或作業系統設計，例如以 Java 的 AWT/Swing 撰寫的 GUI 應用程式，可以透過 Abbot 框架進行測試。

滑鼠和鍵盤操作的自動化工具，也可以進行視窗介面程式的測試。例如

AutoHotKey 軟體可由使用者撰寫 Script 程式，控制視窗介面的各種點擊、按鍵操作，使用這類型的工具進行測試，必須由最終的結果判斷程式功能是否正確。舉例來說，經過多個視窗介面操作的步驟之後，被測試的程式會產生一個輸出檔案，就可以用此輸出檔案判斷程式功能的正確與否，若視窗的介面設計有問題，則會在自動化操作的過程中發生錯誤。

2.5 整合開發工具

儘管支援純文字編輯的軟體即可用於程式碼的撰寫，但對於專業的程式設計人員，使用合適的工具有助於提高品質與生產力。為程式設計用途而開發的編輯工具相當多，功能也良莠不齊，能夠被稱為適合程式設計的編輯器，通常會包含以下列舉的部分或更多功能。

- 程式語法突顯(syntax highlighting)、自動完成(auto completion)
- 支援巨集、程式碼樣板定義
- 程式碼區塊顯示(例如 if { ... } 為一個區塊)
- API 說明、開發指南等文件查閱
- 整合編譯、執行、除錯功能
- 方便的文字編輯功能(複製、貼上、搜尋、取代等)

jEdit 是一套使用 Java 語言發展的工具，具有跨平台(支援 Windows、Mac、Linux 等作業系統)的優點，它是為程式設計用途開發的編輯器，已內建或可由 Plug-in 方式擴充如上述所列舉的功能。在程式語法突顯的部分，jEdit 已支援數十種程式語言的關鍵字標記，也能夠透過 XML 定義檔的方式輕易增加對新語言的支援。即使 jEdit 的功能無法與強大的整合開發環境如 Eclipse 等相提並論，但體積輕巧、啟動快速、容易擴充等優勢，使它仍是一個歷久彌新的程式碼編輯器，簡單而容易使用的操作介面，也能夠減少程式設計初學者的負擔。

PLWeb 將 jEdit 以 Java Web Start 技術與網站平台整合，讓使用者能夠獲得從網頁中就能夠立即執行的好處，而不必事先安裝、設定或進行更新。在網站平台的操作中，課程設定的參數也能夠自動傳入，使得 jEdit 在啟動之後就能夠透過 PLWeb 擴充的 Plug-ins，為使用者建立學習環境。本研究將延續使用 jEdit 作為編輯器發展的基礎，並以 Plug-ins 的方式實作支援多程式語言及教材編輯等功能。

三、系統設計

本章介紹系統設計的概念，將依序描述教學環境、平台架構與教材格式設計，並描述整合運作的系統流程，著重「支援多程式語言」的特色。本研究先定義參與教學的角色及流程，試圖探討教學平台的改良方法，以期盡可能滿足不同程式語言的教學需求。接著再探討如何定義統一的教材的格式，使不同程式語言的細部差異仍能夠以同樣的格式製作教材。

3.1 教學環境設計

依照一般學校授課的需求，教學環境存在的實體有教師、學生、課程，以及課程所採用的教材、編寫教材的作者。將程式設計課程的需求區分為上、中、下游，就包括教材製作、教師授課、學生修課三個層次。教學環境所提供的功能，需要滿足各層次的需求，才能成為完整的平台。

這些實體的關係如下。

- 一位作者可開發多本教材。
- 一位教師可教授多門課程。
- 一門課程可選用多本教材。
- 一門課程可由多位學生參與。
- 一位學生可參與多門課程。

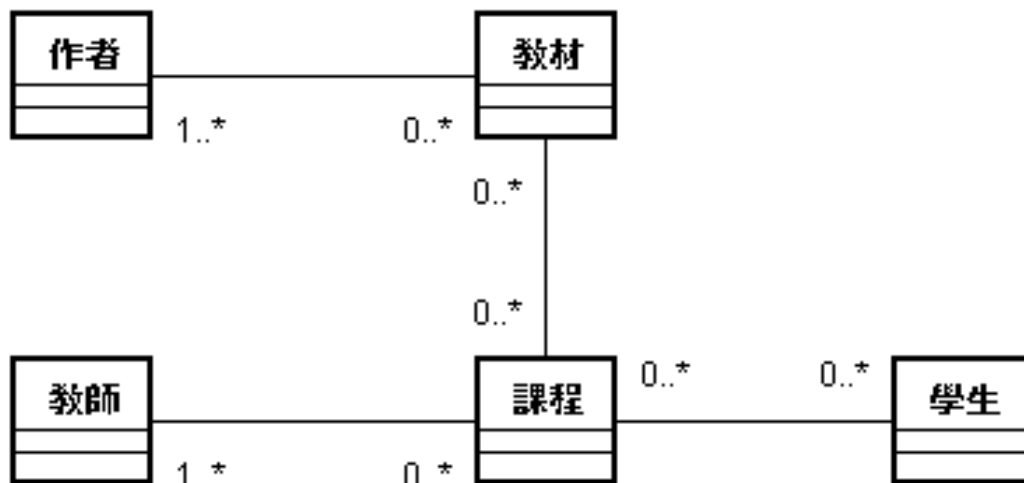


圖 0-1 教學環境實體關係圖

學習環境的實作是一個網路服務平台，可提供教材與課程的相關功能給予作者、教師、學生。這個平台支援在教室集中授課的教學模式，以及教師與學生以非同步的方式進行遠距教學。此平台透過網路傳輸資料，因此使用者必須連線上網才能操作。

3.2 教學平台架構設計

本研究以 PLWeb 的架構設計為基礎，結合 Web-based 的服務平台與 Standalone 的整合開發環境。Server 端的資料庫保存帳號、教材、課程、學習紀錄等資料，在網路連線的狀態下，Client 由網頁傳遞參數進行啟動程序、下載教材資料檔，作者開發完成的教材。而學生的學習紀錄，也是藉由網路傳回至 Server 保存。

為改良教材製作機制，除 PLWeb 的教師及學生兩種角色之外，本研究增加作者的角色，以設計教材編輯的流程，使教材的產生能夠(1)提高效率(2)支援多種程式語言(3)富有擴充彈性。PLWeb 的程式碼編輯器，原本只用於學生的練習，本研究加強編輯器的功能，使作者也能利用編輯器進行教材開發，提高教材製作的便利性，也使得教材的編輯與使用達到整合。

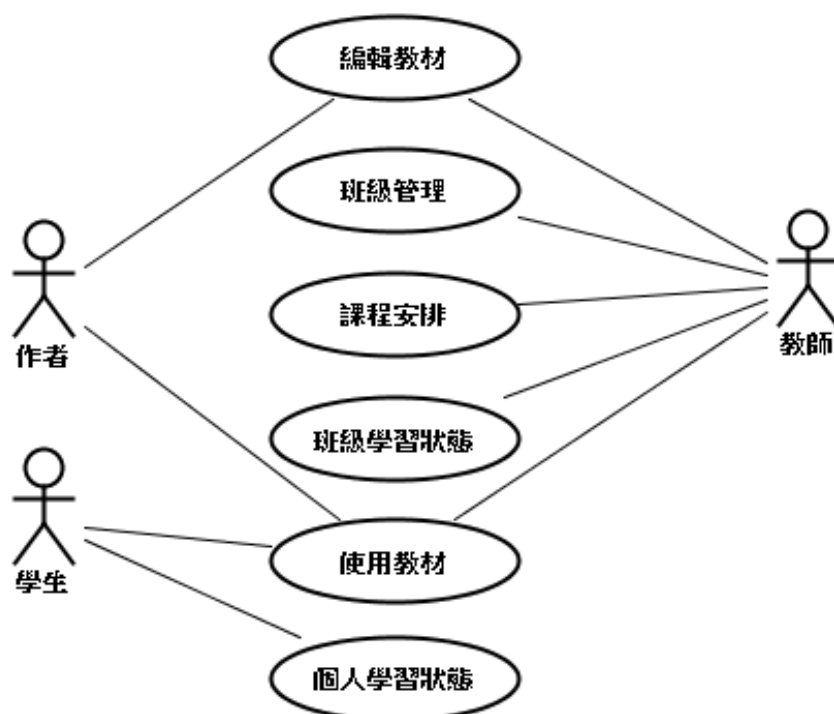


圖 0-2 教學平台使用案例

在作者的角色定義後，編輯教材的流程可以衍生更多功能。PLWeb 的教材由也可身兼作者的教師依課程需求製作完整的教學內容及練習題，並能夠以電子書籍的形式發佈。教材可以由教師與學生共同編輯，或是由專業作者進行開發，最後的完成品則在教學平台發佈與保存。這些製作好的教材，可以經由某種授權，或提供免費共享的方式，給予授課的教師選用。教材依各單元的先後次序，可以提供由淺入深的學習順序，而選用教材的授課教師，則可依照實際課程的需求，另行訂定實際授課採用的單元、重新安排順序，或是與不同的教材單元混合。

教師的角色是負責開設課程、選擇授課內容、安排教學進度，並督導學生進行學習，或是為學生解惑。在選修一門課程之後，學生閱讀教師安排的課程內容，並實際動手練習撰寫程式。學習的過程即產生學習紀錄，由系統選擇有用的資料提供給學生瞭解學習狀態，也方便教師控制教學品質，例如檢視學生所花費的時間，以及完成練習的比例等

下表說明各種資料類型的學習紀錄。

學習紀錄資料類型	說明
狀態代碼	學生對於練習的完成狀態，例如編譯失敗、通過成功、測試失敗等。學生由狀態得知已完成多少練習，教師透過狀態以瞭解全班或個別學生的進度。
輸出訊息	程式碼編譯、執行、測試的過程，有文字輸出訊息，可用於程式碼的除錯或檢視程式測試結果。包含指令執行的輸出訊息，例如編譯器因語法不正確發出錯誤警告的訊息。教師從這項紀錄產生的學習狀態表，可以得知學生是否如期完成程式練習，也能夠在遠端獲悉學生遭遇的困難。
按鍵紀錄	程式碼編輯區的操作歷程，包含增加及移除文字兩種事件，可用來重播學生編寫程式之過程。
紀錄時間	每次學習紀錄產生的時間點。
花費時間	完成每一題練習所用的時間加總。
程式碼	保存學生所寫的完整程式碼，以作為日後繼續編輯或查詢之用。

表 0-1 學習紀錄的主要資料類型

對於作者、教師、學生三個使用者角色，教學平台提供的資料庫，可區分為教材、課程及學習紀錄三大類，其互動方式及資料流向如圖 3-3 所示。作者開發教材，並將完成的教材保存至教材資料庫，由教師選擇教材作為授課內容，學生藉修課可取得教材並開始進行學習任務；在學習過程中，包含表 3-1 所列的各項資料將回傳至學習紀錄資料庫；教師由課程工具所提供的功能，取得學習紀錄彙整統計的結果，以督導學生的學習狀況。

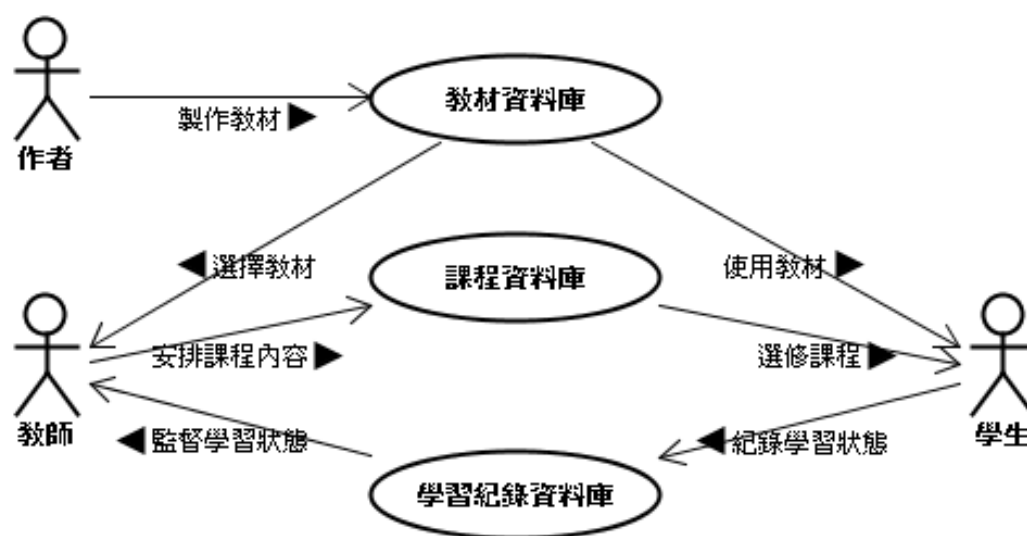


圖 0-3 教材、學習紀錄與使用者角色之間的關係

3.3 教材格式設計

無論教材使用什麼樣的程式語言，只要符合「練習導向」、「從做中學」的教學設計，教材將包含兩大類資源：(1)教學內容、(2)練習題。因此教材格式的設計，需要完整保存這些資源。

本研究參考 PLWeb 原有的教材格式，沿用以 XML 格式儲存的设计，並重新定義其組成元素及標籤語法，建立新的教材格式，在未來不需要修改格式定義的前提下，能夠支援多種程式語言。在教材的 XML 格式制定後，產生教材的工具便可依照不同需求產出能夠符合格式規範的 XML 資料，就可作為此學習環境的教材。

教材可以視為由許多單元組成的電子書籍，各單元皆有相對應的 XML 檔案，這些檔案能夠各自獨立，在檔案中包含單元所需要的完整資源。在各單元的檔案

相互獨立的架構下，教材設計與維護可以獲得如下的便利性：

- ♦ 將教材 A 的單元 1、單元 3、單元 5 複製或搬移到教材 B，作為基礎重新開發新教材。
- ♦ 在教材 C 中混合不同程式語言的教學，以達到特別的教學目的，單元 1、單元 2 教 C++、單元 3、單元 4 教 Java。
- ♦ 把教材 D 匯出成為包含多個 XML 檔案的資料夾。
- ♦ 以 XML 檔案來儲存教材的方式，更容易操作教材內容的新增、複製、修改、移動、刪除等，而不需要同時考慮放置在其他地方的設定或資源，也不必考慮各單元或書籍之間的相依性。

教材單元的概念為書籍中部份教學內容與練習題的集合，設計上必須考量實際的教學案例。例如教學內容的呈現可以是網頁、投影片、多媒體、文件等，而練習題可能包含程式碼、輸出入檔案及驗證答案是否正確的測試工具等，依照實際教材設計的需求不同，這些需要被儲存在教材單元的資料也會有許多差異。

舉例來說，在書籍「C 語言入門」的「檔案系統操作」單元，有一個實作練習「請撰寫程式 SUM.C 可由 INPUT_1.TXT 讀取每一行正整數資料，並將加總的結果儲存為 OUTPUT_1.TXT。」從這個案例描述可以得到，單元中需要一個 HTML 檔案以顯示此題目描述，提供一個 SUM.C 的程式碼檔案給學生撰寫，並附帶一個 INPUT_1.TXT 儲存題目指定的輸入資料，以及一個用來比對 OUTPUT_1.TXT 內容是否正確的 Diff 工具。

而「Java 語言入門」的「類別方法」單元，則可能有這樣的實作練習「請撰寫類別 Counter，包含一個合計的類別變數，初始值為 0，並定義 add 方法，使其接受一個 Long 型別的參數，將參數值加入合計後儲存，並定義 get 方法，使其傳回 Long 型別的合計數值。」在這個案例描述中，除使用 HTML 儲存題目描述，還要有 Counter.java 的程式碼檔案，以及使用 JUnit 進行單元測試的 CounterTest.class 測試程式。

常見的整合開發工具，大多支援 Project 管理的功能，一個 Project 即擁有多個程式碼、文件、設定等。教材中的每一個單元，可以視為各個獨立的 Project。由上述的兩種案例，可以初步推估單元會包含許多檔案資源，例如教學內容、多媒體、程式碼、輸入輸出、測試工具等不同檔案類型，又會包含許多執行指令，以

讓這些檔案能夠自動化運作，例如編譯、執行、驗證結果是否正確的步驟。為能夠支援多程式語言，並不能依照某些程式語言的特性定義 **Project** 的組成元素，而必須找出在大多數程式語言都會存在的共同元素。

本研究以 **Project**、**File**、**Task**、**Command** 的概念定義教材單元，一個單元視為一個 **Project**，由多個 **File** 及 **Task** 所組成，而 **Task** 又可由多個 **Command** 組成。**File** 即單元下所包含的各種檔案，而 **Task** 則為一道練習題，其中包含許多 **Command** 用來編譯、執行程式，並檢驗最後的結果是否符合題目的測試案例要求。**Project** 與 **Task** 的細部定義在不同程式語言之間會有許多差異，因此各加入抽象的 **Properties** 屬性，讓「設定名稱」與「設定值」皆能夠依照各種案例進行擴充調整，而不需要改變根本的教材格式設計。

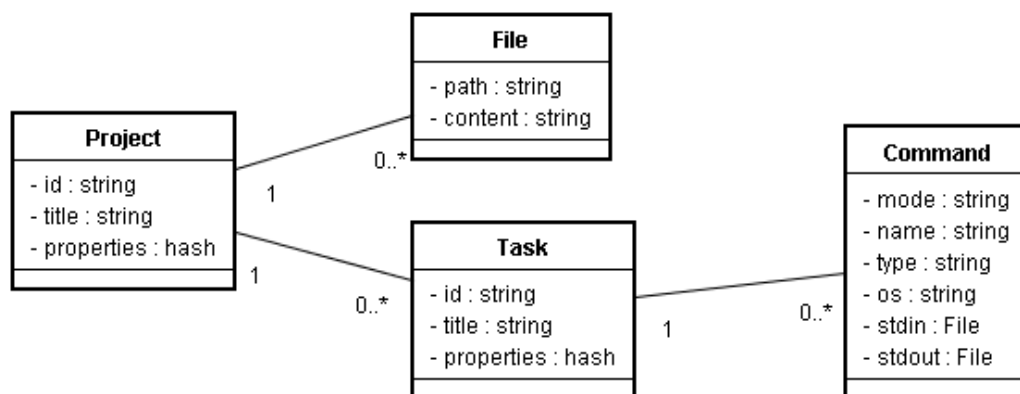


圖 0-4 教材單元的 **Project** 組成元素

開發不同程式語言的教材時，需要考慮程式語言的特性及差異。教材格式能夠適應不同程式語言的需求，作者可依需求自行擴充，才能使教學平台發揮事半功倍的效益。下表列舉不同程式語言之間可能存在的差異：

差異項目	舉例說明
1. 程式碼檔案命名	C 語言的*.c；Java 語言的*.java。
2. 相關檔案	C/C++語言的*.h、Java 語言的*.properties。
3. 直譯器/編譯器	Scheme 使用 petite、C 語言使用 gcc、Java 語言使用 javac。
4. 指令參數	例如指定程式碼的編碼格式、配置記憶體上限、函式庫存在的路徑等。
5. 執行程式	執行編譯產出的檔案(*.exe、*.class)、直譯式語言使用指

	令執行原始碼。
6. 設定檔	有些程式語言需要加入專用的設定檔，例如 PHP 語言的 <code>php.ini</code> 。
7. 結果測試	在終端機畫面執行的程式，可以比對輸出結果；有些程式語言擁有成熟穩定的單元測試框架，例如 Java 可以搭配 JUnit 套件、.NET 則有 NUnit 可以使用。

表 0-2 製作不同程式語言教材可能存在之差異

對於上述的各種差異，應用圖 3-4 的組成元素，可透過 **Command** 與 **Properties** 的調整以滿足不同程式語言。針對某種程式語言設計教材時，可以用 **Command** 定義編譯、執行、測試的步驟(差異 3~7)，搭配 **Properties** 定義編輯器或系統的功能設定(差異 1~4)，如此就能解決程式語言的差異問題。舉例來說，在開發 C 語言的教材時，先以 **Properties** 定義編輯器開啓 `main.c` 程式碼檔案，再以 **Command** 定義執行 `gcc main.c` 的編譯指令。

3.4 教材開發流程設計

對於常用的程式語言，教材製作的需求大多雷同，因此可以預先定義一份示範用的教材，加入練習題的範例程式碼、說明，以及配置合適的測試程序，即可作為學習環境的教材樣板。教材樣板也可以作為編輯流程的引導，例如定義增加一個新題目時，作者需要填入那些設定資料、完成那些檔案。對於不同程式語言，可以提供不同的樣板，而同一個程式語言在不同開發環境或作業系統運作，也可以分別設計不同的樣板。因測試的方法及工具可以內嵌在樣板中，即使其他條件相同，而測試的方法或工具不同時，也可以設計不同的樣板以滿足需求。

對於已有的教學資源，例如書籍所附的範例光碟，已包含程式碼範例、教學電子檔，在製作教材時只需要重複利用這些資源，便可以提高教材開發的效率。轉換的程序從建立一個新的教材單元開始，首先將單元相關的檔案複製一份至教材的路徑，再以 **Properties** 定義哪些檔案是程式碼、哪些檔案是題目說明，並使用 **Command** 定義編譯、執行及測試的流程，必要時也能夠加入讓轉換程序自動化的輔助程式。儘管一份教材可能須要製作數量不少的練習題，但是在不同的練習題間，定義的差異通常都很少，只需要將這些差異以 **Properties** 的變數設定，就可以容易地將定義複製，再修改少許的變數設定後，便成為下一個練習題。

以教材格式的觀點來看，在教材路徑下放置新的檔案，就是增加教材單元的 File 元素，教材單元可以完整保存路徑下的所有檔案。每增加一個新的練習題，就是增加 Task 元素，並以 Task 的 Properties 及 Command 元素保存編譯、執行及測試的定義。教材編輯器所提供的功能，可以輔助這些元素的建立及設定，包含所需要的編碼程序及匯出成 XML 格式。

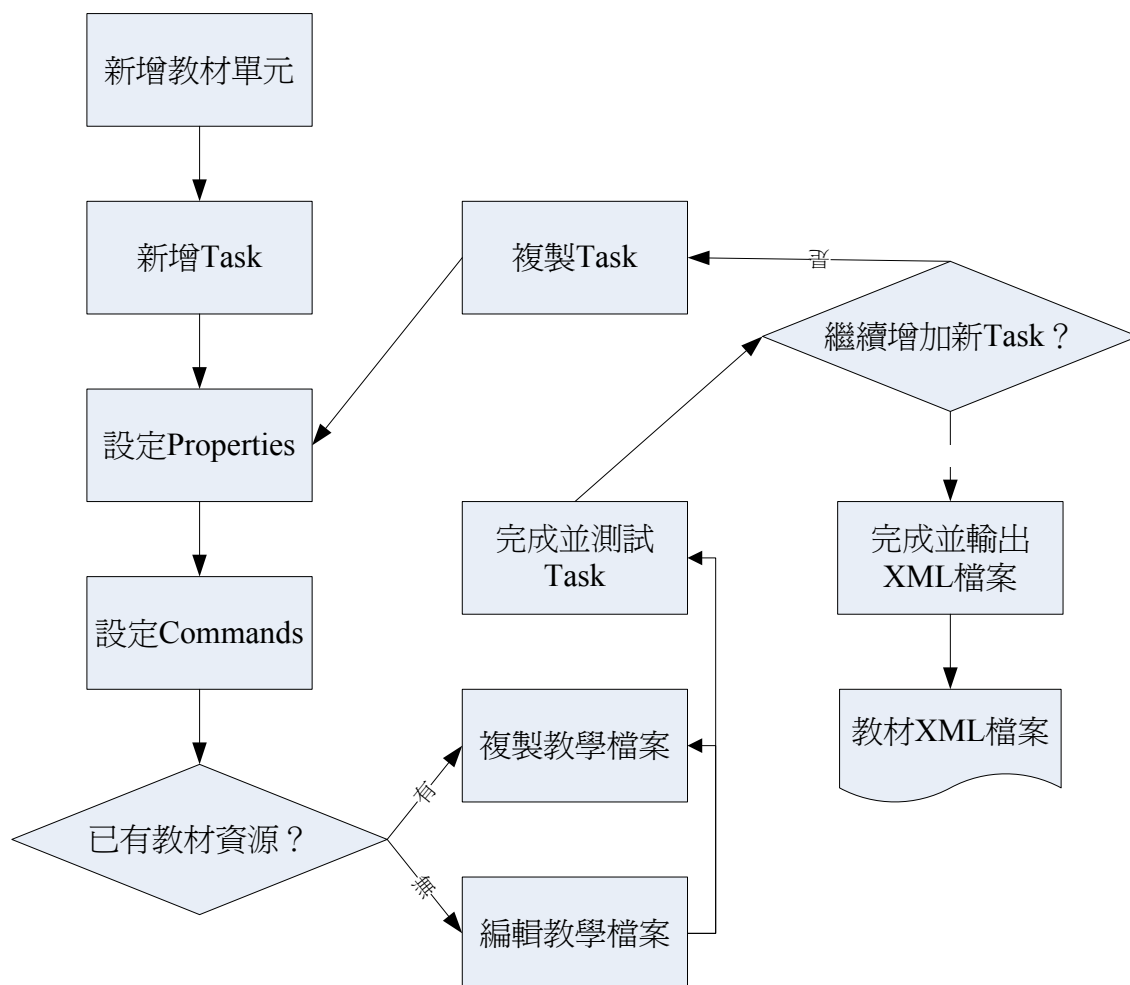


圖 0-5 教材開發流程圖

3.5 測試案例設計

在教材格式中，測試是 Task 所包含的多個 Command 之一，它直接影響到學習者獲得的回饋。每一道練習題進入“執行”的程序時，最後都交由測試程序判

斷練習的結果是否正確，測試的結果由系統儲存後成為學習進度的依據，在練習的結果不正確時，也是由測試程序提供錯誤訊息，通常學習者可以由訊息的指示而修正程式設計的錯誤。

除測試程序外，有許多指令也可以視為練習結果的測試，例如編譯指令可以視為程式碼語法的測試，在編譯失敗時提供錯誤提示訊息。用於測試的指令，必須回傳結束代碼(Exit Code)，一般的作業系統均以此代碼作為 Shell 指令執行結果的判斷依據，依照慣例，程式正確執行無誤，會傳回數值 0 的代碼，若傳回非 0 的數值，則表示有錯誤發生，其代碼的意義則由各程式定義。多數命令列的工具指令，皆依此慣例回傳結束代碼。

開發教材時，可以依照實際的需求，設計各種測試程式以檢驗練習結果，例如檢查輸出檔、分析程式碼、單元測試或功能測試等。這些測試所需要的工具，可以內嵌在教材單元中，或是預先安裝在使用者的電腦系統。

四、系統實作

學習環境主要以 Java 相關技術實作，Web-based 的教學平台使用 Tomcat 伺服器，搭配 Groovlet 技術開發動態網頁，Client 的編輯器則使用 Swing 設計視窗介面，並由 Web Start 技術部署與啟動。平台的實作方式讓使用者不必安裝額外的軟體，只需要 JDK 及開發工具，使用瀏覽器就可以操作。

4.1 資料庫與網站功能實作

教學平台採用 MySQL 關聯式資料庫，資料表的型態皆為 InnoDB，並以 UTF-8 編碼處理字元型態的資料欄位。面對資料量的不斷成長，MySQL 可以採用水平擴充(Scale-out)的策略應對。搭配 MySQL 使教學平台可以在不同規模的環境運作，包括小型的嵌入式裝置，或是大型的伺服器叢集。教學平台的資料庫由數個資料表構成，用於儲存班級、使用者、練習狀態、教材檔案等資料。

下表說明教學平台的主要資料表。

資料表名稱	用途說明
COURSE	教材基本資料。
COURSE_FILE	保存教材 XML 資料檔。
ST_CLASS	班級基本資料。
ST_USER	使用者基本資料。
ST_CODE	學生練習狀態資料。
ST_MESSAGE	學生每一次送出的練習狀態紀錄。

表 0-1 主要資料表用途說明

網站的功能主要分為四大類，包括帳號、班級、教材及學習管理功能，提供使用者透過瀏覽器操作教學平台。使用者在註冊後即擁有學習管理功能，班級管理則需要額外的教師權限，教材管理需要作者權限。

下表說明主要的網站功能。

網站功能名稱	用途說明
帳號管理	新帳號註冊、權限管理、登入及登出、個人資料維護、選修、教材編輯。
班級管理	班級資料維護、安排授課內容及進度、學習狀況統計。
教材管理	教材資料維護、啟動教材編輯器、匯入及匯出教材資

	料檔。
學習管理	課程加退選、啟動學習編輯器、播放教材內容、檢視學習狀態。

表 0-2 主要網站功能用途說明

4.2 教材格式實作

教材格式由 Project、File、Task、Command 及 Properties 等元素組成，在學習環境中運作時，這些元素以 Java 物件的形式儲存在記憶體中，直到需要儲存至資料庫或檔案系統，才轉換成 XML 格式保存。根據教材格式的定義，可以設計出每個元素對應的 XML 標籤，下圖為教材以 XML 格式輸出的範例資料。

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
  <id>1</id>
  <title>Hello</title>
  <property>
    <key>key1</key>
    <value>value1</value>
  </property>
  <file>
    <path>readme.txt</path>
    <content>DQo=</content>
  </file>
  <task>
    <id>1</id>
    <title>Show Hello</title>
    <property>
      <key>WHAT</key>
      <value>Hello</value>
    </property>
    <command>
      <mode>author</mode>
      <name>echo</name>
      <type>echo</type>
      <os>any</os>
      <cmd>echo ${WHAT}</cmd>
      <stdin-file/>
      <stdout-file/>
    </command>
  </task>
</project>
```

圖 0-1 教材格式 XML 範例資料

教材元素 Project、File、Task、Command 及 Properties，實作的 XML 標籤分別為<project>、<file>、<task>、<command>及<property>。以<project>標籤來說，除了<id>及<title>兩個參數標籤外，還包含其他子元素的標籤如<property>、<file>及<task>，其標籤的用途說明如下表。

標籤名稱	定義	說明
<id>	識別碼	教材檔的 Unique ID。
<title>	標題	教材的標題命名。
<property>	屬性設定	教材的全域屬性設定。
<file>	檔案	教材內嵌的檔案。
<task>	習題	對應教材中的每一個程式設計練習。

表 0-3<project>標籤定義與說明

<property>標籤只包含<key>及<value>兩個參數標籤，雖然<property>在 XML 資料中有先後次序，但是載入記憶體後，會以 Hash 的資料型態儲存，而不會依照在 XML 中的次序排列。因 Properties 的次序並無意義，用 Hash 可以減少索引耗費的時間。

標籤名稱	定義	說明
<key>	屬性名稱	使用者或系統訂定的屬性設定名稱。
<value>	屬性值	屬性內容的設定值。

表 0-4<property>標籤定義與說明

在<file>標籤建立時，會以<path>標籤保存在磁碟上的相對路徑，而實際的檔案內容則先以 Base64 編碼，再將結果保存於<content>標籤。教材中包含的<file>標籤，在教材資料被載入記憶體時，可以被還原成磁碟上的實際檔案。

標籤名稱	定義	說明
<path>	檔案路徑	檔案存放於磁碟的相對路徑。
<content>	檔案內容	採用 Base64 編碼的檔案內容。

表 0-5<file>標籤定義與說明

<task>標籤代表教材中的每道練習題，它包含<id>及<title>兩個參數標籤，也有<property>的屬性設定，以及<command>組成的指令集合。其中<id>的內容是練

習題在教材單元中的識別代碼，不可與其它練習題重複。`<task>`在 XML 資料中的先後次序，影響到練習題的題號順序，但是將`<task>`重新排列次序時，所包含的`<id>`並不會隨之改變，在資料庫保存的練習狀態，是以`<id>`的識別代碼作為索引的鍵值。這個設計使教材可以隨時增加新練習題或調整順序，而不會影響學生已作答的狀態紀錄。

標籤名稱	定義	說明
<code><id></code>	練習代碼	在教材檔案內給予每道習題 Unique ID。
<code><title></code>	練習標題	命名每道練習。
<code><property></code>	屬性設定	本習題的屬性設定。
<code><command></code>	指令集合	定義諸如編譯、執行、測試等步驟所需要的指令。

表 0-6`<task>`標籤定義與說明

`<command>`標籤儲存指令，它包含用來指定運作模式的`<mode>`標籤，可以將指令區分為不同的模式，例如 `author` 模式只有在作者編輯教材時才會被執行。`<os>`標籤則可以指定作業系統的代號，對於支援跨平台的教材，可以設定多組功能相同，但適用作業系統不同的指令，例如 `windows` 系統的 `del` 指令在 `linux` 系統則用 `rm`。`<type>`指定指令的類型，它的值影響學習狀態的顯示，例如設定為 `test` 時，依指令執行成功或失敗，會產生 `test_ok` 或 `test_error` 的訊息，這個結果會保存在練習狀態資料庫中，影響學習狀態及進度的統計。

標籤名稱	定義	說明
<code><mode></code>	模式	指令作用於哪一種模式下(例： <code>student</code> 、 <code>author</code> 、 <code>teacher</code> 、 <code>answer</code>)
<code><name></code>	顯示名稱	指令顯示的名稱(例：編譯、執行)
<code><type></code>	類型	指令於教學系統中的作用類型(例： <code>compile</code> 、 <code>run</code> 、 <code>test</code>)，於執行後加上 <code>_ok</code> 、 <code>_error</code> 等執行結果代碼，作為訊息儲存的

		狀態識別碼。
<os>	作業系統	用於哪一種作業系統(例：any、linux、windows)，當教材在不同作業系統下使用不同指令時，可於此設定。
<cmd>	指令	指令內容(例：gcc -o Hello Hello.c)
<stdin-file>	標準輸入檔	指定檔案路徑，作為執行時的標準輸入來源檔案(作用於 I/O Redirection)。
<stdout-file>	標準輸出檔	指定一個<file>節點的 path 作為執行時的標準輸出目的檔案(作用於 I/O Redirection)。

表 0-7<command>標籤定義與說明

教材可以藉由<command>及<property>定義的搭配，以支援多程式語言的特性。這種設計部分需要由學習環境提供支援，主要的實作在編輯器中完成，只需要由系統提供少許的功能，即可由教材開發者自行運用，以滿足多數程式語言的需求。

<property>標籤提供具使用彈性的定義方式，其定義方式為：“屬性名稱 = 屬性值 (例如 prop1 = value1)”，它可以區分為系統屬性與自訂屬性兩種類別。系統屬性的值具有特殊用途，需要學習環境的實作配合，屬性名稱需要符合學習環境已定義的功能，常用的系統屬性說明如下表所示。自訂屬性可由教材設計者自行定義，屬性名稱可自由命名，例如“GccParam = -lm -W”。自訂屬性可以當作變數，用於簡化系統屬性的定義，這種使用方式需要配合特殊符號，其格式為「\${屬性名稱}」，例如先設定自訂屬性“ProgName = ex01”，就可以設定系統屬性“file.main = \${ProgName}.c”，當練習題被執行時，“file.main”的值會被替代成“ex01.c”。

下表說明常用的系統屬性。

屬性名稱	說明
file.main	用於指定主要程式碼檔案的名稱。

	<p>例如：</p> <p><code>file.main = ex01.cpp</code></p>
<code>file.html</code>	<p>用於指定說明網頁之路徑，若與程式碼檔案共存，則直接指定相對路徑；若以 <code>http://</code>或 <code>https://</code>協定名稱起始，則表示網頁位於網際網路，以完整 URL 路徑表示之。</p> <p>例如：</p> <p><code>file.html = ex01.html</code></p> <p><code>file.html = http://hostname/lesson/ex01.html</code></p>
<code>file.part</code>	<p>可以指定一個檔案路徑，學習者在開啓教材時，將以此檔案內容作為基礎程式樣板(部份程式碼、註解、提示說明)，若不指定則提供空白檔案。</p> <p>例如：</p> <p><code>file.part = ex01_part.cpp</code></p>
<code>file.attach.*</code>	<p>用於指定主程式碼檔案需要一併開啓的程式碼或資料(定義檔、輸入資料)。</p> <p>例如：</p> <p><code>file.attach.0 = ex01.h</code></p> <p><code>file.attach.1 = ex01.txt</code></p>
<code>file.author.*</code>	<p>用於指定在教材設計階段，需要與主程式碼檔案一併開啓的程式碼或資料(定義檔、輸入資料)。</p> <p>例如：</p> <p><code>file.author.0 = ex01.html</code></p>
<code>askFor.*</code>	<p>在新增練習題時，會自動跳出對話視窗詢問新練習題的設定值。</p> <p>例如：</p> <p><code>askFor.ExName = 請輸入新的檔案名稱(不含副檔名)</code></p>

表 0-8 常用系統屬性說明

在<command>標籤的設計，同樣可以使用變數的方式定義，提供具有彈性的指令流程定義。例如有兩個自訂屬性 “GccParam = -lm -w” 、 “ProgName = ex01” ，以及一個系統屬性 “file.main = \${ProgName}.c” ，則可以定義編譯的指令內容為 “gcc \${GccParam} -o \${ProgName}.exe \${file.main}” ，在實際執行這道指令時，內容經過處理後即為 “gcc -lm -w -o ex01.exe ex01.c” 。

4.3 編輯器實作

編輯器以 jEdit(stable version 4.2)為基礎，可以在已安裝 JDK/JRE 5.0 以上版本的系統環境運作。固然有許多比 jEdit 功能更強大的編輯器，但是在建構學習環境時，並不能僅考量編輯器的功能多寡，其他因素更可能影響可行性。由於 jEdit 擁有下列的優點，適合學習環境使用。

- ◆ 開放源碼。可以取得編輯器完整的原始碼，能夠依照需求擴充或改寫其功能，並且沒有商業軟體的授權問題。
- ◆ 輕巧。由於 jEdit 壓縮後的檔案體積相當小(小於 4mb)，因此方便使用者於網路下載傳輸。佔用系統資源低，可以在處理器及記憶體配備等級較低的電腦流暢運作。
- ◆ 容易擴充。使用 Java 程式搭配 XML 設定即可開發 jEdit 專用的擴充套件(Plug-ins)，因此很容易將它改造為適合學習環境使用的編輯器。
- ◆ 跨平台。jEdit 可以在任何安裝 JDK/JRE 的平台上使用，包含 Windows、Unix、Mac 等，因此以它建構的學習環境較方便移植，更容易支援其他作業系統。

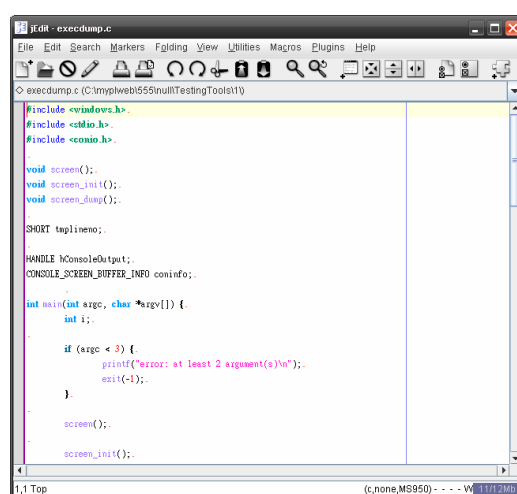


圖 0-2 原始版本的 jEdit 編輯器畫面

使用者以瀏覽器開啟課程或教材的網頁，可以在畫面上點選啟動編輯器的超連結，以執行編輯器的啟動程序。由於網站在 Server 端運作，而編輯器必須下載至 Client 端執行，因此採用 Java Web Start 的機制實作。由網站即時產生 Web Start 所需的 JNLP 檔案，指定需要傳遞給編輯器的參數，包含使用者及課程、教材所需的資料，再由瀏覽器自動執行 javaws 程式進行啟動程序。

啟動程式首先會下載 jEdit 相關檔案，包含學習環境所需的擴充套件、XML 教材檔案等。為節省傳輸頻寬及縮短下載等待時間，啟動程式配合 MD5 驗證碼以實作檔案快取機制，並在 HTTP 傳輸純文字資料時，以 GZIP 進行壓縮/解壓縮。



圖 0-3 啟動程式畫面

下圖為學習環境的編輯器畫面，以 jEdit 為基礎加以擴充。主要增加的兩個擴充套件分別為 Info Viewer 及 Message Console。Info Viewer 為學習者提供教學內容、題目描述及輸入輸出文字訊息等。Message Console 則提供練習與教材設計的各項功能。在擴充套件的控制在下，編輯器會讀取教材檔的內容，依照設定自動開啓需要編輯的程式碼檔案等，並控制 Info Viewer 及 Message Console 顯示學習資訊。

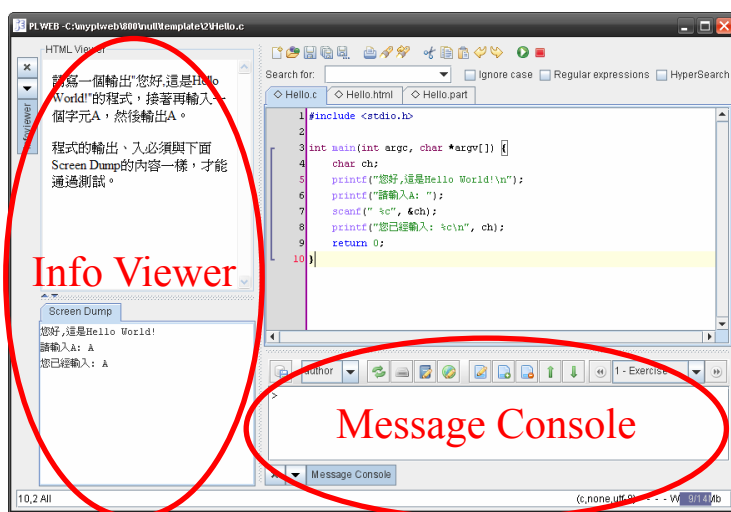


圖 0-4 學習環境的編輯器畫面

五、教材案例實作

若開發的程式可以在 Console 環境運作，教材的設計通常與 Java 相似，可以使用輸出比對或單元測試，驗證學生設計的程式碼是否正確。對於某些網頁與圖形介面的程式設計教學，也能夠以類似的方法建立教材，本章分別以 Java、PHP 及 C# 為例，說明學習環境如何與其它開發工具搭配，並透過屬性及指令的設計，完成支援教學的任務。

5.1 Java 程式設計教學

為驗證學習環境的教材編輯功能，我們使用 Java 程式語言的教學作為示範，實際設計一份教材，並加入三種測試學生作答是否正確的機制。教材中所建立的檔案，其內容在附錄一提供。

5.1.1 軟體安裝與環境設定

編譯及執行 Java 的程式需要安裝 JDK，在這個範例中，使用 Windows 作業系統及 Sun 發行的 JDK 1.6，假設安裝的路徑為「C:\Program Files\java\jdk1.6.0_11」，完成 JDK 的安裝後，需要再設定系統環境變數。

為實現圖形介面的自動化測試，使用者的電腦環境也需要安裝 Abbot Framework，Abbot 未提供自動安裝程式，只需要將檔案解壓縮，假設解壓縮至路徑為「C:\abbot-1.0.2」。

環境變數	設定值
JAVA_HOME	C:\Program Files\java\jdk1.6.0_11
ABBOT_HOME	C:\abbot-1.0.2
CLASSPATH	.;%JAVA_HOME%\lib;%ABBOT_HOME%\lib;%ABBOT_HOME%\lib\costello.jar
PATH	%JAVA_HOME%\bin;%PATH%

表 0-1Java 與 Abbot 相關系統環境變數設定

5.1.2 教學案例描述

為說明 Java 程式在教材中如何被測試，這個範例要求學生設計一個 GCD 類別，提供接受兩個整數參數的 gcdOf 方法，計算並傳回兩個整數的最大公因數值。同時也要求學生使用 Swing 設計視窗圖形介面，加入兩個文字方塊，讓程式的使

用者輸入整數資料，並提供開始計算的按鈕，在按鈕點擊後，讀取使用者輸入的資料，轉換為整數值，並呼叫 `gcdOf` 方法計算兩個整數的最大公因數，計算結果則顯示在第三個文字方塊。假設使用者輸入的整數為 99 及 72，則顯示計算結果為 9，同時也在 Console 輸出「`gcd(99,72)=9`」的文字訊息。

由學生完成的程式碼，在編譯之後，將被進行三種測試。第一種測試是檢查 `gcdOf` 方法的單元測試，由多組測試資料進行檢驗，學生設計的程式若沒有實作正確的處理邏輯，就難以通過這項單元測試。第二種測試由學生自行輸入資料，輸入的資料必須與題目提供的測試資料相符，程式執行後的輸出結果將受到檢驗，必須與題目要求的輸出訊息相符。第三種測試為圖形介面的功能測試，測試工具將開啓學生的程式，自動在文字方塊填入測試資料，再自動點擊按鈕，並檢查圖形介面上的改變是否符合預期。

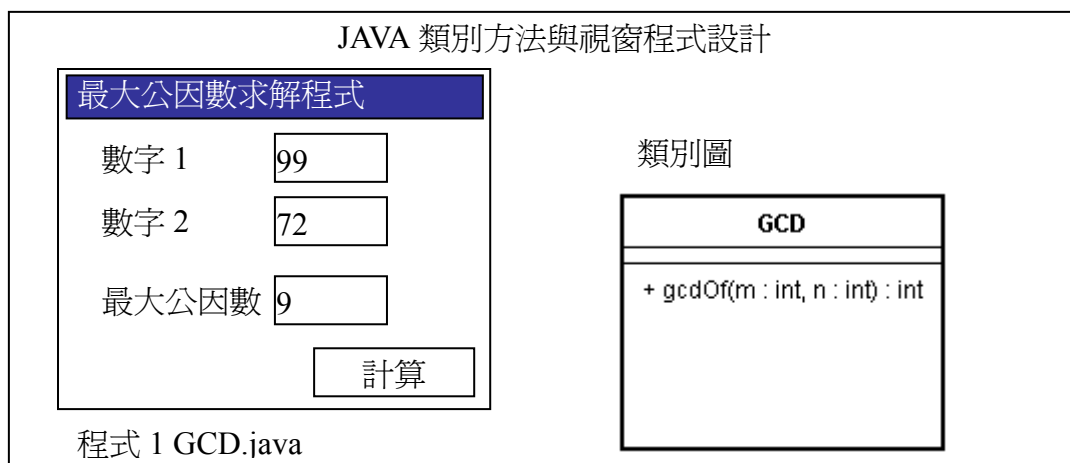


圖 0-1 Java 教學案例描述示意圖

5.1.3 教材實作

在這個教學案例中，作者設計一個 `GCD` 類別，使用網頁說明題目的要求，並提供部份程式碼，讓學習者將程式碼完成。主程式碼的檔案名稱訂為「`GCD.java`」，部份程式碼放置在「`GCD.part`」，而題目說明的網頁則是「`GCD.html`」。作者完成的「`GCD.java`」程式碼即標準答案，在完成編譯後可以執行，並將程式的文字輸出訊息儲存為「`GCD.cond`」。在編輯器的畫面中，題目說明將顯示在左上方的瀏覽器，而文字輸出訊息則顯示在左下方的檢視器。

在學習者開始使用這份教材時，主程式碼的內容被作者提供的部份程式碼取代，學習者需要修改程式碼，讓程式的功能符合題目描述，並且文字輸出訊息也

必須與作者的程式相同，才能完成這個練習的要求。

可以將此教材設計需求撰寫成以下的 Task 屬性表。

Property Name	Property Value	Note
askfor.ExName	請輸入程式碼檔案名稱(不含副檔名)	複製題目設定提示
ExName	GCD	程式碼主檔名
label.view.0	Screen Dump	輸出提示名稱
file.view.0	\${ExName}.cond	輸出提示檔案
file.main	\${ExName}.java	主程式碼
file.part	\${ExName}.part	部份解答
file.html	\${ExName}.html	題目描述
file.author.0	\${ExName}.html	作者編輯題目描述
file.author.1	\${ExName}.part	作者編輯部份解答

表 0-2 Java 教學案例的屬性設定

藉由 Command 的定義，可以協助作者及學生進行編譯、執行，使用 javac 指令將程式碼編譯，並且在使用 java 執行時，同時加上「execdump」的工具指令將程式輸出結果儲存為文字檔案，「execdump」工具可以擷取 Windows Command-line 程式的輸出畫面，並儲存為指定名稱的文字檔案。作者執行程式的輸出儲存為「GCD.cond」，而學生執行程式的輸出則儲存為「GCD.dump」，採用不同的檔案命名以方便後續的測試比對。

Mode	Type	Command
author	compile	javac \${ExName}.java
author	Cond	start /WAIT execdump \${ExName}.cond "java \${ExName}"
student	compile	javac \${JAVAC_ARG} \${ExName}.java
student	Run	start /WAIT execdump \${ExName}.dump "java \${ExName}"

表 0-3 Java 教學案例的指令設定

作者模式的教材編輯器提供 Task Editor 功能，可以編輯屬性及指令的設定，將教材上傳儲存時，這些設定將轉換為 XML 資料保存。

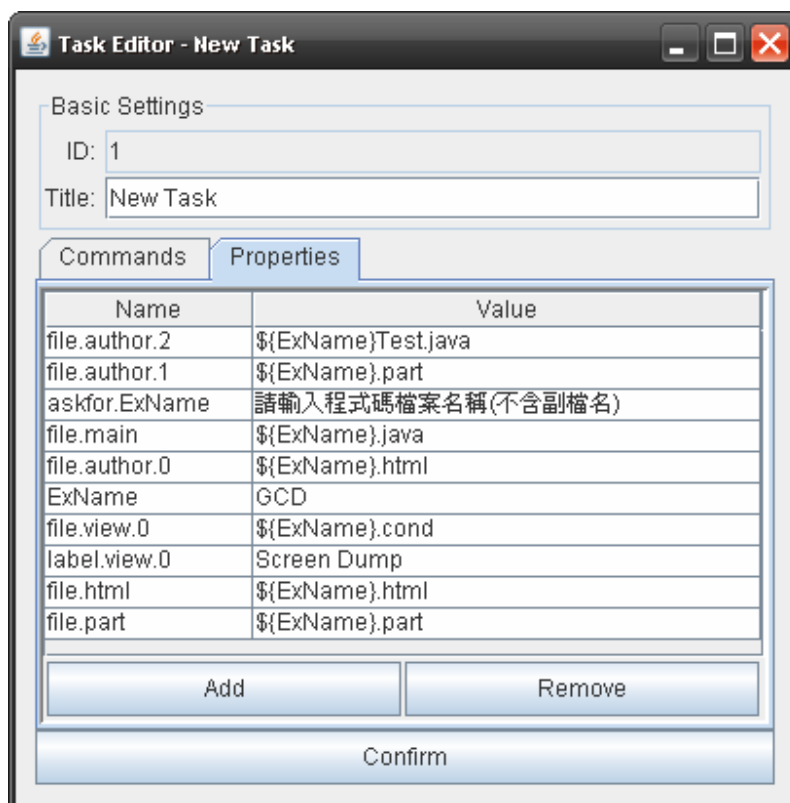
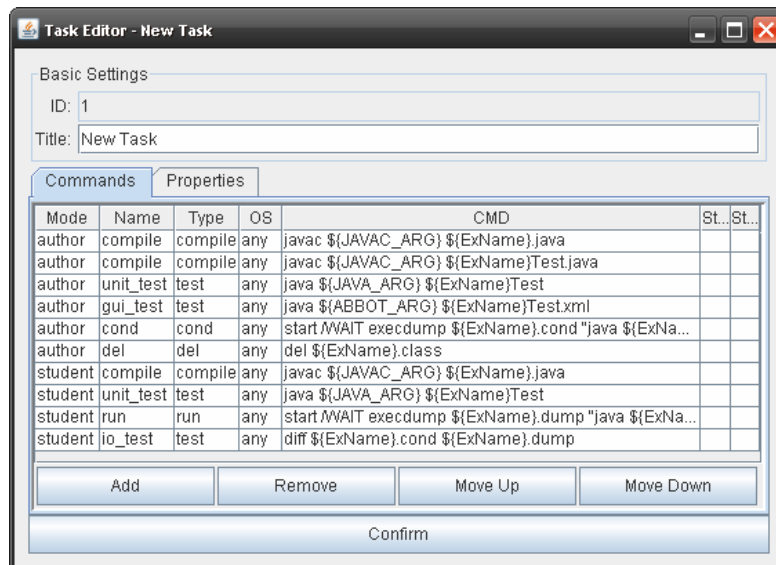


圖 0-2 教材編輯器的 Task Editor 功能

完成屬性及指令的設定後，編輯器會依照屬性設定載入需要編輯的檔案，作者的編輯工作就是完成這些檔案的內容，並透過「執行」產生及測試編輯的結果。

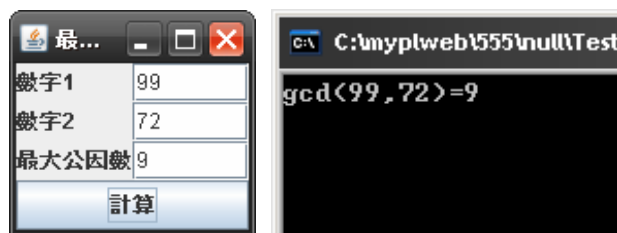
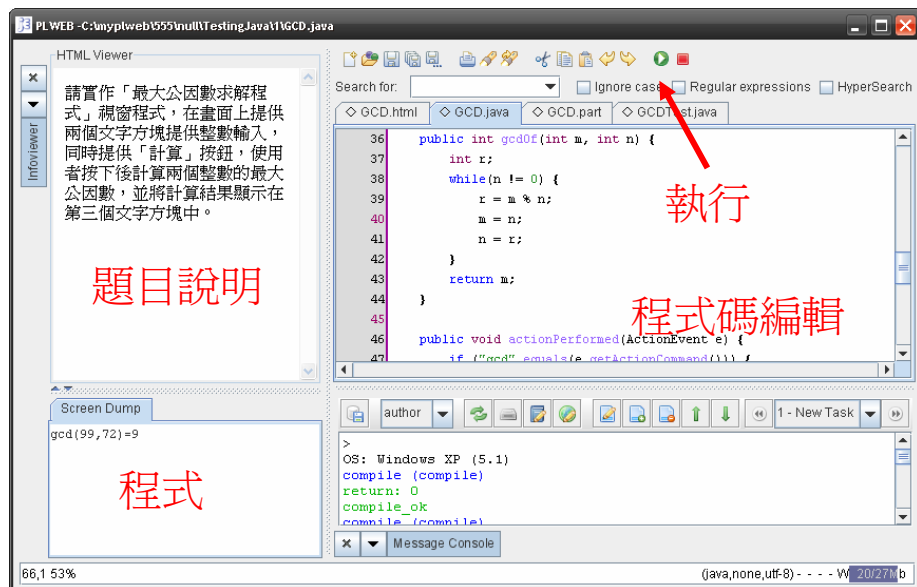


圖 0-3 JAVA 教材的編輯器畫面

5.1.4 測試案例

單元測試需要增加一個測試程式碼，由作者完成程式碼的內容，這個檔案命名為「GCDTest.java」，在 Task 屬性中需要增加相關設定。在這個案例中，Java 的單元測試使用 JUnit Framework，以 assertEquals 進行資料的比對，例如以下的片段程式碼，用兩組測試資料進行對照，檢驗學生設計的 gcdOf 方法是否正確。

- assertEquals(gcdOf(10, 5), gcd.gcdOf(10, 5));
- assertEquals(gcdOf(99, 72), gcd.gcdOf(99, 72));

Property Name	Property Value	Note
file.author.2	\${ExName} Test.java	作者編輯測試程式

表 0-4 測試 Java 教學案例的屬性設定

撰寫圖形介面的自動化測試程式相當麻煩，但 Abbot Framework 提供 Costello Test Script Editor 的工具，可以協助這項工作的進行。Costello 提供動作擷取的功能，由使用者手動操作目標視窗，過程被錄製成每一道測試步驟，這項自動產生

的結果，需要再經過一些調整，才能達到教材要求的自動化測試。Costello 的編輯結果以 XML 檔案保存，可以重複用來進行測試。

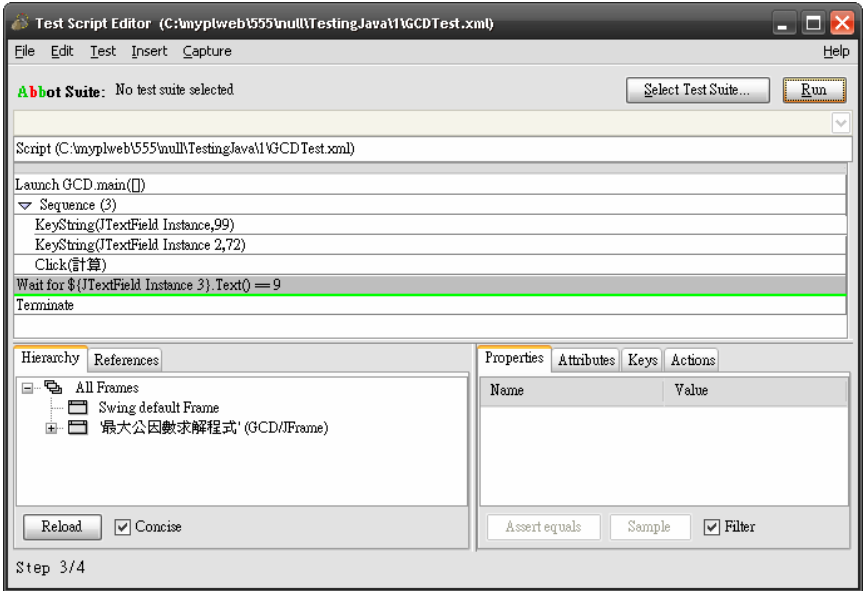


圖 0-4 Abbot Framework 的 Test Script Editor 畫面

輸出結果的比對使用「diff」工具程式，兩個文字檔案內容若一致，則視為通過測試。上述的測試需要增加教材的 Task 指令設定，單元測試使用的測試程式碼，需要在作者模式中進行編譯，而學生模式則需要加入執行三種測試的指令。

Mode	Type	Command
author	compile	Javac \${ExName} Test.java
student	unit_test	java \${JAVA_ARG} \${ExName} Test
student	gui_test	java junit.extensions.abbot.ScriptFixture \${ExName} Test.xml
student	io_test	diff \${ExName}.cond \${ExName}.dump

表 0-5 測試 Java 教學案例的指令設定

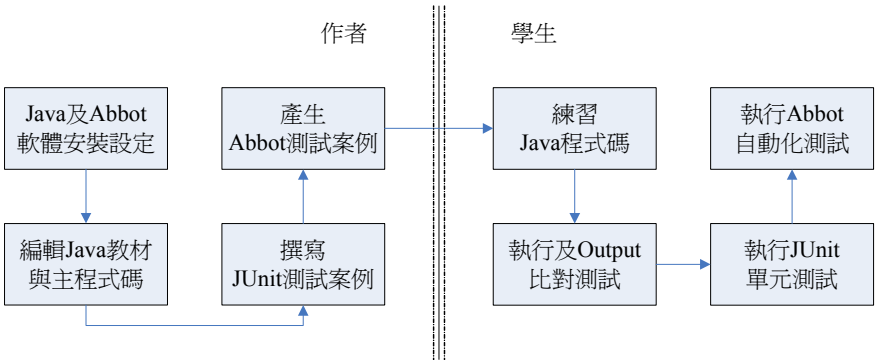


圖 0-5 Java 教材編輯、使用與測試流程

5.2 PHP 網頁程式設計教學

PHP 是熱門的動態網頁開發工具之一，使用 PHP 建立的網頁應用程式，需要搭配伺服器，才能提供使用者以瀏覽器操作。驗證學生撰寫的 PHP 網頁程式，需要配合 Web Testing 的實作，讓測試程式以模擬瀏覽器操作行為的方式，判斷程式是否具有正確的功能。這份教材建立的檔案，請參考附錄二。

5.2.1 軟體安裝與環境設定

由於 PHP 必須搭配網頁伺服器，才能進行動態網頁的撰寫與測試，因此必須在 Client 端預先安裝所需的軟體。

可搭配 PHP 程式使用的網頁伺服器有許多種，例如常見的 Microsoft IIS(Internet Information Server)、Apache Web Server 等。這些伺服器雖然有非常多專業的功能，但以教學的需求而言，並不需要太多複雜功能，只要學習過程所撰寫的程式能夠順利執行即可。因此在這邊所選擇的網頁伺服器，是輕量化的 LightTPD 軟體。WLMP 專案(<http://en.wlmp-project.net/>)同時整合 LightTPD、PHP、MySQL 等軟體，並可以在 Windows 系統運作，提供自動化安裝程式，是一套適合學習者使用的工具。

在學習環境的 Client 端安裝完成 WLMP 後，需要調整的設定包括 HTTP 伺服器的 Port 以及根目錄位置。為避免與電腦上已有其他軟體佔用 80 port，因此可由教材作者決定一個特殊的 Port 號碼，學習者只要設定與教材指定值相同即可。由於每位使用者執行編輯器時，教材練習檔案放置的路徑不盡相同，因此也必須設定伺服器的檔案根目錄，此項設定必須在編輯器啟動後進行修改。假設 WLMP 安裝路徑為 C:\WLMP，則可依下表的路徑及說明調整 LightTPD 設定。

設定檔案路徑	C:\WLMP\LightTPD\conf\lighttpd-inc.conf	
設定名稱	設定值	說明
server.port	8000	網頁伺服器連接埠代號
server.document-root	C:/myplweb/教材單元路徑	網頁檔案存放的磁碟路徑

表 0-6 LightTPD 設定說明

為使教材能夠自動化測試學習者撰寫的 PHP 網頁程式，需要內建網頁測試 (Web Testing) 機制。我們使用 Perl 搭配 WWW::Mechanize 套件撰寫 Web Testing Script，如此網頁測試程式的撰寫可以化繁為簡。

ActivePerl(<http://www.activestate.com/activeperl/>)包含 Perl 直譯器及許多套件，可以在 Windows 平台上安裝使用。假設安裝路徑在 C:\Perl，安裝 ActivePerl 完成後，必須確認系統的 PATH 環境變數已包含「C:\Perl\bin;」，並使用 PPM(Perl Package Manager)檢查 WWW-Mechanize 套件是否已安裝。由於 ActivePerl 的安裝程式相當自動化，通常 PATH 環境變數在安裝後就會自動完成設定，並且也內建 WWW-Mechanize 套件，若使用其他的方式安裝 Perl，則可能需要自行處理這些步驟。

5.2.2 教學案例描述

設計一個表單資料處理的練習，使學習者能夠實際完成表單設計，並使用 PHP 程式接收表單傳遞的資料、進行處理及顯示結果。此練習要求學習者設計 HTML 的表單 form1.html，提供姓名、國文/英文/數學成績輸入，並按下送出後將資料傳送給程式 form1.php，程式除顯示表單輸入的資料外，還必須計算成績的平均分數，若平均大於 60 分則顯示「恭喜！及格。」否則顯示「不及格！加油。」的訊息。

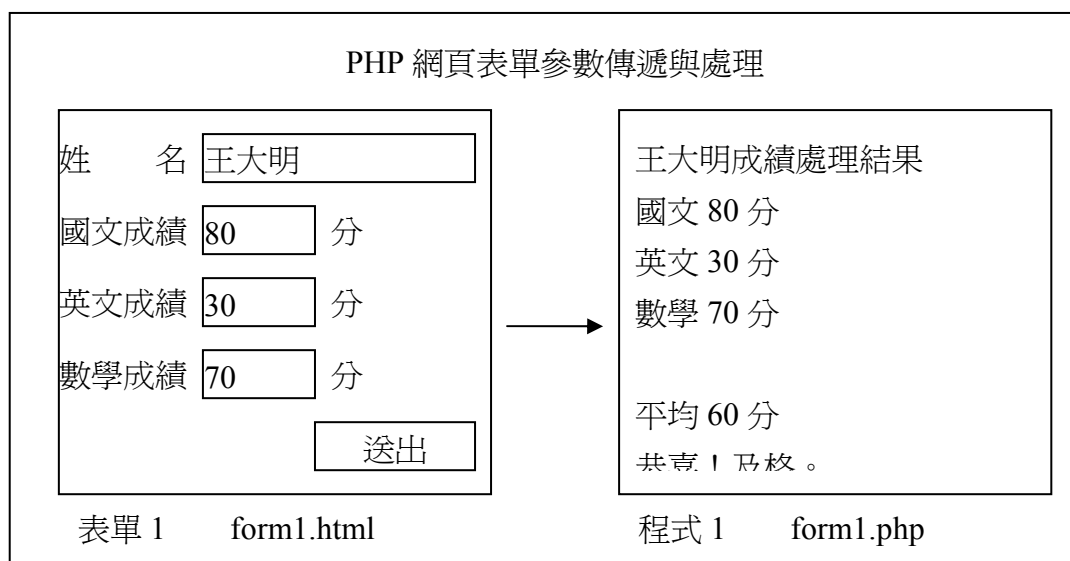


圖 0-6 PHP 教學案例描述示意圖

5.2.3 教材實作

建立一個新的教材單元「動態表單資料處理」，以教材編輯器開啓，透過 open explorer 可以取得教材單元的路徑，例如「C:\myplweb\555\null\TestingPHP\1」，參考 5.1.1 的說明修改 LightTPD 伺服器的根目錄(server.document-root)，將設定值修改為此路徑(分隔符號必須改為倒斜線「/」)。接著啓動 WLMP 服務，用瀏覽器開

啓「http://localhost:8000/」網址，以測試 LightTPD 伺服器是否正常運作。根據教學案例描述完成 Task 屬性設定如下表：

Property Name	Property Value	Note
askfor.ExName	請輸入程式碼檔案名稱(不含副檔名)	複製的設定詢問
ExName	form1	主程式檔案名稱
file.main	\${ExName}.php	主程式碼
file.part	\${ExName}.part	部份解答
file.html	\${ExName}_doc.html	題目描述
file.attach.0	\${ExName}.html	表單網頁
file.author.0	\${ExName}_doc.html	編輯題目描述
file.author.1	\${ExName}.part	編輯部份程式碼

表 0-7 PHP 教材的屬性設計

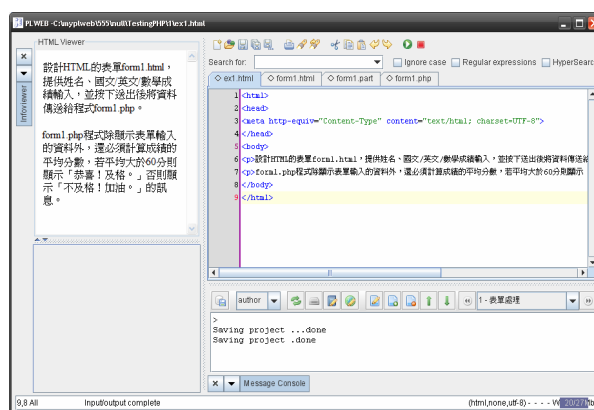


圖 0-7 PHP 教材的編輯畫面

完成以上教學檔案的製作，即可使用瀏覽器開啓網址進行實際測試。在這個範例中，由使用者完成網頁表單的輸入，按下送出後，瀏覽器顯示程式處理後的結果。

測試網址：http://localhost:8000/form1.html

姓 名	王大明	資料送出後	王大明成績處理結果
國文成績	30		國文30分
英文成績	40		英文40分
數學成績	50		數學50分
送出			平均40分 不及格！加油。

圖 0-8 PHP 網頁執行畫面

5.2.4 測試案例

由學生開啓瀏覽器，可以輸入資料並獲得結果，但這個過程發生在瀏覽器與網頁伺服器之間，並無法從中驗證程式的功能。完整的教材需要提供自動化測試，使用 Perl 搭配 WWW-Mechanize 套件，可以撰寫網頁測試程式，藉由模擬瀏覽器的行為，與伺服器的網頁程式進行互動，並判斷網頁程式處理的結果是否正確。

Property Name	Property Value	Note
file.author.2	\${ExName}_test.pl	自動化測試程式

表 0-8 測試 PHP 教材的屬性設定

在學生模式中，網頁測試程式需要被執行，因此需要增加教材的 Task 指令設定。由於 Perl 的程式為直譯式，並不需要編譯步驟，只要一道測試指令即可完成。

Mode	Type	Command
student	Test	perl \${ExName}_test.pl

表 0-9 測試 PHP 教材的指令設定

假設被測試的網址為「http://localhost/form1.html」，以下的 Perl 片段程式碼示範網頁資料如何存取。對於網頁資料的檢驗，則可以使用 Regular Expression。

- \$html = \$mech->get('http://localhost/form1.html') ->content();
- if (\$html =~ m/^text patterns\$/) { print "通過測試\n"; }

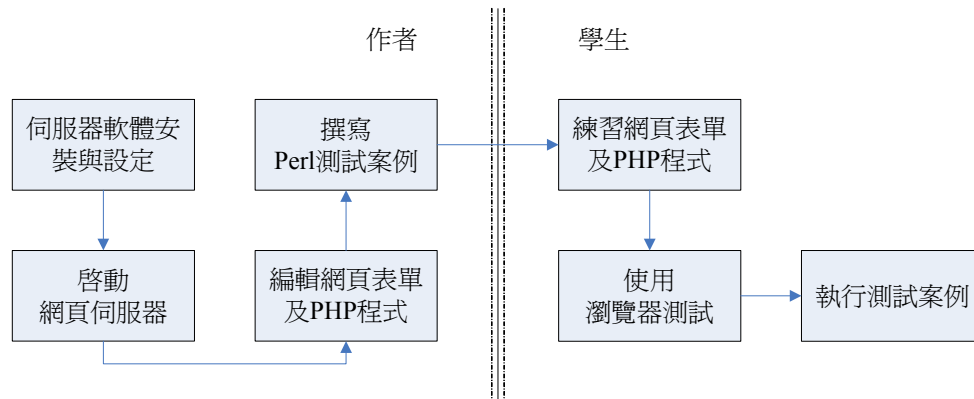


圖 0-9 PHP 教材編輯、使用與測試流程圖

5.3 C#視窗介面程式設計

微軟的.NET 也是時下熱門的開發工具，為此平台所推出的 C#程式語言，可以建立視窗或網頁應用程式。Visual Studio .NET 軟體提供視覺化的 C#整合開發環

境，擁有功能強大的工具支援。儘管學習環境所提供的編輯器，無法整合這類型的開發工具，但仍可以在某些教學案例中，搭配其他工具建立 C#視窗程式設計的教材。建立 C#教材的相關檔案，在附錄三提供。

5.3.1 軟體安裝與環境設定

開發 C#程式需要安裝微軟的 .NET Framework SDK，它包含撰寫、建置、測試及部署 .NET 應用程式需要的工具。用人工方式撰寫 GUI 的程式碼，相當耗時耗力，因此需要搭配其它整合開發工具，以專用的 UI 設計工具完成 GUI 畫面，可以滿足這種需求的軟體，有 Visual C# .NET、Mono Developer 及 SharpDevelop 等。這裡的選擇是較簡單易用的 SharpDevelop，它提供所視即所得的開發方式，可以建立視窗介面應用程式，雖然功能不如 Visual C#，但擁有開放原始碼、使用免付費的好處，且足以滿足教材開發的需求。

由於 .NET 開發的專案包含許多檔案，直接使用指令編譯及部署這些檔案並不容易，為了簡化建立可執行程式的過程，這個教材需要搭配 NAnt 工具，藉由 NAnt 的設定檔，教材可以定義那些程式碼需要被編譯，以及需要產生那些可執行程式。假設 NAnt 安裝的路徑為「C:\nant-0.86-beta1」，則系統需要將「C:\nant-0.86-beta1\bin」加入 PATH 的環境變數。

在這個範例的測試機制，並不使用 GUI 自動化測試的工具，而是在提供給學生的程式碼中，加入視窗狀態紀錄的功能，使程式在執行的過程中，在背景將視窗的狀態儲存為文字檔，進而比對作者及學生的程式所產生的紀錄，當兩個紀錄檔案內容一致時，就表示學生的程式碼通過測試。

5.3.2 教學案例描述

「視窗 1」提供姓名、國文/英文/數學成績輸入的文字方塊，完成「成績處理」按鈕的事件程式碼，使成績處理結果在右方的文字方塊中顯示。顯示的訊息必須包含使用者輸入的資料，並計算三個科目成績的平均分數，若平均大於 60 分則附加顯示「恭喜！及格。」的訊息，否則顯示「不及格！加油。」的訊息。

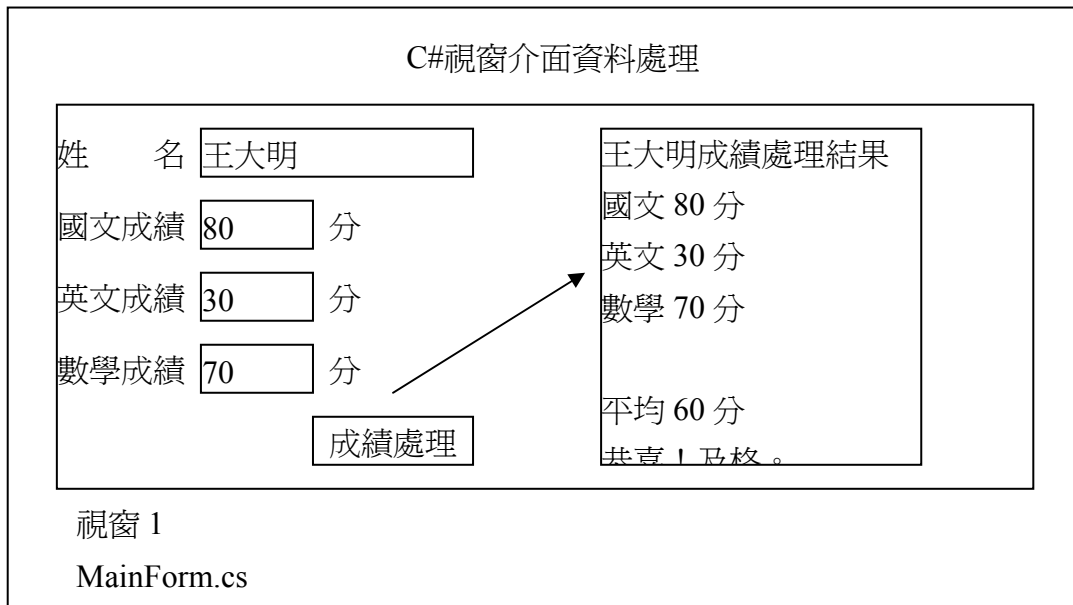


圖 0-10 學習環境的編輯器畫面

5.3.3 教材實作

首先使用 SharpDevelop 軟體建立新專案，完成視窗程式的介面與程式碼，並且將相關檔案放置在教材的路徑中。在這個範例中，建立的檔案包含「MainForm.cs」、「MainFormDesigner.cs」、「MainForm.resx」、「Program.cs」，在 SharpDevelop 軟體編譯、執行，確認功能正常後，就可以開始將這個專案轉換為教材中的練習題。

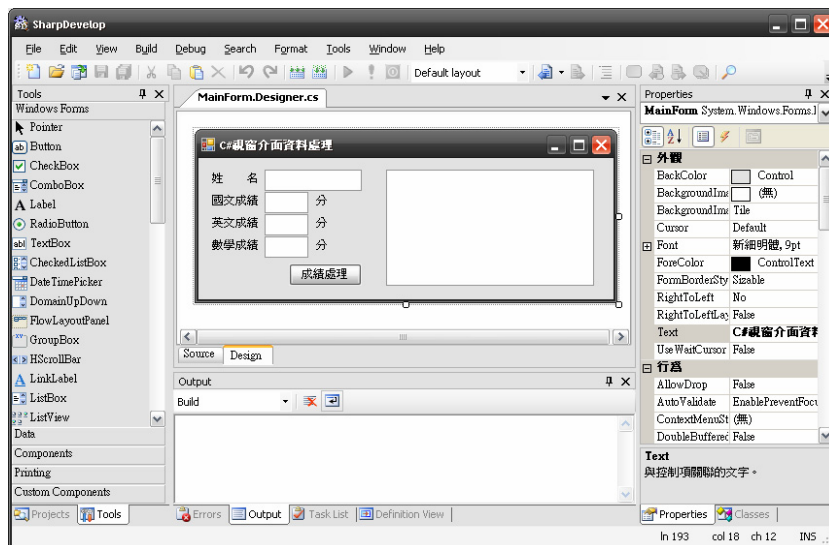


圖 0-11 SharpDevelop 視窗編輯的畫面

由於一個 C#視窗程式專案會包含多個檔案，因此需要以資料夾放置這些檔案。爲了方便後續能夠以複製的方式建立其他題目，程式碼的檔案名稱皆爲固定，而資料夾名稱則視各練習的設定。在這個範例中，使用 ex01 資料夾保存 C#專案的程式碼，並建立一個名爲「ex01.build」的 NAnt 設定檔，定義 ex01 資料夾下的*.cs 檔案需要編譯，並且產生「ex01.exe」的可執行程式。

Property Name	Property Value	Note
askfor.ExName	請輸入程式碼檔案名稱(不含副檔名)	複製的設定詢問
ExName	Ex01	主程式檔案名稱
file.main	\${ExName}/MainForm.Designer.cs	主程式碼
file.part	\${ExName}/MainForm.Designer.part	部份解答
file.html	\${ExName}.html	題目描述
file.author.0	\${ExName}.html	編輯題目描述
file.author.1	\${ExName}.build	編輯 NAnt 設定檔案
file.author.2	\${ExName}/MainForm.Designer.part	編輯相關程式碼
file.author.3	\${ExName}/Program.cs	編輯相關程式碼
file.author.4	\${ExName}/MainForm.cs	編輯相關程式碼

表 0-10 C#教材的屬性設定

教材只需要爲作者及學生增加兩種指令設定，第一個 build 指令用來產生可執行檔，第二個 run 指令則執行建立的程式，執行的畫面如圖 5-7 所示。對於修改幅度較小的調整，作者並不需要再使用 SharpDevelop 編輯，直接在編輯器中修改程式碼即可。

Mode	Type	Command
author	build	nant -buildfile:\${ExName}.build
author	run	\${ExName}.exe
student	build	nant -buildfile:\${ExName}.build
student	run	\${ExName}.exe

表 0-11 C#教材的指令設定

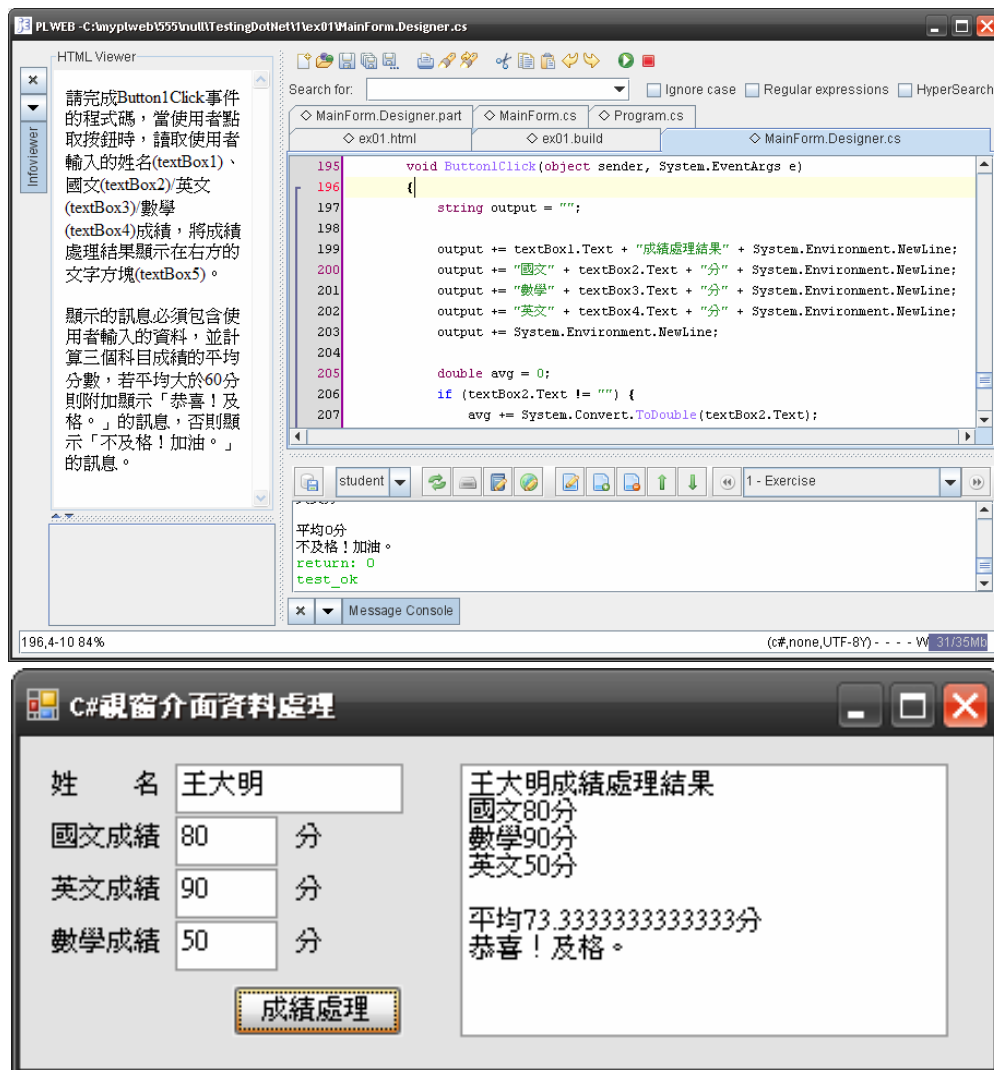


圖 0-12 C#教材的編輯畫面

5.3.4 測試案例

教材對於學生撰寫的程式，並不是採用全自動的測試方式，所以學生需要依照題目的指示輸入測試資料。為加入測試的功能，作者需要修改 `MainForm.cs` 的程式碼的 `MainFormFormClosing` 函式，使程式視窗在結束前，可以將視窗的狀態儲存為一個文字檔。在這個範例中，右方的文字方塊內容，將被儲存為「`ex01.dump`」文字檔案，程式碼的片段如下所示。

- `File.WriteAllText("ex01.dump", textBox5.Text, Encoding.GetEncoding("big5"));`

在作者模式中，以一道複製指令將程式產生的「`ex01.dump`」檔案複製為

「ex01.cond」，作為正確程式輸出的對照檔。學生在執行程式時，同樣產生「ex01.dump」檔案，此時使用「diff」工具與「ex01.cond」進行比對，即可知道學生撰寫的程式是否正確運作。

Mode	Type	Command
author	copy	copy \${ExName}.dump \${ExName}.cond
student	test	diff.exe \${ExName}.cond \${ExName}.dump

表 0-12 C#教材的指令設定

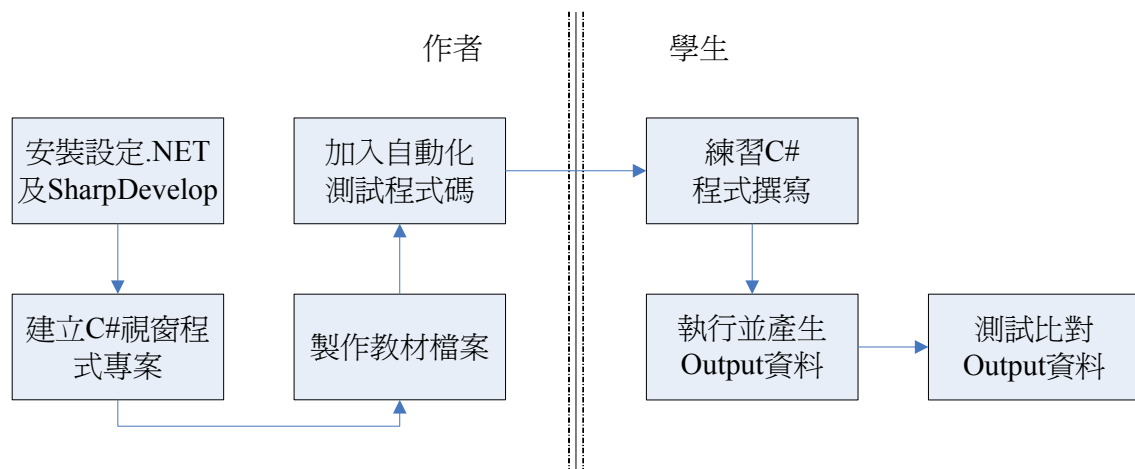


圖 0-13 C#教材編輯、使用與測試流程圖

六、使用狀況

本章節介紹教學平台的使用情況，在 2008 年至 2009 年間，研究中所提出的教材架構，在多門課程中實際用來進行教材開發，並且由學生使用這些教材所提供的練習。這些應用實例，將區分為課程與測驗兩種類型，在以下兩節列出。

6.1 程式語言教學課程

在 2008~2009 年間，共有 11 間學校的 13 個系所，使用本研究建構的教學平台，製作開發 C/C++、Java、Scheme 等程式語言的教材，作為程式語言相關課程的輔助教學工具。

年度	課程名稱	學校/系所	人數
2008	程式設計(一)	弘光科技大學/資訊管理系	58
2008	程式設計_二資一 A	吳鳳技術學院/資訊管理系	33
2008	JAVA 程式設計(一)(二)	僑光技術學院/資訊管理系	106
2008	程式設計(一) 進修部二專一 A	吳鳳技術學院/資訊管理系	33
2008	Java(KYHuang)_4 資 3A 下學期	嶺東科技大學/資訊管理系	35
2008	Java(KYHuang)_2 資 4B 下學期	嶺東科技大學/資訊管理系	47
2008	資料結構程式設計	中州技術學院/資訊管理系	28
2008	程式語言	雲林科技大學/資訊管理系	45
2008	程式設計	親民技術學院/資訊管理系	52
2008	物件導向程式設計	中州技術學院/資訊管理系	48
2008	程式設計觀念與技巧	雲林科技大學/資訊管理系	77
2008	程式語言導論 02	中正大學/資訊工程學系	28
2008	PL(程式語言)	中正大學/資訊工程學系	67
2008	Java(KYHuang)_4 資 3A 上學期	嶺東科技大學/資訊管理系	65
2008	資料結構	雲林科技大學/電腦通訊工程系	55
2008	Java(KYHuang)_2 資 4B 上學期	嶺東科技大學/資訊管理系	51
2009	程式語言	雲林科技大學/資訊管理系	82
2009	程式語言教學	雲林科技大學/資訊管理系	40
2009	程式設計	雲林科技大學/資訊管理系	90

2009	程式設計	育達商業技術學院/資訊管理系	69
2009	物件導向程式設計	中正大學/資訊工程學系	55
2009	計算機概論(二)	雲林科技大學/電子系	109
2009	程式設計	親民技術學院/資訊管理系	78
2009	程式設計實驗課程(一)	雲林科技大學/資訊管理系	26
2009	程式設計實驗課程(二)	雲林科技大學/資訊管理系	25
2009	程式設計實驗課程(三)	雲林科技大學/資訊管理系	28
2009	程式設計	育達商業技術學院/資訊管理系	55

表 0-1 教學平台課程數量統計

6.2 程式設計能力測驗

在受限制的網路環境中，教學平台可以支援測驗功能，在 2009 年共舉辦以下兩個測驗場次。

年度	課程名稱	主辦單位	人數
2009	JAVA 程式競賽	雲林科技大學資訊管理系	21
2009	Java 程式設計實作測驗	中區教學卓越計畫-程式語言學習子計畫	24

表 0-2 測驗數量統計

測驗採集中方式，讓學生在電腦教室進行。硬體的配置需要防火牆，限制學生只能瀏覽測驗平台的網站，若測驗允許參考特定的網路資源時，也能以防火牆設定開放存取。軟體環境需要考慮便利及公平，提供測驗所需的相關開發工具、設定及文件，使用免設定的作業系統可開機光碟，每位學生所使用的軟體環境皆相同且受到控管，這項需求可以使用客製化的 Linux Live CD 達成。

七、結論與未來研究

本章共有兩節，在 7.1 節列出研究的結論與貢獻，而 7.2 節則提供未來可以研究的方向。

7.1 結論

程式語言的教學，需要耗費許多時間準備教材，而傳統教材的設計，並不容易讓學生獲得充分的練習，有些課程甚至需要 1~2 堂課的時間，才能讓學生瞭解開發環境的安裝和設定，以及如何撰寫程式及除錯。一個電腦化的學習環境，應該能夠排除旁枝末節的瑣事，讓教學與學習者能夠更專注在目標上，達到更好的學習成果。

一個好的學習環境，需要有好的教材，在教材的質與量充足的情況下，才能發揮教學的效益。這個研究改善教材製作的困境，讓教學者能夠事半功倍，而學習者也因此能夠得到更好的學習資源。同時，這個研究也提供一個容易擴充的架構，試圖拋磚引玉，讓未來的研究者，能夠在一個功能齊備的平台上，設計出新的教學應用，而不必重新開發教學平台的基礎設施。

這項研究的成果，包含一個教學平台的實作，以及一個容易擴充的教材架構，對於程式語言教學的研究，可以達成本研究設定的兩項目標：

1. 支援多種程式語言。由教材設計者或教材樣板開發者，對於新的程式語言，能夠自訂合適的屬性及指令，以支援新程式語言的教學需求。各種程式語言的教材，可以產生相同的學習紀錄，由教學平台呈現統計結果。
2. 內容產生機制。教材的作者及學生，可以使用相同的整合開發環境，藉由程式設計專用的編輯器，搭配教材編輯的附屬功能，讓作者在完成一項程式開發的專案後，就可以將它作為一份教材提供給學生練習，教材內容的生產效率因此提高。

7.2 未來研究

透過開放 API 介面，讓教學平台成為一個網路服務，方便與其它應用整合。例如，運用多媒體的設計，讓教材的呈現更加豐富多元，在聲音及影像的播放過程中，結合程式碼即時練習的功能，讓學習環境增加互動的效果。方便嵌入新的程式碼測試與檢查工具，自動化提供評量結果。學習紀錄可以由第三方程式存取，

以作資料探勘等其他應用。

有關自動化測試的功能，尚有許多需要克服的地方：

1. GUI 與 Web 的程式設計教材，在測試案例的製作上，目前仍需要由作者撰寫測試用的代碼，即使是簡單的小程式，也需要完成許多測試才能知道功能是否正確，而這些測試的過程需要由作者耗費心力。一個理想的自動化結果，應該能夠分析作者提供的程式，並記錄作者的標準操作過程，由程式產生一份清單，讓作者選擇必要的測試項目。
2. 網頁程式設計所產生的 HTML 結果，存在不容易測試的困難，因為在瀏覽器中的相同顯示結果，可以有非常多種 HTML 代碼的不同寫法，特別是不符合 XHTML 規範的代碼，更有許多難以預期的情況會發生，這個問題增加網頁自動化測試程式的設計困難。
3. 單純的 Console 程式設計，也會存在許多不容易測試的困難，例如教材希望學生使用某一種形式的迴圈或遞迴實作，而學生在不依照題意指示的情況下，仍然撰寫出可以通過測試的程式碼。即使是比對兩個文字輸出是否一致，也存在嚴格程度的問題，例如空白字元、換行符號的使用，嚴格的檢查，可能造成學生練習時的困擾，但寬鬆的檢查又無法滿足某些題目的特別需求。由作者提供更多設定，雖然可以解決這些問題，但同時又帶來教材製作效率的問題。

參考文獻

中文部分

1. 林宗德、董少桓，2006，Java 程式設計教學平台，物件導向技術及應用研討會。
2. 資策會，2008，2008~2010 台灣產業科技人才供需調查-資訊服務產業。
3. 王志弘，2004，支援二進位資料的 XML 物件儲存系統。
4. 徐文杰、林沛傑，2003，數位學習標準與 SCORM 的發展，圖書館學與資訊科學，第二十九卷第一期。
5. 林信成，1999，XML 相關技術與下一代 Web 出版趨勢之研究，教育資料與圖書館學，第三十七卷，第二期，頁 185。
6. 李開復，2008，給中國學生的第四封信---大學四年應該這麼度過，
<http://www.5xue.com/modules/wordpress/?p=1255>。

英文部分

7. ADL, SCORM Documents – What Is SCORM,
<http://www.adlnet.gov/technologies/scorm/> .
8. António Mendes, Valentino Ivanov, Maria Marcelino, 2005, A Web-Based System to Support Java Programming Learning, International Conference on Computer Systems and Technologies.
9. Apache, Ant User Manual, <http://ant.apache.org/manual/index.html> .
10. BlueJ, The interactive Java environment, <http://www.bluej.org/index.html> .
11. Free Software Foundation, GNU Make, <http://www.gnu.org/software/make/> .
12. IEEE, 2002, Draft Standard for Learning Object Metadata,
http://ltsc.ieee.org/doc/wg12/LOM_WD6_4.pdf .
13. JUnit, Resource for Test Driven Development, <http://www.junit.org/> .
14. Kent Beck, 1999, Extreme Programming Explained: Embrace Change, Addison Wesley.
15. Stephen Baudendistel, Grace Hua, 1987, The Desktop Interface in Intelligent Tutoring Systems.
16. Stuart Feldman, 1979, Make --- A Program for Maintaining Computer Programs.
17. TIOBE, TIOBE Programming Community Index,

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> .

18. W3C, XML Schema Part 2, <http://www.w3.org/TR/xmlschema-2/> .

附錄一、JAVA 教材設計附檔

GCD.html

GCD.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<p>請實作「最大公因數求解程式」視窗程式，
在畫面上提供兩個文字方塊提供整數輸入，
同時提供「計算」按鈕，
使用者按下後計算兩個整數的最大公因數，
並將計算結果顯示在第三個文字方塊中。</p>
</body>
</html>
```

GCD.java

GCD.java

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class GCD extends JFrame implements ActionListener {

    private JTextField text1;
    private JTextField text2;
    private JTextField text3;
```

```

public GCD() {

    JPanel p1 = new JPanel(new GridLayout(0, 2));

    p1.add(new JLabel("數字 1"));
    p1.add(text1 = new JTextField());

    p1.add(new JLabel("數字 2"));
    p1.add(text2 = new JTextField());

    p1.add(new JLabel("最大公因數"));
    p1.add(text3 = new JTextField());

    add(p1, BorderLayout.CENTER);

    JButton b1 = new JButton("計算");
    b1.setActionCommand("gcd");
    b1.addActionListener(this);
    add(b1, BorderLayout.SOUTH);

    setTitle("最大公因數求解程式");
    pack();

}

public int gcdOf(int m, int n) {
    int r;
    while(n != 0) {
        r = m % n;
        m = n;
        n = r;
    }
}

```

```

        }
        return m;
    }

    public void actionPerformed(ActionEvent e) {
        if ("gcd".equals(e.getActionCommand())) {
            try {
                int m = Integer.valueOf(text1.getText());
                int n = Integer.valueOf(text2.getText());
                int r = gcdOf(m, n);
                System.out.println("gcd("+m+", "+n+")="+r);
                text3.setText(String.valueOf(r));
            }
            catch (Exception ex) {
            }
        }
    }

    public static void main(String args[]) {
        GCD gcd = new GCD();
        gcd.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        gcd.setVisible(true);
    }
}

```

GCD.part

GCD.part
<pre> import javax.swing.*; import java.awt.event.*; import java.awt.*; </pre>


```
public class GCD extends JFrame implements ActionListener {
```

```
    private JTextField text1;
```

```
    private JTextField text2;
```

```
    private JTextField text3;
```

```
    public GCD() {
```

```
        JPanel p1 = new JPanel(new GridLayout(0, 2));
```

```
        p1.add(new JLabel("數字 1"));
```

```
        p1.add(text1 = new _____());
```

```
        p1.add(new JLabel("數字 2"));
```

```
        p1.add(text2 = new _____());
```

```
        p1.add(new JLabel("最大公因數"));
```

```
        p1.add(text3 = new _____());
```

```
        add(p1, BorderLayout.CENTER);
```

```
        JButton b1 = new JButton("計算");
```

```
        b1.setActionCommand(_____);
```

```
        b1.addActionListener(this);
```

```
        add(b1, BorderLayout.SOUTH);
```

```
        setTitle("最大公因數求解程式");
```

```
        pack();
```

```
    }
```

```
    public int gcdOf(int m, int n) {
```

```

        //計算最大公因數
    }

    public void actionPerformed(ActionEvent e) {
        if (_____.equals(e.getActionCommand())) {
            try {
                //取得輸入並使用 gcdOf 方法進行計算
            }
            catch (Exception ex) {
            }
        }
    }

    public static void main(String args[]) {
        GCD gcd = new GCD();
        gcd.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        gcd.setVisible(true);
    }
}

```

GCDTest.java

GCDTest.java

```

import junit.framework.*;
import junit.textui.*;

public class GCDTest extends TestCase {
    public void testGCD() {
        GCD gcd = new GCD();
        assertEquals(gcdOf(10, 5), gcd.gcdOf(10, 5));
        assertEquals(gcdOf(99, 72), gcd.gcdOf(99, 72));
        gcd.dispose();
    }
}

```

```

    }
    public static int gcdOf(int m, int n) {
        int r;
        while(n != 0) {
            r = m % n;
            m = n;
            n = r;
        }
        return m;
    }
    public static void main(String[] args) {
        int error = 0;
        TestSuite suite = new TestSuite(GCDTest.class);
        TestResult r = TestRunner.run(suite);
        if (!r.wasSuccessful()) {
            error = r.errorCount() + r.failureCount();
        }
        System.exit(error);
    }
}

```

GCDTest.xml

GCDTest.xml
<pre> <?xml version="1.0" encoding="UTF-8"?> <AWTTestScript> <component class="javax.swing.JLayeredPane" id="JLayeredPane Instance" index="1" parent="JRootPane Instance" window="最大公因數求解程式" /> <component class="javax.swing.JPanel" hOrder="1" id="JPanel Instance" index="0" parent="JPanel Instance 2" window="最大公因數求解程式" /> <component class="javax.swing.JPanel" hOrder="1" id="JPanel Instance 2" index="0" parent="JLayeredPane Instance" window="最大公因數求解程式" /> </pre>

```

<component class="javax.swing.JRootPane" id="JRootPane Instance" index="0" parent="最大公因
數求解程式" />
<component class="javax.swing.JTextField" id="JTextField Instance" index="1" parent="JPanel
Instance" window="最大公因數求解程式" />
<component class="javax.swing.JTextField" id="JTextField Instance 2" index="3" parent="JPanel
Instance" window="最大公因數求解程式" />
<component class="javax.swing.JTextField" id="JTextField Instance 3" index="5" parent="JPanel
Instance" window="最大公因數求解程式" />
<component class="GCD" id="最大公因數求解程式" root="true" title="最大公因數求解程式" />
<component class="javax.swing.JButton" id="計算" index="1" parent="JPanel Instance 2" text="計
算" window="最大公因數求解程式" />
<launch args="[]" class="GCD" classpath="C:\myplweb\555\null\TestingJava\1" method="main" />
<sequence>
  <action args="JTextField Instance,99" method="actionKeyString" />
  <action args="JTextField Instance 2,72" method="actionKeyString" />
  <action args="計算" class="javax.swing.AbstractButton" method="actionClick" />
</sequence>
<wait component="JTextField Instance 3" method="getText" value="9" />
<terminate />
</AWTTestScript>

```

附錄二、PHP 教材設計附檔

form1_doc.html

form1_doc.html
<pre><html> <head> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> </head> <body> <p>設計 HTML 的表單 form1.html，提供姓名、國文/英文/數學成績輸入，並按下送出後將資料傳送給程式 form1.php。</p> <p>form1.php 程式除顯示表單輸入的資料外，還必須計算成績的平均分數，若平均大於 60 分則顯示「恭喜！及格。」否則顯示「不及格！加油。」的訊息。 </p> </body> </html></pre>

form1.html

form1.html
<pre><html> <head> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> </head> <body> <form action="form1.php" method="post"> 姓 名 <input name="NAME" />
 國文成績 <input name="CHI" />
 英文成績 <input name="ENG" />
 數學成績 <input name="MAT" />
</pre>

```
<input type="submit" value="送出" />
</form>
</body>
</html>
```

form1.php

form1.php

```
<?php
$NAME = trim($_POST['NAME']);
$CHI = intval(trim($_POST['CHI']));
$ENG = intval(trim($_POST['ENG']));
$MAT = intval(trim($_POST['MAT']));

$AVG = ($CHI+$ENG+$MAT) / 3;

if ($AVG >= 60) {
    $MSG = "恭喜！及格。";
}
else {
    $MSG = "不及格！加油。";
}
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<?php echo $NAME; ?>成績處理結果<br/>
國文<?php echo $CHI; ?>分<br/>
英文<?php echo $ENG; ?>分<br/>
數學<?php echo $MAT; ?>分<br/>
```

```
<br/>
平均<?php echo $AVG; ?>分<br/>
<?php echo $MSG; ?>
</body>
</html>
```

form1_test.pl

form1_test.pl

```
use WWW::Mechanize;
use utf8;

binmode(STDIN, ':encoding(big5)');
binmode(STDOUT, ':encoding(big5)');
binmode(STDERR, ':encoding(big5)');

sub form_test {
    ($url, $fields, @tests) = @_ ;

    my $result = 0;

    my $mech = WWW::Mechanize->new();

    print "開啓網頁 $url\n";

    $mech->get( $url );

    print "表單資料測試..\n";

    $mech->submit_form(
        form_name => "",
        fields => $fields,
```

```

        button => "
    );

    print "進行資料比對..\n";

    $html = $mech->content();

    foreach (@tests) {
        print "搜尋「$_」";

        if ($html =~ m/$_/) {
            print "已發現\n";
        }
        else {
            print "未發現\n";
            $result ++;
        }
    }

    $result;
}

$url = 'http://localhost:8000/form1.html';

$errc = 0;

$errc += form_test(
    $url,
    {
        NAME => '王小明',
        CHI => '80',

```



```

        ENG => '70',
        MAT => '60'
    },
    (
        '王小明成績處理結果',
        '國文 80 分',
        '英文 70 分',
        '數學 60 分',
        '平均 70 分',
        '恭喜！及格。'
    )
);

$errc += form_test(
    $url,
    {
        NAME => '李大華',
        CHI => '50',
        ENG => '70',
        MAT => '30'
    },
    (
        '李大華成績處理結果',
        '國文 50 分',
        '英文 70 分',
        '數學 30 分',
        '平均 50 分',
        '不及格！加油。'
    )
);

```

```
if ($errc > 0) {  
    print "共發現 $errc 次錯誤！\n";  
}  
else {  
    print "沒有發現錯誤，測試通過！"  
}  
  
exit($errc);
```

附錄三、C#教材設計附檔

ex01.html

ex01.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<p>請完成 Button1Click 事件的程式碼，當使用者點取按鈕時，讀取使用者輸入
的姓名(textBox1)、國文(textBox2)/英文(textBox3)/數學(textBox4)成績，將成績處
理結果顯示在右方的文字方塊(textBox5)。</p>
<p>顯示的訊息必須包含使用者輸入的資料，並計算三個科目成績的平均分數，
若平均大於 60 分則附加顯示「恭喜！及格。」的訊息，否則顯示「不及格！加
油。」的訊息。</p>
</body>
</html>
```

ex01.build

ex01.build

```
<?xml version="1.0"?>
<project name="test" default="build">
  <property name="nant.settings.currentframework" value="net-3.5"/>
  <property name="src" value="ex01"/>
  <property name="bin" value="."/>
  <property name="basename" value="ex01"/>
  <property name="debug" value="false"/>

  <target name="clean">
```

```

        <delete file="$(bin)/$(basename).exe" />
    </target>

    <target name="build" depends="clean">
        <csc target="winexe" output="$(bin)/$(basename).exe"
debug="$(debug)">
            <sources>
                <include name="$(src)/*.cs" />
            </sources>
        </csc>
    </target>
</project>

```

Program.cs

```

Program.cs

using System;
using System.Windows.Forms;

namespace test
{
    /// <summary>
    /// Class with program entry point.
    /// </summary>
    internal sealed class Program
    {
        /// <summary>
        /// Program entry point.
        /// </summary>
        [STAThread]
        private static void Main(string[] args)
        {

```

```

        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new MainForm());
    }
}
}

```

MainForm.cs

MainForm.cs
<pre> using System; using System.IO; using System.Text; using System.Collections.Generic; using System.Drawing; using System.Windows.Forms; namespace test { /// <summary> /// Description of MainForm. /// </summary> public partial class MainForm : Form { public MainForm() { // // The InitializeComponent() call is required for Windows Forms designer support. // InitializeComponent(); </pre>

```

        //
        // TODO: Add constructor code after the InitializeComponent() call.
        //
    }

    void MainFormFormClosing(object sender, FormClosingEventArgs e)
    {
        string path = @"ex01.dump";
        File.WriteAllText(path, textBox5.Text,
Encoding.GetEncoding("big5"));
    }
}

```

MainForm.Designer.cs

MainForm.Designer.cs

```

namespace test
{
    partial class MainForm
    {
        /// <summary>
        /// Designer variable used to keep track of non-visual components.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Disposes resources used by the form.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
    }
}

```

```
protected override void Dispose(bool disposing)
```

```
{  
    if (disposing) {  
        if (components != null) {  
            components.Dispose();  
        }  
    }  
    base.Dispose(disposing);  
}
```

```
/// <summary>
```

```
/// This method is required for Windows Forms designer support.
```

```
/// Do not change the method contents inside the source code editor. The  
Forms designer might
```

```
/// not be able to load this method if it was changed manually.
```

```
/// </summary>
```

```
private void InitializeComponent()
```

```
{  
    this.label1 = new System.Windows.Forms.Label();  
    this.label2 = new System.Windows.Forms.Label();  
    this.label3 = new System.Windows.Forms.Label();  
    this.label4 = new System.Windows.Forms.Label();  
    this.textBox1 = new System.Windows.Forms.TextBox();  
    this.textBox2 = new System.Windows.Forms.TextBox();  
    this.textBox3 = new System.Windows.Forms.TextBox();  
    this.textBox4 = new System.Windows.Forms.TextBox();  
    this.label5 = new System.Windows.Forms.Label();  
    this.label6 = new System.Windows.Forms.Label();  
    this.label7 = new System.Windows.Forms.Label();  
    this.button1 = new System.Windows.Forms.Button();  
    this.textBox5 = new System.Windows.Forms.TextBox();  
}
```

```
this.SuspendLayout();  
//  
// label1  
//  
this.label1.Location = new System.Drawing.Point(12, 15);  
this.label1.Name = "label1";  
this.label1.Size = new System.Drawing.Size(59, 23);  
this.label1.TabIndex = 0;  
this.label1.Text = "姓    名";  
//  
// label2  
//  
this.label2.Location = new System.Drawing.Point(12, 38);  
this.label2.Name = "label2";  
this.label2.Size = new System.Drawing.Size(59, 23);  
this.label2.TabIndex = 1;  
this.label2.Text = "國文成績";  
//  
// label3  
//  
this.label3.Location = new System.Drawing.Point(12, 61);  
this.label3.Name = "label3";  
this.label3.Size = new System.Drawing.Size(59, 23);  
this.label3.TabIndex = 2;  
this.label3.Text = "英文成績";  
//  
// label4  
//  
this.label4.Location = new System.Drawing.Point(12, 84);  
this.label4.Name = "label4";  
this.label4.Size = new System.Drawing.Size(59, 23);
```



```
this.label4.TabIndex = 3;
this.label4.Text = "數學成績";
//
// textBox1
//
this.textBox1.Location = new System.Drawing.Point(68, 12);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(100, 22);
this.textBox1.TabIndex = 4;
//
// textBox2
//
this.textBox2.Location = new System.Drawing.Point(68, 35);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(45, 22);
this.textBox2.TabIndex = 5;
//
// textBox3
//
this.textBox3.Location = new System.Drawing.Point(68, 58);
this.textBox3.Name = "textBox3";
this.textBox3.Size = new System.Drawing.Size(45, 22);
this.textBox3.TabIndex = 6;
//
// textBox4
//
this.textBox4.Location = new System.Drawing.Point(68, 81);
this.textBox4.Name = "textBox4";
this.textBox4.Size = new System.Drawing.Size(45, 22);
this.textBox4.TabIndex = 7;
//
```

```
// label5
//
this.label5.Location = new System.Drawing.Point(119, 84);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(59, 23);
this.label5.TabIndex = 8;
this.label5.Text = "分";
//
// label6
//
this.label6.Location = new System.Drawing.Point(119, 61);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(59, 23);
this.label6.TabIndex = 9;
this.label6.Text = "分";
//
// label7
//
this.label7.Location = new System.Drawing.Point(119, 38);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(59, 23);
this.label7.TabIndex = 10;
this.label7.Text = "分";
//
// button1
//
this.button1.Location = new System.Drawing.Point(93, 109);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(75, 23);
this.button1.TabIndex = 11;
this.button1.Text = "成績處理";
```

```

this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.Button1Click);
//
// textBox5
//
this.textBox5.Location = new System.Drawing.Point(193, 12);
this.textBox5.Multiline = true;
this.textBox5.Name = "textBox5";
this.textBox5.Size = new System.Drawing.Size(214, 120);
this.textBox5.TabIndex = 12;
//
// MainForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleModeMode.Font;
this.ClientSize = new System.Drawing.Size(427, 146);
this.Controls.Add(this.textBox5);
this.Controls.Add(this.button1);
this.Controls.Add(this.label7);
this.Controls.Add(this.label6);
this.Controls.Add(this.label5);
this.Controls.Add(this.textBox4);
this.Controls.Add(this.textBox3);
this.Controls.Add(this.textBox2);
this.Controls.Add(this.textBox1);
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Name = "MainForm";
this.Text = "C#視窗介面資料處理";

```

```

        this.FormClosing += new
System.Windows.Forms.FormClosingEventHandler(this.MainFormFormClosing);
        this.ResumeLayout(false);
        this.PerformLayout();
    }
    private System.Windows.Forms.TextBox textBox5;
    private System.Windows.Forms.Button button1;
    private System.Windows.Forms.Label label7;
    private System.Windows.Forms.Label label6;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.TextBox textBox4;
    private System.Windows.Forms.TextBox textBox3;
    private System.Windows.Forms.TextBox textBox2;
    private System.Windows.Forms.TextBox textBox1;
    private System.Windows.Forms.Label label4;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.Label label1;

    void Button1Click(object sender, System.EventArgs e)
    {
        string output = "";

        output += textBox1.Text + "成績處理結果" +
System.Environment.NewLine;
        output += "國文" + textBox2.Text + "分" +
System.Environment.NewLine;
        output += "數學" + textBox3.Text + "分" +
System.Environment.NewLine;
        output += "英文" + textBox4.Text + "分" +
System.Environment.NewLine;
    }

```

```

        output += System.Environment.NewLine;

        double avg = 0;
        if (textBox2.Text != "") {
            avg += System.Convert.ToDouble(textBox2.Text);
        }
        if (textBox3.Text != "") {
            avg += System.Convert.ToDouble(textBox3.Text);
        }
        if (textBox4.Text != "") {
            avg += System.Convert.ToDouble(textBox4.Text);
        }
        avg = avg / 3;

        output += "平均" + avg + "分" + System.Environment.NewLine;

        if (avg >= 60) {
            output += "恭喜！及格。" + System.Environment.NewLine;
        }
        else {
            output += "不及格！加油。" + System.Environment.NewLine;
        }

        textBox5.Text = output;
    }
}

```

MainForm.Designer.part

MainForm.Designer.part
namespace test

```

{
    partial class MainForm
    {
        /// <summary>
        /// Designer variable used to keep track of non-visual components.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Disposes resources used by the form.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing) {
                if (components != null) {
                    components.Dispose();
                }
            }
            base.Dispose(disposing);
        }

        /// <summary>
        /// This method is required for Windows Forms designer support.
        /// Do not change the method contents inside the source code editor. The
Forms designer might
        /// not be able to load this method if it was changed manually.
        /// </summary>
        private void InitializeComponent()
        {

```

```
this.label1 = new System.Windows.Forms.Label();
this.label2 = new System.Windows.Forms.Label();
this.label3 = new System.Windows.Forms.Label();
this.label4 = new System.Windows.Forms.Label();
this.textBox1 = new System.Windows.Forms.TextBox();
this.textBox2 = new System.Windows.Forms.TextBox();
this.textBox3 = new System.Windows.Forms.TextBox();
this.textBox4 = new System.Windows.Forms.TextBox();
this.label5 = new System.Windows.Forms.Label();
this.label6 = new System.Windows.Forms.Label();
this.label7 = new System.Windows.Forms.Label();
this.button1 = new System.Windows.Forms.Button();
this.textBox5 = new System.Windows.Forms.TextBox();
this.SuspendLayout();
//
// label1
//
this.label1.Location = new System.Drawing.Point(12, 15);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(59, 23);
this.label1.TabIndex = 0;
this.label1.Text = "姓    名";
//
// label2
//
this.label2.Location = new System.Drawing.Point(12, 38);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(59, 23);
this.label2.TabIndex = 1;
this.label2.Text = "國文成績";
//
```

```
// label3
//
this.label3.Location = new System.Drawing.Point(12, 61);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(59, 23);
this.label3.TabIndex = 2;
this.label3.Text = "英文成績";
//
// label4
//
this.label4.Location = new System.Drawing.Point(12, 84);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(59, 23);
this.label4.TabIndex = 3;
this.label4.Text = "數學成績";
//
// textBox1
//
this.textBox1.Location = new System.Drawing.Point(68, 12);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(100, 22);
this.textBox1.TabIndex = 4;
//
// textBox2
//
this.textBox2.Location = new System.Drawing.Point(68, 35);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(45, 22);
this.textBox2.TabIndex = 5;
//
// textBox3
```



```

//
this.textBox3.Location = new System.Drawing.Point(68, 58);
this.textBox3.Name = "textBox3";
this.textBox3.Size = new System.Drawing.Size(45, 22);
this.textBox3.TabIndex = 6;
//
// textBox4
//
this.textBox4.Location = new System.Drawing.Point(68, 81);
this.textBox4.Name = "textBox4";
this.textBox4.Size = new System.Drawing.Size(45, 22);
this.textBox4.TabIndex = 7;
//
// label5
//
this.label5.Location = new System.Drawing.Point(119, 84);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(59, 23);
this.label5.TabIndex = 8;
this.label5.Text = "分";
//
// label6
//
this.label6.Location = new System.Drawing.Point(119, 61);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(59, 23);
this.label6.TabIndex = 9;
this.label6.Text = "分";
//
// label7
//

```

```

this.label7.Location = new System.Drawing.Point(119, 38);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(59, 23);
this.label7.TabIndex = 10;
this.label7.Text = "分";
//
// button1
//
this.button1.Location = new System.Drawing.Point(93, 109);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(75, 23);
this.button1.TabIndex = 11;
this.button1.Text = "成績處理";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.Button1Click);
//
// textBox5
//
this.textBox5.Location = new System.Drawing.Point(193, 12);
this.textBox5.Multiline = true;
this.textBox5.Name = "textBox5";
this.textBox5.Size = new System.Drawing.Size(214, 120);
this.textBox5.TabIndex = 12;
//
// MainForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(427, 146);
this.Controls.Add(this.textBox5);
this.Controls.Add(this.button1);

```

```

        this.Controls.Add(this.label7);
        this.Controls.Add(this.label6);
        this.Controls.Add(this.label5);
        this.Controls.Add(this.textBox4);
        this.Controls.Add(this.textBox3);
        this.Controls.Add(this.textBox2);
        this.Controls.Add(this.textBox1);
        this.Controls.Add(this.label4);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Name = "MainForm";
        this.Text = "C#視窗介面資料處理";
        this.FormClosing += new
System.Windows.Forms.FormClosingEventHandler(this.MainFormFormClosing);
        this.ResumeLayout(false);
        this.PerformLayout();
    }
    private System.Windows.Forms.TextBox textBox5;
    private System.Windows.Forms.Button button1;
    private System.Windows.Forms.Label label7;
    private System.Windows.Forms.Label label6;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.TextBox textBox4;
    private System.Windows.Forms.TextBox textBox3;
    private System.Windows.Forms.TextBox textBox2;
    private System.Windows.Forms.TextBox textBox1;
    private System.Windows.Forms.Label label4;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.Label label1;

```

```
void Button1Click(object sender, System.EventArgs e)
{
    //請完成此區程式碼撰寫
}
}
```