

教育部教學實踐研究計畫成果報告
Project Report for MOE Teaching Practice Research Program

計畫編號/Project Number : PEE1110158

學門專案分類/Division : 工程

計畫年度 : 111 年度一年期 110 年度多年期

執行期間/Funding Period : 2022.08.01 – 2023.07.31

計畫名稱: 即時反饋的程式設計學習模式--於教學場域實踐適合
非資訊科系入門的自動程式碼 (C#、VB) 批改系統

配合課程: 計算機概論與程式設計

計畫主持人(Principal Investigator) : 黃智賢

執行機構及系所(Institution/Department/Program) :

國立台灣海洋大學光電與材料科技學系

成果報告公開日期 :

立即公開 延後公開 (統一於 2025 年 7 月 31 日公開)

繳交報告日期(Report Submission Date) : 2023 年 9 月 18 日

計畫名稱：即時反饋的程式設計學習模式--於教學場域實踐適合
非資訊科系入門的自動程式碼（C#、VB）批改系統

一. 本文

1. 研究動機與目的

本人教授程式設計已達二十年，深知程式設計的成功學習，並不是靠教師多高超的教學技巧，主要依賴的是同學積極不斷的練習。

然則，一班四十人的課堂中，當我們指派多一題作業給學生時，助教與教師便多出 40 題的作業量。合理想像若一次課堂僅出五題作業練習，那一學期算 15 周，便有 $40 \times 5 \times 15 = 3000$ 題作業量，如此大的負荷，僅靠一位助教，即便加上教師，也是根本無法負擔的，光是開啟檔案、執行檔案、判斷給分，每題便需要花費至少三分鐘，3000 題等同 9000 分鐘，等於 150 個小時！

宥於現實，作業無法出太多，學生練習次數少，等到考試時，便十分生疏---這便是教學現場的一大問題。

解決這問題，自動化的程式碼批改系統被公認是很有幫助的。然則，現有國內的程式碼自動批改系統多是資訊系相關的老師所設計的。如台大的批改娘與高師大附中江其勳老師的 Zerojudge，主要使用的語言便是 C、C++、Java 與 Python，前兩者適合底層的硬體介面、物聯網或其他作業系統使用，後兩者則分別可推展到網頁撰寫與人工智慧去。

對於非資訊科技的理工科系學生而言，他們多數只會學一種程式語言，把他當成工具。在最常用的 windows 系統上，當然是能夠輕鬆建立 windows 表單的 C# 與 VB.Net 是最適合他們的。此兩種語言基本上是可以完全轉換互通的，為此，電機、光電、機械與部分工程相關科系便十分適合學習 C#，未來可銜接單晶片或 Arduino 用的 C++ 或 C 語言；而其他科系不會接觸到單晶片程式或 MATLAB 設計的，則可由 VB 入門，這部分可與 Office 的 VBscript 銜接使用，有必要時，未來也可輕易轉學 C#。

有鑑於此，過去我便一直在構思用 SQL 寫一個可以自動批改 C# 與 VB 的線上批改系統，命名為自動程式碼批改系統，AutoCodeGrader(簡稱 ACG)。因此，在此我便運用執行教育部實踐計畫的機會，將該 ACG 發展得更完整，補足題庫並視教學現場實際需求設計所需功能。

我們希望可以開發出一個適合教學現場實務的程式碼自動批改系統，協助學生在程式語言的實務學習。

2. 文獻探討

線上批改系統（online judge），或稱線上解題系統，是一種具有自動化批改程式碼的系統，近年來頗為盛行，可以用來考核受測者的程式設計能力，判斷公司是否要錄取某位求職者；也可以用漸進式的方式教育學生自學程式設計，瞭解學生的學習狀況。

最早的線上解題系統是西班牙 Valladolid 大學 Ciriaco Garcia de Celis

於 1995 年所開發的 UVA Online judge，當時應用於該校參加 ACM/ICPC 西南歐區域賽選拔隊員之用¹。其後，陸續有許多線上解題系統被陸續開發上線，目前可說是百家爭鳴。著名的線上解題網站有^{2,3}：西班牙 [UVA online judge](#)、俄羅斯 [Timus online judge](#)、[Leetcode](#)、中國大陸[洛谷](#)、美國 [Top coder](#)…至少超過二十餘家，不勝枚舉（請參考參考文獻[2, 3]中的網站）。

國內方面，早在 2003 年，台大資訊系劉邦鋒老師⁴便幫資訊系的學生建構了批改娘(Judge girl)自動批改網站⁵，不僅可讓學生有更多的練習機會，也讓老師或教學助教能更有餘裕致力於現場教學。2006 年，高師大附中教師江其勳老師自行開發⁶，設計了 ZeroJudge 高中生解題系統網站⁷，使得許多高中生有更多程式練習的機會，目前是台灣第一大評測系統⁸。

然則，就我所知，除了少數需收費線上批改系統，如 Code Judger，支援 C# 外，目前國內並沒有專門針對 C# 與 VB 的線上批改程式碼免費網站，這也是我們要設計這個專門針對 C# 與 VB 的線上批改系統的主要原因。

3. 教學設計與規劃

本計畫執行初期，以為只要有課後作業自動批改系統，就可以讓同學自動練習、學習，結果發現，這是過度簡化的想法，學生反而覺得作業有得抄，便不來上課。之後，迅速調整，改成如圖 1 的教學設計架構。點名，在前幾堂課有立威與認識同學之效，實有必要，每次上課中(3 學分連續)，需加入 3~4 次的課中實測，亦即老師就剛剛教學的內容，出題請同學解答，若同學解答出來後，便由老師或 ACG 登記成績(成績隨解題時間衰減)；同學解題的同時，教師更邊巡堂解答同學遇到的困難與疑惑。課後，則有 ACG 作業請同學解答。前述課中登記之成績與課後作業，都是平時成績的一環。期中期末考試也使用 ACG 來做測驗，但考試時僅允許上微軟 doc 的網站以及課程講義網站，不允許查詢網路或以通訊軟體上網。

4. 研究設計與執行方法

為了配合上述教學設計架構，執行面上，我們的 ACG 系統必須設計有幾個特殊的功能

(1) 時間衰減的評分

由於課中測驗是可以討論的，最早交出的同學勢必是程度最好的，因此分數最高，其後交的同學很可能是與他人討論而解出，或甚至直接拷貝他人，因此，隨時間衰減的計分是有必要的。一般的課中測驗手動登記計分，可以用線性的依序減分，例如從 95 一直登記到 60 為止。

若使用 ACG 則可有更複雜的設計，甚至能運用在期中期末考中。ACG 中此部分功能，是利用自訂輸入公式計算而成，參考公式如下：

[PointGot(得分)] =

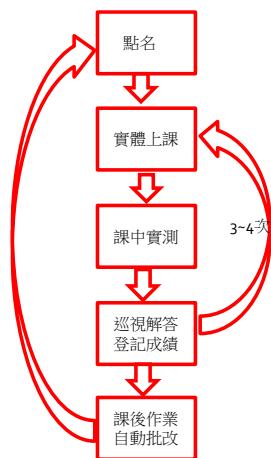


圖 1. 本計畫中的教學設計架構。

[Points(該題分數)] *Exp(ln([DecayRate(衰減率, 如後)]) * ([Tsubmit(繳交時間)] - [N(總題數)] * [AvgSolvingTime(每題平均預計滿分解題時間, 如後)]) / [Tduration(考試時長)]) where [DecayRate] = 70/100,
 [AvgSolvingTime] = 0.2(hr), [.MaxValue(最高分數)] = [PointsSet(每題設定分數)], [InClassLimitIP(限制在課中教室網域中解答)] = 140.121.60.*

上面的公式設計中，假若總計三題(N=3)，[AvgSolvingTime]=0.2(hr)，表示學生若能在 $0.2 \times 3 = 0.6$ 小時，即 36 分鐘解出的題目，皆可得滿分。若某題解答成功時已超過 36 分，如 60 分，則依據每過一段考試總時長(設若為 3 小時)，衰減率 70% 來計算，該題得分為 $[Points(\text{該題分數})] * \text{Exp}(\ln([\text{DecayRate}(\text{衰減率}, 0.7)]) * ([\text{Tsubmit}(\text{繳交時間}=50 \text{ 分鐘})] - [\text{N}(\text{總題數三題})] * [\text{AvgSolvingTime}(12 \text{ 分})]) / [\text{Tduration}(120 \text{ 分鐘})])$ ，即僅得 93.1% 的設定分數。假設他在考試期期限前繳交一題，則該題可得 77.9% 分數。入總分數不到 100 分，最後的總分可均一化成 100 分來表示。上面的情況適合正式考試的情況，因此，只要在考試期限內繳交便有及格分數超過考試期限便無法得分。對於課中測驗，可能只有一題，因此若 15 分鐘是總時長，可以設滿分期限在 5 分鐘，仍依前面公式來計分。

對於作業，我們可以把滿分期限設長一點，如兩三天，超過期限時，則以每周期 60% 或 70% 衰減率衰減。我們的 ACG 還可以設定允許補交選項，如圖 2 中 Check make-up after deadline 選項。使用此選項時，同學即便超過作業時間繳交仍可得分，但所得分數仍是依衰減率衰減後的分數。這是促使同學積極繳作業的設計。

| StudentID | Username | FullName | Department | A0001 Got Pt | A0001 Pt @ Turn-in(hh) | A0002 Got Pt | A0002 Pt @ Turn-in(hh) | A0013 Got Pt | A0013 Pt @ Turn-in(hh) |
|-----------|----------|----------|------------|-----------------|---------------------------|-----------------|---------------------------|-----------------|---------------------------|
| 0091 | | | | 10 | 10@0.02 | 10 | 10@0.03 | 10 | 10@0.06 |
| 0098 | | | | 1@0.4 | 1@0.03 | 10 | 10@0.05 | 0 | 10@ |
| 0099 | | | | 1@0.3 | 1@0.07 | 10 | 10@0.11 | 10 | 10@0.27 |
| 0098 | | | | 1@0.3 | 1@0.04 | 10 | 10@0.16 | 10 | 10@1.35 |
| 0098 | | | | 1@0.3 | 1@0.09 | 10 | 10@0.18 | 10 | 10@0.25 |
| 0098 | | | | 1@0.3 | 1@0.03 | 10 | 10@0.04 | 10 | 10@0.08 |
| 0098 | | | | 0 | 10@ | 0 | 10@ | 0 | 10@ |
| 0108 | | | | 1@0.2 | 1@0.06 | 10 | 10@0.09 | 10 | 10@0.57 |
| 0108 | | | | 1@0.2 | 1@0.11 | 10 | 10@0.18 | 10 | 10@0.16 |
| 0118 | | | | 1@0.1 | 1@0.03 | 10 | 10@0.05 | 10 | 10@0.09 |
| 0118 | | | | 1@0.1 | 1@0.03 | 10 | 10@0.05 | 10 | 10@0.08 |
| 0118 | | | | 1@0.1 | 1@0.03 | 10 | 10@0.04 | 10 | 10@0.08 |
| 0118 | | | | 1@0.1 | 1@0.07 | 10 | 10@0.16 | 10 | 10@0.26 |
| 0118 | | | | 1@0.1 | 1@0.03 | 10 | 10@0.05 | 10 | 10@0.14 |
| 0118 | | | | 1@0.1 | 1@0.03 | 10 | 10@0.05 | 10 | 10@0.37 |
| 0118 | | | | 1@0.1 | 1@0.04 | 10 | 10@0.17 | 10 | 10@0.13 |
| 0118 | | | | 1@0.1 | 1@0.02 | 10 | 10@0.04 | 10 | 10@0.11 |
| 0118 | | | | 1@0.1 | 1@0.02 | 10 | 10@0.04 | 10 | 10@0.15 |
| 0118 | | | | 1@0.1 | 1@0.04 | 10 | 10@0.06 | 10 | 10@0.10 |
| 0118 | | | | 1@0.1 | 1@0.05 | 10 | 10@0.12 | 10 | 10@0.18 |
| 0224524 | | | | 1@0.1 | 1@0.04 | 10 | 10@0.04 | 10 | 10@0.07 |

圖 2. 實際教師設計測驗時的執行畫面，圖中可看到是否要用時間衰減計分的選項與公式。

(2)因人而異的測試資料

前面的衰減設計除區別學生程度外，也是為了緩解抄襲的問題。但為了讓抄襲能進一步緩解，讓每個人解題時有不同的測試資料也是緩解抄襲的一種方式。至少學生無法完全 Copy-Paste---這算是最低級的抄襲。

為了避免作業抄襲，ACG 可設定以帳號為種子的亂數序列當測試資料，如

此便可在產生測試資料時，產生每人不同測資的題目，至少可以避免最低級的抄襲。例如圖 3 中是出題者視窗中，設計求加總的題目，提示程式碼中有設計一個特殊的插入碼 [#Random(0,100,100,3)#]，ACG 看到後便會產生 100 個 $0 \leq x < 100$ 的亂數於陣列中(取小數位三位)。如此設計，同學便無法直接複製貼上別人寫好的程式，至少需要把別人的測資改成自己的測資。

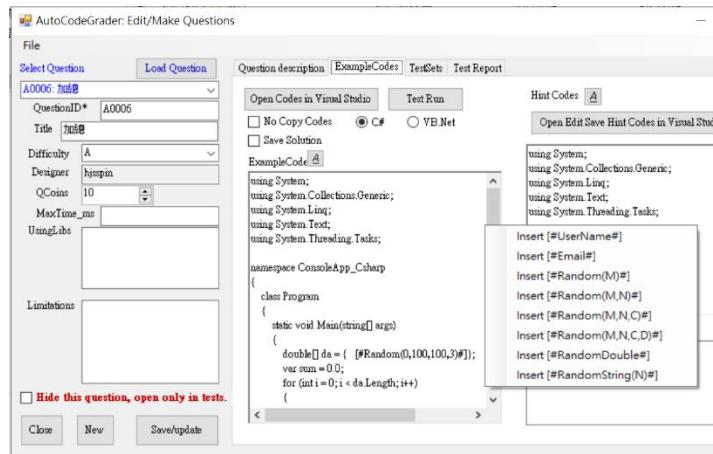


圖 3. 可以在設計題目時插入以帳號為種子的亂數序列。

(3)限制在計中教室裡解答

由於考試或課中測驗都希望同學來現場解答，而不是遠端解答，因此多出這個功能。我們可以在圖 2 中的 [InClassLimitIP] 設定計中 IP，達到限制答題電腦 IP 的效果。

(4)限制上網的考試

期中期末考試時，必須限制同學上網搜尋或以通訊軟體相互串聯，以維持公平性，因此 ACG 必須有限制網路存取的功能。在此，我們運用將 Proxy 設定為本機代表位置 127.0.0.1:80，但本機並無安裝 Proxy，因此，會發生網路錯誤。同學若手動改動 Proxy 則會被 ACG 程式發覺，發出警告。此外，由於 Visual studio 按 F1 可連結到微軟的 Doc 網站，考試時，也允許下載講義，因此可用 Proxy 設定中的 BypassSites 設定允許連線網站，方便同學考試時參考。圖 4 便是編輯設計網路是否可連線時的畫面。

(5)啟動 Visual Studio IDE(VSIDE)練習寫程式

VSIDE 是學習 c# 或 VB 很方便的免費且方便的系統，可有智慧提示，有可能有偵錯功能，因此可以直接寫好程式，送去批改是最為方便的做法。如圖 5，ACG 程式可按 Open Codes in Visual Studio 進入 VSIDE 編輯，編輯完關閉後便可把程式碼帶回。

(6)動畫或圖形輔助理解與趣味化

ACG 每一個題目本身都支援 Rich Text Format(RTF) 編輯，因此都可以設計與 Word 相容的圖形與字型表現。除此之外，還有一個功能是可加入一個代表圖案，支援動畫或圖形，增加趣味性。值得注意的是，我們的動畫或圖形，取自網路的 URL 都有標示，且點選後會進入原本的網站---作為教育教學使用，如此的設計應該已經符合著作權法規基本需求。此外，未來 ACG 也可以考慮

增加解題過關畫面與音效，這是程式自行開發的好處—可依需求自行調整設計。

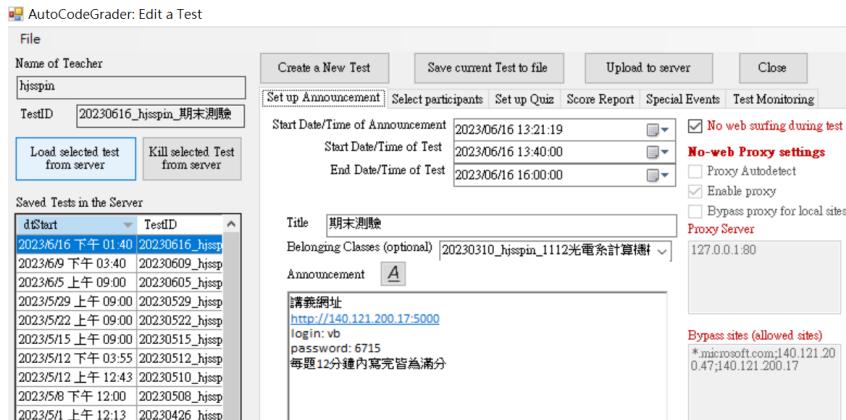


圖 4. 編輯考試時，可運用 proxy 限制網站連接並只允許連接講義或微軟網

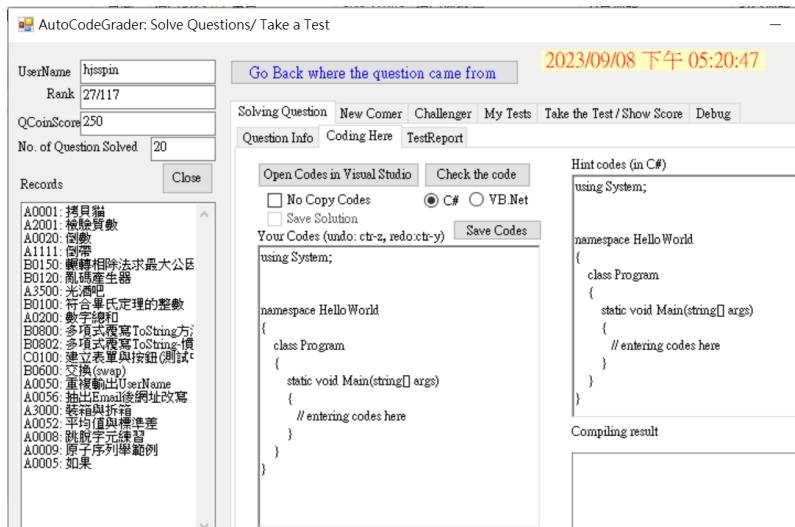


圖 5. 程式可按 Open Codes in Visual Studio 進入 VSIDE 編輯，編輯完
關閉後便可把程式碼帶回。

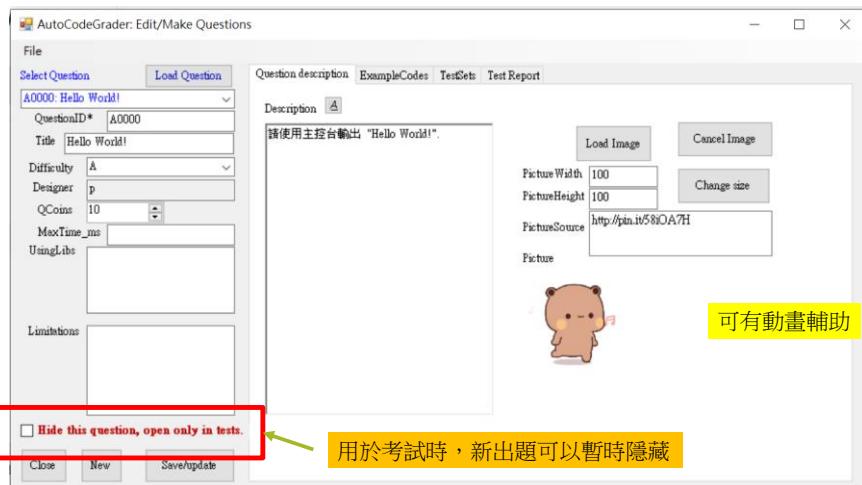


圖 6. 出題時可加入動畫 gif 或一般圖形當代表圖，增加趣味性。另外，出題用於考試
時時或暫時性測試時，可用隱藏選項，題目暫時隱藏。

(7)暫時性隱藏題目

圖 6 中，也顯示設計題目時，可以暫時性隱藏題目。此功能主要用於，出題用於考試時或暫時性開發題目測試時，可用該隱藏選項，將題目暫時隱藏。圖 6 左方也顯示一些未來的設計所需功能。例如 MaxTime_ms 原本是設計做為執行時間的限制得，只是 ACG 因為是本地執行，其執行時間會隨機器而異，故暫時沒有開放。另外，UsingLibs 保留未來限制所用的 NameSpace，暫未使用。Limitations 則是設計一些限制，目前是設計某些題必須解開才可解此題，未來也可以設計成其他限制。

(8)可多人同時上線

由於考試時，可能近百個人同時連線，ACG 必須能處理。因此，我們使用 Win10 內建的 SQL 資料庫來處理，一般而言，SQL 資料庫可同時處理上千上萬筆的資料，超過伺服器附載時頂多只須排隊等一點時間處理，但應不會形成問題。

(9)可線上更新程式

因為 ACG 程式是由使用者下載後執行，若有更新板時，使用者應先下載更新程式，再執行。此部分，所幸 VSIDE 中，其 ClickOnce 部署已經提供此功能，對我們持續除錯更新程式有非常大的幫助。

(10)額外計分功能

因為 ACG 是贏者全拿，期中期末考時，對於部分同學程式碼差不多對，但卡在某些輸出或指令問題，仍只能得零分。因此，我在 ACG 設計時，特別設計教師可以手動瀏覽程式碼，並一題一題給予部分加分，如此，等同手工改考卷，可以分出完全不會跟部分會的差別。

(11)支援 Google 帳號登入

我們設計的 ACG 系統目前可支援 google 帳號登入，因此不需設定密碼，可以簡化登入作為。不過課堂上的修課，還是需使用學校 email 登入。該功能是依據 google 公布的設定模式去寫程式，目前運作良好。

目前，ACG 的下載網站可在 google 搜尋 nanolab autocodegrader 而得，網站也有如何註冊的影音檔說明。

5. 教學暨研究成果

(1) 教學過程與成果

本計畫執行期間，我們必須不斷地隨著課程進度設計新的題目，改進 ACG 的系統，也需不斷地依據學生回報對 ACG 除錯。執行過程可謂忙得不可開交。目前 ACG 系統已經設計了從簡單到繁雜大約 75 個原創題目(部分題目參考圖 7)。值得注意的是，我們每個題目都有部分提示程式碼，以使同學能由淺而深地感受到程式設計的樂趣，而不是充滿挫折感。

在此，也必須坦承，因為時間有限，我們無法兼顧到 VB 部分的開發，此方面的專屬題目與測試上不完整，等待未來持續努力。

教學過程中，我們也發現，點名與課中測驗的重要性。在原本計畫書中，

我們只打算將 ACG 用於課後作業，因此，開始上課時，便給同學講義，專心講解，而在課後給同學設定作業。然而三、四周後，我們發現，學生到課率大幅降到 50%，顯然學生以為有講義可看，作業抄一下同學的，這堂課便可蒙混過關。約第四周後，我緊急改變上課策略，課前先點名(持續數周)，並鄭重登記，另外，在課中，加入實作測驗。亦即，講到一段時間時，請同學測試解某一個題目。我們課中實作是分成兩個階段：

a. 依講義照打：這是因為同學對開發介面 VSIDE 不熟悉，請他們照打，可以讓它們熟悉開發介面，避免弄錯方向。

b. 參考講義，但請其改成另一種功能或表示方式：當同學熟悉介面後，便可請同學參考講義，但做出另一種結果。

課中測驗時，若 ACG 系統有題目可請同學用 ACG 解，但若沒有，則使用手動登記。因為會有部分比較厲害的同學先解出，其他同學在參考他的解出，因此，此時需要用衰減分數的方式登記分數。課中時間若沒解出來的同學，則請他在課後留下來在同學與助教幫助下繳交。

點名與課中實作測驗實施之後，同學看似不情願地陸續回籠上課了，上課率終於回到接近 100%。持續實施後，過去沒來上課的同學，突然覺得自己對開發介面與程式設計十分陌生，學習態度漸漸轉為積極，許多大三大四的重修同學，也改變了消極的學習態度，我便會刻意注意一下他們的學習狀況。

課中測驗時，老師也是不得閒。我必須如蜻蜓點水般幫同學依依解決卡住的問題，從此也可知道同學常見的問題提醒其他同學知道。

課後作業方面，我們使用 ACG 共 10 次課後作業，都使用分數衰減並允許補繳。平時每周補強教學助教協助解題。

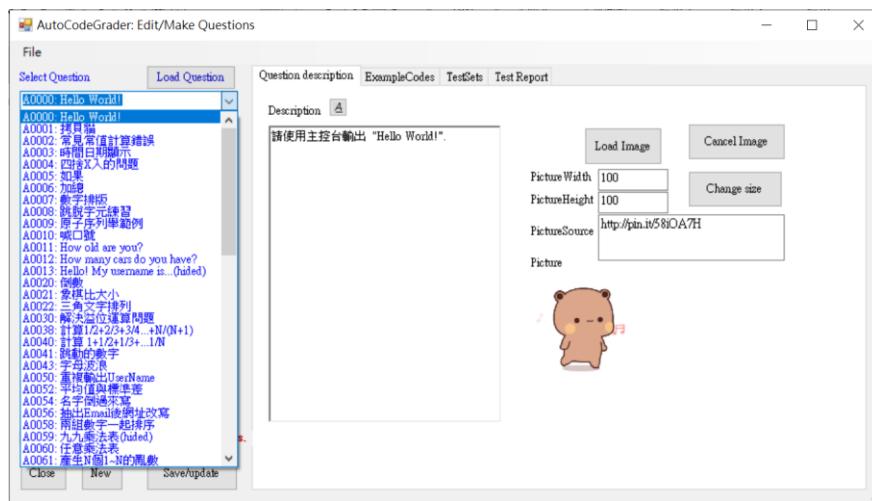


圖 7. ACG 已設計由淺而深 75 題，每題都有部分提示程式碼，上圖左方顯示 A000: Hello World 到 A0061: 產生 N 個 1-N 的亂數只是其中的 30 題。

如此上課方式，到期中期末考，便又是一個挑戰。我本來有點擔心一次五十幾個人同時上線會不會有問題。所幸，雖然略有比平時稍卡一點，但還不至影響 ACG 的運行。此外，考試畢竟要公平性，我們 ACG 可以限制上網，但隔壁

同學間考試時也不應互相觀摩。原本，我們申請購買防窺片，然而，防窺片價個一片上千，六十桌便要六萬，效果好不好也令我們猶豫。所幸，最後，我們運用市售數十元的塑膠瓦楞板，配合強力磁鐵，使瓦楞板夾在電腦上，形成隔間，達成公平的考試。

期中考時，成績有點難看，主要原因是，ACG 批改題目只要一點點錯就全錯。同學反應部分寫對可否部分給分，為此，我們又改了 ACG 系統，將其改成可以額外人工判讀加分，並於期末實施。

整體而言，本計畫中，ACG 在課程中漸漸得到最佳化，更符合需求，題庫也建立完成。此外，同學整體學習態度相較以往更積極。學期末有許多位同學總成績接近滿分，過去則頂多一兩位，顯見同學解題能力已達到整體的提升。



圖 8. 期中期末考時，我們使用塑膠瓦楞板配合磁鐵形成隔間，維持考試公平性。

(2) 教師教學反思

本人反思此計畫執行上，仍有幾個缺憾與心得：

a. 本計畫中，因為時間有限，在 VB 的題目與測試尚未臻完善，尚待未來持續努力。另外，也由於時間不足，目前 ACG 主要聚焦在控制台程式的撰寫，對 winform 仍沒有觸及，是較大的缺憾。未來也許可以繼續努力完成。

b. 優秀的學習環境不只是體現在硬體設備方面，能讓學生在學習有問題時迅速得到教師的反饋也是重要的一環。以程式設計為例，在學習過程中加入測試與練習，便可隨時瞭解同學可能遇到的困難，加以解決。類似的作法，若在傳統的教學課堂中，加入小測驗與提問，也將有助於增強與同學的互動並及時解決同學的問題。

c. 點名的意義不在於扣學生分數，重點是讓老師認識同學，從而讓同學不好意思缺席，這是一種形而外而上的作用，亦即表面上是點名，實際上是加強學生與老師相互的認識，進而讓學生不好意思翹課。

d. 課堂的內容不一定要教太多，但可以有一些延伸內容，請同學自行參閱。

(3) 學生學習回饋

圖 9 是課後的問卷調查結果，基本上同學對本課程與 ACG 使用的肯定意見居多。整體而言，我們在一個 48 人班級中，經過一個學期的 ACG 使用，高達 34 人(71%)認為 ACG 對學習程式有正面幫助遠大於負面意見者(3 人，6%)。此外，並有 65% 的同學比高中時期更滿意自己寫程式的能力。

圖 10 則是修課同學對本課程與 ACG 的回饋意見。基本上，同學對 ACG 有不少正面評價，但也有負面的評價。負面的評價主要集中在 ACG 系統早期不穩定的使用經驗，後期已經大部分修復完成。

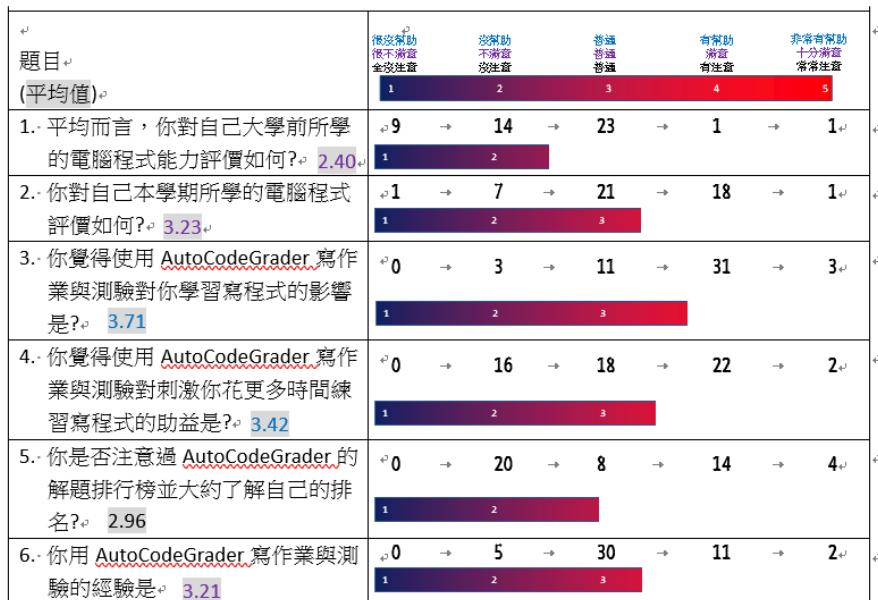


圖 9. 使用 ACG 於 C# 程式教學的問卷結果。

| | |
|--------|--|
| 正面的評價 | 滿意 |
| | 很棒! |
| | 非常好的程式 |
| | Good! |
| | 還行，凡方便的。 |
| | 謝謝老師教 C#，教了剛好用到超開心。 |
| | AutoCodegrader 是個用心的系統，屬於前衛的教學工具。 |
| | 是個很不錯的軟體。 |
| 系統功能缺點 | 有點卡。 |
| | 有時有點卡頓，希望可以修正。 |
| | AutoCodeGrader 有點不太穩，有時會當掉。 |
| | 太多人使用時會有點卡頓，還有大小寫打錯會錯誤有時比較麻煩。 |
| | AutoCodeGrader 有時很卡，有一小段時間會登不進去。 |
| | 有無法儲存的 bug。 |
| | 穩定性不足，在寫程式時，希望能同時看題目。 |
| | 常常好不容易寫好但無法儲存，需重新再寫一次。 |
| | 有時 current output 跟 expected output 都一樣，但還是過不了，不知怎麼修改。 |
| | 有時會有儲存問題導致解題後需要重新再做一次。 |

圖 10. 同學對本課程與 ACG 系統的主要回饋意見。

6. 建議與省思

我們在教學現場發現，若僅靠一般的批改系統當課後作業，同學容易相互抄襲，並影響上課出席率與學習態度。我們所開發的 ACG 經過精心設計後可以某種程度地避免抄襲，並限制網路使用，設計衰減的得分，使同學沒有僥倖心理。

此外，本計畫執行中，我們發覺，在課堂之中加入課中測驗(或練習，有登記分數)是幫助學生參與學習很好的方式。課堂中練習時，同學可以互相討論，並利用競爭心理刺激學習。教師則可在課堂練習間解答同學問題，學習效果會更好。

最後，我們也發現，對於 C#學習者，有了主控台程式的訓練後，同學開始學習多采多姿的 Winforms 設計會更有吸引力。未來，我們深入 Regex(正則表達式)與 reflection 運用，朝向 winform 支援，持續改進 ACG。

二. 參考文獻

- ¹ Revilla M, Manzoor S, Liu RJ. Competitive learning in informatics: the UVa online judge experience (2008,2). Olympiads in Informatics: 131–148.
- ² 參考：
<https://zh.wikipedia.org/wiki/%E5%9C%A8%E7%BA%BF%E8%AF%84%E6%B5%8B%E7%B3%BB%E7%BB%9F>
- ³ 參考 https://code.fandom.com/wiki/Online_judge
- ⁴ 參考 CSIE 通訊，劉邦鋒教授專訪：
<http://csiecomm.blogspot.com/2019/02/blog-post.html>
- ⁵ 批改娘網址：<https://judgegirl.csie.org/>
- ⁶ 參考推動大學程式設計教學網站介紹：
<https://plusdscp.csie.ntnu.edu.tw/index.php/zerojudge/>
- ⁷ Zerojudge 網址：<https://zerojudge.tw/>
- ⁸ 參考：
<https://zh.wikipedia.org/wiki/%E7%AE%97%E6%B3%95%E7%AB%9E%E8%B5%9B>