

導入形成性評量與問題解決教學策略以提升 C/C++ 程式設計課程之學習成效

一、本文 Content (3-15 頁)

1. 研究動機與目的 Research Motive and Purpose

(1) 研究動機

隨著全球帶起一股程式學習風潮，程式素養已被許多先進國家視為必備資訊教育核心能力之一，影響國家經濟與競爭力，被視為是二十一世紀必備的重要技能。近年教育部積極推廣程式教育，落實運算思維向下扎根的理念。教育部 108 課程綱要將資訊科技與程式設計列為高中生必修之課程重點，利用大學程式設計先修檢測(APCS)，由學生透 ITSA 程式能力線上自我評量系統，利用電腦自動評判，以檢測程式設計能力(盧玫燕 & 范斯淳，2019)。國內各大學紛紛開設程式設計課程，積極培養跨領域程式人才，期望大學生通過大學程式能力檢定(CPE)，提升大學生的基本電腦程式素養與技能(國家教育研究院，2020)。程式設計課程培養學生了解基礎程式概念，應用不同的程式設計技巧與資料結構演算法來回應形成式評量上的測驗或作業。

程式設計課程是國內各大學資訊工程學系大一學生必修課程，該課程老師在電腦教室以講述教學方式進行課程教學，上課時老師會講解程式觀念及程式設計技巧。此課程每週有三個小時堂上課，每堂課講完一個程式語法單元概念後，給予學生時間回答該程式單元的隨堂測驗問題。過去教授程式設計課程發現，本計畫想嘗試解決過去課堂上可能的問題。首先，學生的程式語言背景都不盡相同，只有部分的學生有學過程式語言基礎概念，有些新生可能連程式語言都沒有學習過，導致這些新生無法跟上課程進度。接著，過去課程的教學方法都是採課程講述教學法，在課堂上，學生聽老師講述程式語法和撰寫程式方法，如果學生不專心聽講就可能無法理解老師所講授的程式語法及實作流程知識，接著講授新的課程內容就會造成學生學到程式語法知識還未能完全融會貫通，反而學生面對新的課程單元內容就會失去學習信心並增加了學習負荷。因此，本計畫將解決這些問題是有可

能提升本計畫的課程學習成效的重要關鍵。

(2) 研究目的

過去程式設計課程遭遇上述的問題，這些問題是非常值得投入的教學現場研究範疇。為了解決上述的問題，本計畫提出形成性評量結合問題解決教學策略融入程式設計課程進行教學活動，該課程活動是培養新進學生了解 C/C++ 程式語言基本知識概念，能設計與實作基礎程式。過去課程教學活動中，新進學生對於 C/C++ 程式語言的學習普遍感到困難，本計畫運用形成性評量結合問題解決教學策略來擬定陳述問題、瞭解問題、擬定解題計畫、執行計畫，反思與除錯等四個步驟，這些步驟讓新進學生認識到理解 C/C++ 程式語言基礎知識概念，接著利用形成性評量得知學生的高層次認知與學習成效。本計畫運用形成性評量結合問題解決教學策略之優點，這樣的教學策略是有可能達到適性化學習策略，著重於培養學生的問題解決能力，改善學生的學習弱項，進而探討使用形成性評量結合問題解決教學策略對學生學習的學習成效有何影響。

此外，本計畫的相關成果發表於第十一屆工程、技術與科技教育學術研討會(林昱達，劉達全，張清堯，& 蘇育生，2022)，全國師生有機會瞭解形成性評量結合問題解決教學策略融入程式設計課程活動，全國師生能認識到形成性評量結合問題解決教學策略的優點，對程式教學研究注入新的元素，增進與其他學校師生對程式教學現場應用，漸進擴及至全民程式教學方案，以作為教育部推動大專校院教學實踐計畫之參考。

2. 文獻探討 Literature Review

(1) 程式設計教育

隨著全球帶起一股學習程式設計的風潮，先進國家已把電腦程式技能為必備終身程式教育核心能力之一，影響國家經濟與競爭力，程式教育被視為是二十一世紀必備的重要能力。因應教育部十二年國民基本教育之課程綱要中程式設計課程列為國內大學生必修重點課程，國內各大學已納入程式設計課程為高等教育深耕計畫績效指標之一。國內各大學

紛紛開設程式設計課程，積極培養跨領域程式人才，提升學生的程式設計能力，成為將來在職場上或研究所上的關鍵能力(國家教育研究院，2020；張基成，2014)。程式設計課程是能培育大學生的運算思維素養，學生參加大學程式能力檢定(CPE)，經由 ITSA 程式能力線上自我評量系統，以檢測學生的程式能力。

本計畫配合課程是程式設計，以 C/C++ 程式語言為主，學生都會撰寫程式碼、編譯、除錯等過程完成 C/C++ 程式。雖然初學者的程式背景都不相同，他們都有共同點問題是覺得 C/C++ 程式語言是非常難學起來，尤其是學習物件導向概念，他們遭遇不知如何使用物件導向概念設計 C/C++ 程式 (Govender & Grayson, 2006; Gomes & Mendes, 2007; Hung, Liu, Liang, & Su, 2020)。過去程式設計課程是有可能發生學生的程式語言背景差異，學生無法確實掌握課程上每個單元程式語言基礎知識與實作技巧，因而學生跟上課程進度，導致學生的學習低落(Peter & Paul, 2006; Hung, Liu, Liang, & Su, 2020)。

過去程式設計課程是採用課程講述教學法，這是大班級課堂最常用的教學模式。Gomes 和 Mendes (2007)指出，初學者在大班級課堂會遭遇看不懂隨堂測驗題目問題，導致學生不知道使用 C/C++ 程式哪種程式語法來回答隨堂測驗的問題。在隨堂測驗後，老師會解析說明隨堂測驗的每一個題目，然後老師繼續教導新的課程單元內容，對於初學者而言，之前學到程式語法知識還未能完全融會貫通，反而面對新的課程單元內容逐漸失去學習意願(Hung, 2008; Hung, 2012; Su, & Chen, 2020; Uysal, 2014)。

本計畫想解決上述遭遇到教學現況的問題，我們想找到適合的教學策略來教導程式設計課程，方能有利於學生將所習得的程式技能類化到進階程式知識領域。在下一節，我們運用形成性評量結合問題解決教學方法融入程式設計課程進行教學活動，這樣機制是否會影響到學生的學習成效。

(2) 形成性評量結合問題解決教學方法融入程式設計課程

程式教育是具有運算思維，學生學習到實作程式過程中解決問題技巧。在大班級的程式課

程，不同程式背景的學生學習物件導向程式語言是不容易的。過去研究(Bey & Bensebaa, 2012; Govender & Grayson, 2006; Hung, Liu, Liang, & Su, 2020)發現，對於初學物件導向程式語言的學生而言，學生需要步驟式問題解決策略來學習物件導向程式概念，避免學生學習過程中無法獲得建設性回饋而放棄學習。問題解決的教學策略是有助於學生學習物件導向程式實作技巧，增加學生對新的課程單元內容的學習意願(Chung, Chou, Chang, Li, & Hsu, 2016; dos Santos, 2017; Gao, Lu, Zhao, & Li, 2019)。

問題解決教學策略雖然有助於學生學習物件導向程式，但描述性顯然大於規定性，簡而言之該教學策略是缺乏學習成效的評估模式，如何在程式設計課程評估問題解決方法是合適的教學策略，這也是一個重大挑戰(Hung, Liu, Liang, & Su, 2020; Su, Ding, & Lai, 2017; Su, Yang, Hwang, Huang, & Tern, 2014)。依照問題解決方法為架構，以學生為主，而老師扮演協助者的角色，本計畫提出形成性評量結合問題解決教學策略融入程式設計課程教學活動。老師進行問題解決教學過程中，老師提供學生明確的隨堂測驗，依照學生先前學到問題解決步驟來解答隨堂測驗題目，老師會得知學生學習狀況並適度回饋學生。Gomoll、Hmelo-Silver、Sabanovic、和 Francisco (2016)發現，老師採用形成性評量結合問題解決教學策略融入機器人程式課程，學生能拿回學習主導權，仍有意願投入自己的學習。老師實施形成性評量結合問題解決教學方法來得知學生的高層次認知與學習效果，學生使用問題解決步驟能找出可能解決方案，這樣教學方式比起過去教學方法是可能是更加有效的學習效果(Hsia, Su, & Lai, 2020; Lahtinen Ala-Mutka & Jarvinen, 2005; Mutiawani, Juwita, & Novitasari, 2017)。

3. 研究問題 Research Question

綜合上述，本計畫運用形成性評量結合問題解決教學方法融入程式設計課程，透過形成性評量(就像隨堂測驗)來得知學生的高層次認知與學習結果，了解學生對每個課程單元學習效果。進行一個學期程式設計課程活動之後，我們完成了隨堂測驗、期中考試成效、期

末考試成效、以及問卷調查等實驗活動。接著，我們探討老師使用形成性評量結合問題解決教學法融入程式設計課程對學生的形成評量與學習成效有何差異，以及學生在課堂上使用教學系統的感受為何。本計畫提出以下兩項研究問題，如下所示。

1. 運用形成性評量結合問題解決教學法融入程式設計課程對學生的形成評量與學習成效有何差異？
2. 運用形成性評量結合問題解決教學法融入程式設計課程對學生使用教學系統的感受為何？

4. 研究設計與方法 Research Methodology

(1) 課程對象

本計畫的研究對象為校內 67 位大學一年級到四年級的學生修資訊工程學系「程式設計(二)」課程。本計畫課程活動之前，我們實施本計畫知情同意書說明，得到 57 位大學一年級到四年級的學生同意。67 位學生會按照選課規定進行課程教學活動。課程活動結束之後，本計畫使用 57 位學生同意的課程教學數據進行呈現本計畫的教學暨研究成果。

(2) 學習教材

本計畫實施「程式設計(二)」課程的學習教材設計來自於深度學習 C++ (第三版) (子由, 2011)。該課程設計 16 週 C 程式語言系列單元與 C++ 程式語言系列單元的學習教材。我們設計 C 程式語言系列單元包含了 C 語言的基本結構、資料型態與運算式、基本的輸出與輸入、流程控制、遞迴、陣列與字串、函式、巨集、指標、動態記憶體配置、以及鏈結串列、堆疊、動態規劃。我們設計 C++ 程式語言系列單元就有從 C 到 C++、物件導向 C++、類別繼承、虛擬函式、運算子覆載、異常處理、標準樣板函式介紹、堆疊、佇列、向量、串列等演算法。從表一課程大綱得知，課程老師教授 C 程式設計為期六週、教授 C++ 物件導向程式設計為期十週、最後兩週實施實體期中成效測驗與線上期末成效測驗，最後問卷調查工作。

表一、課程大綱

週次	授課主題	教學大綱
第一週 至 第六週	C 程式設計	<ul style="list-style-type: none"> ● C 語言的基本結構 ● 資料型態與運算式 ● 基本的輸出與輸入 ● 流程控制 ● 陣列與字串 ● 函式 ● 巨集 ● 指標 ● 動態記憶體配置 ● 遞迴、鏈結串列、堆疊、動態規劃 ● 以上單元有含形成性評量(隨堂測驗)
第七週 至 第八週 、 第十週 至 第十七週	C++程式設計	<ul style="list-style-type: none"> ● 從 C 到 C++ ● 物件導向 C++ ● 類別繼承 ● 虛擬函式 ● 運算子覆載 ● C++字串 ● 異常處理 ● 標準樣板函式 ● 堆疊、佇列、向量、串列等演算法 ● 以上單元有含形成性評量(隨堂測驗)
第九週 第十八週	期中與期末成效測驗、以及問卷調查活動	<ul style="list-style-type: none"> ● 第九週期中成效測驗 ● 第十八週期末成效測驗 ● 問卷調查活動

(3) 實施程序

本計畫進行校內資訊工程學系一門「程式設計(二)」課程活動，這門課程為期 18 週，

每週一次，每週為 150 分鐘。

課程的第一週，老師講解課程教學目標和成績考核方式，接著介紹 ITSA 程式能力線上自我評量系統以及相關程式編輯工具使用方式。

接著，我們使用形成性評量結合問題解決教學法進行 16 週的「程式設計(二)」課程教學活動。老師於課堂上使用問題解決教學法結合形成性評量進行步驟式解析方式，包括第一步驟的陳述問題、第二步驟的瞭解問題、第三步驟的擬定解題計畫、以及第四步驟的執行計畫、反思與除錯。老師於課堂上講授某課程單元教材後，老師會實施隨堂測驗。學生完成隨堂測驗後，老師進行問題解決教學法來步驟式解析隨堂測驗答案。因為疫情影響，第六週進行同步線上課程教學活動，而期中成效測驗採實體紙筆考試。在期中成效測驗後，本課程持續進行同步線上課程教學活動到學期末，而期末成效測驗是採線上考試。

課程的最後一週，老師實施使用教學系統感受問卷調查活動。最後，我們收整隨堂測驗成效、期中成效測驗、期末成效測驗、問卷等課程數據等數據。後續進行分析數據研究工作，最後呈現本計畫的教學暨研究成果。

(4) 工具

本計畫實施課程研究工具，那就是課程單元簡報、課程單元的隨堂測驗、期中與期末成效、以及使用教學系統感受問卷。

課程單元簡報

課程單元簡報是課程老師參考深度學習 C++ (第三版) (子由, 2011) 自編課程學習教材，如表一課程大綱所示。

課程單元的隨堂測驗

課程老師使用問題解決教學法來設計每個課程單元的隨堂測驗題目，隨堂測驗題目包含選擇題、是非題、填充題等，總共 109 題，如表二的課程單元隨堂測驗之題目數量所示。老師透過隨堂測驗來得知學生對課程單元教材的高層次認知歷程學習狀況。

表二、課程單元的隨堂測驗題目數量

課程單元	隨堂測驗之題目數量
資料型態與運算式	4 題
基本的輸出與輸入	3 題
流程控制	5 題
陣列與字串	5 題
函式、巨集	5 題
指標、動態記憶體配置	8 題
遞迴、鏈結串列、堆疊、動態規劃	5 題
從 C 到 C++	4 題
C++類別、物件及覆載	7 題
類別繼承、虛擬函式	14 題
運算子覆載	10 題
C++字串	14 題
異常處理	6 題
標準樣板函式	7 題
堆疊、佇列、向量、串列等演算法	12 題

期中與期末成效

期中與期末測驗試卷是由經驗豐富的程式設計課程老師設計而成。

期中測驗試卷是能了解學生的 C 程式記憶、理解及程式綜合知識學習成效，期中測驗試卷有 7 題選擇題與 21 題問答題，總分為 100 分。

期末測驗試卷是能得知學生的 C++程式記憶、理解及程式綜合知識學習成效，期末測驗

試卷有 16 題問答題，總分為 100 分。

使用教學系統感受問卷

我們是依據 Chang 與 Su (2021) 提出使用教學系統感受問卷來修改成本計畫的使用教學系統感受問卷。該問卷題目有 4 題是「使用教學系統觀看教材時感受」、4 題「使用教學系統進行隨堂測驗時感受」以及 2 題「使用教學系統時感受」，共 10 題。使用李克特量表為 5 分，可分成非常符合(5 分)、符合、尚符合、不符合、非常不符合(1 分)。

5. 教學暨研究成果 Teaching and Research Outcomes

(1) 教學過程與成果

在課程教學過程與成果，老師使用問題解決教學法的四步驟程序在課堂上進行形成性評量活動。這四步驟程序分別為第一步驟的陳述問題、第二步驟的瞭解問題、第三步驟的擬定解題計畫、以及第四步驟的執行計畫、反思與除錯。

第一步驟是陳述問題：老師會引導學生理解隨堂測驗題目的意義。例如：學生有了資料型態與運算式、基本的輸出與輸入、流程控制、陣列、字串、函式、指標、動態記憶體、結構體等 C/C++ 程式語言基礎，老師先教導推疊的 Push 功能(把東西放入 Stack)與 Pop 功能(把最上面的東西從 Stack 取出)等基礎概念後，接著，老師提出隨堂測驗題目「請問 a、b、c、d、e、f 區要填入什麼程式碼，才能完成 Stack 的 Push_Stack 函式與 Pop_Stack 函式」。老師之前教過 Stack 的 Push 功能與 Pop 功能，解答隨堂測驗題目之目的。如果學生要完成 Stack 的 Push_Stack 函式，請填入 a、b、c 區正確程式碼。如果學生要完成 Stack 的 Pop_Stack 函式，請填入 d、e、f 區正確程式碼。

第二步驟是瞭解問題：老師會提醒隨堂測驗題目的回答重點所在，並確認學生瞭解該題解題之要求。例如：老師提醒學生使用 isFull 函式(這個函式是檢查 Stack 是否為空)與 isEmpty 函式(這個函式是檢查 Stack 是否已滿)的時機與搭配 value 整數變數與 *top 整數指標變數，接著 a、b、c、d、e、f 區可能搭配整數變數與指標變數，讓學生知道這樣 Stack

的 Push 功能與 Pop 功能才能正常運作。

第三步驟是擬定解題計畫：當學生理解隨堂測驗题目的解題要求時，老師會教導學生擬定一份解題計畫，解題計畫過程中，老師會給予學生適當彈性，以因應解題期間可能因某些無法預測之原因而導致課程活動之延長或縮短。例如：學生會先回顧老師先前教過推疊的 Push 與 Pop 功能概念。學生將學到推疊的 Push 與 Pop 功能運作知識來呈現示意圖，就像範例值帶入推疊進行 Push 與 Pop 功能運作過程。接著，學生將已知 Push_Stack 函式與 Pop_Stack 函式預設程式碼的變數改成範例值。

第四步驟是執行計畫、反思與除錯：學生實施這份解題計畫來解答隨堂測驗题目，學生會使用程式教學系統來實作並確認解題的正確性，發生實作錯誤會進行反思後更正式錯誤。例如：對應解題計畫，學生會從解題計畫找到 a、b、c、d、e、f 區可能搭配整數變數與指標變數，學生會使用程式教學系統來驗證程式碼的正確性。如果學生驗證程式過程中發生錯誤，老師會觀看程式教學系統來得知學生的程式碼錯誤地方，老師會再學生適合的解題提示，讓學生修正解題計畫與程式碼，最後學生解答了隨堂測驗题目。

(2) 教師教學反思

在這個章節，本計畫探討老師運用形成性評量結合問題解決教學法融入程式設計課程對學生的形成評量與學習成效有何差異。因此，我們使用統計分析期中學習成效與形成式評量成效，以及分析期中學習成效與形成式評量成效，分析結果如下。

期中學習成效與形成式評量成效之分析結果

在上半學期教學活動中，本計畫實施隨堂測驗與期中測驗的题目有些是不相同的，我們比較相同或相似的题目中成績有何差異。我們使用成對樣本 t 檢定進行分析，結果顯示了學生的期中學習成效與形成式評量成效是有達顯著性($p < .001$)。

結果顯示，學生的期中學習成效比起學生的形成式評量成效有顯著高的成績，這代表本計畫使用形成性評量結合問題解決教學法能改善學生的期中學習成效。

期末學習成效與形成式評量成效之分析結果

在下半學期教學活動中，本計畫實施隨堂測驗與期末測驗的题目有些是不相同的，我們比較相同或相似的题目中成績有何差異。我們使用成對樣本 t 檢定進行分析，結果

顯示了學生的期末學習成效與形成式評量成效是有達顯著性($p < .05$)。結果顯示，學生的期末學習成效比起學生的形成式評量成效有顯著高的成績，這代表本計畫使用形成性評量結合問題解決教學法能改善學生的期末學習成效。

(3) 學生學習回饋

在這個章節，我們探討老師運用形成性評量結合問題解決教學法融入程式設計課程對學生使用教學系統的感受為何。本計畫在課程最後一週實施使用教學系統感受問卷，該問卷有 10 題問題，採用李克特 5 點量表，那就是非常符合(5 分)、符合、尚符合、不符合、非常不符合(1 分)，共回收 57 份問卷。

在使用教學系統感受問卷中，我們發現使用教學系統觀看教材時感受向度、使用教學系統進行隨堂測驗時感受向度、及使用教學系統時感受向度的平均數皆大於 3.38，這表示課堂學生普遍認為使用教學系統來觀看教材以及進行隨堂測驗是有助於程式設計課程學習。

6. 建議與省思 Recommendations and Reflections

本計畫運用了形成性評量結合問題解決教學策略融入程式設計課程教學活動。結果發現，學生的期中/期末學習成效與形成式評量成效是有達顯著性，這表示期中/期末學習成效比起形成式評量成效有顯著高的成績，在隨堂測驗未獲得高分的同學，在期中/期末測驗時分數大幅提高。此外，我們發現下半學期形成式評量成效的平均數是高於上半年形成式評量成效的平均數，而且下半學期末成效的平均數是高於上半年期中成效的平均數。然而，學習成效與形成式評量成效的標準差差異不大，部分學生注重實作能力，並不在意理論知識，所以學習成效的成績提升有限(Achilleos et al., 2019)。在學生學習回饋中，學生認為教學策略與教學教具有助於程式設計課程互動及尋求協助，就像學生認為使用教學系統是有助於程式設計課程學習。我們期望其他老師能使用形成性評量結合問題解決教學活動對融入程式設計課程教學確實有幫助。

本研究的過程中有需改善的地方，以下提出兩點建議，分述如下：

1. 本研究缺乏準實驗研究設計，沒有規劃實驗組與對照組用於呈現比較實施形成性評量結合問題解決教學策略於程式設計課程的成效差異，因此較難以判斷學習成效是否提升。未來的研究可以實施準實驗研究設計於程式設計課程中比較實驗組與對照組學生之間學習表現有何差異。
2. 課程活動中發現學生使用不同的作業系統與程式編輯工具，無法取得學生實際撰寫程式碼歷程數據。未來希望能收整這部分數據分析學生於課程中撰寫程式學習行為，有助於了解學生對程式設計課程的學習表現。

一. 參考文獻 References

- Achilleos, A.P., Mettouris, C., Yeratziotis, A., Papadopoulos, G.A., Pllana, S., Huber, F., Jäger, B., Leitner, P. Ocsovszky, Z., & Dinnyés, A. (2019). SciChallenge: A social media aware platform for contest-based STEM education and motivation of young students. *IEEE Transactions on Learning Technologies*, 12(1), 98-111.
- Bey, A., & Bensebaa, T. (2012). Towards an e-assessment approach of algorithmic problem-solving skills using plan-based program understanding approach. International Conference on Education and e-Learning Innovations, Sousse, 1-4. doi: 10.1109/ICEELI.2012.6360666
- Chang C.Y. & Su Y.S. (2021). Effects of Programming Teaching System with Problem-based Learning Approach on STEM Learning Perceptions, Learning Performance, and Learning Behaviors in STEM Coding Education. Proceedings of International Conference of Innovative Technologies and Learning (ICITL 2021), virtual conference.
- Chung, I. -L., Chou, C. -M., Chang, C. -Y., Li, D. -K., & Hsu, C. -P. (2016). Adaptive problem-solving oriented programming learning system. 2016 International Conference on Applied System Innovation (ICASI), Okinawa, 1-3. doi:10.1109/ICASI.2016.7539851
- dos Santos, S. C. (2017). PBL-SEE: An authentic assessment model for PBL-based software engineering education. *IEEE Transactions on Education*, 60(2), 120-126. doi:10.1109/TE.2016.2604227
- Gao, P., Lu, M., Zhao, H. & Li, M. (2019). A new teaching pattern based on PBL and visual programming in computational thinking course. 2019 14th International Conference on Computer Science & Education (ICCSE), Toronto, ON, Canada, 304-308. doi:10.1109/ICCSE.2019.8845503
- Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. In proceeding of 10th International Conference on Engineering Education.
- Govender, I., & Grayson, D. (2006). Learning to program and learning to teach programming: A closer look. In proceeding of the ED-MEDIA 206-World Conference on Educational Multimedia, Hypermedia & Telecommunications.
- Gomoll, A., Hmelo-Silver, C. E., Sabanovic, S., & Francisco, M. (2016). Dragons, ladybugs, and softballs: Girls' STEM engagement with human-centered robotics. *Journal of Science Education and Technology*, 25(6), 899-914. doi:10.1007/s10956-016-9647-z
- Hsia, C. H., Su, Y. S., & Lai, C.F. (2020). Impact of using ARCS model and problem-based leaning on human interaction with robot and motivation. *Library Hi Tech*. Vol., ahead-of-print No. ahead-of-print.
- Hung, Y. C. (2008). The effect of problem-solving instruction on computer engineering majors'

- performance in Verilog programming. *IEEE Transaction on Education*, 51(1), 131-137.
doi:10.1109/TE.2007.906912
- Hung, Y. C. (2012). Combining self-explaining with computer architecture diagrams to enhance the learning of assembly language programming. *IEEE Transactions on Education*, 55(4), 546-551.
doi:10.1109/TE.2012.2196517
- Hung, H. C., Liu, I. F., Liang, C. T., & Su, Y. S. (2020). Applying educational data mining to explore students' learning patterns in the flipped learning approach for coding education. *Symmetry*, 12(2), 213.
- Lahtinen, E., Ala-Mutka, K., & Jarvinen, H. M. (2005). A study of the difficulties of novice programmers. In proceeding of the 10th annual SIGCSE conference on Innovation and technology in computer science education.
- Mutiawani, V., Juwita, R. P. F. A., & Novitasari, D. (2017). Implementing problem-solving method in learning programming application. 2017 International Conference on Electrical Engineering and Informatics (ICELTICS), Banda Aceh, 211-215. doi: 10.1109/ICELTICS.2017.8253263
- Peter, B., & Paul, R. (2006). Program annotations: Feedback for students learning to program. In proceeding of Eighth Australasian Computing Education Conference (ACE2006). 19-23.
- Su, Y. S., Yang, J. H., Hwang, W. Y., Huang, S. J., & Tern, M. Y. (2014). Investigating the role of computer-supported annotation in problem solving based teaching: An empirical study of a scratch programming pedagogy. *British Journal of Educational Technology*, 45(4), 647-665.
doi:10.1111/bjet.12058
- Su, Y. S., Ding, T. J., & Lai, C. F. (2017). Analysis of students engagement and learning performance in a social community supported computer programming course. *Eurasia Journal of Mathematics, Science & Technology Education*, 13(9), 6189-6201.
doi:10.12973/eurasia.2017.01058a
- Su, Y. S., & Chen, H. R. (2020). Social Facebook with Big Six approaches for improved students' learning performance and behavior: A case study of a project innovation and implementation course. *Frontiers in Psychology*, 11, 1166.
- Uysal, M. P. (2014). Improving first computer programming experiences: The case of adapting a web-supported and well-structured problem-solving method to a traditional course. *Contemporary Educational Technology*, 5(3), 198-217.
- 張基成。(2014)。數位化學習歷程檔案對學生知識蓄積之影響。課程與教學季刊，17(4)，207-228。
- 盧玟燕 & 范斯淳。(2019)。國中資訊科技導入 APCS 檢定之可行性與評量策略分析。工業科技教育學刊，(12)，67-81。

林昱達、劉達全、張清堯、蘇育生。(2022)。應用教育資料探勘技術探究學生學習特徵及預測學習表現。第十一屆工程、技術與科技教育學術研討會，屏東科技大學，屏東。

國家教育研究院(2020)。十二年國民基本教育課程綱要國民中學暨普通型高級中等學校-科技領域。

子由 (2011)。深度學習 C++ (第三版)。台北市：藍海文化。