

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：PHE1100361  
學門專案分類/Division：民生  
執行期間/Funding Period：2021/8/1~2022/7/31

專題式學習在非資訊科系程式設計課程之教學策略研究  
(配合課程名稱：行動裝置程式設計)

計畫主持人(Principal Investigator)：莊益瑞

執行機構及系所(Institution/Department/Program)：景文科技大學/資訊工程系

成果報告公開日期：

立即公開

延後公開(統一於 2024 年 9 月 30 日公開)

繳交報告日期(Report Submission Date)：2022 年 9 月 12 日

# 專題式學習在非資訊科系程式設計課程之教學策略研究

## 一、研究動機與目的

自 107 學年度開始，教育部在高教深耕計畫中要求每位大專生，在畢業前都要能修過程式設計相關課程，將大學生應具備基本程式設計能力的任務訂為各校的重點工作，並訂於 108 年度時達到 50% 學生修習過程式設計相關課程的量化目標。本校為配合此一目標，也於 107 學年度在各學院訂定一門程式設計方面的必修課，研究者也被指派，首次在非資訊科系教授程式設計課程。

由於當時對教育部推動此政策之目的不甚明瞭，僅簡單解釋為「從學習程式設計中培養邏輯思考能力」，授課內容仍以程式設計基礎課程為主，類似於資訊相關科系的初級程式設計課，包含變數宣告、運算式、條件判斷式、迴圈結構、副程式等。但實際授課時，發現學生的學習動機與學習成效，並不如預期。尤其原本就對程式設計沒有興趣的非資訊科系學生，上到條件判斷與迴圈結構單元時，較無法從拆解問題中規劃出程式設計的邏輯結構，在變數的運用上也卡關，更不用說後續的「副程式」單元。

此外，學生在學習程式設計時，較難運用運算思維所學習到的四種能力，包含問題拆解、模式辨識、抽象化和演算法等。在講解這四種運算思維能力時，大多數學生都能從非程式相關的範例中學習，但運用在寫程式的範例時就會顯得困難。這中間的轉換過程中似乎缺乏了中介的教材。其實，運算思維和程式設計是相輔相成的，熟練程式設計也會更熟練運算思維，若能順利地銜接好兩者之間的學習落差，則學生更能體會運算思維的問題解決模式。

研究者當時參考過去的教學研究成果，採用翻轉教學的策略，並在課堂上採用練習式教學法 (Practice and Drill)，以步驟式的教學，教師做一步，學生做一步，或者請學生依照書本一步步做。但大部分學生只會照著做，不知其所以然，也就無法舉一反三。在進一步做類似題或進階題時，就不會做了。從學生在期末填寫的問卷觀察，學生對於學習程式設計的目的質疑，也覺得程式設計很難，表示為什麼還是要修這門課，心中有所不解。其實，當教師對非資訊科系學生上程式設計課的意義沒有定位清楚，也會導致學生學習動機缺乏支持。其教材和教法也都應該不能和一般程式設計課程一樣，以培養問題解決能力為主，程式設計為手段，專業應用為發展。

基於上述在教學現場發現的問題，回想過去曾經上過畢業專題課程，在引導學生進入狀況時，曾經使用過「專題式學習」的模式教學，在引起動機和增進同儕合作有不錯的成效。專題式學習 (Project-Based Learning) 是課堂教學的一種綜合方法，旨在讓學生積極參與現實世界和對個人有意義的專題來學習 (BIE, 2017)，是一種情境學習的形式，基於建構主義的研究結果。當學生透過運用自己的構想積極地建立自己的理解時，他們會對教材有更深刻的理解，也能增強學習動機和求知慾。在探究專題的過程中，透過教師和同儕提供的支持，能使其動機和思考得以持續 (Krajcik & Blumenfeld, 2006)。在專題式學習中，學生會遇到對他們來說很重要且有意義的問題，這些問題類似於科學家、數學家、作家和歷史學家所做的事情，允許學生設定要研究的問題，提出假設和解釋，討論他們的想法，挑戰他人的想法，並嘗試新的想法。據研究文獻表明，專題式學習教室中的學生比傳統教室中的學生能得更高分 (Marx et al., 2004; Rivet & Krajcik, 2004; William & Linn, 2003)。

本研究以學生為中心規劃非資訊科系的程式設計課程，以貼近學生生活或所屬科系專業方面的專題，作為規劃運算思維和程式設計教材的情境，採用專題式學習的策略，啟發學生學習興趣，提供適當的鷹架，讓學生有足夠的參考資源，減少太多挫折的探索。並採用合作學習，彌補無法兼顧不同程度學生的缺憾。希望讓學生體會上這門課的好處，也能提升邏輯

思考和創造思考的能力，並培養帶得走的解決問題能力。因此，本研究以專題式學習法應用於非資訊科系之程式設計課程，透過教材與教法的修正與創新，預期能提升學生的學習動機、課堂參與度、運算思維能力和問題解決能力。

## 二、文獻探討

### 1. 運算思維與程式教育

自學者 Jeannette Wing (2006) 在 Communications of the ACM 期刊提出運算思維 (Computational Thinking) 一詞後，中小學的程式設計課程的議題逐漸被重視且推廣，許多國家的教育政策也逐漸納入程式設計教育，例如愛沙尼亞是全球推動程式設計教育最積極的國家，是最早將程式設計納入課綱的國家，自 2012 年起，學童從 7 歲開始從遊戲中學程式設計。前總統伊爾韋斯說：學習程式是為了善盡人力資源，讓每個人都能發揮所長（引自：親子天下文章 <https://flipedu.parenting.com.tw/article/4808>）。日本於 2017 年提出「未來投資戰略」計畫，敲定 2020 年的中小學義務教育中納入程式設計教育，文部科學省將其定義為培養“程式思維”，即通過進程式教育培養孩子們的思考能力（引自：社團法人台灣快樂學程式推廣學會文章 <https://coding4taiwan.org/news/1>）。英國在 2014 年將程式設計列入課綱，學童從 5 歲起開始學習 Scratch，期待 11 歲就能使用兩種語言（英語和程式語），鼓勵孩子從有趣的環境中玩程式，進而創新開發新程式，影響了當代的科技教育（張瀨文、賓靜蓀、程遠茜，2016; Inside 網路文章：<https://www.inside.com.tw/article/5137-coding-education>）。美國前總統歐巴馬在 2016 年提出「全民資訊科學 (Computer Science for All)」倡議，旨在倡導使一代美國學生能掌握在蓬勃發展的數位經濟中所需的計算機科學技能，而不僅僅只是消費者，在技術驅動的世界發展中，成為一個主動參與者。（引自：美國白宮部落格 <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>）。

台灣也在國民教育 108 新課綱中納入程式設計課程，在中等教育階段的科技領域課綱中，訂定資訊科技必修課程，學習內容包含「演算法」、「程式設計」等，目的為培養學生與時俱進的資訊科技基本能力，成為主動、積極且負責任的數位公民。國民中學教育階段以問題解決為主軸，強調培養學生利用資訊科技與運算思維解決問題之能力；高級中等學校教育階段則更重視整合運用，藉由資訊科學的初步探索，讓學生進一步理解運算思維之相關原理，以培養整合資訊科技與運算思維以有效解決問題之能力（引自：[https://www.k12ea.gov.tw/files/class\\_schema/課綱/13-科技/13-1/十二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.pdf](https://www.k12ea.gov.tw/files/class_schema/課綱/13-科技/13-1/十二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.pdf)）。

教育部為了未來大學程式教育能與中小學的資訊教育銜接，在引導各大學規劃高教深耕計畫時，已將大學生應具備基本程式設計能力的任務訂為各校應進行的重點工作，並訂有 108 年需達到 50% 學生修習過程式設計課程的量化目標。此計畫目的在使所有學生都有機會學習程式設計之能力，進而將所學應用在自己的專業或未來的職場，期待除了能讓學生具備程式設計或運算思維的基本素養外，也希望非資訊領域學生在學習程式設計後，能進一步擴大其所屬專業領域在資訊應用上的想像空間，進而達到其專業領域的典範轉移 (paradigm shift)，創造更多跨領域發展加值應用的可能，提升國家資訊力與整體競爭力（引自：<https://www.openedu.tw/plus.jsp>）。

推廣前期在實施上遇到許多問題，包含學生學習動機低落、教學教材內容過於艱澀、師資人數與經驗不足、學習環境配套有待提升及各校學生特質與需求差異大等。因此，教育部委託國立政治大學等十三所大學成立了「推動大學程式設計教學計畫」跨校協助機制。透過鼓勵各校制訂程式設計教育的推動策略與非資訊領域學生的修習課程地圖，希望能確保各校

程式設計相關課程的永續發展，以有效提升學生未來參與數位經濟的基本能力（引自：<http://plus.pro.edu.tw/about/project.jsp>）。

具體而言，我們希望將運算思維視為一種思維模型，它更多的是思維而非計算。儘管程式設計可以作為運算思維的一部分，但不應局限於計算機科學，而應廣泛應用於各種專業領域和日常生活中。例如有人運用計算模型用於匯總和分析新冠病毒危機的趨勢數據，以幫助預測多個國家冠狀病毒危機。如果沒有特別關注資訊處理的效率和品質的提高，我們可能會失去機會來培養學生正確的運算思維觀念，以及應對全球危機做準備的技能。更重要的是，學校課程和教學方法必須將運算思維納入學生的主要學習內容中，不僅是資訊科技和數學，應該還要包括 STEM 學科及其他學科（Li, 2020b）。

綜觀國內外在運算思維教學或程式設計教育的推展脈絡，運算思維不能單純地等同於資訊科技，程式設計課程也並非要把所有人都變成開發者，程式是一種「解決問題」的過程，希望能夠從小培養「運算思維」，包含數學、邏輯、演算法，學習把大的問題拆解成一系列更小、更容易處理的問題，最後組合成可以運作的解決方案。本研究以此重點為設計教材和教法的方向，清楚地讓學生明瞭學習運算思維的意義，體驗程式設計的樂趣，也能應用在自己的專業領域上。

## 2. 專題式學習

專題式學習（Project-Based Learning; PrBL）是課堂教學的一種綜合方法，學生可以通過積極參與現實世界和個人有意義的專題來學習，透過長時間工作來調查與回應真實、有趣且複雜的問題或挑戰，從而獲得知識和技能（Blumenfeld et al., 1991; BIE, 2017）。在專題式學習中，學習者彼此合作以解決具有挑戰性的問題，而這些問題都是真實世界的問題，是基於課程規劃的內容，並且經常是跨學科的。學習者可以自己決定如何解決問題以及進行什麼探究活動。從各種來源收集資訊，然後綜合分析，並從中獲取知識。其真正的價值在於，因為它與真實的事物聯繫在一起，能學習到合作與反思的技能。在整個過程中，教師的作用是引導和建議，而不是直接指導和管理（Krajcik & Blumenfeld, 2006; Solomon, 2003）。

專題式學習的科學教室可讓學生探索現象，調查問題，討論他們的想法，挑戰他人的想法並嘗試新的想法。專題式學習是一種設計學習環境的整體方法，此基於專題的學習環境具有五種特點（Krajcik & Blumenfeld, 2006）：

- (1) 從一個需要解決的主導問題（driving question）開始。
- (2) 學生通過參與真實情境來探索主導問題，過程中將學習與應用該學科的重要概念。
- (3) 學生、老師、甚至是社區成員一起參與協作活動，以找到解決問題的方法。
- (4) 在進行探究過程中，學生可藉由學習科技的鷹架（scaffolding），協助他們參與超出能力範圍的活動。
- (5) 學生會建立一個解決問題的方案，此方案以公開分享方式呈現。

依 Bransford、Brown 和 Cocking（1999）等學習科學（learning science）專家的研究，學習者在下列四個關鍵情境中可以產生學習動機與行為：

第一、主動建構知識：學習科學研究發現，當學習者根據自己的經驗和在世界上的互動而主動建構其含義時，就會產生深刻的理解（understanding），而當學習者被動地吸收從老師、電腦或書本中傳遞的資訊時，只會發生表面學習。理解的發展是一個連續的過程，需要學生建構和重構他們從新的經驗和想法以及先前的知識和經驗中學到的知識。教師和教材不會向學習者透露知識；相反，學習者在探索周圍世界，觀察現象並與之互動，吸收新想法，在新舊想法之間建立聯繫，以及與他人討論和互動時，會積極地積累知識。在專題式學習中，學生通過參加類似於專家參與現實世界中的探究活動，來積極地構建自己的知識，以解決問

題並產出結果。

第二、營造情境學習：學習科學研究顯示，最有效的學習是在真實的環境中進行的。在某些科學類學科中，科學家在實驗室中進行實驗。有系統地觀察自然世界，並從觀察中得出結論。在科學中進行的情境學習將使學生參與各種科學實踐，例如設計調查、進行解釋、建立模式、以及向他人展示自己的想法，從而體驗其產生的現象。在專題式學習中，當學生按照自己規劃的步驟進行科學實驗時，這幾乎比被動地聽課更好。當他們創建自己的探究計畫來解決問題時，他們可以看到如何將科學應用於解決重要問題。

第三、增進社會互動：從學習科學研究中得出最可靠的發現之一是社會互動在學習中的重要作用，最好的學習源自一種特殊的社會互動，當老師、學生和社區成員共同開展一項現場活動，以建立共識時，學習者通過與他人共享、使用和辯論想法，發展對原理和想法的理解 (Blumenfeld et al., 1996)。專題式學習的探究過程中，參與人員之間的互動是必要且珍貴的，因為這是產生學習行為的重要活動。

第四、使用認知工具：學習科學研究已經證明了工具在學習中的重要作用 (Salomon, Perkins, & Globerson, 1991)。認知工具 (cognitive tools) 可以擴大學生的學習範圍。例如統計圖是認知工具，可以幫助學習者查看數據中的模式。各種形式的電腦軟體都可以視為認知工具，因為它們可以使學習者執行沒有該軟體的幫助和支持就不可能完成的任務。專題式學習過程中，透過學習科技的使用，可以拓展學生探索的範圍，體驗更多種的現象。

教師若能依據上述四個情境，設計有意義的專題式教材與教法，將有助於學生在學習運算思維或程式設計時，能更具體且深入了解其意義，進而可以運用運在專題探究上。

### 3. 結合運算思維與專題式學習的文獻探討

經檢索與分析含有運算思維（或程式教學、程式教育）及專題式學習這兩個關鍵詞的教學實踐文獻，有的運用運算思維於專題式學習，也有運用專題式學習於運算思維。在年齡層上從國小到大學都有，但以大學生為例者居多，但篇數不多。

Benakli 等人 (2016) 設計了 9 項計算機實驗專題，每一項計畫都劃分成幾個小部分，並提供鷹架支持學生一步步順利的解題。過程中，利用統計語言 R 模擬、視覺化和數據分析來促進 STEM 的運算思維，目標是嘗試 R 語言提供的功能，而不是全面性的教導 R 語言。程式設計可幫助學生注意細節並真正理解數學問題，並遵循正確的數學邏輯實施解決方案。將程式設計視為解決計算問題的一種形式，與純粹的數學形式的問題解決交織在一起。研究發現，這些活動的確能使學生保持積極參與學習和解決問題的過程，並發展出更好的直覺來理解複雜的數學概念。

Volcz (2018) 運用運算思維在「數位素養」課程中的專題式學習，參與者是大學預科 (pre-university) 學生，此課程並非要訓練學生數位技術，而是要學生應用運算思維四種技巧 (拆解、辨識特徵、抽象化和演算法) 來創建合作專案，培養學生問題解決能力。透過問卷調查，有 85% 的學生正向回饋表示運算思維技巧有助於解題，有助於學習其他科目，以及將上大學後會持續使用此技巧。

學者 Lu 和 Lin (2019) 以運算思維為基礎，結合專題式學習進程式設計教學活動設計。對象是大學生，主題是旅遊導覽的專題。學生透過老師的引導將製作專題之流程進行拆解，學習程式設計邏輯。接著運用積木程式語言進行教學，讓學生在短時間內製作出自己預想的內容，學生會因為製作出成果後獲得成就感，並與同儕進行分享。從教學記錄中也可發現，受試者對於專題設計很感興趣，也會進行自主性思考。研究結果指出，透過專題導向學習的方式可有效提升學生自主學習的動力、成就感、以及參與感。

在國家圖書館博碩士論文知識加值系統 (2020 年 12 月 12 日於 <https://ndltd.ncl.edu.tw/>)

中檢索運算思維（或程式設計、程式語言）及專題式學習這兩個關鍵詞的教學實踐論文，在國中端有 1 篇，高中職端有 2 篇，大學端則有 3 篇（黃柏軒，2020；陸香如，2019；賴信豪，2019；楊依蓉，2010；嚴尹晨，2019；葉佳忠，2019）。綜論其研究成果，專題式學習使用在程式設計相關課程時，有助於提升學習動機、學習成就及滿意度。在黃柏軒（2020）的研究中設計跨域學生於運算思維的教學上，並以專題式學習進行教學，對提升學生高層次思考有顯著幫助，印證了專題式學習在「思維」（thinking）教學上有正面的影響，鼓舞了本研究的信心。在賴信豪（2019）的研究中專題式學習對學生的問題解決能力和學習態度並沒有產生正向的作用，作者在文末檢討研究成果時，發現教師在專題式學習的課前準備不足，以及課堂上對學生如何進行小組討論及問題解決，教師的協助都不夠。此外，在發表與評量階段，教師應採取多元評量方式，設計同儕互評表時也都需要有不同向度，並給予適當意見。這些經驗都可以提供本研究在設計專題式學習時很重要的參考。另外，葉佳忠（2019）的研究結果顯示不論是否有資訊背景的學生，在學習成績表現上並無差異，足見專題式學習可以應用在具有不同知識背景的學生，提供一個改善的可能性。

### 三、研究問題

依研究目的及上述研究架構中的假說，本研究待答的問題如下：

1. 採用專題式學習法是否能提升學生學習程式設計的興趣？
2. 採用專題式學習法是否能提升學生學習程式設計課程的課堂參與度？
3. 採用專題式學習法是否能提升學生的運算思維能力？
4. 採用專題式學習法是否能提升學生的問題解決能力？
5. 非資訊相關科系的程式設計課程採用專題式學習法後，學生的學習興趣、課堂參與度、運算思維能力和問題解決能力等，是否存在相互影響的關係？

### 四、研究設計與方法

本計畫將以本校行動商務與多媒體應用系在大二開設的「行動裝置程式設計」課程為例進行教學實驗，由於該系目前大二只有一個班級，在無法找到相同背景及無法同時開設兩班的情況下，將採用「前實驗設計（pre experimental design）」之研究設計，整學期的前八週將採用「學思達教學法」（如圖 2），期中考後的八週則採用「專題式學習」（如圖 3）。

#### 1. 研究設計

依本研究前述之研究目的來設定研究架構，以學生在不同教學方法（學思達教學法和專題式學習法）中的相關表現當作自變項，研究目的中欲提升的學習興趣、學習成就、課堂參與度和問題解決能力等為依變項，進行量化與質化的研究（H1~H4），並進一步探討四個依變項之間是否存在相互影響的關係（H5~H10）（參考圖 1）。研究假說如下：

- H1：採用專題式學習法比學思達教學法更能引起學生的「學習興趣」。
- H2：學生在採用專題式學習法課程上的「問題解決能力」比學思達教學法來得好。
- H3：學生在採用專題式學習法課程上的「課堂參與度」比學思達教學法來得高。
- H4：學生在採用專題式學習法課程中的「運算思維能力」比學思達教學法來得好。
- H5：「學習興趣」愈高，「問題解決能力」就愈高，反之亦同。
- H6：「學習興趣」愈高，「課堂參與度」愈高，反之亦同。
- H7：「學習興趣」愈高，「運算思維能力」愈高，反之亦同。
- H8：「問題解決能力」愈高，「運算思維能力」愈高，反之亦同。
- H9：「問題解決能力」愈高，「課堂參與度」愈高，反之亦同。

H10：「課堂參與度」愈高，「運算思維能力」愈高，反之亦同。

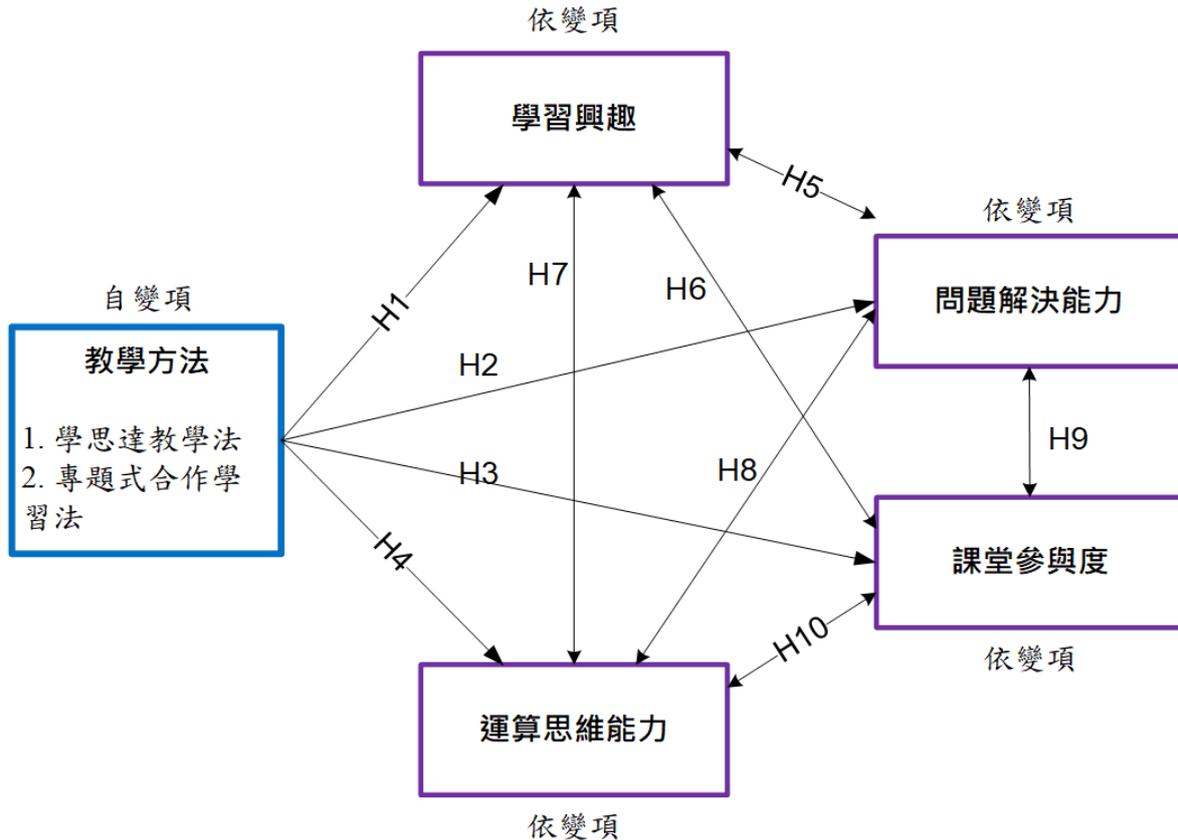


圖 1. 研究架構

## 2. 研究方法與工具

在量化研究方面：

- (1) 研究者參考 Pintrich & De Groot (1990) 的「動機與自我調節學習量表 (Motivational and self-regulated learning scale)」中動機信念 (motivational beliefs) 的題目、葉昭吟 (2014) 的「學習興趣與課堂參與調查問卷」、以及自行編製了一份 Likert-5 點量表，稱「教學法評價量表」，包含學習興趣、課堂參與度 (包含參與感、同儕互動和師生互動)、思考能力、自信心和成就感等向度。此量表於期中評量和期末評量各填寫一次，用來評估學生在對兩個教學方法的想法與感受。
- (2) 在「運算思維能力」方面，本研究依學生程度挑選「國際運算思維挑戰賽 (Bebras)」(<http://bebras.csie.ntnu.edu.tw/>) 題庫中的題目，依挑戰賽規則進行測驗。教師依難易度挑選 10 題，編製成「運算思維能力測驗」，於學期末進行測驗。
- (3) 在「問題解決能力」方面，以學習成就評量為主，由教學者出題，包涵選擇題和實作題，在期中考及期末考進行測驗，用以比較實施不同的教學法之後，問題解決能力的差異。另外，專題成果採用教師評分及同儕互評的機制評量之。

上述量表及測驗於施測時均採用匿名方式，而為了之後方便進行成對樣本 T 檢定，每位學生在施測前均隨機給予一個編號，在前後測時填寫之，以作為識別之用。

在質化研究方面，使用焦點團體訪談，以學生立場來看專題式學習法進行過程中的問題，包含學習興趣、學習成就、專題式學習和整體評價等，形成訪談大綱，選擇低、中、高成就學生各 3 名，以及在觀察紀錄中學習態度較佳與較差的同學各 1 名，進行焦點團體訪談，釐清學生實際的想法。此外，我們也透過觀察學生分組討論及課堂參與情形，包括學生如何彼此協調與協助、如何解決問題、表現態度等來評估學生的表現。

## 五、教學暨研究成果

### 1. 教學過程

本研究以本校商管學院行動商務與多媒體應用系大二開設的「行動裝置程式設計」課程為例進行教學實驗。課程分二階段。第一階段：第 2~8 週（共 7 週）以學思達教學法教授運算思維及程式設計的基礎（教學流程如圖 2），第二階段：第 10~17 週（共 8 週）以專題式學習進行專題製作（教學流程如圖 3）。

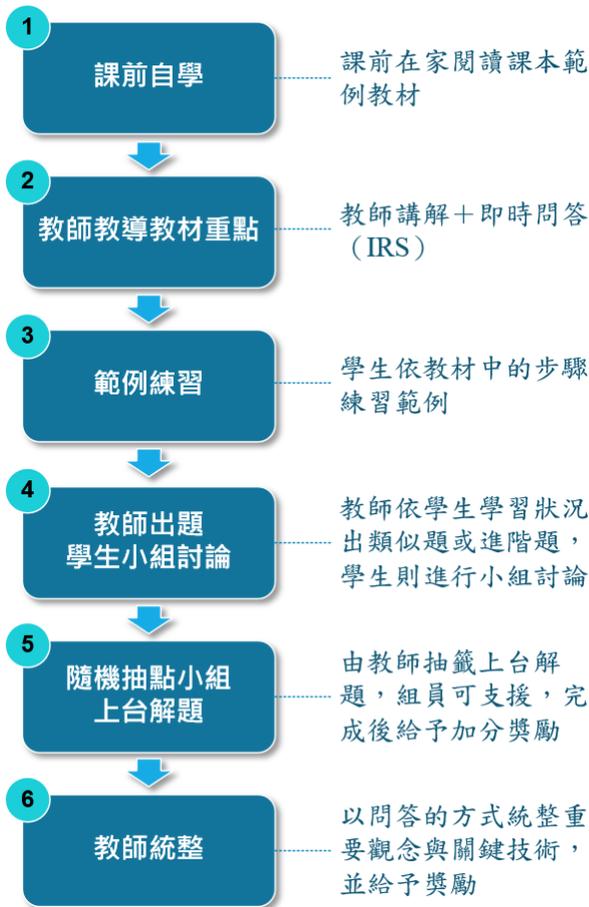


圖 2. 學思達教學法實施步驟



圖 3. 專題式學習法實施步驟

第二階段規劃較大且與學生切身相關的主題，沒有指定的功能或製作步驟，會以專題式學習的實施步驟引導學生製作專題的教材為主。而且教師會以「運算思維」和「程式設計」來引導學生進行專題的企劃與實作，使學生能學習應用此兩項知識與技能於問題解決上。

專題式學習實施步驟方面，從文獻 (Edutopia, 2006; Hayes, 2013; Indarti, 2016; Jalinus, Nabawi, & Mardin, 2017; Stix, & Hrbek, 2006) 上歸納專題式學習的實施步驟分為 6 個步驟 (如圖 3)：

步驟 1：介紹主題與專題進行方式。教師於課前準備好主題背景與需求說明，於課堂上引導學生體會製作此專題的重要意義，並說明未來幾週要如何進行小組合作，規劃與執行此專題。

步驟 2：師生確認探究主題並擬定執行計畫。各小組依步驟 1 的說明內容提出欲進行的專題題目，擬訂探究議題、學習任務及待解決問題，透過口頭報告，徵詢老師和同儕的建議。

步驟3：執行計畫。學生在進行計畫過程中，教師每週授課時可以簡要提醒學生運用運算思維來解決問題，並加入小組討論，必要時提供鷹架支持。

步驟4：進度報告與追蹤。此步驟與步驟3重複執行多次，要求學生每週撰寫進度報告，老師給予指導。

步驟5：成果發表與師生評量。本教學執行期間共9週，安排期中和期末兩次成果發表。期中發表內容主要以「我愛景文 App」的功能規劃與人機界面設計為主，期末則發表實作作品 App 展示為主，過程中教師和同儕共同給予評價與回饋。

步驟6：檢討與省思。此步驟是專題式學習的重要步驟，會安排各小組回顧並分享專題進行前、中、後的酸甜苦辣，遇到什麼問題，學習到什麼。

本教學實驗對象是商管專業的學生，可以運用自己所學習的商務專業知識，規劃與設計一個相關的 App。藉由專題式學習的模式，跨域學習與思考相關的議題，並以運算思維來思考解決問題的方法，最後透過線上 App 設計工具 App Inventor 2 完成所規劃的專題內容，印證運算思維在專業上的有效應用。

## 2. 教學與研究成果

由於兩種教學法的對象是同一個班級的學生，分別於實施後填寫的「教學法評價量表」，因此以成對樣本 T 檢定進行資料統計與評估其差異，得到結果如表 1。

本量表施測時均採用匿名方式，並且做好前後追蹤之成對樣本 T 檢定，每位學生在施測前均隨機給予一個編號，此編號只有學生自己知道，在前後測時填寫之，以作為識別之用。

表 1. 「教學法評價量表」的成對樣本 T 檢定結果

相關的 依變項	向度 (n=35)	成對變數差異					t	自由 度	顯著性 (雙尾)
		平均數	標準差	平均數 的標準 誤	差異的 95%信 賴區間				
					下界	上界			
學習興趣	學習興趣	.343	.906	.153	.032	.654	2.240	34	.0318*
課堂參與度	參與感	.286	.667	.113	.056	.515	2.533	34	.0161*
	同儕互動	.486	.818	.138	.205	.767	3.513	34	.0013**
	師生互動	.229	1.003	.169	-.116	.573	1.349	34	.1863
運算思維	思考能力	.343	.938	.158	.021	.665	2.163	34	.0376*
問題解決 能力	自信心	.086	.742	.126	-.169	.341	.683	34	.4992
	成就感	.343	.873	.147	.043	.643	2.325	34	.0262*

註：\* $p < .05$ , \*\* $p < .01$

### (1) 學習興趣 (H1):

從表 1 中呈現的結果得知學習興趣向度的 T 檢定結果，顯示有達到顯著 ( $p = 0.0318 < .05$ )，表示學生對專題式學習的學習興趣顯著高於學思達教學法。在焦點團體訪談時，學生也曾表示能在專題式學習中透過小組討論將程式設計應用在自己專業領域，比較有真實的感受到學習程式設計的用意，進而對程式設計課程的興趣有所改變，學習動機也因此提升。

## (2) 課堂參與度 (H3)

課堂參與度在表 1 教學法評價量表上共有三個向度，包括參與感、同儕互動與師生互動，其中參與感和同儕互動達到顯著，表示學生認為專題式學習比學思達教學法有更好的參與感與同儕互動，而在師生互動上沒有顯著差異。透過焦點團體訪談的內容得知學生在專題式學習時，因為有大量的小組討論機會，提升了較高的參與感；而學思達教學法時大多數時間都是老師在與學生討論，有較多師生之間的問答。兩者比較下來，兩種教學法各有其好處，也各有其不同的互動程度。

## (3) 問題解決能力 (H2)

以期中評量作為學思達教學法的學習成就評量，期末評量作為專題式學習的學習成就評量和專題成果評量來計算，用來評估學生的問題解決能力。由於有部分同學僅參加其中一次評量，無法進行前後比對來進行成對樣本 T 檢定，故刪除這類受測者後餘 33 位。兩次評量的題目類型皆為選擇題和實作題，選擇題大多屬於記憶與理解的問題較多，實作題則需要學生思考解決問題的方法，透過程式設計技巧解題。將測驗成績進行成對樣本 T 檢定的結果如表 2，顯示期末評量與期中評量之間有顯著的差異 ( $p < .05$ )，期末評量顯著優於期中評量，亦即專題式學習的學習成效優於學思達教學法。此外，從表 1 教學法評價量表中的成就感向度來看，學生在專題式學習中所得到的成就感優於學思達教學法，更驗證了專題式學習能提升學生的問題解決能力。不過，在自信心方面，兩個教學法之間沒有顯著差異。

表 2. 期中評量與期末評量的成對樣本 T 檢定結果

成對樣本統計量				
	平均數	個數	標準差	平均數的標準誤
期末評量	82.65	33	6.421	1.118
期中評量	74.14	33	17.594	3.063

成對樣本檢定								
n=33	成對變數差異					t	自由度	顯著性 (雙尾)
	平均數	標準差	平均數的 標準誤	差異的 95%信賴區間				
				下界	上界			
期末評量 - 期中評量	8.512	20.016	3.484	1.415	15.610	2.443	32	.020*

註：\* $p < .05$

## (4) 運算思維能力 (H4)

在期中考和期末考期間除了進行上述成就評量外，學生還進行了「運算思維能力測驗」，由於其前後測的人員不太一樣，去除僅做單一測驗的受測者後剩下 39 位，將其前後測的平均分數進行成對樣本 T 檢定，相關統計數據請參見表 3。雖然從平均數來看，後測（使用專題式學習之後）平均數高於前測（使用學思達教學法）平均數高了 4.872 分，但從顯著性來看並未達到顯著 ( $p > .05$ )，表示專題式學習期間學生的運算思維能力並沒有顯著提升。

此情形可以解釋為學思達教學法期間主要課程內容在於培訓學生具備程式設計能力，而專題式學習期間，主要的教學重點在於應用前半學期所學，致力於專題的製作，因此想在這期間再次提升運算思維能力的確是有限的。不過，至少學生沒有退步，還能將所學運用在專

題製作上，也算是某種程度上的提升。此外，從表 1 中思考能力向度的施測結果來看，學生認為專題式學習比學思達教學法更有助於提升其思考能力。

表 3. 運算思維能力測驗前後測之城對樣本 T 檢定結果

成對樣本統計量				
	平均數	個數	標準差	平均數的標準誤
後測	63.85	39	11.149	1.785
前測	58.97	39	13.726	2.198

成對樣本檢定								
	成對變數差異					t	自由 度	顯著性 (雙尾)
	平均數	標準差	平均數的標 準誤	差異的 95%信賴區間				
				下界	上界			
後測 - 前測	4.872	15.873	2.542	-0.274	10.017	1.917	38	0.063

### (5) 相關性 (H5~H10)

我們將實施專題式學習的教學法評價量表、期末評量和運算思維後測的數據，刪除有缺漏的施測資料列，共有 31 筆資料列，進行 Pearson 相關係數考驗，相關係數計算結果如表 4。其中學習興趣與課堂參與度達到顯著相關，表示學習興趣愈高則課堂參與度愈高，也符合一般認知。

表 4. 依變數之間的 Pearson 相關係數考驗結果

		學習興趣	課堂參與度	問題解決能力	運算思維能力
學習興趣	Pearson 相關	1	-	-	-
	顯著性 (雙尾)	-	-	-	-
	個數	31	-	-	-
課堂參與度	Pearson 相關	<b>.793**</b>	1	-	-
	顯著性 (雙尾)	<b>.000</b>	-	-	-
	個數	31	31	-	-
問題解決能力	Pearson 相關	-.267	-.160	1	-
	顯著性 (雙尾)	.147	.389	-	-
	個數	31	31	31	-
運算思維能力	Pearson 相關	.048	-.067	.162	1
	顯著性 (雙尾)	.799	.719	.383	-
	個數	31	31	31	31

註：\*\* $p < .01$

### 3. 教師教學反思

從研究結果來反思使用專題式學習對學生的學習興趣、問題解決能力、課堂參與度和運

算思維能力的提升，均達顯著水準，整體來說，可以看到有不錯的成效。其實，專題式學習所強調的論點是基於建構主義的研究結果，學生透過運用自己的構想積極地建立自己的理解時，會對教材有更深刻的理解，也能增強學習動機和求知慾。在探究專題的過程中，透過教師和同儕提供的支持，能使其動機和思考得以持續 (Krajcik & Blumenfeld, 2006)，因此實施過程中教師給予學生的鷹架 (scaffoldings) 必須拿捏好涉入的程度。給太少的話，學生會陷入放棄的狀態；給太多的話，減少了學生思考的空間和創新的可能性，甚至降低學習興趣。

以此論點來看研究問題中的問題解決能力，我們看重的是學生製作專題的過程是否有產生學習行為與結果，以前半學期所學習與練習得到的知識與技能，可以運用在專題上，甚至願意探索未知的知識與技能，建構屬於自己的新知識。因此，未來在評估問題解決能力時，不僅要從學習成就測驗來檢驗，更要從平時的小組合作學習中觀察學生的改變，寫下研究日誌或週誌，累積自己的收穫與發現。

此外，專題式學習中必要的分組合作學習也是一大挑戰，尤其在學生背景多元、程度不一的情況下，對於共同合作完成專案的經驗生澀，團體中若有不合作者，常會拖累其他組員，勞逸不均容易造成更嚴重的社會性懈怠 (social loafing) 狀況。在進行專題式學習時，教師可能無法很仔細的觀察到小組的討論狀況，無法即時發現上述現象，學生也可能礙於情面而未能及時反應回報老師。

研究者為了改善此一現象，會在每次有小組討論時，於課程結束前請同學填寫同儕互評的表單，評量組內每一位同學的表現，題目是以 Likert-5 點量表 (1-非常不同意, 2-不同意, 3-沒意見, 4-同意, 5-非常同意) 加上一題問答題來呈現，如下：

- (1) 這位同學參與討論的積極度很好。
- (2) 這位同學很願意配合小組討論結果。
- (3) 這位同學對小組的貢獻很多。
- (4) 這位同學很願意幫助其他組員。
- (5) 這位同學能力很強。
- (6) 關於這位同學，我有話要說：\_\_\_\_\_

一方面可以藉此提醒每一位同學要積極參與小組活動，一方面可以發現小組中有哪些同學無法融入群組，教師可進一步關心與溝通協調。

#### 4. 學生學習回饋

透過問卷和焦點團體訪談，摘錄學生的原始回饋內容，分正向與負向如表 7。

表 7. 學生正負向回饋摘錄

正向回饋	負向回饋
<ul style="list-style-type: none"> <li>• 比較能自己找到重點</li> <li>• 可以一起跟同學製作</li> <li>• 可以得知同學學的課程能不能活用</li> <li>• 能和組員一同討論學習</li> <li>• 能跟同學一起想出專案並做出來，會很有成就感</li> <li>• 好處是可以自己摸索，更容易學習</li> <li>• 可以透過跟同學的討論一起製作期末專題</li> <li>• 可以自己學到東西</li> <li>• 想做的東西有時候會超越自己學過的知識</li> </ul>	<ul style="list-style-type: none"> <li>• 一定會遇到沒貢獻的學生</li> <li>• 要認真學習 要報告很心痛</li> <li>• 學生有可能進度拖欠讓課程很難進行</li> <li>• 有些同學無法成功做出成果</li> <li>• 很看組員不然都會變一個人在做</li> <li>• 時間拉長就會導致惰性的出現</li> <li>• 如果卡住會比較難以前進</li> <li>• 要報告</li> <li>• 可能會遇到擺爛的組員</li> <li>• 困難增加，變得更不想學這門課程，後面</li> </ul>

- 
- 做出的成品可能失敗或偏題
  - 非常自由可以大多由學生來互動完成的模式
  - 能展現出同學不同的創意
  - 可以透過報告是學習知道自己的程度
  - 可以讓學生們比較融入當下，但是比較厲害的同學就會做的比較多
  - 透過討論能讓人更加投入在學習內容上
  - 跟同學互動變多可以培養團隊合作與實作
  - 能檢討、修正完善報告
  - 多參與課程多表達
  - 可以親身體驗到老師教的功能的實用性，學以致用
- 變成專題式設計腳本教學就已經放棄這科了
  - 不會還是沒辦法做
  - 很常需要講話發表可能會讓內向的人非常有壓力尤其是不常參與製作的同學沒辦法很好的報告
  - 偏向單一個主題，只能從別組學到不同概念
  - 對於頻繁報告參與度不高
- 

綜合歸納學生的回饋，採用專題式學習對學生的自主學習、合作學習、學以致用、同儕互動、表達能力、學習動機等有所幫助，呼應本研究的初衷欲改善學生對學習運算思維或程式設計的迷思，也能將所學應用於自己本科專業領域上。不過，學生對於與他人合作時的溝通協調的經驗不足，當組員對專題製作的品質、進度和要求有所不同，常難以達成共識，甚至有組員不配合或已經放棄、擺爛的情形，需要教師適時介入與關心，必要時得重新分組。此外，學生的表達能力有待加強，會擔心開口分享、上台報告和跟老師回報進度時表現不佳，畏懼在眾人面前講話。教師可以適時的以一問一答的方式慢慢引導學生思考與分享，再慢慢退出這樣的輔助，畢竟表達能力需要一點一滴的累積經驗，再透過老師和同儕的鼓勵來建立自信，應可漸入佳境。

## 六、建議與省思

教育部推動大學生必修運算思維相關課程的政策，除了任課老師要很清楚之外，更要謹慎向學生解釋該課程的「弦外之音」，若僅簡單解釋為「從學習程式設計中培養邏輯思考能力」，則非資訊科系的學生仍會帶著抱怨與無奈來修課。教師必須從教材、教法上改變，讓學生樂於學習、樂在學習和啟發學習。也就是能將運算思維或程式設計的課程內容換成另一種形式的呈現，以及結合學生本科專業背景的應用，讓學生體驗跨學科、多元智能的融合可以創造出意想不到、創新、創意的作品，有助於未來職場上的發揮。

專題式學習意圖從一個主導問題開始發想解決策略，透過團隊合作找到解決方法，在實踐的過程有老師的指導和同儕的互助，發揮潛能完成專題。其所重視的不只是結果，更重要的是過程，能在過程中體驗運算思維和程式設計的知識，可以為他們帶來不同的思維模式與解決方法。本研究運用專題式學習於非資訊科系學生的程式設計課，實驗結果顯示專題式學習能有效提升學生學習興趣、課堂參與度、運算思維能力、問題解決能力等，同時也能促進學生更多思考、討論與表達的機會，可以提供未來非資訊科系學習運算思維或程式設計課程的教學參考模式。

然而，在實施專題式學習的策略時，要多關注學生分組合作學習的情形，不可避免的會發生社會性懈怠的現象，而因為學生的溝通協調能力尚有不夠圓融之處，容易產生紛爭與勞逸不均的抱怨，教師需要多給予輔導和練習的機會。此外，學生在自我探索及問題解決能力不足的情況下，剛開始要給予較多的鷹架輔助，提供較多的資源，降低困難度，之後再慢慢拆除鷹架，給學生發揮創意的機會。以開放性的問題引導思考，鼓勵多元化的解決方案。

最後，本研究發現學生的表達能力較弱，上台分享意願低，尤其在疫情嚴峻遠距同步教學時期，不願露臉也不願開麥克風融入課程。建議教師在實施專題式學習策略時，若是實體課的話，可以先讓小組內部彼此分享的方式進行，降低上台分享的焦慮與壓力。之後再由各組推派一位或由老師抽籤一位代表小組上台分享，並給予適度的問題引導，與老師對話的方式進行之，再逐漸讓學生自由發揮。至於遠距同步教學的情境下，可以讓各組代表在課前先自拍分享影片，於遠距教學時播放，播放後再由教師詢問問題，學生自在回答，看看是否可以改善無人回應的窘境。

以上建議仍待未來持續的教學實踐研究，找出可行的教學策略與方法，以輔助使用專題式學習的不足，亦可從中發現創新的、改良的專題式學習模式。

## 參考文獻

- Benakli, N., Kostadinov, B., Satyanarayana, A., & Singh, S. (2016). Introducing computational thinking through hands-on projects using R with applications to calculus, probability and data analysis, *International Journal of Mathematical Education in Science and Technology*, 48(3), 393-427.
- BIE (2017). What is project-based learning (PBL). 2020/12/10 Retrieved from [http://www.bie.org/about/what\\_pbl](http://www.bie.org/about/what_pbl)
- Blumenfeld, P. C., Marx, R. W., Krajcik, J. S., & Soloway, E. (1996). Learning with peers: From small group cooperation to collaborative communities. *Educational Researcher*, 25(8), 37-40.
- Blumenfeld, P.C., Soloway, E., Marx, R.W., Krajcik, J.S., Guzdial, M., Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist*, 1991, 26 (3 & 4), 369-398.
- Bransford, J., Brown, A. L., & Cocking, R. R. (1999). *How people learn: Brain, mind, experience, and school*. Washington, DC: National Academy Press.
- Edutopia (2006). 2020.12.12 retrieved from: <https://edut.to/3mdfmPT>
- Hayes (2013). 2020.12.12 retrieved from: <https://bit.ly/2Wn8Cod>
- Indarti (2016). Implementing project-based learning (PBL) in final collection to improve the quality of fashion design student. *Innovation of Vocational Technology Education*, xii(1), 22-30.
- Jalinus, N., Nabawi, R.A., & Mardin, A. (2017). The seven steps of project based learning model to enhance productive competences of vocational students. *Advances in Social Science, Education and Humanities Research*, 102, 251-256.
- Krajcik, J.S., & Blumenfeld, P.C. (2006). Project-based learning. In Sawyer, R.K. (Eds), *The Cambridge Handbook of the Learning Sciences*. Cambridge University.
- Li, Y., Schoenfeld, A.H., diSessa, A.A., Graesser, A.C., Benson, L.C., English, L.D., & Duschl, R.A. (2020b). On Computational Thinking and STEM Education, *Journal for STEM Education Research*, 3, 147-166.
- Lu, Z.Y., & Lin, S.C. (2019). A preliminary study of project-based learning teaching activity for programming based on computational thinking. (基於運算思維之PBL程式設計教學活動成效初探). Investigating the Elementary School Students' Skills of Computational Thinking and Self-Efficacy through a Robot Programming Project. In Kong, S.C. et al. (Eds.). *Proceedings of the Conference on Computational Thinking Education 2019*. Hong Kong: The Education University of Hong Kong. 50-52.
- Marx, R. W., Blumenfeld, P. C., Krajcik, J. S., Fishman, B., Soloway, E., Geier, R., & Revital T. T. (2004). Inquiry-based science in the middle grades: Assessment of learning in urban systemic reform. *Journal of Research in Science Teaching*, 41(10), 1063-1080.
- Mazur (1997). *Peer instruction: A user's manual series in educational innovation*. Prentice Hall, Upper Saddle River, NJ.
- Pintrich, P.R. & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology*, 82(1), 33-40.
- Rivet, A., & Krajcik, J. (2004). Achieving standards in urban systemic reform: An example of a sixth grade project-based science curriculum. *Journal of Research in Science Teaching*, 41(7), 669-692.
- Salomon, G., D. N. Perkins, & Globerson, T. (1991). Partners in cognition: Extending human intelligence with intelligent technologies. *Educational Researcher*, 20, 2-9.
- Solomon, G. (2003). Project-Based Learning: A primer. *TechLEARNING*, 2020/12/10 Retrieved from:

[https://free.openeclass.org/modules/document/file.php/ENG155/Projects%20online/PBL-Primer-www\\_techlearning\\_com.pdf](https://free.openeclass.org/modules/document/file.php/ENG155/Projects%20online/PBL-Primer-www_techlearning_com.pdf)

- Stix, A., & Hrbek, F. (2006). *Teachers as classroom coaches: How to motivate students across the content areas*. VA: ASCD.
- Volcz, I. (2018). The use of computational thinking to advance learning in the pre-university subject of digital literacies. In Kong, S.C. et al. (Eds.). *Proceedings of the Conference on Computational Thinking Education 2018*. Hong Kong: The Education University of Hong Kong. 68-71.
- Williams, M., & Linn, M. (2003). WISE Inquiry in fifth grade biology. *Research in Science Education*, 32 (4), 415–436.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–36.
- 陸香如 (2019)。物聯網專題式學習導入高中職程式設計課程。樹德科技大學資訊工程系碩士班碩士論文，高雄市。取自 <https://hdl.handle.net/11296/qdgznr>
- 黃柏軒 (2020)。國中跨域課程對學生運算思維與學習成效之影響。國立成功大學教育研究所碩士論文，台南市。取自 <https://hdl.handle.net/11296/34y65z>
- 楊依蓉 (2010)。結合專題導引機制之程式設計合作學習平台建置與實證。國立臺南大學數位學習科技學系碩士班碩士論文，台南市。取自 <https://hdl.handle.net/11296/uka97a>
- 葉佳忠 (2019)。使用合作專案式學習法探討學生背景在 Arduino 模組程式設計學習成效之關聯— 以北台灣某科大為例。龍華科技大學資訊管理系碩士班碩士論文，桃園縣。取自 <https://hdl.handle.net/11296/zerjg3>
- 葉昭岑 (2015)。學思達教學法運用於國中九年級地理科學生學習之研究。逢甲大學公共政策研究所，台中市。取自 <https://hdl.handle.net/11296/p5678h>
- 賴信豪 (2019)。專題導向學習應用於程式設計實習課程對技術型高中學生問題解決能力與學習態度之影響。國立臺北科技大學技術及職業教育研究所碩士論文，台北市。取自 <https://hdl.handle.net/11296/9fp33c>
- 嚴尹晨 (2019)。應用翻轉教室及問題導向學習於程式語言之研究。南臺科技大學資訊管理系碩士論文，台南市。取自 <https://hdl.handle.net/11296/ttrt3z>