airiti

Interdisciplinary STEAM Instruction: Learning Programming by Music Composition

Si-Ru Chen, Liang-Yin Ye, Sheng-Che Hsiao, Yu-Tzu Lin and Sheng-Tsang Chuo

OVERVIEW

This study developed and implemented an interdisciplinary STEAM instruction for programming that integrates music composition to enhance students' programming achievement, creativity, and learning attitude.

KEY POINTS

- Developed an interdisciplinary STEAM instruction integrating programming and music composition.
- Employed a quasi-experimental design to compare the effects of interdisciplinary STEAM and traditional instruction on programming achievement and creativity.
- The instructional stage "music appreciation and rule identification" fostered pattern recognition and algorithmic thinking.
- The interdisciplinary STEAM instruction significantly enhanced students' programming creativity.
- Artistic contexts connected programming to real-world applications.

INTRODUCTION

STEAM education integrates the arts into STEM to foster creativity alongside logical thinking. This study developed an interdisciplinary STEAM instruction for computer programming using JythonMusic, enabling students to learn programming through creative music composition. Using a quasi-experimental design, the research investigated the impact on programming achievement, creativity, and learning attitudes among 11th-grade students. The experimental group received the interdisciplinary STEAM instruction, while the control group followed traditional programming instruction. The purpose of this study was to examine the effects of this instructional approach on students' programming achievement, programming creativity, and learning attitudes.

FINDINGS

The experimental results revealed that this approach effectively supported students in developing pattern recognition, data representation, and algorithmic thinking skills, and enhanced their programming creativity and learning attitudes. Through "music appreciation and rule identification," students successfully recognized the attributes of various program structures and their functional roles within code, thereby significantly enhancing their pattern-recognition skills. The

airiti

"creative programming for music composition" stage provided an authentic context that encouraged students to develop flexible problem-solving strategies by incorporating musical elements. Although no statistically significant difference was found between the experimental and control groups in learning attitudes, qualitative interviews revealed that students who received programming instruction generally expressed positive attitudes toward programming and could relate their programming learning to future career possibilities. These results suggest that interdisciplinary STEAM instruction can meaningfully support conceptual understanding and creative engagement in programming learning.

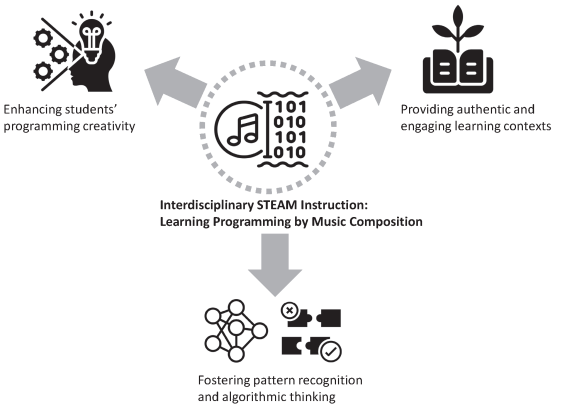


Figure 1. The key benefits of the proposed interdisciplinary STEAM programming instruction

TAKEAWAYS

This study demonstrates that integrating music composition into programming instruction can effectively promote creativity and algorithmic thinking. By using an interdisciplinary STEAM instruction, educators can provide students with a tangible medium—music—to ground abstract programming constructs, making the learning process more intuitive and engaging. Future research should expand this interdisciplinary model to other artistic domains, such as visual arts and digital media, to further validate the role of STEAM-based instruction in developing holistic problem-solving abilities. This study serves as a foundational reference for designing curricula that balance logical rigor with creative flexibility, ultimately preparing students for the interdisciplinary demands of the modern workforce.

airiti

科學教育學刊
2025, 第三十三卷第三期, 195-225
DOI:10.6173/CJSE.202509_33(3).0001

Contemporary Journal of Science Education
2025, 33(3), 195-225

STEAM科際整合程式設計教學——以音樂創作學習程式設計

陳思如　葉亮吟　蕭聖哲　林育慈*　卓聖倉
國立臺灣師範大學　資訊教育研究所

摘要

科學、技術、工程和數學(science, technology, engineering, and mathematics, STEM)領域教育強調科際整合，透過動手實作培養真實世界之問題解決能力，科學、技術、工程、藝術和數學(science, technology, engineering, arts, and mathematics, STEAM)教育則是於STEM中加入藝術，藉以在STEM的邏輯與系統思考中加入創意元素。本研究發展STEAM科際整合程式設計教學模式，使用JythonMusic作為程式設計工具，讓學生透過音樂創作學習程式設計。研究探討此教學對程式設計能力、程式創造力與程式設計學習態度的影響。研究採準實驗研究法，實驗組學生施以STEAM科際整合程式設計教學，控制組學生則施以傳統程式設計教學。研究結果顯示STEAM科際整合程式設計教學模式能幫助學生將具體的樂曲聲音對照抽象的程式概念與流程，藉以瞭解程式概念與邏輯(演算法思維)，並在利用樂曲規則進行創作的過程中，辨識各種程式結構的屬性及其於程式中扮演的角色(模式辨識)。藝術及生活化的創作情境亦能幫助學生發展問題解決策略，學生在創作中融入音樂元素，以更靈活的方式解決問題，透過程式設計體現藝術，進而提升創造力。儘管實驗組與控制組學生在學習態度上無顯著差異，然而，從訪談結果可發現，STEAM科際整合教學可幫助學生建立較為正向的程式設計學習態度，並促進其連結程式設計與未來職涯。本研究期能提供未來欲從事STEAM科際整合教學與程式設計教學之教師與研究者作為課程設計之參考。

關鍵詞：STEAM科際整合教學、程式設計教學、創造力

壹、緒論

程式設計是一項複雜的認知活動，學生不僅需要具備程式設計知識，亦須具備邏輯思維、抽象推理能力，以及問題解決能力，才能撰寫程式以解決問題(Du Boulay, 1989; Jenkins, 2002; Lin et al., 2022; Pea & Kurland, 1984)。然而，傳統程式設計教學著重於程式語法知識及語意說明，因此導致學生無法學習高階的邏輯思維與程式解題策略(Perkins & Martin, 1986; Robins et al., 2003; Soloway & Ehrlich, 1984)。再者，傳統程式設計教材缺乏與真實生活情境的連結，易造成學生學習動機低落，且難以將習得的概念應用到真實的情境以解決問

*通訊作者：林育慈，linyt@ntnu.edu.tw；ORCID：0000-0002-2138-538X
投稿：2025/2/15，修訂：2025/6/12，接受：2025/7/20，線上出版：2026/1/26

airiti

題(Gomes & Mendes, 2007; Guzdial & Soloway, 2002)。研究指出若學生能在真實的問題情境脈絡中進行學習，將可提升學習態度與學習成效，並能增進其問題解決能力(Beier et al., 2019; Glatthorn, 1994)。

然而，真實問題情境經常相當複雜且涉及不同領域，因此許多研究提出科學、技術、工程和數學(science, technology, engineering, and mathematics, STEM)領域科際整合教學，讓學生體驗複雜而涉及多種知識的真實問題情境。因此許多研究於數學、物理等課中實施STEM教學，學生須整合不同領域的知識，以解決真實且複雜的問題(Daugherty et al., 2021)，藉此可激發學生的學習動機，並在問題解決中更有機會活用知識，並提升推理、抽象思考等高階思維(Dominguez et al., 2023; K. H. Kim & Coxon, 2016; Tiryaki & Adıgüzel, 2021)。為了在STEM課程中提升創新思考的機會，有學者提出應於課程中加入藝術元素(Connor et al., 2015; Guyotte et al., 2014)，成為科學、科技、工程、藝術和數學(science, technology, engineering, arts, and mathematics, STEAM)科際整合教學，以幫助學生在解決複雜的問題時，除了以聚斂性思考(convergent thinking)進行問題拆解(problem decomposition)與分析，亦能以擴散性思考(divergent thinking)將解決方案應用到各種真實世界的問題(Land, 2013)。

愈來愈多學者投入STEAM科際整合教學相關研究，探討如何以藝術為STEM領域的學生創造更多的學習機會與效益，並提高學習動機，使其更願意投入STEM領域之學習(Guyotte et al., 2015)，並在STEM領域中注入創造力的元素，以幫助學生未來在職場中能運用創造性思考更有效發展解題策略(Maeda, 2012; Perignat & Katz-Buonincontro, 2019)。在電腦科學或程式設計方面，基於其能應用於各領域的屬性，相當適合採用STEAM科際整合教學(Ajaykumar et al., 2022; Castledine & Chalmers, 2011; Chung et al., 2014; Dagdilelis et al., 2005)。近年開始有研究將藝術融入程式設計教學：有研究讓學生透過機器人程式設計，搭配故事情境畫面的繪製，以敘說故事情節(Ozobot, 2020)；也有研究將程式設計結合科學學習，並進行藝術設計以互動方式呈現科學概念，此可增進學習參與度並提升創造力(Dúo-Terrón, 2023)。除了視覺藝術，亦有研究讓學生透過音樂作曲學習程式設計(Freeman et al., 2019; Horn et al., 2020; S.-W. Kim & Lee, 2016; Manaris et al., 2016; Wang et al., 2015)。事實上，音樂與運算有許多共通特性(如形式符號表徵與邏輯結構等) (Bell & Bell, 2018)，音樂創作原本就需要作曲者的運算思維(Edwards, 2011)。過往的相關研究已發現音樂程式設計教學在學生學習態度與學習成就上的優勢：一、真實的音樂創作問題情境讓學生更加認同程式設計課程的價值，因而能提升其課堂參與度及學習動機(Freeman et al., 2019; Jones et al., 2024; Liu et al., 2025; Zhang et al., 2024)，並對於運算領域可擁有更高的自信心與持續學習的意願(Freeman et al., 2019; Jones et al., 2024; Liu et al., 2025; Magerko et al., 2016; Wanzer et al., 2020; Zhang et al., 2024)；二、透過對照具象的音樂與抽象的運算概念，學生更能瞭解程式的表徵形式及運算邏輯，因此可提升其程式設計學習成效(Bell & Bell, 2018; Jones et al., 2024)。

除了學習態度與學習成就，許多研究認為藉由音樂創作學習程式設計能提升學生的創造力(Freeman et al., 2019; Siva et al., 2018)。開放的問題情境與自由的創作環境可以激發學生產出更有創意的解題策略，亦可令學生在更廣的思考空間中依照音樂與程式

airiti

設計的邏輯進行創作(Freeman et al., 2019)。創造力是一種可以產出新穎、有價值的想法的能力(Boden, 2009)，包含流暢力(fluency)、獨創力(originality)、變通力(flexibility)、精進力(elaboration)以及敏覺力(sensitivity)等重要特徵(Guilford, 1950, 1966)。透過創造力的擴散性思考，可針對問題提出多樣化的想法以及替代性解決方案(Guilford, 1966; Runco & Acar, 2012)。發明、設計、作曲以及設計規劃等皆屬於創意行為(Guilford, 1950, 1966)，而電腦科學的成就表現亦與創造力有關(Giza, 2021; Saunders & Thagard, 2005)。研究指出學習程式設計能提升創造力(Clements, 1986, 1995)，而適切的程式設計教學模式亦能提升學生創造力(Ferreira, 2013)。創造力教學的重要原則即為真實情境，但必須設計適切的教材方可達到培養創造力的目的(Kind & Kind, 2007)。

儘管許多音樂程式設計教學的相關研究會提及對創造力的正面影響，僅少數研究實際分析創造力的變化(Magerko et al., 2016; Siva et al., 2018)，且為自評問卷所得，較無法客觀評量創造力的發展。相關研究亦鮮少探討如何設計適切的教學引導以提升創造力，及此等教學如何幫助創造力的發展。再者，現有的音樂程式設計教學多著重於探討學生學習態度的提升(Freeman et al., 2019; Jones et al., 2024; Liu et al., 2025; Zhang et al., 2024)，少數討論學習成就的研究僅檢驗學生學習前後的成就差異(Freeman et al., 2019; Magerko et al., 2016; Siva et al., 2018)，對於融入音樂的STEAM科際整合程式設計教學是否比起傳統教學，更能幫助學生學習程式設計，以及如何幫助學生理解程式設計概念等議題，較缺乏確切的研究證據。研究中所設計的教學活動與引導如何幫助學生理解程式設計概念，亦須更進一步的研究探討(Petrie, 2022)。專此，本研究希冀發展結合音樂之STEAM科際整合程式設計教學策略，以音樂創作與運算思維的引導培養學生的程式設計能力、創造力與正向的學習態度，並探討其成效。

貳、研究目的

本研究探討結合音樂之STEAM科際整合教學對程式設計能力與創造力的影響。具體而言，本研究的主要研究問題為下列三項：

一、結合音樂之STEAM科際整合教學是否能提升學生的學習成就？

二、結合音樂之STEAM科際整合教學是否能提升學生的創造力？

三、結合音樂之STEAM科際整合教學是否能改善學生的學習態度？

參、文獻探討

一、程式設計教學

許多研究指出初學者在學習程式設計時，經常遭遇到許多困難，例如：不熟悉程式語言的語法或無法瞭解程式語意與邏輯(Gomes & Mendes, 2007; Mannila et al., 2006; Qian & Lehman, 2018)。因此傳統教學多著重語法的介紹與語意的說明，較缺乏引導學生發展解題策略或除錯技巧(McCauley et al., 2008; Pea & Kurland, 1984; Perkins & Martin, 1986)。如此著重程式語法而缺乏概念應用的教學，將造成程式設計知識的破碎化，導致誤用相關語法(Perkins & Martin, 1986)，並且導致面對新問題時無法發展解題策略(Robins et al., 2003; Soloway & Ehrlich, 1984)。因此，許多學者提出：學習程式設計就是要學習專家的解題策略，需要引導學生將學過的程式

airiti

片段組織成為有效的解題程式碼(Spohrer & Soloway, 1986)。

教學策略以及教材內容也是影響程式設計教學成效的因素之一(Cheah, 2020)。傳統程式設計教學活動常與真實情境脫軌，容易造成學生無法將程式設計實際應用於日常生活(Baist & Pamungkas, 2017; Lahtinen et al., 2005)。給予適當的情境問題，方能讓學生可以應用學會的語法。有研究者建議可採用專題式的教學策略來教導學生撰寫程式(Salomon & Perkins, 1987)，動手實作亦是程式設計教學的重要方法，利用製作軟硬體產品的方式，學生能運用理論於實際問題解決，進而獲得跨領域的知識，雖然這樣的課程設計較具挑戰性，卻更能提高學生學習興趣(Sprague, 2013)。

事實上，程式設計兼具了抽象化、理論化以及實務性的概念，亦同時具有數學以及工程等跨領域的特性，因此很適合導入STEM科際整合教學模式。透過STEM科際整合教學的方式，亦能讓學生利用程式設計工具(例如：Python的數學運算、VPython圖形製作等)探索科學或工程議題，使學生更能體驗程式設計在真實問題中的應用方式(Hambrusch et al., 2009)。此外，創造力與創新是電腦科學中很重要的元素之一，因此進行程式設計教學時，應注意是否具備促進創造力的活動，例如：合作、自由探索、做中學以及視覺回饋等(Sharmin, 2021)。

程式設計的自我效能對程式設計學習來說亦相當重要，不僅影響程式設計學習表現，亦能幫助程式設計知識的遷移(Ramalingam et al., 2004)。然而許多學生在初學程式設計的過程中會有負向的自我評價，導致有較低的自我效能(Gorson & O'Rourke, 2020)。欲提升自我效能，除了須幫助學生理解概念，亦須提高其學習動機。真實的問題情境能讓學生產生想要解決困難的內在動機，視覺化的程式設計環境亦能幫助學生減少挫折感並讓低自我效能的學生更願意主動學習(Tsai, 2019)。STEM科際整合的教學方式，亦可促進學生將程式設計與STEM領域職涯作連結，並對STEM領域之職業產生興趣(Ogegbo & Aina, 2024)。

二、STEAM科際整合教學

STEM是指科學、技術、工程與數學四個領域，而進行STEM教育時，應整合不同領域間的知識，並提供學生真實問題情境，以提升學習動機與發展問題解決的技能(Daugherty et al., 2021)。但真實問題情境往往相當複雜，進行教學設計時須留意如何將學習內容與問題情境適切連結。然而，如何設計跨科際之STEM教學主題是值得探討的議題(Cahill & Petersen, 2022)。

傳統STEM的教學多以工程研究或科學探索為主，例如：討論有關於製作生質材油的研究及教育，使學生瞭解生物材油燃料的永續發展以及可行方式，以整合生物及化學課程(Burrows et al., 2014)，或透過大數據分析天文資料或氣象資料等以進行天文或氣象研究(Hambrusch et al., 2009)。但上述局限於STEM領域的教學，無法培養創意思維以有效解決問題(Guyotte et al., 2014)，因為在STEM科際整合教學中，學生需進行創意性思考以整合跨領域知識，方能有效解決複雜的實體情境問題(Katanski, 2013)。然而，一般的STEM科際整合教學中並未強調創造性思考力(Maeda, 2012)。

為使STEM科際整合教學更有效，有學者提出將藝術與STEM整合，以成為STEAM科際整合教學。透過藝術的包裝，讓學生有

airiti

機會運用創意性思考，並更願意投入STEM領域的學習。STEAM教學可視為在STEM教學中加入博雅與人文教育(Ge et al., 2015)，以培養學生運用創意、批判性思考、合作、有效溝通等高階能力解決真實世界問題(Quigley & Herro, 2016)。此外，藝術與STEM的屬性相當不同，藝術是主觀、直覺、獨特與感覺的，而STEM則是客觀、可重複、邏輯且為分析性的，兩者皆為教育上不可缺少的成分，可互相彌補不足(Sousa & Pilecki, 2013)。STEM科際整合教學不僅要促進學生問題解決能力，還要培養尋找問題的技巧，並讓學生保持學習興趣，最好的方式是在STEM中加入藝術領域課程內容(Land, 2013)，並且導入自主學習策略與探索學習法(Connor et al., 2015)。愈來愈多學者投入STEAM科際整合教學相關研究，探討如何以藝術為STEM領域的學生創造更多的學習機會與效益，並提高學生的學習動機，讓其更願意投入STEM領域之學習(Guyotte et al., 2015)。透過藝術作品的創作，例如：繪製圖畫、音樂作曲等，可讓學生在STEM領域的學習中得以發揮創造力(Connor et al., 2015; Kanematsu & Barry, 2016)。亦有研究使用XronoBeat結合數學以及音樂學習，發現學生認為與XronoBeat進行互動，有助於視覺化與理解關於旋律與節拍的數學理論，並能發現數學與音樂的相關性(Milne & Calilhanna, 2019)。

三、音樂整合電腦科學教學

電腦科學教育領域整合音樂領域的STEAM科際整合教學的研究也逐漸增加。由於音樂與運算有許多共通特性，例如：皆須使用形式符號表徵、皆包含序列、重複等概念，透過音樂的真實經驗，能幫助學生對照並理解運算中的抽象概念，進而學習運算思維(Bell & Bell, 2018)。有學者甚至認為音樂創作原本就需要作曲者的運算思維，樂曲的結構與形式表徵皆與運算思維相關(Edwards, 2011)，演算法作曲(algorithmic compostion)便是基於這樣的概念發展而來，作曲家可以藉著電腦與演算法表達音樂邏輯(Simoni & Dannenberg, 2013; Taube, 2004)。因著音樂與電腦科學的相關性，許多研究讓學生透過程式作曲軟體進行音樂創作：利用TunePad、JythonMuisc或EarSketch等音樂創作軟體可以Python或Javascript撰寫程式進行音樂創作，以此設計STEAM課程可提供學生自由表達創作的機會，能有效鼓勵學生發揮創意(Freeman et al., 2019; Horn et al., 2020; S.-W. Kim & Lee, 2016; Manaris et al., 2016; Wang et al., 2015)。Monterey Mirror則於音樂創作之外，亦可製作視覺化的效果，以進行互動音樂創作與表演(Manaris et al., 2016)。除了基於Python程式語言的音樂製作平臺，Sonic Pi為使用Ruby語言的實況程式設計(live coding)平臺，可進行電子音樂製作與演奏，更加強調互動性並可以即興演奏。有研究比較不同的音樂程式設計工具(EarSketch與TunePad)是否對學生學習運算思維有不同的影響，發現對國中生來說，雖然TunePad比起EarSketch更容易使用，但EarSketch提供較高的語法自由度，更能幫助學生學習模組化與抽象概念，並能創作更多種類的音樂風格(Petrie, 2022)。

許多研究利用上述音樂創作工具進行程式設計教學，發現結合音樂與程式設計的科際整合教學具備二項主要的優點：

(一)真實的應用情境可提升學生的學習態度：音樂創作的真實情境讓學生將程式設計與真實世界連結，可感受到程式設計的實用性並體驗有意義的學習，因而能認同程式設計的價值，進而提升課堂參與度與學習動機(Freeman et al., 2019;

airiti

Petrie, 2022; Zhang et al., 2024)，而因挫折感與焦慮感降低，學生更願意不斷嘗試而不放棄，因此也提升了對運算的自信心(Freeman et al., 2019; Jones et al., 2024; Liu et al., 2025; Zhang et al., 2024)。此外，學生在音樂程式設計的過程中能享受製作音樂的愉悅與成就感，因而亦表達未來願意持續學習程式設計或電腦科學(Magerko et al., 2016; Wanzer et al., 2020)；

(二)具象的音樂與抽象程式設計或運算概念的對照連結，幫助學生理解概念：學生將音樂創作想法以程式碼呈現，須將音樂的表徵形式轉換為程式的表徵形式，並須顧及作品結構及美感的呈現，因此更能掌握程式結構與資料的表示，使程式編寫成為一種音樂分析的過程(Roberts & Horn, 2025)。而在創作過程中聆聽自己編寫的程式碼如何影響到作品的成果，也幫助學生更具體地理解程式碼的意義，並更能掌握運算概念(Bell & Bell, 2018; Jones et al., 2024)。

儘管有研究提及自由的音樂創作與真實的問題情境可增進學生創造力(Freeman et al., 2019; Siva et al., 2018)，然而多數僅提到作曲活動的創造性意涵(Freeman et al., 2019; Jacob, 1996)，或僅透過問卷方式檢驗學生的創造力(Magerko et al., 2016; Siva et al., 2018)，並未真正檢驗學生的通用創造力或領域特定創造力。再者，多數研究著重學習態度的探討，且少數針對程式設計或電腦科學學習成就加以討論者，卻僅作前、後測比較(Freeman et al., 2019; Magerko et al., 2016; Siva et al., 2018)，較無法準確反映此種教學相較於傳統教學之效果。

四、創造力的教學與評量

創造力是一種可以產出新穎、有價值的想法的能力(Boden, 2009)，創造力的重要特徵包含流暢力、變通力、獨創力、敏覺力與精進力(Guilford, 1950, 1966)，而依據接觸到的媒材不同會有不同的表現。創意思考(creative thinking)包含了擴散性思考能力以及轉化(transformation)的能力(Guilford, 1967)。創意思考可以簡易地分為三個階段：獲取並組織資訊(gathering and structuring of information elements)、構思(ideation)以及檢核(verification)，它和Polya (2014)提出的問題解決的歷程相仿。在高度科技化及資訊化的社會，具備創新及創造力的人才更顯重要，因此培養學生創造力成為教育上的重大議題(K. H. Kim & Coxon, 2016; Land, 2013)。創造力教學的重要原則之一是提供真實問題情境，在教材設計上亦須考量到學生的先備知識以及能力(Kind & Kind, 2007)，尤其是提供學生實作的機會以培養創造力(Jeffrey & Craft, 2004)。電腦科學中的創造力涉及抽象與實務兩種面向，因此在教學中需提供理論與實務結合之機會，並引發內在動機以驅動探索與學習(Giza, 2021)。若能在STEM教學中加入藝術成為STEAM科際整合課程，可提供學生想像、探索、實驗、操弄等活動，以讓學生體驗科學家創建新知識所進行的事項，也就是創造的歷程，但若缺乏適切的教學設計，對創造力的提升效果並不顯著(Conradty & Bogner, 2019)。STEAM科際整合課程若設計得當，可提供學生開放的學習環境，同時，課程中的問題情境也會刺激他們的想像力，另外，加入藝術也對於創造力的發展有正向的影響，可以增進學生創造力(Ozkan & Topsakal, 2021)。

airiti

亦有研究將藝術創作程序加入STEAM課程，以使學生融入藝術創作時的再設計與再修正過程，此可促進學生的創造力發展(Leavy et al., 2023)。STEAM課程融入創造力相關的學習活動，能提供學生一個審視問題解決歷程的新觀點，也讓學生有正向的學習態度來面對困難(Žižic et al., 2017)。音樂創作亦為一種藝術創作，Engelman等(2017)指出使用EarSkectch此音樂程式設計工具的STEAM科際整合課程，可促進學生的學習動機並提升創造力，且可讓學生維持對電腦科學的興趣(Freeman et al., 2019; McKlin et al., 2018)。此外，在STEAM中結合Sonic Pi或是TunePad進行程式設計與音樂創作的教學，均能促進學生對於程式設計的學習動機(Horn et al., 2020; Petrie, 2022)。不過，STEAM科際整合課程對於學生的程式設計相關的創造力有何影響、影響為何，均需進一步的探討。

創造力的評量亦為創造力教學中的重要環節。Torrance的創造思考測驗(Torrance Tests of Creative Thinking, TTCT)是根據Guilford (1950)提出的創造力相關理論所發展，採心理計量(psychometric)的模式，將創造力分為流暢性、變通性、獨創性與精進性，屬於較通用的測驗。Sternberg與Lubart (1998)則認為不同領域有其特定的創造力，須以領域知識為基礎。Hu與Adey (2002)便根據科學的屬性建立科學創造力的結構模型(scientific structure creativity model, SSCM)，模型中包含產品、特性與過程三個向度，因此後續許多研究依此設計科學創造力評量(Majumdar, 1975)。除了科學創造力，葉玉珠(2004)亦納入科技領域知識，開發了科技創造力測驗，強調科技產品的設計創意，並以流暢力、變通力、獨創力、精進力與視覺造型進行評量。亦有研究透過真實情境的問題解決探究學生於科技創造力方面的認知歷程，發現內在動機能影響創造力的表現(Yeh & Wu, 2006)。在數位科技方面，有研究透過問卷評量學生的數位創作成就(製作行動App等)、學校日常數位創作(製作班級多媒體專題等)與自我表達創作(製作個人部落格等) (Hoffmann et al., 2016)，然而，此問卷著重的是一般性的數位素養。儘管創造力在電腦科學中相當重要(Giza, 2021)，目前仍較缺乏其教學與評量的相關研究。

肆、研究方法

為了驗證STEAM科際整合程式設計教學對學生的程式設計能力、程式創造力及學習態度的影響，本研究採準實驗研究法，比較所提出的STEAM科際整合程式設計教學與傳統程式設計教學的差異。茲將研究對象、研究流程、STEAM科際整合教學設計與研究資料蒐集與分析分述如下：

一、研究對象

本研究於新北市的高中進行教學實驗，實驗對象為高中二年級修習資訊科技課程的學生(平均年齡17歲)，多數未具備程式設計基礎能力。此屆學生入學時學測分數在北區79所高中排名約為中段，並依照入學成績以S型排列方式分班。參與教學實驗的班級有兩班，一班為控制組(男生14人，女生20人)，另一班為實驗組(男生14人，女生20人)，每班人數均為34人，總計68位學生。

二、研究流程

實驗組與控制組為同一教師教學，實驗組採STEAM科際整合程式設計教學，控制組採傳統講述式程式設計教學，以投影片講授

airiti

概念，教師並親自示範程式撰寫，再由學生進行練習。實驗為期13週，每週進行兩節課(一節課為50分鐘)。實驗組與控制組均具有相同的學習內容及學習時數。前、後測皆包含程式設計能力、程式設計學習態度。後測施以程式創造力測驗。研究流程及內容如表1所示。

如表1所示，實驗組與控制組會在第1週進行前測，前測包含程式設計能力、程式設計學習態度問卷。第2、第3週使用code.org進行程式闖關遊戲，目的是讓學生體驗程式設計的撰寫與執行。第4 ~ 8週實驗組採STEAM科際整合程式設計教學，控制組則採傳統程式設計教學(教師講授搭配學生上機練習)。第9週實驗組與控制組皆進行程式設計的形成性評量，目的是瞭解程式設計學習狀況。第10 ~ 12週進行程式專題製作。第13週進行後測，後測除程式創造力測驗以外，其餘測驗皆與前測項目相同。在程式設計工具方面，兩組都使用Python程式語言，實驗組使用JythonMusic進行音樂程式創作，控制組則使用IDLE (Integrated Development and Learning Environment)開發環境進行程式練習。

三、研究工具

(一)程式設計能力測驗

程式設計能力測驗分為前測與後測，前測試題是以Bebras (n.d.)國際運算思維歷年考古題為基礎，從中挑選與STEAM科際整合教學之程式設計能力相關的三個主題：模式辨識、資料表示與演算法，並結合專家意見設計而成。前測題目共有十題選擇題，每題10分，總分為100分，前測試題可參閱Bebras國際運算思維官網之範例。後測試題是參考高中程式解題系統的題目，並根據專家意見修訂設計而成，因此具專家效度。測驗皆為上機測驗，共三題，總分100分，圖1為後測範例試題。

表1：實驗組與控制組的研究流程及內容

週次	控制組	實驗組
1	前測：程式設計能力(50分鐘)、程式設計學習態度(20分鐘)	
2 ~ 3	透過code.org遊戲瞭解模式辨識、函式、迴圈與條件判斷	
4	1.電腦程式的應用 2.程式語言的簡介與特性 3.資料型別(數值／字串)	1.譜號 2.音階
5	1.資料型別(串列) 2.算術運算子、布林值 3.變數	1.調號 2.音高
6	1.函式 2.資料／選擇排序、氣泡演算法	1.樂段 2.樂句
7	1.關係／邏輯運算子 2.if條件	1.A-B-A音樂架構 2.A-A-B-A音樂架構
8	1.for/while迴圈 2.巢狀結構	現代流行音樂編曲歷程
9	程式設計能力形成性評量(50分鐘)	
10 ~ 12	專題製作(學生從108課綱19項議題自由挑選，資訊與科技除外)	
13	後測：程式設計能力(50分鐘)、程式設計學習態度(20分鐘)、程式創造力測驗(30分鐘)	

airiti

(二)程式創造力測驗

本研究探討STEAM所發展之教學是否能提升學生靈活應用程式設計概念，以創作自己程式作品的能力，因此研究者參考Guilford (1950, 1966)提出的創造力特徵與Torrance (1974)發展的TTCT (語文版)，定義本研究程式創造力的內涵及檢測向度如表2所示，可分為流暢力、變通力、獨創力、敏覺力與精進力五個向度。測驗以Bebras (n.d.)國際運算思維歷年考古題為基礎，並經兩位電腦科學教育專家檢視以確定專家效度，根據檢測向度設計程式創造力測驗試題，題目範例如表3。程式創造力測驗共有10題，每題十分，總分為100分。

各向度評分方式，茲分述如下：

1.流暢力：學生若能在規定的時間內或是在

1.阿信是一位高中電腦老師，這學期正在教學生寫 Python 程式。他的評分標準是依照每一位學生在 google classroom 上解出的題數，計算出對應的得分。為了不讓分數落差太大，因此他並不是採取每一題固定得分的方式，而是隨著題數增加而調整每題的得分。規則如下：

(1) 答對題數在 0~10 者，每題給 6 分。

(2) 題數在 11~20 者，從第 11 題開始，每題給 2 分。(前 10 題還是每題給 6 分)

(3) 題數在 21~40 者，從第 21 題開始，每題給 1 分。(前 10 題還是每題給 6 分；11-20 題每題給 2 分)

(4) 題數在 40 以上者，一律 100 分。

如此一來，只要輸入答對的題數，程式就會自動計算，並輸出這名同學的得分，以下為輸入/輸出的範例，請同學們依照範例的格式撰寫程式。

範例一：答對 10 題，總計 60 分

輸入：10

輸出：60

範例二：答對 40 題，總計 100 分

輸入：40

輸出：100

不過問題來了，雖然學生們因為這樣的計分公式而大大地提升了及格率，但因為阿信有 600 多位學生，一個一個去計算真的是一件很吃重的工作，所以現在想請你幫他寫個程式解決這個問題。

圖1：程式設計能力測驗後測範例試題

表2：程式創造力定義

向度	程式創造力定義
流暢力	在有限的時間或有條件的情況下，針對問題情境，提出多種可能解決答案的能力
變通力	可以將相同的程式指令或結構套用在各種問題情境
獨創力	使用不同於課堂所學的程式指令或結構，並將其應用在問題情境當中
敏覺力	能夠找出與說明問題情境所需解決的問題，並提出解決方法
精進力	修改課堂所學的程式指令或結構，並將其應用在程式情境當中

表3：程式創造力五向度題目範例

向度	程式創造力題目範例
流暢力	小海狸想要利用溫控系統管理溫度，使得溫室的溫度不會低於16°C或高於26°C，小海狸所設計溫控系統，可能會用到哪些程式指令或結構呢？用在哪裡？
變通力	小海狸希望可以在設計好的溫控系統中，擴充新的系統。如果你是小海狸，你會想要擴充哪些新的系統？以及你會運用哪些程式指令或結構來擴充這些新系統？(至少舉出兩個系統)
獨創力	擴充的新系統，還需要運用哪些額外的程式指令或結構？用在哪裡？
敏覺力	自從引進小海狸設計的電腦化程式管理溫室後，小海狸種植的花朵品質愈來愈好。然而，最近小海狸發現，早上溫室的溫度較晚上溫室的溫度要來的高，小海狸聽了這個狀況後，便開始檢查程式。請問程式可能是哪些方面的問題，假如你是小海狸，你會如何解決？可能是程式的那部分出錯？
精進力	小海狸覺得自從利用電腦程式管理溫室後，帶來諸多的便利，因此，他也想要在居家中設置一些便利的系統。假如你是小海狸，你會想要製造哪些系統？以及你會運用哪些程式指令或結構製造系統？運用在哪裡？

airiti

有條件的情況下，針對程式情境與問題寫出多種可能解決的答案，且所寫之答案正確或合理，則將根據標準予以給分，最後加總即為學生該向度所得之總分，如圖2所示。

2.變通力：此向度在測驗學生是否具備將相同的程式指令或結構套用在各種問題情境的能力，評分方式根據學生作答符合正確性或合理性的個數，依據所設立的標準予以評分，最後加總即為學生該向度所得之總分，如圖3所示。

3.獨創力：此向度旨在測驗學生是否具備使用不同於課堂所學的程式指令或結構，並將其應用在程式情境當中的能力，評分方式根據學生作答符合合理性的個數，依據評分標準予以給分，最後加總即為學生該向度所得總分，如圖4所示。

4.敏覺力：此向度在評測學生對於問題的洞察力，看其是否具備能夠察覺與說明程式問題之所在，並提出相對應的程式設計解決方法。評分方式乃根據學生作答符合合理性的個數，依據評分標準予以給分，最後加總即為學生該向度所得之總分，如圖5所示。

5.精進力：此向度在評測學生是否具備修改程式指令或結構(皆為課堂所學過的片段)，並將其應用在程式情境當中的能力。評分標準為能夠修改課堂已教授的程式指令或



圖2：程式創造力測驗作答情形(流暢力)

註：本圖顯示學生於流暢力題項中列舉多種透過程式碼控制溫度的方案。



圖3：程式創造力測驗作答情形(變通力)

註：本圖顯示學生於變通力題項中回答濕度與亮度的情境與情境可用的指令。

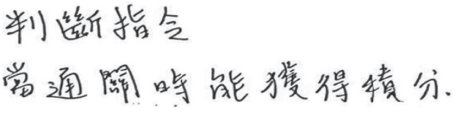


圖4：程式創造力測驗作答情形(獨創力)

註：本圖顯示學生於獨創力題項中回答判斷積分獲得方式。

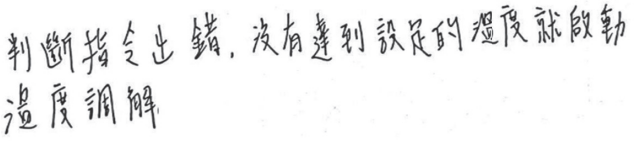


圖5：程式創造力測驗作答情形(敏覺力)

註：本圖顯示學生於敏覺力題項中回答觀察到的溫度設定問題與解決方式。

airiti

結構，以應用在符合其程式情境中的個數，如圖6所示。

(三)學習態度問卷

學習態度問卷是參考Wiebe等(2003)所開發的電腦科學態度量表設計而成，包含程式設計自我效能、程式設計學習動機與程式設計在STEAM領域之應用三個向度，並以李克特(Likert)五點量表進行作答，包含非常同意、同意、普通、不同意、非常不同意。問卷經兩位電腦科學教育專家檢閱修訂以確認其專家效度，且信度分析其Cronbach's α係數為.731，題目如表4。

(四)學習單

學習單是根據研究所設計的STEAM科際整合程式設計教學程序逐步引導學生在音樂創作中進行運算思維(詳見「四、資料蒐集與分析」)，示例如圖7，學生每節課皆須填寫學習單以進行相對應的學習步驟。

四、資料蒐集與分析

本研究探討STEAM科際整合教學對學生程式設計能力與態度的影響，除此之外也希望瞭解透過STEAM科際整合教學，學生透過音樂的創作是否能增進其彈性應用所學的程式概念以解決問題的能力，因此也加入學生程式創造力的評鑑。蒐集之資料包含量化資料與質性資料，量化資料以SPSS進行數據統計分析，質性資料則以課堂學習單、訪談紀錄進行分析。茲將本研究蒐集之資料及分析方法分述於下：

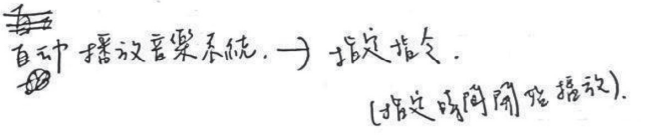


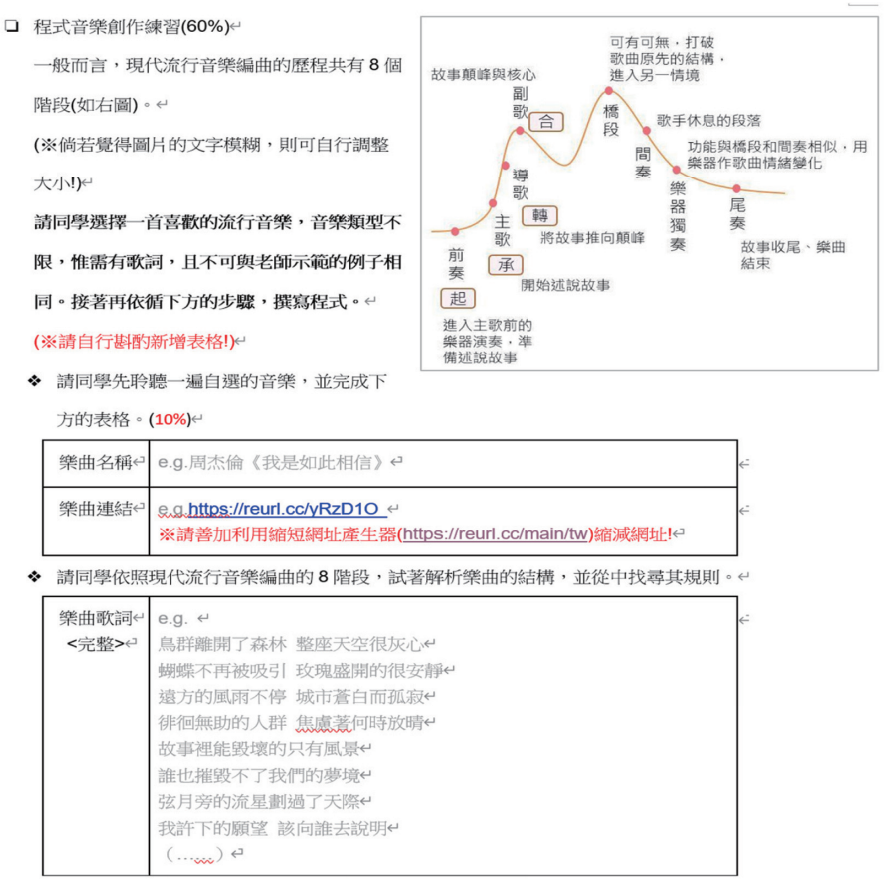
圖6：程式創造力測驗作答情形(精進力)

註：本圖顯示學生於敏覺精進力題項中回答可做的改變與適用的指令。

表4：學習態度問卷三向度題目

向度	學習態度問卷題目
程式設計自我效能	1.學習程式語言對我來說是件困難的事情。 2.從無到有寫出一個程式常讓我感到焦慮。 3.我無法獨立寫出我想要寫的程式，通常需要借助老師或同學的協助。 4.我喜歡花時間解決程式問題，這會讓我有成就感。
程式設計學習動機	1.我希望能多學習程式設計相關的知識與技能。 2.以我的經驗來說，程式設計課程會讓我感到開心。 3.程式設計是枯燥的理論與複雜的指令組合而成的技能。 4.程式設計對我未來的職業、工作是重要的。
程式設計在STEAM領域之應用	1.程式設計是按指令逐步操作，並不需要創造性的思考。 2.比起數理學科，程式設計與藝術的關係較遠。 3.程式設計可幫助人們解決其他領域的問題(數學、生物、人文社會)。 4.程式設計是一種創作的歷程。 5.電腦與藝術的結合對未來科技的發展是重要的。

airiti



❑ 程式音樂創作練習(60%)

一般而言，現代流行音樂編曲的歷程共有 8 個階段(如右圖)。

(※倘若覺得圖片的文字模糊，則可自行調整大小!)

請同學選擇一首喜歡的流行音樂，音樂類型不限，惟需有歌詞，且不可與老師示範的例子相同。接著再依循下方的步驟，撰寫程式。

(※請自行斟酌新增表格!)

❖ 請同學先聆聽一遍自選的音樂，並完成下方的表格。(10%)

樂曲名稱	e.g.周杰倫《我是如此相信》
樂曲連結	e.g.https://reurl.cc/yRzD1O ※請善加利用縮短網址產生器(https://reurl.cc/main/tw)縮減網址!

❖ 請同學依照現代流行音樂編曲的 8 階段，試著解析樂曲的結構，並從中找尋其規則。

樂曲歌詞 <完整>	e.g. 鳥群離開了森林 整座天空很灰心 蝴蝶不再被吸引 玫瑰盛開的很安靜 遠方的風雨不停 城市蒼白而孤寂 徘徊無助的人群 焦慮蒸何時放晴 故事裡能毀壞的只有風景 誰也摧毀不了我們的夢境 弦月旁的流星劃過了天際 我許下的願望 該向誰去說明 (……)

圖7：學習單示例

(一)量化資料分析

1.程式設計能力：為驗證STEAM科際整合程式設計教學與傳統程式設計教學對學生程式設計能力學習成效的影響(研究問題一)，施以程式設計能力測驗的前測與後測(施測時間皆為50分鐘)，並以Quade Test比較兩組在歷經不同的教學策略後，程式設計能力是否有顯著差異。

2.程式創造力：為驗證STEAM科際整合程式設計教學與傳統程式設計教學對學生程式創造力表現的影響(研究問題二)，施以程式創造力之後測，測驗(施測時間為30分鐘)，並以曼惠特尼*U* (Mann-Whitney *U*)檢定比較兩組在歷經不同的教學策略後，程式創造力的表現是否有顯著差異。

3.學習態度：為驗證STEAM科際整合程式設計教學與傳統程式設計教學對學生程式設計學習態度的影響(研究問題三)，施以程式設計學習態度問卷前後測(施測時間皆為20分鐘)，並以Quade Test比較兩組在歷經不同的教學策略後，程式設計學習態度的表現是否有差異，以前測為共變項排除程式設計起始學習態度的影響。

(二)質性資料分析

為蒐集質性資料以輔助解釋量化資料，

airiti

研究亦透過學生填寫的學習單與半結構式訪談瞭解兩組學生在歷經不同教學策略的學習表現、感受與想法。學習單示例如圖7，共3份，是根據研究所設計的教學程序逐步引導學生在音樂創作中進行運算思維(詳見「五、STEAM科際整合程式設計教學」)。教師每節課回收學習單並進行評分，以瞭解學生在本研究提出的STEAM科際整合程式設計教學中，每個步驟的學習表現如何影響其程式設計能力。

半結構式訪談問題包含：

1.你對這學期的程式設計課程的想法；

2.你對這學期的程式設計課程印象最為深刻的活動或環節；

3.程式設計課程安排的輔助學習工具，例如學習單，是否及如何幫助你學習；

4.程式設計課程之教學活動，是否能幫助你激發創意。

訪談學生共計8人，選自實驗組與控制組各4人，每人10分鐘。訪談學生選取依據為程式設計能力前、後測成績，自後測高、中、低程式設計能力學生中各選取1 ~ 2人，並納入學習狀況有值得進一步探究之受試者(例如：前測表現不佳但後測有較佳的表現者，或反之)。根據訪談結果，分析實驗組與控制組學生對程式設計學習的感受、對教師教學方式的感受，並瞭解實驗組學生在STEAM科際整合程式設計教學中之感受、是否在教學引導中獲益、是否／如何進行運算思維。學習單之評分結果與訪談資料等彙整後將與量化資料進行交叉分析討論，以佐證相關結果。

五、STEAM科際整合程式設計教學

此部分根據上述研究流程及內容，詳細說明實驗組所採用STEAM科際整合程式設計教學程序，此教學程序修改自陳怡芬等(2018)提出的運算思維導向程式設計教學模式，其教學程序分為四項程序，每一程序皆有對應欲培養的運算思維。

根據表5，四個教學步驟分別為音樂賞析與尋找規則、程式概念講授、程式體驗與音樂聆聽、音樂程式創作。其中程式概念講授為教師講解程式設計的概念，其餘三個教學步驟示例如下：

(一)音樂賞析與尋找規則

在此步驟中，教師帶領學生進行音樂賞析，引導學生辨識音樂中的曲式，並對應到適合的程式結構，以瞭解程式結構的特色。如圖8與圖9所示，圖8為音樂曲式講解說明，教師帶領學生賞析音樂，引導學生辨識音樂中的多次使用的曲式，帶入程式設計的函式概念，將該段曲式命名；在下次使用時只需要呼叫曲式名稱，不需要整段曲式重寫，透過曲式規則的辨識瞭解函式的必要性與運作方式，建立學生模式辨識的運算思維能力。

表5：實驗組的STEAM程式設計教學程序

教學步驟	教學活動	運算思維概念
音樂賞析與尋找規則	在開始撰寫程式之前，先讓學生聆聽樂曲，透過熟悉樂曲的旋律與脈絡，觀察與記錄規則	模式辨識、資料表示
程式概念講授	教師講解程式概念	
程式體驗與音樂聆聽	聆聽音樂對照程式邏輯	模式辨識、演算法思維
音樂程式創作	透過音樂創作	模式辨識、演算法思維

airiti

透過教學引導，學生搭配填寫圖9的音樂曲式學習單，在音樂曲式辨識的過程中做到模式辨識。

(二)程式體驗與音樂聆聽

學生在前一教學步驟中已學到模式辨識以掌握程式結構，接續在此教學程序中，教師引導學生從具體的音樂聆聽中對照抽象的程式設計概念。圖10為程式體驗與音樂聆聽學習單，在學習單上有兩段程式，都是同一首兒歌的歌詞，不同的是其中一段為從頭到尾以print方式印出歌詞，而另一段則結合函式相對簡短。教師帶領學生聆聽兒歌並對照學習單上的歌詞，引導學生將聽到的音樂對照學習單上的程式碼，學生在過程中除了瞭解不同程式的撰寫方式，更會發現多次出現的歌詞可以應用函式，每當需要同樣歌詞時，呼叫函式即可表達。如此可幫助抽象能力不佳的學生透過具體的音樂瞭解抽象概念，並能在測試、修改、聆聽中澄清概念。

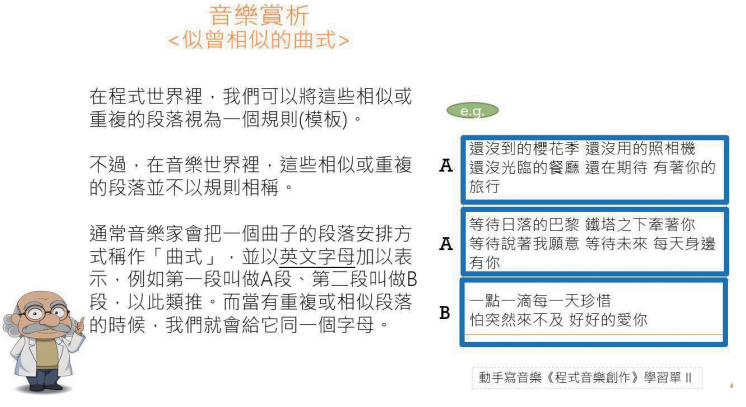


圖8：音樂曲式講解

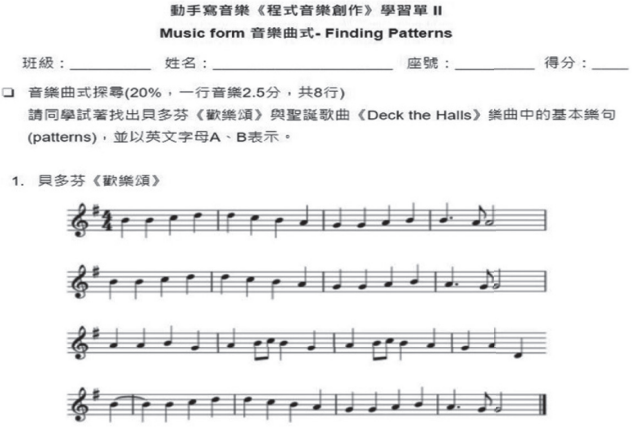


圖9：音樂曲式學習單

airiti

(三)音樂程式創作練習

學生在前面的教學步驟中已能夠進行模式辨識，並能夠將具體的音樂對照抽象的程式設計概念，此步驟則希望學生能應用所學到的程式概念透過自行設計的演算法進行創作。圖11為音樂程式創作的教學簡報。教學過程中，學生可以選擇喜歡的流行歌曲，教師先引導學生練習剖析樂曲的結構，再進一步加入程式設計的函式概念，漸進式的引導學生應用習得的概念進行演算法設計並撰寫程式。

前述為實驗組採用的STEAM科際整合程式設計教學，在整個教學中所使用的程式設計教學工具為JythonMusic (Manaris et al.,

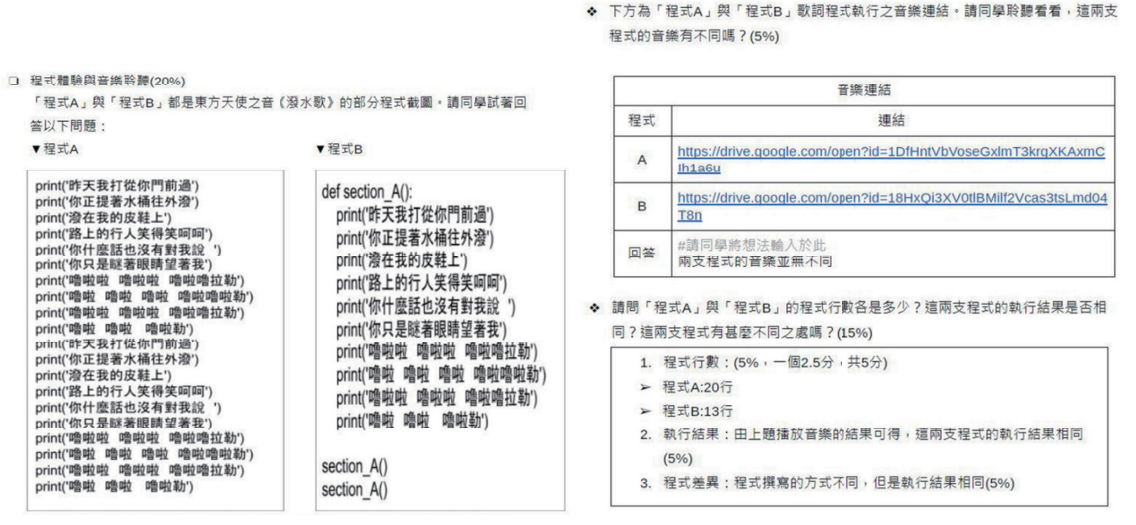


圖10：程式體驗與音樂聆聽學習單

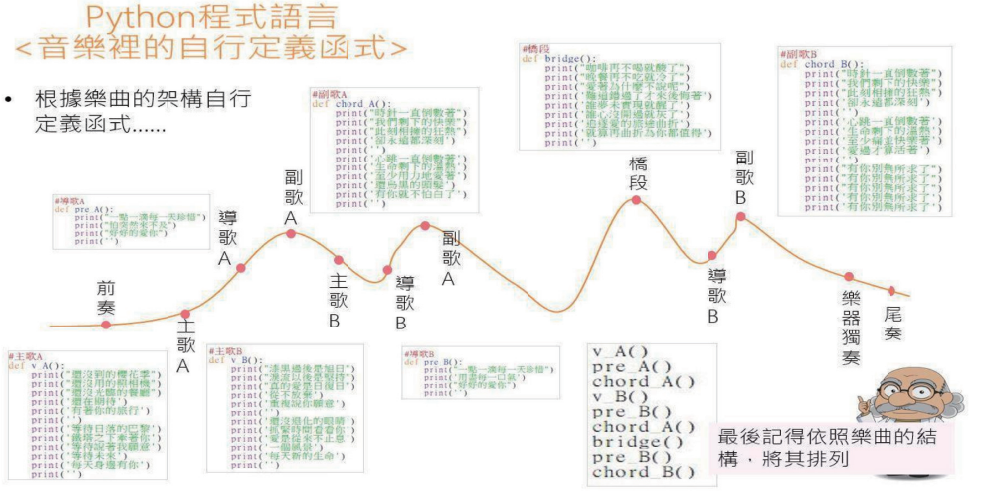


圖11：音樂程式創作教學簡報

airiti

2016)，JythonMusic是利用Python程式語言進行音樂演算法作曲的整合性工具。此工具除了提供一般的程式設計編譯功能，更提供多種音樂相關的模組，以及影像製作、建立圖形化介面等功能。學生使用該工具只需要匯入音樂模組即可呼叫音樂相關的多種函式，有利於專注在程式重要概念進行創作樂曲，進而達到程式設計的學習目的。以往音樂程式設計的研究在中學階段多使用EarSketch (Engelman et al., 2017; Freeman et al., 2019; Magerko et al., 2016)，但在本研究之前導實驗中，學生反映EarSketch之創作自由度較低，因此正式實驗中選用JythonMusic。JythonMusic程式編輯畫面如圖12所示。

根據上述運算STEAM科際整合程式設計教學程序，逐步引導學生在音樂創作中同時進行運算思維。主要的程式設計教學主題、音樂創作內容、與運算思維之對應如表6所示，學生能透過音符的表徵、辨識樂曲重複的模式，或判斷樂曲演奏的條件等，學習運算思維與程式設計。

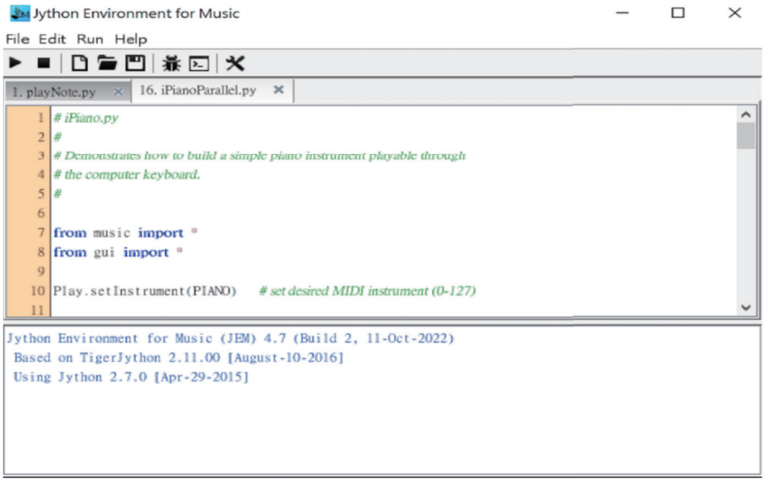


圖12：JythonMusic程式編輯畫面

表6：STEAM科際整合程式設計教學各教學主題之音樂創作與運算思維的對應

週次	教學主題	音樂創作內容	運算思維概念
5	內建函式	聆聽一段貝多芬《歡樂頌》，請學生先將音符數值化，並練習使用JythonMusic內建函式撰寫一段歡樂頌的程式碼規則。	1.資料表示(用數值表徵音符並存放到串列)。 2.演算法思維(用內建函式完成程式碼)。
6	自訂函式	1.節錄兩首歌(歡樂頌、Deck the Halls)的樂句，讓學生觀察樂句的規則。 2.相同的歌(潑水歌)以兩種不同的方式撰寫程式碼，請學生分別聆聽後，觀察與比較兩種程式碼的不同。	1.模式辨識(觀察程式碼的模式以決定函式內容)。 2.演算法思維(用自訂函式完成程式碼)。
7	if條件	請學生分別輸入不同的數值，聆聽與觀看程式執行的結果(音樂、圖片)。	演算法思維(用if條件完成程式碼)。

airiti

伍、研究結果與討論

一、研究結果

本研究對象人數扣除未完成程式設計前、後測之樣本，因此總人數為61人，其中實驗組為30人，控制組為31人。

(一)STEAM科際整合教學對程式設計能力的影響

以Quade Test檢定分析STEAM科際整合教學對學生程式設計能力的影響。分析結果如表7所示，*F*值為.185，*p*值為.669 (> .05)，未達顯著水準，表示兩組學生在整體的程式設計的學習成就並未因教學策略的不同而有改變。為瞭解本研究設計之STEAM程式設計教學程序(程式概念講授之外的其他三個程序：音樂賞析與尋找規則、程式體驗與音樂聆聽、音樂程式創作)對實驗組學生的程式設計能力的影響，進一步以Spearman相關係數分析實驗組學生於三個教學程序之學習表現(教師針對學生每個填寫的學習單進行評量)與程式設計能力的相關性，結果顯示：「音樂賞析與尋找規則」的*r*值為.341，*p*值為.032 (< .05)、「音樂程式創作」的*r*值為.409，*p*值為.012 (< .05)，說明STEAM程式設計教學程序之學習能幫助學生提升程式設計能力，而如表8所示，音樂賞析與尋找規則步驟能培養「模式辨識」與「資料表示」之運算思維概念，音樂程式創作步驟能培養「模式辨識」之運算思維概念。

(二)STEAM科際整合教學對程式創造力的影響

以曼惠特尼*U*檢定分析STEAM科際整合教學對學生程式創造力的影響。分析結果如表9所示，*Z*值為-4.827，*p*值為.000 (< .001)，達到顯著水準，表示本研究設計之STEAM科際整合教學有助於提升程式創造力表現。

為瞭解實驗組與控制組學生在不同的教學策略下，於程式創造力五個向度之表現是否有差異，研究者再進一步以曼惠特尼*U*檢

表7：以Quade Test分析檢定教學策略對程式設計能力之影響

		平均數		標準差		Quade Test	
組別	人數	前測	後測	前測	後測	*F*	*p*
實驗組	30	67.667	41.650	17.555	13.953	.185	.669
控制組	31	65.161	44.548	16.906	18.164		

表8：以Spearman相關係數檢定程式設計教學程序之學習表現(以學習單內容評分)與程式設計能力之相關性

STEAM程式設計教學程序	相關係數	*p*	運算思維
音樂賞析與尋找規則	.341	< .05	模式辨識、資料表示
程式體驗與音樂聆聽	.014	.470	模式辨識、演算法思維
音樂程式創作	.409	< .05	模式辨識、演算法思維

表9：以曼惠特尼*U*檢定教學策略對程式設計創造力之影響

		後測					
組別	人數	平均數	標準差	曼惠特尼*U*	Wilcoxon *W*	*Z*	*p*
實驗組	30	38.233	19.638	133	629	-4.827	< .001
控制組	31	11.677	13.948				

airiti

定分析兩組學生之創造力各向度之表現結果如表10，檢定結果為流暢力的Z值為-5.164，p值為.000 (< .001)、變通力的Z值為-4.097，p值為.000 (< .001)、獨創力的Z值為-3.311，p值為.001 (< .010)、敏覺力的Z值為-4.752，p值為.000 (< .001)、精進力的Z值為-4.473，p值為.000 (< .001)，各向度皆達到顯著水準，表示STEAM科際整合教學對創造力各向度皆有所助益。

(三)STEAM科際整合教學對學習態度的影響

以Quade Test分析STEAM對程式設計學習態度。自變項為教學策略，依變項為程式設計學習態度結果。分析結果如表11所示，結果顯示：「程式設計自我效能」F值為0.000，p值為.997；「程式設計學習動機」F值為0.371，p值為.545，「程式設計與STEAM之關聯性」F值為1.098，p值為.299，均未達顯著水準，表示STEAM科際整合教學未影響學習態度。

二、討論

(一)STEAM科際整合教學對程式設計能力的影響

實驗結果發現接受STEAM科際整合教學與接受傳統教學的學生在程式設計能力上並無顯著差異。這可能是因為實驗組上課時間為早上一、二節，而控制組為三、四節，研究者發現實驗組部分學生第一堂課常無法準

表10：以曼惠特尼U檢定教學策略對程式設計創造力各項度後測之差異

向度	組別	人數	平均數	標準差	曼惠特尼U	Wilcoxon W	Z	p
流暢力	實驗組	30	13.700	5.273	111.500	607.500	-5.164	< .001
	控制組	31	4.419	5.315				
變通力	實驗組	30	7.833	4.594	186.500	682.500	-4.097	< .001
	控制組	31	2.741	3.786				
獨創力	實驗組	30	3.500	3.684	256.500	752.500	-3.311	< .010
	控制組	31	1.000	2.082				
敏覺力	實驗組	30	6.833	4.793	150.500	646.500	-4.752	< .001
	控制組	31	1.290	2.479				
精進力	實驗組	30	8.233	5.799	165.000	661.000	-4.473	< .001
	控制組	31	2.225	3.008				

表11：以Quade Test分析檢定教學策略對學習態度之影響

			平均數		標準差		Quade Test	
程式設計學習態度向度	組別	人數	前測	後測	前測	後測	F	p
程式設計自我效能	實驗組	30	9.867	7.433	3.598	3.866	0.000	.997
	控制組	31	9.742	7.613	3.286	3.383		
程式設計學習動機	實驗組	30	12.000	10.000	3.648	3.991	0.371	.545
	控制組	31	12.161	9.677	3.338	4.721		
程式設計與STEAM之關聯性	實驗組	30	18.433	17.900	2.800	5.435	1.098	.299
	控制組	31	18.968	17.581	2.601	4.682		

airiti

時上課，導致影響學習表現。然而，深入分析實驗組之STEAM科際整合教學程序對學生程式設計能力的影響，發現學生在「音樂賞析與尋找規則」與「音樂程式創作」上的表現愈好，程式設計能力愈好。推測原因可能是在「音樂賞析與尋找規則」步驟中，學生能透過具體的樂曲音律對照抽象的程式結構與流程，以辨識並理解程式結構與邏輯，並透過音符的呈現瞭解資料結構的應用與資料表示方法。此亦運算思維中「模式辨識」與「資料表示」的培養。學生的訪談資料可佐證此發現：

> 透過播放音樂的過程，讓我能夠快速地找到程式規則。(E01)

> 可從音樂播放出來的結果立即知道自己的程式是否撰寫正確。在聆聽音樂的過程中，不僅有助於規則的探尋，也有助於理解較為抽象化的知識與概念及瞭解程式是否能夠順利執行。(E02)

> 尤其是遇到具有明確規則，或是重複大量事情的情況時，可以應用迴圈，比較方便。(E03)

過去的研究表明透過多元表徵、觀察與動手驗證能幫助抽象概念具體化(Bell & Bell, 2018; Jones et al., 2024)，而在實驗組的教學中，程式的邏輯可透過音律表徵的形式呈現，學生在聆聽音樂時，亦同時可觀察並驗證程式邏輯，以幫助抽象的程式邏輯具體化。透過具體的音樂情境，可幫助學生連結音樂與程式設計共通的概念(形式符號表徵、序列、重複結構等)，進而更直觀地學習運算思維(Bell & Bell, 2018)。

在「音樂程式創作步驟」中，學生則能在自由創作的思考活動中，辨識樂曲的特色並連結至相關的程式結構，以靈活應用所學的程式結構設計自己的演算法以進行音樂創作，並體驗程式應用的真實情境，因此經歷了模式辨識」與「演算法思維」的運算思維：

> 最吸引我的環節是利用程式進行音樂創作，雖然先前也曾經聽過，能夠用電腦軟體創作音樂，但是它只能照著給定的音符輸入，不像課堂教授的一樣可以自行創作。此外，我覺得程式與音樂很相似，音樂也是有規則的，這跟程式很像。(E02)

> 我覺得可以在日常生活中應用課堂所學的程式知識，例如迴圈，尤其是碰到重複、龐大的事情，這樣會方便許多。(E03)

以上皆說明STEAM程式設計教學程序中透過「模式辨識」、「資料表示」與「演算法思維」等運算思維的引導，能幫助學生發展程式設計能力。此亦回應了過去研究建議應探究教學引導於音樂程式設計課程中之有效性的建議(Petrie, 2022)。

(二)STEAM科際整合教學對程式創造力的影響

實驗結果顯示，接受STEAM教學模式的學生在程式創造力中的各向度皆有比控制組更好的表現。此結果可能是由於本研究STEAM科際整合教學設計採用實境式學習(authentic learning)，教學活動與真實世界相關且是學生所熟悉的情境，能幫助學生應用所學。訪談內容亦反映了生活化的學習情境能幫助學生應用知識靈活解決問題：

> 專題報告跟現在的議題結合，又可以達到團體討論的目標，也可以學到程式設計，接觸到很多不一樣的內容。(E04)

airiti

> 課程是實用的，可以用程式解決生活上的事情，讓我們的生活更方便。(E03)

> 創造的靈感來源來自生活，課程方面就是生活化的題目，自然就會運用課堂學習到的程式知識。(E01)

而藝術的自由創作亦能激發更多創造力，先前研究亦有相似結果，學習情境的真實性與個人創造力間存在關聯性(Siva et al., 2018)。此外，學生在藝術創作結合STEAM課程中有促進創造力之表現(Leavy et al., 2023)。本研究於教學中允許學生自行創作樂曲，在專題中學生可以自訂主題與情境並加入音樂元素，運用程式設計展現藝術元素，如圖13是學生以防災為主題進行故事情境畫面編排與音樂設定，達到自由運用學到的知識發展問題解決方法與策略(Herrington et al., 2009; Mayer, 1989; Neo et al., 2012; Reeves et al., 2002)。先前研究亦指出創作具有文化關聯性與個人意義的音樂，能激發學生的創造力(Magerko et al., 2016)。

透過創意性思考使其面對實際問題時產生相對應的思考與解法(Katanski, 2013)。學生在訪談中反映了自由創作的變化性及藝術的感官特性皆可幫助學生發揮靈感，並於面對問題時能找到程式設計解法：

> 音樂賞析，聆聽音樂，找尋規則，可以知道我接下來的事情，以及讓我比較容易找到規則。(E03)

> 程式與音樂相像處，雖然每個人接觸的音樂類型不同，但是大家都會接觸。音樂是有規則的，這跟程式相像，音樂與程式有所對應。(E02)

故事情境	畫面	音樂
場景一	在一個寧靜的夜晚，可以看到有人家裡燈還亮著，有人已經熄燈休息，大家都過著正常的生活。 簡報第2頁為場景一	from mmsic import* p = Phrase() p.setTempo(150) pi1 = [C4,C4,G4,G4,A4,A4,G4,REST,F4,F4,E4,E4,D4,D4,C4] du1 = [QN,QN,QN,QN,QN,QN,HN,0.1,QN,QN,QN,QN,QN,QN,HN] pi2 = [G4,G4,F4,F4,E4,E4,D4,REST,G4,G4,F4,F4,E4,E4,D4] du2 = [QN,QN,QN,QN,QN,QN,HN,0.1,QN,QN,QN,QN,QN,QN,HN] pi3 = [C4,C4,G4,G4,A4,A4,G4,REST,F4,F4,E4,E4,D4,D4,C4] du3 = [QN,QN,QN,QN,QN,QN,HN,0.1,QN,QN,QN,QN,QN,QN,HN] p.addNoteList(pi1,du1) p.addNoteList(pi2,du2) p.addNoteList(pi3,du3) Play.midi(p)



圖13：學生設計故事情境畫面與音樂之程式創作結果

airiti

過去研究亦發現在開放式的創作過程中，學生透過多樣化的音樂風格表達藝術的自由度，但又須考量更廣泛的音樂與運算上的限制(Freeman et al., 2019)，此學習過程令學生在更廣的思考空間中依照音樂與程式設計的邏輯進行創作，可在領域知識的限制下進行創意思考，以創作符合該領域需求的作品(Sternberg & Lubart, 1999)。

(三)STEAM科際整合教學對學習態度的影響

根據量化資料分析結果，本研究設計的STEAM科際整合教學模式對學生的學習態度未產生顯著影響。實驗組與控制組學生在「程式設計自我效能」、「程式設計學習動機」及「程式設計與STEAM之關聯性」三個構面上，均未有顯著差異，推測可能因教學時間較短，且STEAM教學涉及跨領域的教學內容，對音樂內容的接受程度可能影響對本課程的學習態度。

量化資料結果雖未能呈現實驗組與控制組學生學習態度的差異，但在課後訪談中發現，接受STEAM科際整合程式設計教學的學生反映更多對程式設計與生活及未來職業連結的想像(McKlin et al., 2018; Petrie, 2022)，亦即對於程式設計在各領域的應用有更深的體會，而能讓學生更加認可課程的價值(Beier et al., 2019)：

> 我認為這學期的程式設計課程是實用的，因為可以用程式解決生活上的事情，讓我們的生活更方便。(E03)

> 程式設計是一種興趣，雖然未來沒有想要繼續往程式設計領域走，但是有想過繼續往程式音樂創作深造，因為能夠自己創作是一件很棒的事。(E02)

此外，實際於藝術領域進行程式創作，亦幫助學生體驗程式應用的真實情境，進一步提升對程式設計的興趣：

> 雖然課堂所學的程式知識感覺跟生活比較沒有連結，不過，因為對於程式設計與音樂都感興趣，因此有意願繼續朝程式音樂創作方面深造。(E02)

許多過往研究亦發現提供真實的應用情境能幫助學生感受作品的意義與成就感，進而提升其參與度與學習動機，甚至願意持續學習(Freeman et al., 2019; Magerko et al., 2016; Siva et al., 2018; Wanzer et al., 2020)。

陸、結論與建議

一、結論

本研究提出STEAM科際整合程式設計教學模式，透過音樂創作以學習程式設計，實驗結果發現此教學能幫助學生進行模式辨識、資料表示與演算法思維，並有效提升程式創造力與學習態度。研究發現如下：

(一)雖然STEAM科際整合教學在此次實驗未能使實驗組學生的程式設計能力顯著優於控制組學生，但可發現本研究所發展的STEAM科際整合教學程序能逐步帶領學生透過音樂創作學習程式設計；透過「音樂賞析與尋找規則」步驟，能提升學生程式設計能力。學生能辨識程式結構的模式與規則(模式辨識)並瞭解資料結構的意義與用法(資料表示)；透過「音樂程式創作」理解程式概念的應用並掌握程式邏輯，在自由創作的過程中，辨識以選取適用的程式結構於程式作品(模式辨識)，並掌握程式邏輯以設計演算法以進行創作(演算法思維)。

airiti

(二)在程式創造力方面，STEAM科際整合教學能提供學生真實的程式設計應用情境，讓他們更能將程式設計與問題情境做連結，進一步覺察問題情境所需解決的問題，並能透過選取適切的程式概念提出解決方法。此教學亦能幫助學生透過樂曲與程式邏輯的對照更掌握程式指令與結構的意義，並能於樂曲的自由創作中更有彈性地將所學應用於問題解決，亦更能找到各種合適的解法，因此相較於傳統教學下的學生，他們能具備更好的創造力。

(三)STEAM科際整合教學能讓學生體驗程式設計於真實生活情境與不同領域的應用，將學習與生活經驗結合使學生能覺知程式設計與生活或未來職涯的關聯性，進而認可課程的價值並增強其學習動機。

二、建議

本研究所發展之STEAM科際整合程式設計教學雖能幫助學生掌握程式邏輯、有效提升學生的程式創造力與學習態度，然而仍有可改進之處：

本研究所使用的STEAM科際整合程式設計教材，橫跨電腦科學與音樂兩個領域，儘管這樣的課程能有效幫助學生掌握程式邏輯、提升程式創造力與學習態度，但跨科整合的教材內容讓部分學生產生認知負荷，覺得學習時間太短，較難同時掌握音樂與程式設計。因此未來在教學設計上，可考慮規劃較長的課程時間，以降低學生的認知負荷。除了教學時間的調整，亦可透過協同教學的方式分散教學與學習負荷，由音樂等藝術科目的專業教師負責該主題知識的教學，並由資訊科技與藝術領域教師共同研發合適的科際整合教材，以綜整考量不同學科、領域的特性，進而設計更具整合性、更符合需求的教材，如此亦可避免課程讓學生誤認為是兩塊不同領域內容的拼湊(Henriksen, 2014)。此外，JythonMusic軟體有其限制，雖然JythonMusic是以Python程式語言進行程式設計，但需透過特殊指令以完成音樂創作，此特性侷限了學生的程式應用領域。建議未來可將JyhonMusic這樣的工具當作入門使用，且逐漸銜接至一般的Python開發環境，以使學生有更具彈性的創作空間。

對於音樂程式設計平臺開發者，則建議可考量音樂與程式設計等不同知識背景的使用者，設計具備漸進式難度的操作工具與指令，並提供更多元的函式庫，以滿足不同使用者的創作需求。亦可針對學習需求，開發具教育意涵之音樂創作工具，提供中、小學教學與學習者更多選擇。此外，可於平臺中加入自訂教材內容與即時回饋等教學輔助工具，使其在中、小學教學現場更具實用性。

對於未來STEAM教學的研究者，建議可延伸探討學生在程式設計與音樂創作雙重任務下的認知負荷、不同音樂或藝術創作任務對程式設計創造力的影響，以及此等教學對音樂(或其他藝術主題)知識的學習與音樂／藝術創造力的影響，以更全面理解科際整合教學的效益，及不同藝術主題對於STEM學科學習的影響。此外，未來研究亦可延伸至具藝術背景之學生群體，例如音樂班學生，以探討結合音樂的STEAM教學是否更能引發音樂背景或其他藝術領域學生的學習動機，並進一步評估本教學設計之適用性與延展性。

airiti

參考文獻

Bebras (n.d.)。**國際運算思維挑戰賽**。https://reurl.cc/NNlome

陳怡芬、林育慈、翁禎苑 (2018)。運算思維導向程式設計教學——以「動手玩音樂」模組化程式設計為例。**中等教育**，**69**(2)，127-141。https://doi.org/10.6249/SE.201806_69(2).0023

[Chen, Y.-F., Lin, Y.-T., & Weng, C.-Y. (2018). Modular programming with computational thinking. *Secondary Education*, *69*(2), 127-141. https://doi.org/10.6249/SE.201806_69(2).0023]

葉玉珠 (2004)。"科技創造力測驗"的發展與常模的建立。**測驗學刊**，**51**(2)，127-162。https://doi.org/10.7108/PT.200412.0127

[Yeh, Y.-C. (2004). The development of the "Technological Creativity Test" and the construction of its scoring norms. *Psychological Testing*, *51*(2), 127-162. https://doi.org/10.7108/PT.200412.0127]

Ajaykumar, G., Steele, M., & Huang, C.-M. (2022). A survey on end-user robot programming. *ACM Computing Surveys*, *54*(8), 1-36. https://doi.org/10.1145/3466819

Baist, A., & Pamungkas, A. S. (2017). Analysis of student difficulties in computer programming. *VOLT: Jurnal Ilmiah Pendidikan Teknik Elektro*, *2*(2), 81. https://doi.org/10.30870/volt.v2i2.2211

Beier, M. E., Kim, M. H., Saterbak, A., Leautaud, V., Bishnoi, S., & Gilberto, J. M. (2019). The effect of authentic project-based learning on attitudes and career aspirations in STEM. *Journal of Research in Science Teaching*, *56*(1), 3-23. https://doi.org/10.1002/tea.21465

Bell, J., & Bell, T. (2018). Integrating computational thinking with a music education context. *Informatics in Education*, *17*(2), 151-166. https://doi.org/10.15388/infedu.2018.09

Boden, M. A. (2009). Computer models of creativity. *AI Magazine*, *30*(3), 23-34. https://doi.org/10.1609/aimag.v30i3.2254

Burrows, A. C., Breiner, J. M., Keiner, J., & Behm, C. (2014). Biodiesel and integrated STEM: Vertical alignment of high school biology/biochemistry and chemistry. *Journal of Chemical Education*, *91*(9), 1379-1389. https://doi.org/10.1021/ed500029t

Cahill, M., & Petersen, J. (2022). Taking STEM to STEAM and enhancing creativity. In K. J. Murcia, C. Campbell, M. M. Joubert, & S. Wilson (Eds.), *Children's creative inquiry in STEM* (pp. 273-287). Springer. https://doi.org/10.1007/978-3-030-94724-8_16

Castledine, A.-R., & Chalmers, C. (2011). LEGO robotics: An authentic problem-solving tool? *Design and Technology Education*, *16*(3), 19-27. https://doi.org/10.24377/DTEIJ.article1703

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, *12*(2), Article ep272. https://doi.org/10.30935/cedtech/8247

airiti

Chung, C. J., Cartwright, C., & Chung, C. (2014, March). *Robot music camp 2013: An experiment to promote STEM and computer science*. 2014 IEEE Integrated STEM Education Conference, Princeton, NJ. https://doi.org/10.1109/ISECon.2014.6891012

Clements, D. H. (1986). Effects of Logo and CAI environments on cognition and creativity. *Journal of Educational Psychology*, *78*(4), 309-318. https://doi.org/10.1037/0022-0663.78.4.309

Clements, D. H. (1995). Teaching creativity with computers. *Educational Psychology Review*, *7*(2), 141-161. https://doi.org/10.1007/BF02212491

Connor, A. M., Karmokar, S., & Whittington, C. (2015). From STEM to STEAM: Strategies for enhancing engineering & technology education. *International Journal of Engineering Pedagogy*, *5*(2), Article 37. https://doi.org/10.3991/ijep.v5i2.4458

Conradty, C., & Bogner, F. X. (2019). From STEM to STEAM: Cracking the code? How creativity & motivation interacts with inquiry-based learning. *Creativity Research Journal*, *31*(3), 284-295. https://doi.org/10.1080/10400419.2019.1641678

Dagdilelis, V., Sartatzemi, M., & Kagani, K. (2005). Teaching (with) robots in secondary schools: Some new and not-so-new pedagogical problems. In P. Goodyear, D. G. Sampson, D. J.-T. Yang, K., T. Okamoto, R. Hartley, & N.-S. Chen (Eds.), *The fifth IEEE international conference on advanced learning technologies*, (pp. 757-761). Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICALT.2005.255

Daugherty, M., Carter, V., & Sumner, A. (2021). Standards for technological and engineering literacy and STEM education. *Technology and Engineering Teacher*, *80*(5), 32-37.

Dominguez, A., De la Garza, J., Quezada-Espinoza, M., & Zavala, G. (2023). Integration of physics and mathematics in STEM education: Use of modeling. *Education Sciences*, *14*(1), Article 20. https://doi.org/10.3390/educsci14010020

Du Boulay, B. (1989). Some difficulties of learning to program. In E. Soloway & J. C. Spohrer (Eds.), *Studying the novice programmer* (pp. 283-299). Psychology Press.

Dúo-Terrón, P. (2023). Analysis of scratch software in scientific production for 20 years: Programming in education to develop computational thinking and steam disciplines. *Education Sciences*, *13*(4), Article 404. https://doi.org/10.3390/educsci13040404

Edwards, M. (2011). Algorithmic composition: Computational thinking in music. *Communications of the ACM*, *54*(7), 58-67. https://doi.org/10.1145/1965724.1965742

Engelman, S., Magerko, B., McKlin, T., Miller, M., Edwards, D., & Freeman, J. (2017). Creativity in authentic STEAM education with EarSketch. In S. H. Edwards, M. E. Caspersen, T. Barnes, & D. D. Garcia (Eds.), *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 183-188). Association for Computing Machinery. https://doi.org/10.1145/3017680.3017763

airiti

Ferreira, D. J. (2013). Fostering the creative development of computer science students in programming and interaction design. *Procedia Computer Science*, *18*, 1446-1455. https://doi.org/10.1016/j.procs.2013.05.312

Freeman, J., Magerko, B., Edwards, D., Mcklin, T., Lee, T., & Moore, R. (2019). EarSketch: Engaging broad populations in computing through music. *Communications of the ACM*, *62*(9), 78-85. https://doi.org/10.1145/3333613

Ge, X., Ifenthaler, D., & Spector, J. M. (Eds.). (2015). *Emerging technologies for STEAM education: Full STEAM ahead*. Springer. https://doi.org/10.1007/978-3-319-02573-5

Giza, P. (2021). Creativity in computer science. *Creativity Studies*, *14*(2), 444-460. https://doi.org/10.3846/cs.2021.14699

Glatthorn, A. A. (1994). Constructivism: Implications for curriculum. *International Journal of Educational Reform*, *3*(4), 449-455. https://doi.org/10.1177/105678799400300407

Gomes, A., & Mendes, A. J. (2007, September). *Learning to program—Difficulties and solutions*. International Conference on Engineering Education—ICEE 2007, Coimbra, Portugal.

Gorson, J., & O'Rourke, E. (2020). Why do CS1 students think they're bad at programming? Investigating self-efficacy and self-assessments at three universities. In A. Robins, A. Moskal, A. J. Ko, R. McCauley, A. Robins, & A. J. Ko (Eds.), *ICER '20: Proceedings of the 2020 ACM conference on international computing education research* (pp. 170-181). Association for Computing Machinery. https://doi.org/10.1145/3372782.3406273

Guilford, J. P. (1950). Creativity. *American Psychologist*, *5*(9), 444-454. https://doi.org/10.1037/h0063487

Guilford, J. P. (1966). Measurement and Creativity. *Theory Into Practice*, *5*(4), 185-189. https://doi.org/10.1080/00405846609542023

Guilford, J. P. (1967). Creativity: Yesterday, today and tomorrow. *The Journal of Creative Behavior*, *1*(1), 3-14. https://doi.org/10.1002/j.2162-6057.1967.tb00002.x

Guyotte, K. W., Sochacka, N. W., Costantino, T. E., Kellam, N. N., & Walther, J. (2015). Collaborative creativity in STEAM: Narratives of art education students' experiences in transdisciplinary spaces. *International Journal of Education & the Arts*, *16*(15).

Guyotte, K. W., Sochacka, N. W., Costantino, T. E., Walther, J., & Kellam, N. N. (2014). STEAM as social practice: Cultivating creativity in transdisciplinary spaces. *Art Education*, *67*(6), 12-19. https://doi.org/10.1080/00043125.2014.11519293

Guzdial, M., & Soloway, E. (2002). Teaching the Nintendo generation to program. *Communications of the ACM*, *45*(4), 17-21. https://doi.org/10.1145/505248.505261

Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2009). A multidisciplinary approach towards computational thinking for science majors. *ACM SIGCSE Bulletin*, *41*(1), 183-187. https://doi.org/10.1145/1539024.1508931

airiti

Henriksen, D. (2014). Full STEAM ahead: Creativity in excellent STEM teaching practices. *STEAM*, *1*(2), 1-9. https://doi.org/10.5642/steam.20140102.15

Herrington, J., Reeves, T. C., & Oliver, R. (2009). *A guide to authentic e-learning*. Routledge. https://doi.org/10.4324/9780203864265

Hoffmann, J., Ivcevic, Z., & Brackett, M. (2016). Creativity in the age of technology: Measuring the digital creativity of millennials. *Creativity Research Journal*, *28*(2), 149-153. https://doi.org/10.1080/10400419.2016.1162515

Horn, M. S., Banerjee, A., West, M., Pinkard, N., Pratt, A., Freeman, J., Magerko, B., & McKlin, T. (2020). *TunePad: Engaging learners at the intersection of music and code*. https://reurl.cc/k8OK6b

Hu, W., & Adey, P. (2002). A scientific creativity test for secondary school students. *International Journal of Science Education*, *24*(4), 389-403. https://doi.org/10.1080/09500690110098912

Jacob, B. L. (1996). Algorithmic composition as a model of creativity. *Organised Sound*, *1*(3), 157-165. https://doi.org/10.1017/S1355771896000222

Jeffrey, B., & Craft, A. (2004). Teaching creatively and teaching for creativity: Distinctions and relationships. *Educational Studies*, *30*(1), 77-87. https://doi.org/10.1080/0305569032000159750

Jenkins, T. (2002). On the difficulty of learning to program. In *3rd annual conference of the LTSN centre for information and computer sciences* (vol. 4, pp. 53-58). LTSN-ICS.

Jones, W. M., Hansen, K., Krug, D. L., Schad, M. L., Whittington, N., & Liu, X. (2024). Digital music composition: The effects of computer coding musical compositions on adult learners' attitudes toward computer science. *Computer Science Education*, *35*(3), 635-667. https://doi.org/10.1080/08993408.2024.2429059

Kanematsu, H., & Barry, D. M. (2016). STEM and creativity. In H. Kanematsu & D. M. Barry (Eds.), *STEM and ICT education in intelligent environments* (pp. 15-23). Springer. https://doi.org/10.1007/978-3-319-19234-5_3

Katanski, D. (2013, June). *Bridging the creativity and STEM crisis*. 2013 ASQ Advancing the STEM Agenda Conference, Section 4-2, Grand Rapids, MI.

Kim, K. H., & Coxon, S. V. (2016). Fostering creativity using robotics among students in STEM fields to reverse the creativity crisis. In M. K. Demetrikopoulos & J. L. Pecore (Eds.), *Interplay of creativity and giftedness in science* (pp. 351-365). Sense Publishers. https://doi.org/10.1007/978-94-6300-163-2_19

Kim, S.-W., & Lee, Y. (2016). The analysis on research trends in programming based STEAM education in Korea. *Indian Journal of Science and Technology*, *9*(24), 1-11. https://doi.org/10.17485/ijst/2016/v9i24/96102

Kind, P. M., & Kind, V. (2007). Creativity in science education: Perspectives and challenges for developing school science. *Studies in Science Education*, *43*(1), 1-37. https://doi.

airiti

org/10.1080/03057260708560225

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. In J. Cunha, W. Fleischman, V. K. Proulx, & J. Lourenço (Eds.), *ITiCSE '05: Proceedings of the 10th annual SIGCSE conference on innovation and technology in computer science education* (pp. 14-18). Association for Computing Machinery. https://doi.org/10.1145/1067445.1067453

Land, M. H. (2013). Full STEAM ahead: The benefits of integrating the arts into STEM. *Procedia Computer Science*, *20*, 547-552. https://doi.org/10.1016/j.procs.2013.09.317

Leavy, A., Dick, L., Meletiou-Mavrotheris, M., Paparistodemou, E., & Stylianou, E. (2023). The prevalence and use of emerging technologies in STEAM education: A systematic review of the literature. *Journal of Computer Assisted Learning*, *39*(4), 1061-1082. https://doi.org/10.1111/jcal.12806

Lin, Y.-T., Yeh, M. K.-C., & Tan, S.-R. (2022). Teaching programming by revealing thinking process: Watching experts' live coding videos with reflection annotations. *IEEE Transactions on Education*, *65*(4), 617-627. https://doi.org/10.1109/TE.2022.3155884

Liu, Z., Zhang, S., Israel, M., Smith, R., Xing, W., & Minces, V. (2025). Engaging K-12 students with flow-based music programming: An experience report on its impact on teaching and learning. In J. A. Stone, T. Yuen, L. Shoop, S. A. Rebelsky, & J. Prather (Eds.), *Proceedings of the 56th ACM Technical Symposium on Computer Science Education, Vol. 1*, (pp. 708-714). Association for Computing Machinery. https://doi.org/10.1145/3641554.3701902

Maeda, J. (2012). *STEM to STEAM: Art in K-12 is key to building a strong economy*. https://reurl.cc/1kvRnQ

Magerko, B., Freeman, J., Mcklin, T., Reilly, M., Livingston, E., Mccoid, S., & Crews-Brown, A. (2016). Earsketch: A STEAM-based approach for underrepresented populations in high school computer science education. *ACM Transactions on Computing Education*, *16*(4), Article 14. https://doi.org/10.1145/2886418

Majumdar, S. K. (1975). A systems approach to identification and nurture of scientific creativity. *Journal of Indian Education*, *1*, 17-23.

Manaris, B., Stevens, B., & Brown, A. R. (2016). JythonMusic: An environment for teaching algorithmic music composition, dynamic coding and musical performativity. *Journal of Music, Technology & Education*, *9*(1), 33-56. https://doi.org/10.1386/jmte.9.1.33_1

Mannila, L., Peltomäki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer Science Education*, *16*(3), 211-227. https://doi.org/10.1080/08993400600912384

Mayer, R. E. (1989). Cognitive views of creativity: Creative teaching for creative learning. *Contemporary Educational Psychology*, *14*(3), 203-211. https://doi.org/10.1016/0361-

airiti

476X(89)90010-6

McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: A review of the literature from an educational perspective. *Computer Science Education*, *18*(2), 67-92. https://doi.org/10.1080/08993400802114581

McKlin, T., Magerko, B., Lee, T., Wanzer, D., Edwards, D., & Freeman, J. (2018). Authenticity and personal creativity: How EarSketch affects student persistence. In T. Barnes, D. Garcia, E. K. Hawthorne, & M. A. Pérez-Quiñones (Eds.), *SIGCSE '18: Proceedings of the 49th ACM technical symposium on computer science education* (pp. 987-992). Association for Computing Machinery. https://doi.org/10.1145/3159450.3159523

Milne, A. J., & Calilhanna, A. M. (2019). Teaching music with mathematics: A pilot study. In M. Montiel, F. Gomez-Martin, & O. A. Agustín-Aquino (Eds.), *Mathematics and computation in music* (pp. 383-389). Springer. https://doi.org/10.1007/978-3-030-21392-3_34

Neo, M., Neo, K. T.-K., & Tan, H. Y.-J. (2012). Applying authentic learning strategies in a multimedia and web learning environment (MWLE): Malaysian students' perspective. *Turkish Online Journal of Educational Technology*, *11*(3), 50-60.

Ogegbo, A. A., & Aina, A. Y. (2024). Exploring young students' attitude towards coding and its relationship with STEM career interest. *Education and Information Technologies*, *29*(8), 9041-9059. https://doi.org/10.1007/s10639-023-12133-5

Ozkan, G., & Topsakal, U. U. (2021). Exploring the effectiveness of STEAM design processes on middle school students' creativity. *International Journal of Technology and Design Education*, *31*(1), 95-116. https://doi.org/10.1007/s10798-019-09547-z

Ozobot. (2020). *Lessons & activities*. https://reurl.cc/VmzdxQ

Pea, R. D., & Kurland, D. M. (1984). *On the cognitive and educational benefits of teaching children programming: A critical look*. https://reurl.cc/0ab4EA

Perignat, E., & Katz-Buonincontro, J. (2019). STEAM in practice and research: An integrative literature review. *Thinking Skills and Creativity*, *31*, 31-43. https://doi.org/10.1016/j.tsc.2018.10.002

Perkins, D. N., & Martin, F. (1986). Fragile knowledge and neglected strategies in novice programmers. In E. Soloway & S. Iyengar (Eds.), *Papers presented at the first workshop on empirical studies of programmers on Empirical studies of programmers* (pp. 213-229). Ablex Publishing.

Petrie, C. (2022). Programming music with Sonic Pi promotes positive attitudes for beginners. *Computers & Education*, *179*, Article 104409. https://doi.org/10.1016/j.compedu.2021.104409

Polya, G. (2014). *How to solve it: A new aspect of mathematical method*. Princeton University Press. https://doi.org/10.1515/9781400828678

Qian, Y., & Lehman, J. (2018). Students' misconceptions and other difficulties in introductory pro-

airiti

gramming: A literature review. *ACM Transactions on Computing Education*, *18*(1), Article 1. https://doi.org/10.1145/3077618

Quigley, C. F., & Herro, D. (2016). "Finding the joy in the unknown": Implementation of STEAM teaching practices in middle school science and math classrooms. *Journal of Science Education and Technology*, *25*(3), 410-426. https://doi.org/10.1007/s10956-016-9602-z

Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. In R. Boyle, M. Clark, & A. Kumar (Eds.), *ITiCSE '04: Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 171-175). Association for Computing Machinery. https://doi.org/10.1145/1007996.1008042

Reeves, T. C., Herrington, J., & Oliver, R. (2002). Authentic activities and online learning. In T. Herrington (Ed.), *Research and development in higher education Vol. 25: Quality conversations* (pp. 562-567). Higher Education Research and Development Society of Australasia.

Roberts, C. L., & Horn, M. S. (2025). Computational musicking: Music + coding as a hybrid practice. *Behaviour & Information Technology*, *44*(5), 993-1013. https://doi.org/10.1080/0144929X.2024.2402533

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), 137-172. https://doi.org/10.1076/csed.13.2.137.14200

Runco, M. A., & Acar, S. (2012). Divergent thinking as an indicator of creative potential. *Creativity Research Journal*, *24*(1), 66-75. https://doi.org/10.1080/10400419.2012.652929

Salomon, G., & Perkins, D. N. (1987). Transfer of cognitive skills from programming: When and how? *Journal of educational computing research*, *3*(2), 149-169. https://doi.org/10.2190/6F4Q-7861-QWA5-8PL1

Saunders, D., & Thagard, P. (2005). Creativity in computer science. In J. C. Kaufman & J. Baer (Eds.), *Creativity across domains: Faces of the muse* (pp. 171-186). Psychology Press.

Sharmin, S. (2021). Creativity in CS1: A literature review. *ACM Transactions on Computing Education*, *22*(2), Article 16. https://doi.org/10.1145/3459995

Simoni, M., & Dannenberg, R. B. (2013). *Algorithmic composition: A guide to composing music with nyquist*. University of Michigan Press. https://doi.org/10.3998/mpub.3702280

Siva, S., Im, T., McKlin, T., Freeman, J., & Magerko, B. (2018). Using music to engage students in an introductory undergraduate programming course for non-majors. In T. Barnes, D. Garcia, E. K. Hawthorne, & M. A. Pérez-Quiñones (Eds.), *SIGCSE'18: Proceedings of the 49th ACM technical symposium on computer science education* (pp. 975-980). Association for Computing Machinery. https://doi.org/10.1145/3159450.3159468

Soloway, E., & Ehrlich, K. (1984). Empirical studies of programming knowledge. *IEEE Transactions*

airiti

on Software Engineering, *SE-10*(5), 595-609. https://doi.org/10.1109/TSE.1984.5010283

Sousa, D. A. & Pilecki, T. (2013). *From STEM to STEAM: Using brain-compatible strategies to integrate the arts*. Corwin Press.

Spohrer, J. C., & Soloway, E. (1986). Novice mistakes: Are the folk wisdoms correct? *Communications of the ACM*, *29*(7), 624-632. https://doi.org/10.1145/6138.6145

Sprague, N. (2013). Arduino as a platform for a computer organization course. *Journal of Computing Sciences in Colleges*, *28*(3), 53-60.

Sternberg, R. J. & Lubart, T. I. (1998). The Concept of Creativity: Prospects and Paradigms. In R. J. Sternberg (Ed.), *Handbook of Creativity* (pp. 3-15). Cambridge University Press.

Taube, H. (2004). *Notes from the metalevel: An introduction to computer composition*. Routledge. https://doi.org/10.4324/9781315078182

Tiryaki, A., & Adıgüzel, S. (2021). The effect of STEM-based robotic applications on the creativity and attitude of students. *Journal of Science Learning*, *4*(3), 288-297. https://doi.org/10.17509/jsl.v4i3.29683

Torrance, E. P. (1974). *Torrance tests of creative thinking*. Personnel Press.

Tsai, C.-Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior*, *95*, 224-232. https://doi.org/10.1016/j.chb.2018.11.038

Wang, G., Cook, P. R., & Salazar, S. (2015). ChucK: A strongly timed computer music language. *Computer Music Journal*, *39*(4), 10-29. https://doi.org/10.1162/COMJ_a_00324

Wanzer, D. L., McKlin, T., Freeman, J., Magerko, B., & Lee, T. (2020). Promoting intentions to persist in computing: An examination of six years of the EarSketch program. *Computer Science Education*, *30*(4), 394-419. https://doi.org/10.1080/08993408.2020.1714313

Wiebe, E., Williams, L. A., Yang, K., & Miller, C. S. (2003). *Computer science attitude survey*. https://reurl.cc/9b3DbO

Yeh, Y.-C., & Wu, J.-J. (2006). The cognitive processes of pupils' technological creativity. *Creativity Research Journal*, *18*(2), 213-227. https://doi.org/10.1207/s15326934crj1802_7

Zhang, Y., Tabares, S. O., Patt, R., Krug, D. L., Mead, H., Mouza, C., Shepherd, D., & Pollock, L. (2024). Examining participation and outcomes among middle school students in a virtual camp on coding with music. In R. Lindgren, T. Asino, E. A. Kyza, C. K. Looi, D. T. Keifert, & E. Suárez (Eds.), *18th International Conference of the Learning Sciences (ICLS)*, pp. 594-601. International Society of the Learning Sciences. https://doi.org/10.22318/icls2024.102665

Žižic, A., Granić, A., & Lukie, M. P. (2017). What about creativity in computer science education? *International Journal for Talent Development and Creativity*, *5*(1&2), 95-108.

airiti

科學教育學刊
2025, 第三十三卷第三期, 195-225
DOI:10.6173/CJSE.202509_33(3).0001

Contemporary Journal of Science Education
2025, 33(3), 195-225

Interdisciplinary STEAM Instruction: Learning Programming by Music Composition

Si-Ru Chen[1], Liang-Yin Ye[1], Sheng-Che Hsiao[1], Yu-Tzu Lin[1,*], Sheng-Tsang Chuo[1]

[1]Graduate Institute of Information and Computer Education, National Taiwan Normal University

Abstract

STEAM education is an interdisciplinary approach to learning that combines science, technology, engineering, arts, and mathematics. It aims to develop students' problem-solving ability and creativity by engaging them in an authentic learning context. The aim of this study is to implement an interdisciplinary STEAM instruction for computer programming through music composition using JythonMusic. In addition, a quasi-experimental research was conducted to examine the effectiveness of the proposed STEAM instruction on students' programming achievement, creativity, and learning attitude. The experimental group learned programming through the STEAM instruction, and the control group learned in a traditional method. The experiment results show that the STEAM instruction helped students grasp program flow and logic by mapping the perceivable melody to abstract program constructs, which involves algorithmic thinking. Students in the experimental group could also recognize the features of program constructs to understand the roles the constructs played in the program, which is pattern recognition. The music composition application also provided an authentic context for the students, in which they could create artifacts based on their ideas and perceive the problems more easily, thereby enhancing their sensitivity and creativity. Moreover, the art context of this course made students aware of the connection between computer programming and art, which also helped them recognize the usefulness of programming in various fields.

Key words: STEAM Education, Computer Programming, Creativity

* Corresponding author: Yu-Tzu Lin, linyt@ntnu.edu.tw; ORCID: 0000-0002-2138-538X
Received: 2025/2/15, Revised: 2025/6/12, Accepted: 2025/7/20, Available Online: 2026/1/26