

自我解釋適性化聊天機器人影響程式設計能力之初探

The Preliminary Exploration of the Impact of Self-Explaining Adaptive Chatbots on
Programming Ability

林子白, 洪暉鈞

台灣中央大學 網路學習科技研究所

abcde12345326@gmail.com

【摘要】 本研究針對結合生成式 AI 聊天機器人與自我解釋策略的適性化自動反饋系統，探討其對學生程式設計能力及運算思維的影響。透過在台灣北部一所大學開設的 16 周「Python 教育資料探勘實作」課程，研究以 Wilcoxon 符號檢定方式，比較學生在參與課堂前後程式設計能力、運算思維與學習動機的差異。結果顯示，學生的 Python 程式設計分數於課程結束後明顯提升。顯示學生普遍生成式 AI 技術在程式教育中不僅能作為解答工具，更能透過自我解釋機制引導學生反思與修正錯誤。

【關鍵詞】 自我解釋；運算思維；程式學習；生成式人工智慧；聊天機器人

Abstract: This study investigates the impact of an adaptive automated feedback system that integrates generative AI chatbots and the self-explanation strategy on students' programming abilities and computational thinking. Through a 16-week "Python Educational Data Mining Practice" course offered at a university in northern Taiwan, the study employs the Wilcoxon signed-rank test to compare students' programming proficiency, computational thinking, and learning motivation before and after participating in the course. The results indicate a significant improvement in students' Python programming scores after completing the course, demonstrating that students generally recognized the benefits of this instructional model. Generative AI technology in programming education not only serves as an answering tool but also facilitates students' reflection and error correction through the self-explanation mechanism.

Keywords: Self-explanation, Computational Thinking, Programming Learning, Generative Artificial Intelligence, Chatbot

1. 前言

隨著科技快速發展，學生需培養批判思考及運用新技術解決問題的能力(Ogegbo & Ramnarain, 2021)，而學習運算思維(Computational Thinking, CT)被視為提升此類能力的有效方式(Cansu & Cansu, 2019)。因其重要性，CT 已成為因應世界變化的基礎能力，並被許多國家納入正式教育體系(Grover & Pea, 2018; Heintz et al., 2016)。其中，程式設計與電腦科學領域息息相關，使學生能透過撰寫程式訓練和實踐 CT (Grover & Pea, 2013; Tang et al., 2020)，程式設計能力也遂成為教育趨勢，不僅包含傳統語言，也延伸至機器學習和資料科學領域。

在學習過程中，反饋是關鍵要素(Shute, 2008)。程式教育中，作業反饋可協助教師與學生掌握學習狀況，但評分往往耗時且難以同時滿足多數學生的學習需求(Alves et al., 2019; Hao et al., 2022; Slavuj et al., 2017)。學生間的差異包括知識背景、認知能力、學習速度與風格(Cook et al., 2018)；因此需要即時反饋與個別化支持，使教育者探索創新方法以提升學習效果。然而，目前多數自動反饋系統雖可檢查作業正確性與品質(Keuning et al., 2018)，但在實現真正的適性化學習與深度反饋方面仍顯不足(Combéfis & Schils, 2016; Keuning et al., 2018)。教師在此類系統中的調整空間也相對有限(Keuning et al., 2018)。此外，因缺乏對學生理解的

關注，易導致不良學習習慣或錯誤認知，特別在生成式 AI 技術興起後，問題更形嚴峻 (Auvinen, 2015; Kapur, 2016; Lehtinen et al., 2023)。

Lehtinen et al. (2021)提出的 Questions about Learners' Code (QLCs)策略，強調學生透過「自我解釋」(Bisra et al., 2018)來反思程式碼與背後原理，促進由表面模仿轉向深度理解並提升後續程式撰寫、除錯與擴充能力。然而，此策略若要落實於自動化與適性化系統，仍面臨技術挑戰。近年來，ChatGPT 等生成式 AI 的出現 (Popovici, 2023)，開啟了在程式課程中提供個別化對話、深度討論和反饋的新可能 (Hwang et al., 2020; Popovici, 2023)，例如，Chiu (2024) 採用自我決定理論 (Self-Determination Theory, SDT) 探討 ChatGPT 如何支持學生的學習需求，發現 AI 可透過提供自主性支援與即時反饋來增強學習者的內在動機。

因此，本研究旨在開發一套結合生成式 AI 和 QLC 方法的輔導系統，以回應既有自動化反饋工具在適應性及深度理解上的不足。系統將透過個人化反饋協助學生從基礎概念深化至程式設計思維，同時提供教師更靈活的反饋設定與課程調整。透過實驗研究與使用者反饋，本研究將探究該系統在增進學習成效與教學品質方面的潛力。

在現今的教學情境中，反饋被視為學習過程中的關鍵要素(Shute, 2008)，透過即時與客觀的反饋，學生能了解自己與目標之間的差距，並進一步調整學習策略(Paiva et al., 2022)。然而，若完全仰賴教師進行手動批改與反饋，隨著班級規模擴大，往往會面臨「成本高、耗時長」的問題(Alves et al., 2019; Slavuj et al., 2017)。特別是在程式教育中，由於學生的學習背景、認知能力與學習速度各不相同，教師要同時提供深入且個別化的反饋並不容易(Cook et al., 2018; Hao et al., 2022)。手動評分也可能受評分者一致性不足、疲勞或主觀偏見的影響(Zen et al., 2011)，顯示自動化反饋系統在協助教師即時且客觀地評量學生程式作業上，具有極高的應用潛力(Combéfis & Schils, 2016)。

在教學策略方面，自我解釋(self-explanation)已被證實是能顯著提升學習成效的方式(Chi, 2013)。此策略強調學習者以「說明什麼、為什麼與如何」的方式，將新知與既有知識聯繫並融會貫通，有助於鞏固所學並促進遷移能力(Pirolli & Recker, 1994)。自我解釋也被視為一種後設認知策略(Chi, 2009, 2018)，讓學生得以持續監控並調整自己的學習過程，增進對知識差距的覺察與解決方式，進而強化學習動機與持續投入(Yu, 2022)。在程式設計教育中，透過要求學生對程式碼的功能、邏輯與設計原理進行自我解釋，有助於他們更有效地理解並內化程式概念，培養解決問題與運算思維的能力(Pirolli & Recker, 1994; Wylie & Chi, 2014)。

近年來，聊天機器人(chatbot)技術的成熟，讓自動化系統在教學上的應用更具彈性與互動性(Pham et al., 2018)。在教育領域，聊天機器人不僅能迅速回應學生疑問，還能在測驗、作業與論文等反饋活動中提供即時且個別化的反饋(Lalwani et al., 2018)。在程式教學方面，Yin et al. (2021)與 Benotti et al. (2017)的研究皆發現，聊天機器人可應用於大規模線上課程，透過自然語言對話為學生解答程式相關問題，提供自動化的形成性反饋，顯著提高學生的任務完成率。這些研究結果顯示，將聊天機器人應用於程式教育，不僅能因應大班教學需求，還可進一步促進學生的即時反思與深度學習。

綜觀上述文獻，可以發現：「自動化反饋」與「自我解釋」結合「聊天機器人」的互動機制，對於提升學生在程式教育的學習動機與成效具有相當大的潛力。本研究基於此基礎，計畫開發一個適性化自動反饋系統，整合自我解釋策略與聊天機器人技術，為學生提供即時且個性化的問題提示與反饋。系統將引導學生透過自我解釋的方式進程式碼思維的梳理與反思，期望能在多元化的學習情境下，有效提升學生的程式設計能力與運算思維。

2. 自我解釋適性化學伴系統

本研究開發的自我解釋適性化學伴系統提供適性化題目生成、即時自我解釋反饋功能。在課堂教學的前期，教師透過傳統講授或示範等方式進行基礎理論與實作指導。教師會在課堂的各階段結束後發布該階段主題的程式作業，使學生在課後進行更深入的練習。當學生完成課堂練習後，提交作業後，本系統的 AI 出題模組會根據學生實際撰寫的程式碼內容，

生成適性化的題目。並於隔周課堂時間開放學生進行系統活動，讓學生在回答適性化題目的過程中透過聊天機器人的回應，進行自我解釋的檢驗與修正，強化學生對程式原理的掌握，還能透過不斷反思與修正，達到更深層的學習成效。

系統活動介面如圖 1 所示，分為自我解釋區塊與反饋互動區塊，自我解釋測驗區塊根據學生程式碼自動生成與章節概念相關的題目，以開放式問題詢問學生對程式碼邏輯或概念的理解程度。如 Koedinger et al. (2010)研究的 ASSISTments 系統，提供即時反饋及輔助鷹架，學生可於反饋互動區塊與聊天機器人取得即時反饋並進行後續討論或釐清，即時反饋包含自我解釋評分與評分原因及修改建議，自我解釋評分基於修改自 Nakamoto et al. (2023)的自我解釋標準，詳見表 1 自我解釋評分分級。

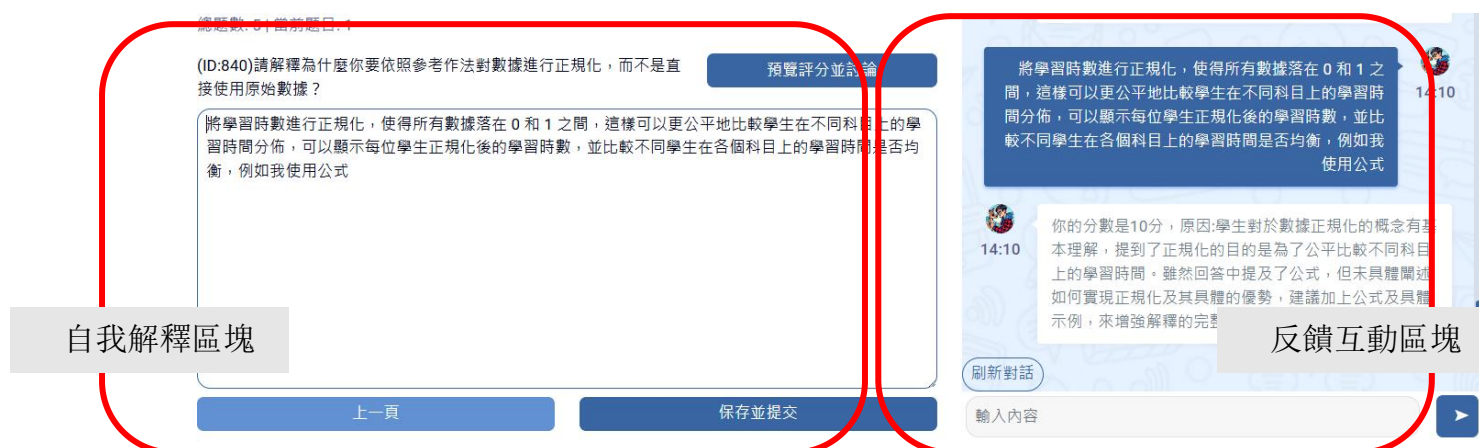


圖 1 系統活動介面圖

表 1 自我解釋評分分級

級別	分數	描述
不可接受	0~3	回答有多處錯誤，表達不清晰，未能準確回答問題，或與問題不相關。
差	4~7	回答部分接近問題所需的內容，但多為表面概括，缺乏深入詳盡的解釋。
尚可	8~11	回答基本符合要求，提供問題所需步驟的基本解釋，達到一般標準水平。
非常好	12~15	大部分回答正確且恰當，但在邏輯推理或表達上還有一定的改進空間。
優秀	16~20	回答不僅正確，而且邏輯清晰，表達流暢，充分解釋了相關細節和概念。

整個系統的環境部署於微軟 Axure 雲端服務提供的 Ubuntu Linux 虛擬機上，以 Nginx 作為反向代理伺服器，並使用 Django 與 MySQL 負責後端邏輯及資料庫管理。透過 Django ORM 工具可有效處理並記錄學生互動過程；同時，後端會與 OpenAI 的 ChatGPT API 串接，將學生的程式碼與解釋傳遞給 ChatGPT，並接收回應後以 AJAX 即時反饋給前端。前端採用 HTML5、CSS3、Bootstrap 等技術實作響應式設計，並透過 jQuery 與 AJAX 實現即時互動機制。

3. 研究方法

3.1. 研究對象

本研究的教材為 16 周的「Python 教育資料探勘實作」課程。課程內容包含從 Python 程式設計到資料探勘實作的教學，包和基礎程式語法、資料分析、機器學習模型等章節。

研究對象為台灣北部一所大學課程的碩博士生，這些學生來自不同的專業背景，包括工程、教育和統計等領域，參與本研究之學生人數約為 40 人。在該大學的工程學院之實體教室進行，並在實驗開始前均簽署了知情同意書。每位學生皆配備電腦進行課堂活動，並且在

完成程式作業後，會與課堂系統進行互動。系統會提供平台讓學生對程式作業進行自我解釋，而機器人可以針對學生的自我解釋進行即時反饋，這樣的互動旨在驗證學生是否真正理解他們所寫的程式，同時也促進學生進行自我解釋和反思，進而達到更深層學習。

3.2. 實驗流程

本研究在實驗準備期確定了課程內容和作業的設計，並與課堂教師討論 Python 程式能力測驗，測驗內容為總分 100 的 20 項選擇題，並參考 Tsai et al. (2021) 的運算思維量表

(Computational Thinking Scale, CTS) 及 Pintrich (1991) 所編制的《學習動機策略量表》

(Motivated Strategies for Learning Questionnaire, MSLQ)，旨在準確反饋學生在實驗前後的相关指標變化。隨後完成了系統架構的設計與開發，該系統包括前端界面、後端伺服器、資料庫架構等，確保能夠有效支援課堂的系統互動與行為數據收集。

在教學實踐期，本研究正式將開發完成的自我解釋系統導入教學中，並根據預定計劃進行實施。第一周的課程主要包括課程介紹、Python 程式能力的前測以及問卷填寫，問卷均採用了五點量表設計，非常同意為 5 分、同意為 4 分、普通為 3 分、不同意為 2 分、非常不同意為 1 分，旨在了解學生的程式設計能力與先備知識。前測部分設計了 20 題選擇題。此外，還對系統進行了詳細說明，並指導學生如何操作該系統，以確保後續使用的順利進行。

從第二周至第十五周，每週課程分為三節，每節包含 50 分鐘的授課時間與下課時間，在每個階段的課程完成後，會針對該階段主題發放回家作業，並在下一次課堂上進行自我解釋活動，自我解釋題目根據學生的作業進行適性化出題。學生在活動中需要對題目進行自我解釋，並可通過與聊天機器人反饋互動來幫助完成解釋過程。

最後一周，重複進行與第一周相同的 Python 程式能力測驗與問卷填寫，旨在反饋學生在教學實踐期中的進步與學習動機的變化以及對系統的感知。此外，還將進行學生訪談，以收集對整個教學過程的反饋和建議。

在資料分析期中，首先收集了來自學生的問卷資料以及他們在系統的使用數據。接著，對這些資料進行了深入分析，以探討學生的系統使用情況是否對他們的學習成效與學習動機產生影響。分析結果將用來驗證本研究的假設，並為後續的教學改進提供依據。

3.3. 資料分析

由於研究數據多為前測與後測的配對資料，且部分數據未滿足正態分布的假設 (Shapiro-Wilk 檢定結果 $p < 0.05$)，因此採用無母數檢定中的 Wilcoxon 符號檢定進行分析。該檢定方法適用於兩組相關數據中位數差異的檢驗，能有效反饋學生在課程介入前後各項能力的提升。

在學習動機與方面，本研究比較了學生在內在目標導向、外在動機導向等面項的前後測表現，反饋課程是否對學生的學習動機產生影響。運算思維能力與 Python 程式設計能力方面採用問卷與測驗方式調查前測與後測數據，並進而分析學生在程式設計基礎知識及資料科學應用能力上的成效差異。

4. 結果與討論

4.1. Python 程式能力的變化

為了檢驗學生在參與學習活動前後 Python 程式能力變化，測驗成績最高為 100 分，最低為 0 分，使用 Wilcoxon 符號檢驗進行分析。表 2 結果顯示，後測成績的中位數 ($M = 90.000$) 顯著高於前測成績 ($M = 65.000$)，中位數差值為 -25.000 ($Z = 5.384$, $p = 0.000$)。這表明學生的 Python 程式能力在參與學習活動後有顯著提升。

4.2. 運算思維的變化

運算思維的問卷均採用了五點量表設計，非常同意為 5 分、同意為 4 分、普通為 3 分、不同意為 2 分、非常不同意為 1 分，後測運算思維的中位數 ($M = 4.250$) 顯著高於前測 ($M = 4.000$)，中位數差值為 -0.250 ($Z = 3.199$, $p = 0.001$)。這一發現支持了 Cansu and Cansu (2019) 及 Grover and Pea (2013) 的觀點，即程式設計教育有助於培養學生的運算思維。本研

究中的聊天機器人引導學生進行自我解釋,促使學生深入理解程式碼的邏輯結構,這與 Bisra et al. (2018)所述的自我解釋策略能有效促進深度學習的結論相一致,生成式 AI 的個性化提問有助於學生將複雜問題分解為更易處理的部分,從而提升整體運算思維能力。

4.3. 學習動機的變化

在學習動機問卷均採用了五點量表設計,非常同意為 5 分、同意為 4 分、普通為 3 分、不同意為 2 分、非常不同意為 1 分,後測學習動機的中位數 ($M = 4.188$) 顯著高於前測學習動機 ($M = 4.000$), 中位數差值為 -0.188 ($Z = 3.141$, $p = 0.002$)。進一步分析內在目標與外在動機的變化,結果顯示後測內在目標的中位數 ($M = 4.250$) 顯著高於前測 ($M = 4.000$), $Z = 3.081$, $p = 0.002$ 。同樣,後測外在動機的中位數 ($M = 4.250$) 也顯著高於前測 ($M = 3.875$), $Z = 2.149$, $p = 0.032$ 。這些結果表明,學生的學習動機在內在與外在層面均有所提升,可能得益於教學設計和學習環境的改善。

5. 結論

本研究開發的自我解釋適性化學伴系統,將自我解釋策略與聊天機器人技術深度結合,學生透過系統所提供的即時反饋與個性化引導,不但能充分認識並修正程式碼錯誤,亦能培養更穩固的運算思維能力。自我解釋和 AI 生成式對話的結合,促使學生於思考程式邏輯時更能聚焦於理解本質,進而提升解決問題的深度和廣度。從實證結果來看,此教學設計對程式設計學習成效與學習動機皆產生顯著助益,也為未來教學技術整合帶來更多發展機會

致謝

本研究感謝國科會專題研究計畫(計畫編號: NSTC113-2628-H-008-001-MY3)與教育部教學實踐研究計畫(計畫編號: PSK1136863)以及中央大學學習科技研究中心支持。

參考文獻

- Alves, N. D. C., Von Wangenheim, C. G., & Hauck, J. C. (2019). Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study. *Informatics in Education*, 18(1), 17. <https://doi.org/10.15388/infedu.2019.02>
- Auvinen, T. (2015). Harmful study habits in online learning environments with automatic assessment. 2015 International Conference on Learning and Teaching in Computing and Engineering, 50-57. <https://doi.org/10.1109/LaTiCE.2015.31>
- Benotti, L., Martnez, M. C., & Schapachnik, F. (2017). A tool for introducing computer science with automatic formative assessment. *IEEE transactions on learning technologies*, 11(2), 179-192. <https://doi.org/10.1109/TLT.2017.2682084>
- Bisra, K., Liu, Q., Nesbit, J. C., Salimi, F., & Winne, P. H. (2018). Inducing self-explanation: A meta-analysis. *Educational Psychology Review*, 30, 703-725. <https://doi.org/10.1007/s10648-018-9434-x>
- Cansu, F. K., & Cansu, S. K. (2019). An overview of computational thinking. *International Journal of Computer Science Education in Schools*, 3(1), 17-30. <https://doi.org/10.21585/ijcses.v3i1.53>
- Chi, M. T. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in cognitive science*, 1(1), 73-105. <https://doi.org/10.1111/j.1756-8765.2008.01005.x>
- Chi, M. T. (2013). Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. In *Advances in instructional Psychology, Volume 5* (pp. 161-238). Routledge. <https://doi.org/https://doi.org/10.1111/j.1756-8765.2008.01005.x>

- Chi, M. T. (2018). Learning from examples via self-explanations. In *Knowing, learning, and instruction* (pp. 251-282). Routledge.
<https://doi.org/https://doi.org/10.4324/9781315044408-8>
- Chiu, T. K. (2024). A classification tool to foster self-regulated learning with generative artificial intelligence by applying self-determination theory: a case of ChatGPT. *Educational Technology Research and Development*, 72(4), 2401-2416.
- Comb  fis, S., & Schils, A. (2016). Automatic programming error class identification with code plagiarism-based clustering. *Proceedings of the 2nd International Code Hunt Workshop on Educational Software Engineering*, 1-6. <https://doi.org/10.1145/2993270.2993271>
- Cook, C. R., Kilgus, S. P., & Burns, M. K. (2018). Advancing the science and practice of precision education to enhance student outcomes. *Journal of School Psychology*, 66, 4-10.
<https://doi.org/10.1016/j.jsp.2017.11.004>
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer science education: Perspectives on teaching and learning in school*, 19(1), 19-38.
<https://doi.org/10.5040/9781350057142.ch-003>
- Hao, Q., Smith IV, D. H., Ding, L., Ko, A., Ottaway, C., Wilson, J., Arakawa, K. H., Turcan, A., Poehlman, T., & Greer, T. (2022). Towards understanding the effective design of automated formative feedback for programming assignments. *Computer Science Education*, 32(1), 105-127. <https://doi.org/10.1080/08993408.2020.1860408>
- Heintz, F., Mannila, L., & F  rnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K–12 education. *Teoksessa 2016 IEEE frontiers in education conference (FIE)*. <https://doi.org/10.1109/FIE.2016.7757410>
- Hwang, G.-J., Sung, H.-Y., Chang, S.-C., & Huang, X.-C. (2020). A fuzzy expert system-based adaptive learning approach to improving students' learning performances by considering affective and cognitive factors. *Computers and Education: Artificial Intelligence*, 1, 100003.
<https://doi.org/10.1016/j.caeai.2020.100003>
- Kapur, M. (2016). Examining productive failure, productive success, unproductive failure, and unproductive success in learning. *Educational Psychologist*, 51(2), 289-299.
<https://doi.org/10.1080/00461520.2016.1155457>
- Keuning, H., Jeurig, J., & Heeren, B. (2018). A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education (TOCE)*, 19(1), 1-43. <https://doi.org/10.1145/3231711>
- Koedinger, K. R., McLaughlin, E. A., & Heffernan, N. T. (2010). A quasi-experimental evaluation of an on-line formative assessment and tutoring system. *Journal of Educational Computing Research*, 43(4), 489-510. <https://doi.org/10.2190/EC.43.4.d>
- Lalwani, T., Bhalotia, S., Pal, A., Rathod, V., & Bisen, S. (2018). Implementation of a Chatbot System using AI and NLP. *International Journal of Innovative Research in Computer Science & Technology (IJRCST)* Volume-6, Issue-3.
<https://dx.doi.org/10.2139/ssrn.3531782>
- Lehtinen, T., Santos, A. L., & Sorva, J. (2021). Let's ask students about their programs, automatically. *2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*, 467-475. <https://doi.org/10.1109/ICPC52881.2021.00054>
- Lehtinen, T., Sepp  l  , O., & Korhonen, A. (2023). Automated Questions about Learners' Own Code Help to Detect Fragile Prerequisite Knowledge. *Proceedings of the 2023 Conference on*

Innovation and Technology in Computer Science Education V. 1, 505-511.

<https://doi.org/10.1145/3587102.3588787>

- Nakamoto, R., Flanagan, B., Yamauchi, T., Dai, Y., Takami, K., & Ogata, H. (2023). Enhancing Automated Scoring of Math Self-Explanation Quality Using LLM-Generated Datasets: A Semi-Supervised Approach. *Computers*, 12(11), 217.
<https://doi.org/10.3390/computers12110217>
- Ogegbo, A. A., & Ramnarain, U. (2021). A systematic review of computational thinking in science classrooms. *Studies in Science Education*, 58(2), 203-230.
<https://doi.org/10.21585/ijcses.v3i1.53>
- Paiva, J. C., Leal, J. P., & Figueira, Á. (2022). Automated assessment in computer science education: A state-of-the-art review. *ACM Transactions on Computing Education (TOCE)*, 22(3), 1-40. <https://doi.org/10.1145/3513140>
- Pham, X. L., Pham, T., Nguyen, Q. M., Nguyen, T. H., & Cao, T. T. H. (2018). Chatbot as an intelligent personal assistant for mobile language learning. *Proceedings of the 2018 2nd International Conference on Education and E-Learning*, 16-21.
<https://doi.org/10.1145/3291078.3291115>
- Pintrich, P. (1991). A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ). National Center for Research to Improve Postsecondary Teaching and Learning.
- Pirolli, P., & Recker, M. (1994). Learning strategies and transfer in the domain of programming. *Cognition and instruction*, 12(3), 235-275. https://doi.org/10.1207/s1532690xci1203_2
- Popovici, M.-D. (2023). ChatGPT in the Classroom. Exploring Its Potential and Limitations in a Functional Programming Course. *International Journal of Human-Computer Interaction*, 1-12. <https://doi.org/10.1080/10447318.2023.2269006>
- Shute, V. J. (2008). Focus on formative feedback. *Review of educational research*, 78(1), 153-189.
<https://doi.org/10.3102/0034654307313795>
- Slavuj, V., Meštrović, A., & Kovačić, B. (2017). Adaptivity in educational systems for language learning: a review. *Computer Assisted Language Learning*, 30(1-2), 64-90.
<https://doi.org/10.1080/09588221.2016.1242502>
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798.
<https://doi.org/10.1016/j.compedu.2019.103798>
- Tsai, M.-J., Liang, J.-C., & Hsu, C.-Y. (2021). The computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, 59(4), 579-602.
<https://doi.org/10.3390/computers12110217>
- Wylie, R., & Chi, M. T. (2014). 17 the self-explanation principle in multimedia learning. *The Cambridge handbook of multimedia learning*, 413-432.
<https://doi.org/10.1017/CBO9780511816819.018>
- Yin, J., Goh, T.-T., Yang, B., & Xiaobin, Y. (2021). Conversation technology with micro-learning: The impact of chatbot-based learning on students' learning motivation and performance. *Journal of Educational Computing Research*, 59(1), 154-177.
<https://doi.org/10.1177/0735633120952067>
- Yu, F.-Y. (2022). An online learning system supporting student-generated explanations for questions: design, development, and pedagogical potential. *Interactive Learning Environments*, 30(5), 782-802. <https://doi.org/10.1080/10494820.2019.1686398>

Zen, K., Iskandar, D. F. A., & Linang, O. (2011). Using Latent Semantic Analysis for automated grading programming assignments. 2011 International Conference on Semantic Technology and Information Retrieval, 82-88. <https://doi.org/10.1109/STAIR.2011.5995769>