

同伴评价的协作编程策略对不同计算思维水平学生编程学习成绩及问题解决倾向的影响研究

A study of the effects of peer-assessed collaborative programming strategies on programming academic performance and problem-solving tendencies of students with different levels of computational thinking

荣健康¹, 刘海英¹, 曾秋蓉², 李伟^{1*}, 刘城烨³, 黄信艺⁴

¹温州大学 STEM 教育研究中心

²中华大学资讯工程学系

³香港中文大学课程与教学系

⁴温州市瓯海区瞿溪第一小学

*liweiwzu@wzu.edu.cn

【摘要】协作编程旨在通过合作学习提升学生的编程成绩，但在协作过程中可能出现搭便车现象。为此，本研究提出了一种结合同伴评价的协作编程学习方法，探讨其对不同计算思维水平学生的学习成绩和问题解决倾向的影响。实验组采用结合同伴评价的协作编程学习方法，对照组采用常规协作编程学习方法。研究结果表明，不同计算思维水平的学生与学习方法在学习成绩和问题解决倾向方面均存在显著的交互作用，其中结合同伴评价的协作编程学习方法对高计算思维水平学生更具显著促进作用。

【关键词】同伴评价；协作编程；计算思维

Abstract : Collaborative programming aims to improve students' programming performance through cooperative learning, but free-riding may occur during the collaborative process. For this reason, this study proposes a collaborative programming learning method incorporating peer assessment to explore its effects on the academic performance and problem solving tendencies of students with different levels of computational thinking. The experimental group used the collaborative programming learning method incorporating peer assessment, and the control group used the conventional collaborative programming learning method. The results of the study show that there are significant interactions between students with different levels of computational thinking and the learning method in terms of academic performance and problem-solving tendency, in which the collaborative programming learning method with peer assessment has a more significant effect on students with high levels of computational thinking.

Keywords: Peer evaluation, collaborative programming, computational thinking

1. 引言

随着信息技术的发展，编程在 K-12 教育中受到越来越多的关注，被认为是数字时代培养劳动力人才的关键(Sabarinath & Quek, 2020)。学习编程不仅可以使学生获得知识，而且在他们的高阶思维表现方面也有好处(Li et al., 2023; Strawhacker & Bers, 2019)。然而，对于大多数学生来说，学习计算机编程是一项具有挑战性的任务(Grover et al., 2017)。编程作为一种逻辑和解决问题的活动，需要复杂的思维技能，学生往往很难独立完成(Shadiev et al., 2014)。而协作编程作为一种教学方式，可以让学生在编程过程中通过交流讨论编程程序和任务来解决编程中遇到的问题(吴忭 & 王戈, 2019)。

然而，在协作编程过程中，一些学生可能会努力解决问题，而另一些学生可能不愿意主动参与小组项目(Chang & Chen, 2009)。这对勤奋的学生是不公平的，并导致学生对小组作业感到不满(Freeman & McKenzie, 2002)。同时教师很难密切监控小组中每个学生的努力，因此，教师无法评估每个学生的贡献(Boud & Holmes, 1981)。如何在小组中评估学生的表现是一个需要解决的重要问题。当面对大量学生时，同伴评价可以解决教师评估学生作业的能力有限的问题，并及时提供详细的评价(Comer et al., 2014)。同伴评价是学生评估同伴的表现或成就的过程(Topping et al., 2010)。该方法是学生或同伴群体的定性评论，并通过鼓励反思和评估同伴的表现，从而有益于学生的学习(Harris & Brown, 2013)。同伴评价作为一种教学策略，可以使学生增强自己的反思能力，另一方面学生可以认识到自己的缺点(Ion et al., 2018)。

不同计算思维水平的学生可能在实际表现上有差异。具有较高计算思维水平的学生往往能够更有效地分析问题并提出创新性的解决方案，他们可能表现出更高的学习成绩，因为他们能够更快速地理解和实施复杂的编程任务(Jun et al., 2017)。计算思维水平较低的学生可能更倾向于依赖传统的、较为机械的解决方法，需要更多的指导和支持(Zhang et al., 2024)。尽管越来越多的研究探索了K-12教育中学生的计算思维和相关变量，但在同一个学习策略的影响下，不同计算思维水平的学生可能会获得不同程度的好处。同时关于同伴评价的研究主要集中在同伴评价的有效性等方面，而与计算思维的交互影响却鲜少研究。因此，本研究旨在探究计算思维倾向与同伴评价的交互对学生协作编程的学习成绩和问题解决倾向的影响。

2.方法

为了评估同伴评价的协作编程学习方法的有效性，我们在台湾一所国中开展了准实验研究，探究该方法对不同计算思维水平学生学习成绩和问题解决倾向的影响。

2.1. 试验对象

本研究以台湾156名初中三年级的学生为研究对象，平均年龄为15岁，他们来自12个不同的班级。其中，六个班的79名学生被随机分配为实验组，采用结合同伴评价的协作编程方法进行学习。另外六个班的77名学生被分配为对照组，采用常规的协作编程方法进行学习。

2.2. 实验设计

本实验共持续了10周时间，每周1节课，每节课45分钟。本研究的学习系统参考了Li et al. (2024)学者的研究。该系统包括协作学习模块、同伴评价模块。协作学习模块包含参与讨论机制、合作规则流程图机制、合作编写代码机制。同伴评价模块包含自评机制、组内互评机制和组间互评机制（评价标准见图1）。图2显示了实验过程。在第1周，实验组和对照组的学生用30分钟参加学习成绩前测，计算思维倾向和问题解决倾向的前问卷，从第2周开始，对照组的学生通过协作学习模块，在聊天室讨论学习任务，合作完成流程图的绘制和代码的编写。实验组的学生另外使用同伴评价模块按照评价标准进行自我评价、组内互评和组间互评。在第10周，所有学生完成学习成绩后测，计算思维倾向和问题解决倾向的后问卷。

評量尺規			非常好	好	一般	較差	非常差
評價維度	配分	評價標準描述					
正確性	40	運行流暢，結果正確，沒有語法錯誤。能準確的繪出解決問題的步驟。	5	4	3	2	1
邏輯性	40	結構清晰，邏輯正確，可有條理地呈現出解決問題的過程。	5	4	3	2	1
可讀性	20	命名及符號符合規範，容易理解。	5	4	3	2	1

图1 评价维度及评价标准描述

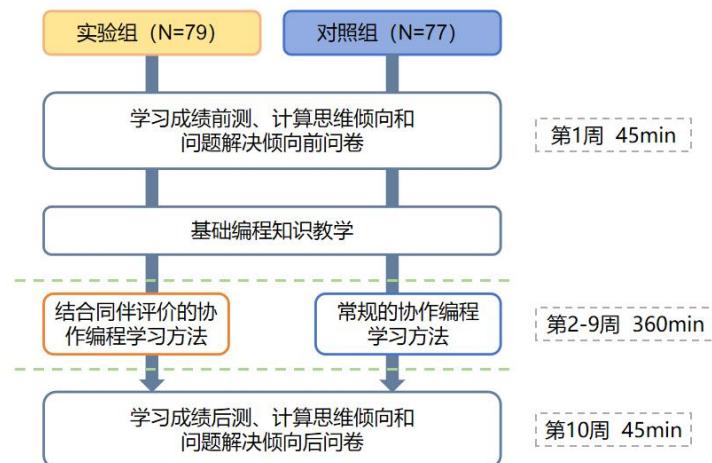


图 2 学习活动实验流程图

2.3. 测量工具

本研究使用的测量工具包括学习成绩前后测试题、计算思维倾向和问题解决倾向前后问卷。

学习成绩前测试题的目的是检验参与实验的学生关于 python 编程的先前知识水平，后测试题是为了考察学生在整个协作编程过程中所获得的学习收益。前、后测试题均为 10 道选择题，每道题 10 分，总分为 100 分。两份测试题都是由具有丰富教学经验的信息技术教师进行设计和修改。

计算思维倾向问卷改编自 Hwang et al. (2020) 开发的方法。该问卷采用 5 点李克特量表 (1=完全不同意；5=完全同意)。

问题解决倾向问卷改编自 Lai and Hwang (2014) 开发的方法。该问卷采用 5 点李克特量表 (1=完全不同意；5=完全同意)。

3. 实验结果

本研究采用了一个结合同伴评价的协作编程学习系统来支持学生的编程学习。分析采用了 two-way ANCOVA 方法。参考研究人员(Chen & Lin, 2014; Kelley, 1939)，高计算思维水平组为前测前 33% 的学生，低计算思维水平组为前测后 33%。表 1 为描述性统计分析结果。

表 1 学习成绩描述性统计表

组别	计算思维高低	平均值	标准差	个数
对照组	高	58.89	22.14	36
	低	65.12	15.02	41
实验组	高	68.95	17.98	38
	低	63.66	21.42	41

3.1. 学习成绩

为了探究同伴评价对不同计算思维倾向学生学习成绩的影响，本研究采用了双因素协方差分析方法 (two-way ANCOVA)。结果如表 2 所示，计算思维水平和学习方法对学生的学习成绩的影响存在显著的交互作用 ($F = 4.22, p = 0.043$)。

表 2 学习成绩双因素协方差结果

变量	离均差平方和	自由度	均方	F	η^2
学习成绩前测	2634.61	1	2634.61	7.39***	0.05
学习方法	737.36	1	737.36	2.07	0.01
计算思维倾向	3.68	1	3.68	0.01	0
学习方法*计算思维倾向	1506.03	1	1506.03	4.22*	0.03

误差	53854.45	151	356.65
----	----------	-----	--------

* $p < 0.05$, *** $p < 0.001$

为了探究不同计算思维水平对采用不同学习方法学习的学生学习成绩的影响,因此进一步采用了简单主效应分析方法。结果如表3所示,实验组中不同计算思维水平的学生的学习成绩不具有显著差异($F = 1.94, p = 0.13$)。同样,对照组中不同计算思维水平的学生的学习成绩也不具有显著差异($F = 2.30, p = 0.17$)。因此,结果显示在协作编程中采用同伴评价时,不同计算思维水平学生的学习成绩不具有显著差异。

表3 计算思维倾向对学生学习成绩的简单主效应分析结果

变量	离均差平方和	自由度	均方	F	η^2
实验组	组内 690.53	1	690.53	1.94	0.01
	组间 53854.45	151	356.65		
	共计 54544.98	152			
对照组	组内 819.68	1	819.68	2.30	0.02
	组间 53854.45	151	356.65		
	共计 54674.13	152			

* $p < 0.05$

表4显示了学习方法对不同计算思维水平学生学习成绩影响的简单主效应分析结果。结果显示,高计算思维水平的学生在学习过程中采用不同学习方法,其学习成绩具有显著差异($F = 5.80, p = 0.01$)。而低计算思维水平的学生在学习过程中采用不同学习方法,不具有显著差异($F = 0.2, p = 0.65$)。结果表明,采用不同的学习方法对高计算思维水平的学生比低计算思维水平的学生的学习成绩影响大。

表4 学习方法对学生学习成绩的简单主效应分析结果

变量	离均差平方和	自由度	均方	F	η^2
高	组内 2068.64	1	2068.64	5.80*	0.04
	组间 53854.45	151	356.65		
	共计 55923.09	152			
低	组内 72.28	1	72.28	0.2	
	组间 53854.45	151	356.65		
	共计 53926.73	152			

* $p < 0.05$

图3显示了不同的学习方法和不同计算思维水平与学生学习成绩之间的交互作用。结果表明,使用同伴评价的协作编程方法学习的学生学习成绩显著优于使用常规协作编程方法学习的学生。此外,高计算思维水平中的实验组的学习成绩要优于对照组,而低计算思维水平的学生,两组之间学习成绩差异不显著。

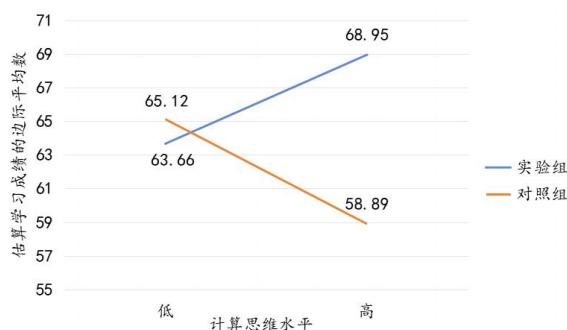


图3 学习方法与计算思维水平对学习成绩的交互作用

3.2. 问题解决倾向

为了探究同伴评价对不同计算思维水平学生问题解决倾向的影响,本研究采用了双因素协

方差分析方法 (two-way ANCOVA)。结果如表 5 所示, 计算思维水平和学习方法对学生的问题解决倾向的影响存在显著的交互作用($F = 4.45, p = 0.036$)。

表 5 问题解决倾向双因素协方差结果

变量	离均差平方和	自由度	均方	F	η^2
问题解决倾向前	2.18	1	2.18	10.49***	0.05
学习方法	4.04	1	4.04	19.45***	0.10
计算思维倾向	2.52	1	2.52	12.15***	0.06
学习方法*计算思维倾向	0.92	1	0.92	4.45*	0.02
误差	31.33	151	0.21		

* $p < 0.05$, *** $p < 0.001$

为了探究不同计算思维水平学生对采用不同学习方法学习的学生的问题解决倾向的影响, 因此进一步采用了简单主效应分析方法。结果如表 6 所示, 实验组中不同计算思维水平的学生问题解决倾向具有显著差异 ($F = 16.50, p = 0.001$)。另一方面, 对照组中不同计算思维水平的学生问题解决倾向不具有显著差异 ($F = 0.99, p = 0.32$)。因此, 结果显示在协作编程中采用同伴评价时, 不同计算思维水平学生的问题解决倾向具有显著差异。

表 6 计算思维倾向对学生问题解决倾向的简单主效应分析结果

变量	离均差平方和	自由度	均方	F	η^2
实验组	组内 3.42	1	3.42	16.50***	0.1
	组间 31.33	151	31.33		
	共计 33.75	152			
对照组	组内 0.2	1	0.2	0.99	0.01
	组间 31.33	151	31.33		
	共计 31.53	152			

* $p < 0.05$, *** $p < 0.001$

表 7 显示了学习方法对不同计算思维水平学生问题解决倾向影响的简单主效应分析结果。结果发现, 高计算思维水平的学生, 采用不同学习方法进行学习, 其问题解决倾向具有显著差异($F = 20.13, p = 0.001$)。而低计算思维水平的学生采用不同的学习方法学习, 问题解决倾向不存在显著差异($F = 2.72, p = 0.10$)。结果表明, 采用不同的学习方法对高计算思维水平的学生比低计算思维水平的学生的问题解决倾向影响大。

表 7 学习方法对学生问题解决倾向的简单主效应分析结果

变量	离均差平方和	自由度	均方	F	η^2
高	组内 4.18	1	4.18	20.13***	0.12
	组间 31.33	151			
	共计 35.51	152			
低	组内 0.57	1	0.57	2.72	0.02
	组间 31.33	151			
	共计 31.90	152			

*** $p < 0.001$

图 4 显示了不同的学习方法和不同计算思维水平对学生的问题解决倾向的交互作用。结果表明, 使用同伴评价的协作编程方法学习的学生问题解决倾向显著优于使用常规协作编程方法学习的学生。此外, 高计算思维水平中的实验组的问题解决倾向要显著优于对照组, 而低计算思维水平中的实验组和对照组的学生, 问题解决倾向无显著差异。

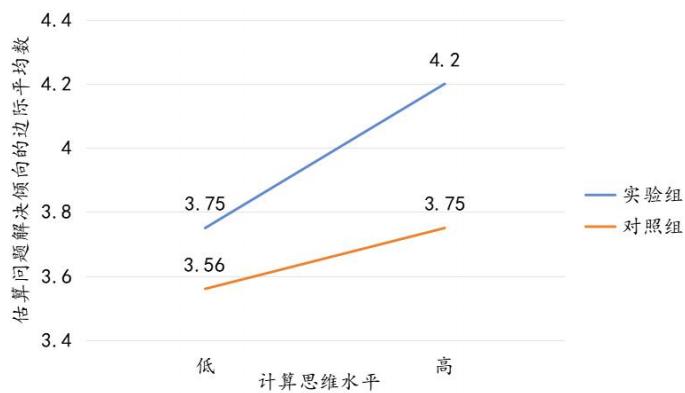


图 4 学习方法与计算思维水平对问题解决倾向的交互作用

4.讨论与结论

本研究探讨了同伴评价策略对不同计算思维水平学生学习成绩及问题解决倾向的影响，结果显示采用同伴评价的协作编程方法对不同计算思维水平学生的学习成绩和问题解决倾向均存在交互，且能够显著提高高计算思维水平学生的学习成绩和问题解决倾向。

4.1. 同伴评价的协作编程学习方法如何影响不同计算思维水平学生的学习成绩？

在学习成绩方面，同伴评价的协作编程学习方法对不同计算思维水平的学生存在显著的交互作用，尤其能显著提高高计算思维水平学生的学习成绩，而对于低计算思维水平学生的学习成绩提升效果则相对有限。高计算思维水平学生所具备的抽象思维、算法处理和系统思考能力，使他们能够轻松理解编程语言的抽象概念(Grover & Pea, 2013)。同伴评价环节要求学生评价同伴的作品，这使他们能够在评价过程中进一步深化对编程知识的理解，从而提高了他们的积极性和学习效果(Yin et al., 2022)。

然而，对于低计算思维水平的学生而言，他们在理解编程概念、构建算法逻辑等方面存在较大困难(Kalelioğlu, 2015)。同伴评价过程中需要分析和评价的作品对他们来说可能增加了额外的认知负担，而难以像高计算思维水平学生那样从中获得有效的学习助力。

4.2. 同伴评价的协作编程学习方法如何影响不同计算思维水平学生的问题解决倾向？

在问题解决倾向方面，同伴评价的协作编程学习方法对不同计算思维水平的学生存在交互作用，尤其能显著提高高计算思维水平学生的问题解决倾向。高计算思维水平的学生具有更强的抽象和推理能力(Sun et al., 2022)，他们会更有逻辑的思考以及寻找最合适的方法解决编程中遇到的问题(Yadav et al., 2014)。同伴评价环节需要学生分析同伴的作品并提出适当的解决方案(Voon et al., 2022)，高计算思维水平的学生通过参与同伴作品的分析和评价，能够有效激发其逻辑推理能力和问题解决能力(刘敏 & 汪琼, 2022)。

然而，对于低计算思维水平的学生而言，他们的逻辑分析能力不足，在解决编程问题时会感到困难(Rojas-López & García-Peñalvo, 2019)。同伴评价需要他们能够识别同伴程序中的问题和给出相应的建议，这对他们来说也是一种挑战。他们可能只关注到同伴程序存在的浅层问题(吴忭 & 王戈, 2019)，而不能给出有效的建设性意见，因此不利于问题解决能力的提高。

本研究结合同伴评价的协作编程学习方法，结果表明，对学生的学习成绩和问题解决倾向均有不同程度影响，尤其是对高计算思维水平的学生有显著积极影响，但对低计算思维水平学生的影响较小。当前研究的结果也存在着一定的局限性。首先，同伴评价的协作编程学习方法对高计算思维水平学生的积极作用高于低计算思维水平的学生，未来可以针对不同计算思维水平的学生在系统中设计不同学习脚手架支持。其次，未量化互评参与频次，代码修改次数等动态学习行为，未来可以增加这方面的研究。最后，本研究只关注了同伴评价的协作编程学习方法对不同计算思维水平学生的学习成绩和问题解决倾向，未来可以关注更多方面。

参考文献

- 刘敏, & 汪琼. (2022). 结对编程:中小学编程教育的首选教学组织形式. *现代教育技术*, 32(3), 102-109. <https://doi.org/10.3969/j.issn.1009-8097.2022.03.011>
- 吴忭, & 王戈. (2019). 协作编程中的计算思维发展轨迹研究——基于量化民族志的分析方法. *现代远程教育研究*(02), 76-84+94.
<https://link.cnki.net/urlid/51.1580.G4.20190322.1631.006>
- Boud, D. J., & Holmes, W. H. (1981). Self and peer marking in an undergraduate engineering course. *IEEE Transactions on Education*, 24(4), 267-274.
- Chang, T.-Y., & Chen, Y.-T. (2009). Cooperative learning in E-learning: A peer assessment of student-centered using consistent fuzzy preference. *Expert Systems with Applications*, 36(4), 8342-8349. <https://doi.org/10.1016/j.eswa.2008.10.050>
- Chen, C.-M., & Lin, S.-T. (2014). Assessing effects of information architecture of digital libraries on supporting E-learning: A case study on the Digital Library of Nature & Culture. *Computers & Education*, 75, 92-102. <https://doi.org/10.1016/j.compedu.2014.02.006>
- Comer, D. K., Clark, C. R., & Canelas, D. A. (2014). Writing to learn and learning to write across the disciplines: Peer-to-peer writing in introductory-level MOOCs. *International Review of Research in Open and Distributed Learning*, 15(5), 26-82.
- Freeman, M., & McKenzie, J. (2002). SPARK, a confidential web - based template for self and peer assessment of student teamwork: benefits of evaluating across different subjects. *British journal of educational technology*, 33(5), 551-569.
- Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A Framework for Using Hypothesis-Driven Approaches to Support Data-Driven Learning Analytics in Measuring Computational Thinking in Block-Based Programming Environments. *ACM Transactions on Computing Education*, 17(3), 1-25. <https://doi.org/10.1145/3105910>
- 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>
- Harris, L. R., & Brown, G. T. (2013). Opportunities and obstacles to consider when using peer-and self-assessment to improve student learning: Case studies into teachers' implementation. *Teaching and Teacher Education*, 36, 101-111. <https://doi.org/10.1016/j.tate.2013.07.008>
- Hwang, G.-J., Li, K.-C., & Lai, C.-L. (2020). Trends and strategies for conducting effective STEM research and applications: A mobile and ubiquitous learning perspective. *International Journal of Mobile Learning and Organisation*, 14(2), 161-183. <https://doi.org/10.1504/IJMLO.2020.106166>
- Ion, G., Sánchez Martí, A., & Agud Morell, I. (2018). Giving or receiving feedback: which is more beneficial to students' learning? *Assessment & Evaluation in Higher Education*, 44(1), 124-138. <https://doi.org/10.1080/02602938.2018.1484881>
- Jun, S., Han, S., & Kim, S. (2017). Effect of design-based learning on improving computational thinking. *Behaviour & Information Technology*, 36(1), 43-53. <https://doi.org/10.1080/0144929X.2016.1188415>
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in human behavior*, 52, 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>
- Kelley, T. L. (1939). The selection of upper and lower groups for the validation of test items. *Journal of educational psychology*, 30(1), 17. <https://doi.org/10.1037/h0057123>

- Lai, C. L., & Hwang, G. J. (2014). Effects of mobile learning time on students' conception of collaboration, communication, complex problem - solving, meta - cognitive awareness and creativity. *International Journal of Mobile Learning and Organisation*, 8(3-4), 276-291. <https://doi.org/10.1504/IJMLO.2014.067029>
- Li, W., Huang, J.-Y., Liu, C.-Y., Tseng, J. C., & Wang, S.-P. (2023). A study on the relationship between student learning engagements and higher-order thinking skills in programming learning. *Thinking Skills and Creativity*, 49, 101369. <https://doi.org/10.1016/j.tsc.2023.101369>
- Li, W., Liu, C. Y., & Tseng, J. C. (2024). Development of a metacognitive regulation - based collaborative programming system and its effects on students' learning achievements, computational thinking tendency and group metacognition. *British Journal of Educational Technology*, 55(1), 318-339. <https://doi.org/10.1111/bjet.13358>
- Rojas-López, A., & García-Peñalvo, F. J. (2019). Initial learning scenarios based on the computational thinking evaluation for the course Programming fundamentals at INACAP. Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality. <https://doi.org/10.1145/3362789.3362802>
- Sabarirath, R., & Quek, C. L. G. (2020). A case study investigating programming students' peer review of codes and their perceptions of the online learning environment. *Education and Information Technologies*, 25(5), 3553-3575.
- Shadiev, R., Hwang, W.-Y., Yeh, S.-C., Yang, S. J. H., Wang, J.-L., Han, L., & Hsu, G.-L. (2014). Effects of Unidirectional vs. Reciprocal Teaching Strategies on Web-Based Computer Programming Learning. *Journal of Educational Computing Research*, 50(1), 67-95. <https://doi.org/10.2190/EC.50.1.d>
- Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational technology research and development*, 67, 541-575.
- Sun, L., Hu, L., & Zhou, D. (2022). Programming attitudes predict computational thinking: Analysis of differences in gender and programming experience. *Computers & Education*, 181, 104457. <https://doi.org/10.1016/j.compedu.2022.104457>
- Topping, K. J., Smith, E. F., Swanson, I., & Elliot, A. (2010). Formative Peer Assessment of Academic Writing Between Postgraduate Students. *Assessment & Evaluation in Higher Education*, 25(2), 149-169. <https://doi.org/10.1080/713611428>
- Voon, X. P., Wong, S. L., Wong, L. H., Khambari, M. N. M., & Abdullah, S. I. S. S. (2022). Role of peer assessment in facilitating computational thinking among pre-service teachers.
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1-16. <https://doi.org/10.1145/2576872>
- Yin, S., Chen, F., & Chang, H. (2022). Assessment as Learning: How Does Peer Assessment Function in Students' Learning? *Frontiers in Psychology*, 13, 912568. <https://doi.org/10.3389/fpsyg.2022.912568>
- Zhang, W., Guan, Y., & Hu, Z. (2024). The efficacy of project-based learning in enhancing computational thinking among students: A meta-analysis of 31 experiments and quasi-experiments. *Education and Information Technologies*, 1-33.