International Journal of Evaluation and Research in Education (IJERE)
Vol. 15, No. 5, October 2026, pp. 4114~4121
ISSN: 2252-8822, DOI: 10.11591/ijere.v15i5.39518

The relationship between creativity and programming skills among lower secondary school students

Rattanawadee Muensripoom[1], Mana Sopa[2]
[1]Innovation in Computer Education, Udon Thani Rajabhat University, Udon Thani, Thailand
[2]Faculty of Education, Udon Thani Rajabhat University, Udon Thani, Thailand

Article Info

Article history:

Received Mar 15, 2026
Revised Aug 5, 2026
Accepted Aug 29, 2026

Keywords:

Computational thinking
Creativity
Practical skills
Programming skills
Synectics model

ABSTRACT

This study addresses the limited emphasis on creativity in lower secondary programming classrooms, where instruction often relies on imitation-based coding practices. To address this issue, the Synectics instructional model was applied to enhance students' creativity and programming skills. A quasi-experimental design was conducted with 60 students divided into experimental and control groups. Data were collected using creativity and programming skill assessments and analyzed using independent samples t-test and Pearson correlation. The results showed that the experimental group achieved significantly higher scores in creativity ($M=9.91$, $SD=0.95$) and programming skills ($M=34.67$, $SD=1.83$) than the control group ($p<0.001$). A strong positive correlation was found between creativity and programming skills ($r=0.710$, $p<0.001$). These findings indicate that creativity is a key factor in programming performance and that the Synectics model is effective in enhancing both creative thinking and technical skills. The study provides practical implications for improving programming education in 21st-century classrooms.

Corresponding Author:

Rattanawadee Muensripoom
Innovation in Computer Education, Udon Thani Rajabhat University
64 Thahan Road, Mak Khaeng Subdistrict, Mueang Udon Thani District, Udon Thani 41000, Thailand
Email: 65120612106@udru.ac.th

1. INTRODUCTION

In the 21st century, educational paradigms emphasize developing essential skills for digital economies. Creativity, innovation, and computational thinking are now core competencies required for sustainable development and global competitiveness. Regionally and internationally, this shifting paradigm is heavily driven by comprehensive educational policy frameworks and foundational core curricula, which systematically prioritize learners' creativity, critical thinking, and technological competence to cultivate a resilient future workforce [1], [2].

Creativity enables individuals to generate novel ideas and solve problems effectively. Torrance [3] defined creativity through fluency, flexibility, originality, and elaboration, while Guilford [4] connected it to divergent thinking. In computer education, programming skills represent practical competencies integrating cognitive and psychomotor domains. Simpson [5] psychomotor taxonomy outlines that competence is developed through structured practice and progressive acquisition rather than rote memorization.

To promote creative thinking, Gordon [6] developed the Synectics model, which uses analogical processes, including direct analogy, personal analogy, and compressed conflict, to connect unfamiliar concepts with familiar experiences. This model is widely recognized for stimulating creativity and problem-solving [7]. Scholars suggest that creativity can be systematically nurtured. Beghetto and Kaufman [8] argued that classroom environments encouraging exploration foster creativity, while Sawyer [9] highlighted that collaborative thinking supports deeper innovation. This aligns with Sweller [10] instructional design principles regarding creativity and Wade [11] analytical thinking strategies. Furthermore, Davies [12] emphasized that structured instructional processes influence learning outcomes, and Joyce *et al.* [13] noted that instructional models systematically improve cognitive and practical skills. Furthermore, establishing standardized benchmarks for educational outcomes requires robust systematic frameworks, as emphasized by Walbesser [14], whose evaluation models clarify how behavioral objectives and structured curriculum designs directly dictate measurable learning achievements.

Developing practical skills requires continuous engagement, as mapped by Harrow [15] psychomotor stages. In computer education, these skills grow through active problem-solving. Westcott and Smith [16] suggested that creative teaching approaches improve conceptual understanding in technical subjects like mathematics. Thus, embedding Synectics into programming may enhance both creative and technical competence. Previous research proves Synectics' efficacy in creative domains, including dance production [17], literacy skills [18], and creative writing [19]. Recent studies also emphasize its role in technical areas; Pease *et al.* [20] noted that computational creativity drives scientific and mathematical discovery. Other global frameworks support integrating creativity into computational tasks to improve problem-solving [21]–[23].

Despite its value, limited research applies Synectics to lower secondary programming instruction. Many classrooms still rely on imitation-based instruction, where students follow step-by-step coding without independent problem-solving [24], [25]. Consequently, students master programming syntax but struggle to design original solutions or apply divergent thinking [26]. This creates a critical gap where creativity is neglected alongside technical knowledge [27]. While educational policies advocate for innovation [28], actual classroom practice remains disconnected [25], [29].

Applying Synectics as a formal model to simultaneously develop creative cognition and technical programming proficiency in lower secondary students represents a novel approach. This study addresses this gap by investigating if a creative-thinking intervention can directly enhance a technical, practical skill. It makes three contributions: applying Synectics to computer programming, providing empirical evidence on the creativity-programming relationship, and focusing on a critical developmental stage in secondary education. Thus, this study aims to: i) investigate the effect of the Synectics model on students' creativity and programming skills, and ii) examine the relationship between creativity and programming performance. Based on the research objectives, the following research questions were formulated:

- Does the Synectics instructional model significantly improve students' creativity and programming skills compared to conventional instruction? (RQ1)
- Is there a significant relationship between students' creativity and their programming skills? (RQ2)

2. METHOD

2.1. Research design

This study employed a static-group comparison design, which is a type of quasi-experimental, post-test-only control group design. The purpose of this design is to compare the outcomes between a group that receives the experimental intervention and a group that does not, in order to determine the effect of the treatment. Table 1 presents the research design used in this study, illustrating the static-group comparison structure, including the experimental and control groups, the applied treatment, and the post-test measurements.

Table 1. Research design static group comparison design

Group	Treatment (X)	Post-test measurement
Experimental	X	O_1
Control	-	O_2

X represents the instructional intervention using the Synectics model.
O_1 represents the post-test results of the experimental group, and
O_2 represents the post-test results of the control group.

2.2. Population and sampling

The population for this study consisted of lower secondary school students enrolled at the Demonstration School of Udon Thani Rajabhat University, under the Ministry of Higher Education, Science, Research and Innovation, during the first semester of the 2025 academic year. A total of 270 students across nine classrooms were included in the population. The sample was divided into two groups as experimental and control group.

2.2.1. Experimental group
This group comprised 30 lower secondary school students from the demonstration school of Udon Thani Rajabhat University who participated in instructional activities based on the Synectics teaching model combined with practical application. The participants were selected using voluntary sampling from students enrolled in the course learning skills.

2.2.2. Control group
This group consisted of 30 lower secondary school students from the demonstration school of Udon Thani Rajabhat University who received conventional instruction. The participants were selected using voluntary sampling from students enrolled in the course learning skills. It should be noted that because a non-probability voluntary sampling approach was adopted due to institutional scheduling constraints, participants could not be randomly assigned to the groups, which represents a methodological limitation regarding potential selection bias.

2.3. Instruments
2.3.1. Instructional activities
Six 2-hour lesson plans (12 hours total) based on the 6-step Synectics instructional model were developed, covering story design, storyboarding, and Scratch animation. Content validity was verified by three computer education experts, yielding index of item-objective congruence (IOC) values of 0.95, 0.93, 0.97, 1.00, 0.98, and 0.92, respectively [30].

2.3.2. Creativity assessment instrument
The creativity assessment instrument was a subjective testing form validated by three experts (IOC=1.00) designed to evaluate divergent thinking skills based on Torrance’s core components: fluency, flexibility, originality, and elaboration. To establish high psychometric transparency, a pilot test was conducted with 30 non-sampled demographic peers, yielding an internal consistency Cronbach’s alpha coefficient of 0.81. Furthermore, to eliminate individual grading subjectivity, all student test outcomes were evaluated independently by a panel of three blinded expert assessors using predefined, objective rubric criteria. The inter-rater reliability among the three evaluators was statistically computed using the intraclass correlation coefficient (ICC) two-way mixed effects model under the consistency definition, yielding an average measures ICC value of 0.821 (95% confidence interval: 0.672, 0.909; $F(29, 58)=5.592$, $p<0.001$), indicating an excellent and highly standardized level of inter-rater consistency.

2.3.3. Programming skill assessment instrument
The programming skill assessment instrument utilized a standardized performance-based rubric designed to measure computing product quality and algorithmic task execution in Scratch (focusing on script logic, control structures, events, and debugging proficiency). The rubric consisted of five core assessment criteria scored on a 4-point scale (totaling 20 points). Content validity was confirmed by three experts (IOC=1.00). In a pilot test with 30 non-sampled peers, the rubric criteria achieved a Cronbach’s alpha coefficient of 0.87. To ensure strict objectivity across the experimental and control groups, all programming products were independently evaluated by the same panel of three blinded expert assessors. The inter-rater reliability was calculated using the ICC two-way mixed effects model, yielding an average measures ICC value of 0.757 (95% confidence interval: 0.554, 0.876; $F(29, 58)=4.115$, $p<0.001$). This result demonstrates a statistically significant and good level of grading stability and standardization among the evaluators.

2.4. Data analysis procedure
The researcher implemented the study according to the following steps: the instructional activities were conducted sequentially according to the lesson plans. After each instructional session, students were assessed using the creativity assessment form and the practical programming skills assessment form, and their scores were recorded. Data analysis:
- Comparative analysis: an independent samples t-test was used to compare the mean post-test scores for creativity and practical programming skills between the experimental group and the control group to determine if a statistically significant difference existed.
- Analysis of the relationship between creativity and practical programming skills the relationship between students’ creativity and their practical programming skills following the implementation of the Synectics-based instructional activities was analyzed using the Pearson product-moment correlation coefficient (r). The correlation coefficient ranges from -1 to +1, where the strength of the correlation was interpreted according to Cohen [31]: i) a negative value indicates an inverse relationship; ii) a positive value indicates a direct linear relationship; and iii) a value of 0 indicates no relationship between the variables.

3. RESULTS AND DISCUSSION

3.1. Results

3.1.1. Results of the comparison of mean scores of creativity and programming practice skills between the experimental and control groups

As shown in Table 2, it was found that:

- Creativity: the experimental group achieved a mean creativity score of 9.91 with a standard deviation of 0.95, whereas the control group obtained a mean score of 8.69 with a standard deviation of 1.55. The t-test analysis revealed a t-value of 3.676 and a p-value less than 0.001 ($p<0.001$), which is below the statistical significance level of 0.05. Therefore, it can be concluded that there is a statistically significant difference in creativity between the experimental and control group. The experimental group demonstrated significantly higher creativity than the control group.
- Practical programming skills: the experimental group had a mean programming performance score of 34.67 with a standard deviation of 1.83, while the control group had a mean score of 29.19 with a standard deviation of 4.85. The t-test results indicated a t-value of 5.790 and a p-value less than 0.001 ($p<0.001$), which is statistically significant at the 0.05 level. This indicates a statistically significant difference between the two groups, with the experimental group exhibiting substantially higher programming skills than the control group.

The effect size analysis showed a large effect of the Synectics instructional model on creativity (Cohen's d=0.94) and practical programming skills (Cohen's d=1.48), indicating that the intervention had a substantial impact on students' learning outcomes.

Table 2. Comparison of mean scores of creativity and programming practice skills of students in the experimental group and the control group

Variables Studied	Experimental group (n=30)		Control group (n=30)		t	P
	$\bar{x}$	S.D.	$\bar{x}$	S.D.		
Creativity	9.91	0.95	8.69	1.55	3.676	<0.001
Practical programming skills	34.67	1.83	29.19	4.85	5.790	<0.001

3.1.2. Results of the analysis of the relationship between creativity and programming skills following the implementation of Synectics-based learning activities for lower secondary school students

This section presents the findings regarding the relationship between students' creativity and programming skills after the intervention. Table 3 presents the correlation analysis between creativity and programming skills following the implementation of the Synectics instructional model. The results presented in Table 3 indicate that, based on the Pearson correlation analysis, creativity was found to have a statistically significant, strong positive correlation with programming performance among students in the experimental group who participated in the Synectics-based instructional activities ($r=0.710$, $p<0.001$). A Pearson correlation value of 0.710 demonstrates a high level of positive association between creativity and programming skills. This finding suggests that higher levels of creativity are significantly associated with higher levels of programming performance. According to Cohen's [31] interpretation criteria, the obtained correlation coefficient ($r=0.710$) indicates a large positive relationship between creativity and programming skills.

Table 3. Correlation between creativity and programming skills

Correlations			
Variables		Practical programming skills	Creativity
Practical programming skills	Pearson correlation	1	0.710**
	Sig. (2-tailed)		0.000
	N	30	30
Creativity	Pearson correlation	0.710**	1
	Sig. (2-tailed)	0.000	
	N	30	30

**Correlation is significant at the 0.01 level (2-tailed).

The scatter plot in Figure 1 illustrates a clear positive influence of creativity on programming skills. The trendline and regression equation indicate a statistically significant predictive relationship. Creativity accounts for 50.4% of the variance in programming skills ($R^2=0.504$), suggesting that creativity is a moderate to strong predictor of students' practical programming performance. This finding demonstrates that higher levels of creativity are associated with higher levels of programming proficiency.

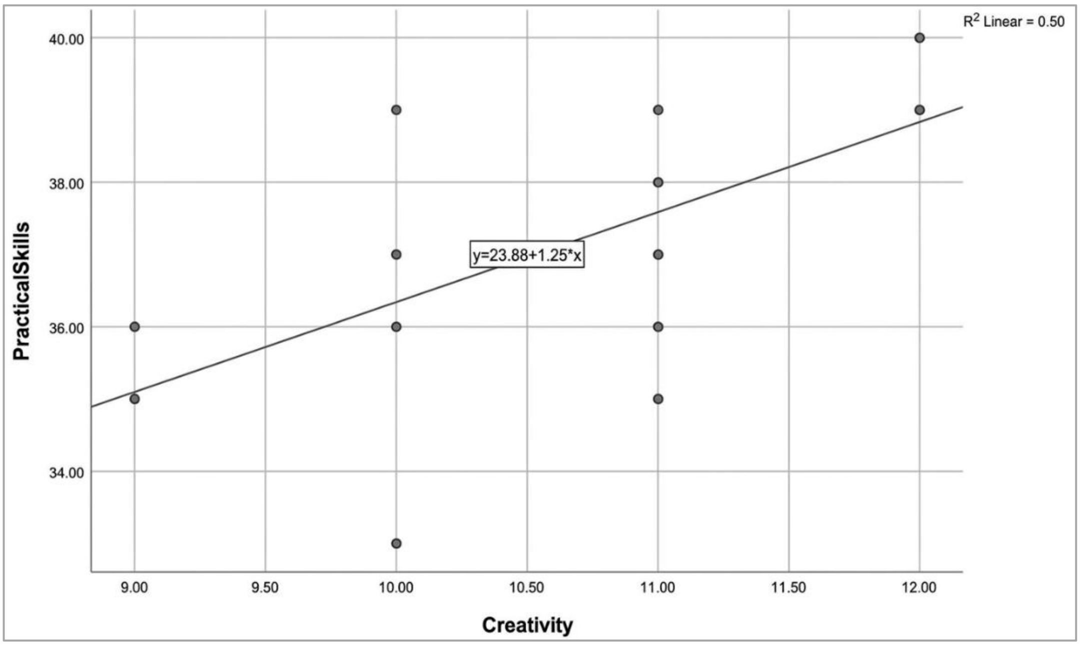

Figure 1. Scatter plot illustrating the relationship between creativity and programming skills

3.2. Discussion

The findings of this study provide empirical evidence for the efficacy of the Synectics instructional model in bridging the gap between creative cognition and technical skill development. Students undergoing Synectics-based instruction achieved significantly higher scores in both creativity and programming proficiency compared to those in the conventional "follow-the-example" group. This core finding validates Gordon [6] foundational theory that structured analogical thinking can break cognitive fixation and directly enhance problem-solving capacities in practical domains.

The descriptive statistics and large effect sizes (Cohen's d=0.94 for creativity; d=1.48 for programming) demonstrate that the intervention exerted a transformative impact rather than a marginal improvement. While traditional programming instruction often limits lower secondary students to rote syntax imitation [24], [25] leaving them ill-equipped for independent algorithmic design [26], the Synectics model offers an explicit mechanism for generative thinking. Specific phases, such as personal analogy (e.g., conceptualizing oneself as a line of code) and compressed conflict (e.g., resolving opposing conceptual forces within a program's logic), forced students to transcend standard design paradigms. By actively destabilizing familiar assumptions, the model systematically guided learners to construct original algorithmic pathways, confirming prior assertions that creative classroom environments promote deeper technical innovation [8], [9], [16].

Critically, the strong positive correlation ($r=0.710$, $p<0.001$) and regression analysis ($R^2=0.504$) redefine the interplay between creative thinking and computational performance. Creativity is frequently mischaracterized as a soft skill separate from technical science, technology, engineering, and mathematics (STEM) competencies. However, these data indicate that creative capacity accounts for over half of the variance in students' practical programming skills. This predictive relationship suggests that the cognitive flexibility and originality nurtured by Synectics are foundational to the psychomotor and logical demands of debugging, structuring logic, and writing code [5], [15]. Consequently, expanding computational creativity from artistic fields [17]–[19] into technical disciplines is not merely adjacent to computer science education; it is a pedagogical prerequisite for high-level technical mastery [20].

To contextualize the broader academic impact of these findings, the outcomes of this study were systematically compared with a network of prior global research. The empirical alignment observed across multiple domains strongly corroborates the universal adaptability of the Synectics model. Specifically, the significant leap in creative performance resonates with the milestones achieved by Wiharsih *et al.* [17] in performative arts, Sunendar *et al.* [18] in linguistic literacy, and Saputro and Waraulia [19] in creative writing, thereby substantiating that analogical destabilization effectively nurtures cognitive fluency across both artistic and technical curricula.

Furthermore, the robust predictive relationship between creativity and computational skill (R^2=0.504) directly parallels the theoretical assertions of Pease *et al.* [20] regarding computational creativity, as well as the empirical frameworks of Grover and Pea [24] and Resnick [26], who argued that programming mastery cannot thrive on syntax acquisition alone but requires high-level divergent problem-solving. This study also strengthens the pedagogical positions of Beghetto and Kaufman [8] and Sawyer [9], proving that structured creative interventions mitigate the limitations of traditional, imitation-heavy computer education as critiqued by Lye and Koh [25]. Lastly, the psychomotor stability captured in the programming rubrics bridges the foundational taxonomies of Simpson [5] and Harrow [15], demonstrating that technical execution is a direct downstream byproduct of structured conceptual origination. Beyond academic synthesis, this research delivers three foundational contributions to the field of computer education:

- Theoretical contribution: it expands the boundaries of Gordon [6] Synectics theory by proving that a model originally designed for the humanities is highly applicable to algorithmic and deterministic STEM fields, effectively transforming how the intersection between creative cognition and logical programming is conceptualized.
- Practical/empirical contribution: it provides frontline computer science educators with a validated, turnkey pedagogical strategy (the 6-step Synectics framework) to transcend mechanical coding practices, offering concrete evidence that nurturing student creativity directly improves independent debugging and code quality.
- Methodological contribution: by rigorously deploying a dual-instrument validation process backed by solid internal consistency testing (α=0.81 and 0.87) and utilizing a rigorous triple-assessor panels approach to guarantee inter-rater objectivity for both creativity and technical tasks (ICC=0.821 and 0.757, respectively), this study establishes a highly reliable and replicable empirical blueprint for future researchers seeking to quantify soft-skill variances within highly technical performance tasks.

4. CONCLUSION

This study investigated the impact of the Synectics instructional model on the creativity and practical programming skills of lower secondary school students, while exploring the underlying relationship between these two domains. The empirical evidence demonstrates that integrating analogical and metaphorical thinking into programming tasks yields superior outcomes compared to conventional, imitation-heavy instruction. Furthermore, the study uncovers a powerful, positive predictive relationship where creative capacities account for approximately 50.4% of the variance in programming performance. This proves that creative thinking operates as a primary cognitive driver for technical and algorithmic problem-solving in computer education.

The findings offer several critical implications for 21st-century educational reform and classroom practice: i) pedagogical shift: computer science curricula must transition away from mechanistic, rote coding exercises. Instructors should formally embed generative thinking frameworks—such as direct and personal analogies—into standard technical lessons to cultivate both technical literacy and creative cognitive capacity simultaneously; ii) curriculum cohesion: educational policymakers aiming to fulfill national mandates regarding computational innovation must actively align high-level policies with classroom activities by embedding cognitive flexibility rubrics directly into STEM and computing grading criteria; and iii) active learning environments: to facilitate psychomotor and cognitive skill acquisition in computer education, learning environments must be designed to tolerate experimentation, encourage divergent problem-solving, and provide structured frameworks for collaborative risk-taking.

Despite these contributions, certain limitations must be acknowledged. First, this study utilized a non-probability voluntary sampling method to recruit participants from a single demonstration school, and the sample size was relatively small (n=60). Because random assignment was not feasible, potential selection bias may be present; for instance, students who volunteered might have possessed higher initial baseline motivation, distinct prior learning characteristics, or greater interest in computer technology than the general student population. These factors may limit the immediate internal validity and the generalizability of the findings to broader, more diverse public school settings. Future research should address these constraints by employing larger, randomly assigned random samples across multiple geographical locations. Additionally, longitudinal studies are recommended to track the long-term retention of these creative problem-solving skills and to evaluate whether the cognitive benefits of Synectics transfer successfully to adjacent technical fields such as advanced mathematics, physics, and robotics.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the expertise and valuable contributions of the expert panel, Asst. Prof. Dr. Kanisorn Jeekratok, Asst. Prof. Dr. Suttikarn Bojakrapan, and Asst. Prof. Nipol Sungsutti, for their assistance in validating the research instruments. Finally, the authors thank the administrators, teachers, and students of the Demonstration School of Udon Thani Rajabhat University for their cooperation and participation in this study.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rattanawadee Muensripoom	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mana Sopa	✓	✓		✓				✓	✓	✓		✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

The research related to human use has been complied with all the relevant national regulations and institutional policies in accordance with the tenets of the Helsinki Declaration and has been approved by the authors' institutional review board, No. 0622.7/112.

DATA AVAILABILITY

Derived data supporting the findings of this study are available from the corresponding author [RM] on request.

REFERENCES

[1] UNESCO, *Reimagining Our Futures Together: A New Social Contract for Education*. Paris, France: UNESCO Publishing, 2021.
[2] OECD, "PISA 2022 creative thinking framework," in *PISA 2022 Assessment and Analytical Framework*. OECD, Ed., Paris: OECD Publishing, 2023, pp. 140–196.
[3] E. P. Torrance, *Education and the creative potential*. Minneapolis, MN: University of Minnesota Press, 1963.
[4] J. P. Guilford, *The nature of human intelligence*. New York, NY: McGraw-Hill, 1967.
[5] E. J. Simpson, *The classification of educational objectives in the psychomotor domain*. Washington DC: Gryphon House, 1972.
[6] W. J. Gordon, *Synectics: the development of creative capacity*. New York: Harper and Row, 1961.
[7] B. S. Bloom, J. T. Hastings, and G. F. Madaus, *Handbook on formative and summative evaluation of student learning*. New York, NY: McGraw-Hill, 1971.
[8] R. A. Beghetto and J. C. Kaufman, *Nurturing creativity in the classroom*. Cambridge, U.K.: Cambridge University Press, 2010.
[9] R. K. Sawyer, *Explaining Creativity: The Science of Human Innovation*, 2nd ed. New York, NY: Oxford University Press, 2012.
[10] J. Sweller, J. J. G. van Merriënboer, and F. Paas, "Cognitive architecture and instructional design: 20 years later," *Educational Psychology Review*, vol. 31, pp. 261–292, 2019, doi: 10.1007/s10648-019-09465-5.

[11] C. A. Wade, "Using writing to develop and assess critical thinking," *Teaching of Psychology*, vol. 22, no. 1, pp. 24–28, 1995, doi: 10.1207/s15328023top2201_8.
[12] I. K. Davies, *The management of learning*. London, U.K.: McGraw-Hill, 1971.
[13] B. R. Joyce, M. Weil, and E. Calhoun, *Models of teaching*, 9th ed. Boston, MA: Pearson, 2015.
[14] H. H. Walbesser, "Reviews: Tyler, Ralph W., Gagné, Robert M., and Scriven, Michael. Perspectives of Curriculum Evaluation," *American Educational Research Journal*, vol. 5, no. 1, pp. 120–121, 1968, doi: 10.3102/00028312005001120.
[15] A. J. Harrow, *A taxonomy of the psychomotor domain: A guide for developing behavioral objectives*. New York, NY: Longman, 1972.
[16] A. M. Westcott and J. A. Smith, *Creative teaching of mathematics in the elementary school*. Boston, MA: Allyn and Bacon, 1967.
[17] R. Wiharsih, Kusnadi, and T. M. Sofa, "The effect of the synectic model on creativity development of students' dance works," in *4th International Conference on Arts and Arts Education (ICAAE 2020)*, 2021, pp. 248–252, doi: 10.2991/assehr.k.210602.049.
[18] D. Sunendar, T. I. Hardini, and I. S. Karimah, "Self-literacy synectic writing model and challenge on 21st century skills," in *Fifth International Conference on Language, Literature, Culture, and Education (ICOLLITE 2021)*, 2021, pp. 163–169, doi: 10.2991/assehr.k.211119.026.
[19] A. N. Saputro and A. Waraulia, "Synectics multiliteracy learning model in writing short stories," in *Proceedings of the 3rd International Conference on Education and Technology (ICETECH 2022)*, 2023, pp. 367–375, doi: 10.2991/978-2-38476-056-5_39.
[20] A. Pease *et al.*, "The importance of applying computational creativity to scientific and mathematical domains," in *Proceedings of the 10th International Conference on Computational Creativity, ICCC 2019*, 2019, pp. 250–257.
[21] M. Csikszentmihalyi, *Creativity: flow and the psychology of discovery and invention*. New York, NY: HarperPerennial, 1997.
[22] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in *Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA)*, Apr. 2012, pp. 1–25.
[23] R. Israel-Fishelson and A. Hershkovitz, "Studying interrelations of computational thinking and creativity: A scoping review (2011–2020)," *Computers & Education*, vol. 176, p. 104353, 2022, doi: 10.1016/j.compedu.2021.104353.
[24] S. Grover and R. Pea, "Computational thinking in K–12," *Educational Researcher*, vol. 42, no. 1, pp. 38–43, Jan. 2013, doi: 10.3102/0013189X12463051.
[25] S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: what is next for K-12?" *Computers in Human Behavior*, vol. 41, pp. 51–61, Dec. 2014, doi: 10.1016/j.chb.2014.09.012.
[26] M. Resnick, "All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten," in *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, 2007, pp. 1–6, doi: 10.1145/1254960.1254961.
[27] P. Mishra and M. J. Koehler, "Technological pedagogical content knowledge: a framework for teacher knowledge," *Teachers College Record: The Voice of Scholarship in Education*, vol. 108, no. 6, pp. 1017–1054, Jun. 2006, doi: 10.1111/j.1467-9620.2006.00684.x.
[28] OECD, *The future of education and skills 2030*. Paris: OECD Publishing, 2018.
[29] M. U. Bers, *Coding as a playground: programming and computational thinking in the early childhood classroom*, 2nd ed. New York, NY, USA: Routledge, 2020.
[30] R. J. Rovinelli and R. K. Hambleton, "On the use of content specialists in the assessment of criterion-referenced test item validity," *Dutch Journal of Educational Research*, vol. 2, no. 2, pp. 49–60, 1977.
[31] J. Cohen, *Statistical power analysis for the behavioral sciences*, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

BIOGRAPHIES OF AUTHORS

Rattanawadee Muensripoom is a teacher at the Demonstration School of Udon Thani Rajabhat University, Thailand. She received her Bachelor of Education (B.Ed.) in 2015 and her Master of Education (M.Ed.) in Innovation in Computer Education in 2025, both from Udon Thani Rajabhat University. Her primary research interests include computer education, pedagogical interventions for creativity, instructional design, and enhancing programming skills for secondary school students. This paper is derived from her master's thesis. She can be contacted at email: 65120612106@udru.ac.th.

Mana Sopa is an assistant professor in the Faculty of Education at Udon Thani Rajabhat University, Thailand, specializing in computer education. He served as the principal thesis advisor for this research. His expertise lies in educational technology, innovation in computer education, instructional design for 21st-century skills, and research in learning processes. He can be contacted at email: manasopa@udru.ac.th.