

國中運算思維與程式邏輯課程對學習動機與滿意度之影響

溫廷宇*、鄭國明

中文摘要

研究目的：本研究旨在探討「運算思維與程式邏輯課程」對國中學生學習動機與學習滿意度之影響。隨著程式設計已成為 21 世紀核心素養的重要組成部分，瞭解此類課程如何影響學生的情意參與，對於課程改革與教學創新具有關鍵意義。

研究方法：本研究以建構主義學習理論與自我決定理論為基礎，採準實驗研究設計。研究對象為臺灣某國中兩班八年級學生，分為實驗組與控制組。實驗組接受為期四週之課程，內容包含運算思維框架、積木式程式設計，以及真實情境問題解決任務；控制組則依照現行科技課程教學。研究以驗證過的量表測量學生的學習動機與滿意度，並使用獨立樣本 t 檢定進行資料分析。

研究發現：研究結果顯示，實驗組學生在學習動機與學習滿意度兩項指標上均顯著高於控制組，且具有大效應量，顯示課程對學生學習具有顯著正向影響。具互動性、真實關聯性與以學生為中心的學習環境，有助於提升學生的參與感、學習樂趣與對課程的認同價值。

研究價值：本研究提供實證證據，證明將運算思維與程式邏輯融入國中課程，能有效促進學生的情意學習成效。研究強調學習自主性與問題導向學習在提升動機與滿意度方面的重要性，並為教育現場與政策制定者提供具體建議，以設計具吸引力與成效的運算課程，培養學生的數位素養與對程式設計的持久興趣。

關鍵詞：運算思維、程式邏輯、學習動機、學習滿意度、Scratch

溫廷宇(通訊作者)，國立高雄師範大學工業科技教育系所博士生。Email: 811071002@mail.nknu.edu.tw
鄭國明，國立高雄師範大學工業科技教育系所教授。Email: jgm0730@gmail.com

The Impact of Computational Thinking and Programming Logic Curriculum on Junior High School Students Learning Motivation and Satisfaction

Ting-Yu Wen* , Guo Ming Jeng

Abstract

Purpose—This study aims to explore the effects of a computational thinking and programming logic curriculum on junior high school students' learning motivation and satisfaction. As programming becomes a core component of 21st-century competencies, understanding how such curricula influence students' affective engagement is crucial for curriculum reform and instructional innovation.

Design/methodology/approach—Grounded in constructivist learning theory and Self-Determination Theory, this study adopted a quasi-experimental design. Two eighth-grade classes from a junior high school in Taiwan were divided into an experimental group and a control group. The experimental group engaged in a four-week course incorporating computational thinking frameworks, block-based programming, and real-world problem-solving tasks. The control group followed the standard technology curriculum. Learning motivation and satisfaction were measured using validated scales, and the data were analyzed using independent samples t-tests.

Findings—The results showed that students in the experimental group scored significantly higher in both learning motivation and satisfaction compared to the control group. Large effect sizes were observed, indicating strong positive impacts. The interactive, relevant, and student-centered learning environment contributed to increased engagement, enjoyment, and perceived value of the learning experience.

Originality/value/implications—This study provides empirical evidence that integrating computational thinking and programming logic into junior high school curricula can enhance students' affective learning outcomes. The findings highlight the value of learner autonomy and problem-based learning in promoting motivation and satisfaction. The study offers practical insights for educators and policymakers to develop effective, engaging computing curricula that support digital competence and sustained interest in programming.

Keywords: Computational Thinking, Programming Logic, Learning Motivation, Learning Satisfaction, Scratch

Ting-Yu Wen (Corresponding Author), Student, Department of Industry Technology Education, National Kaohsiung Normal University, Email:811071002@mail.nknu.edu.tw
Guo Ming Jeng, Professor, Department of Industry Technology Education, National Kaohsiung Normal University, Email:jgm0730@gmail.com

1. Introduction

1-1. Background

In the era of digital transformation, the ability to think computationally and engage in problem-solving using algorithmic processes has emerged as a vital literacy. Internationally, educational systems have embraced this paradigm by embedding programming and computational thinking into national curricula (Bocconi et al., 2016). Taiwan's 2019 Curriculum Guidelines (commonly known as the 108 Curriculum) are aligned with these global trends, introducing programming into junior high school technology courses with the intention of cultivating digital competence and future-ready skills.

From a theoretical standpoint, this curriculum reform is grounded in constructivist learning theory (Piaget, 1970; Papert, 1980), which posits that learners actively construct knowledge through experiences that are meaningful and contextually relevant. Programming, when approached through problem-based and project-based learning, offers such a context by promoting active engagement, logical reasoning, and iterative thinking. Furthermore, Self-Determination Theory (Deci & Ryan, 1985) suggests that learning environments that support autonomy, competence, and relatedness are more likely to foster intrinsic motivation—an essential component for sustained learning, especially in cognitively demanding subjects such as programming, where sustained engagement is critical.

This study seeks to explore whether a curriculum designed with these pedagogical and theoretical insights can indeed influence two key affective variables: learning motivation and learning satisfaction. These variables serve not only as indicators of student engagement but also as predictors of long-term academic persistence and success (Pintrich & De Groot, 1990). Computational thinking, motivation, and attitude are strongly interrelated and jointly influence programming achievement (Choi & Choi, 2023).

1-2. Research Objectives

This study aims to explore the impact of a computational thinking and programming logic course on junior high school students. The specific objectives are as follows:

1. To examine whether the course can enhance students' learning motivation.
2. To assess whether the course can improve students' learning satisfaction.
3. To provide empirical data and suggestions for future programming instruction and curriculum planning in junior high schools.

1-3. Research Questions

In line with the above objectives, this study addresses the following research questions:

1. Is there a significant difference in learning motivation between students who participate in the computational thinking and programming logic course and those who do not?

2. Is there a significant difference in learning satisfaction between students who participate in the computational thinking and programming logic course and those who do not?
3. What is the overall learning attitude of students towards the coding course?

2. Literature Review

2-1. Computational Thinking

Computational thinking (CT) is a methodical problem-solving process that involves formulating problems in a way that enables the use of computer-based tools for solving them. Wing (2006) emphasized that CT is a "fundamental skill for everyone, not just computer scientists." CT comprises components such as abstraction, decomposition, pattern recognition, and algorithmic design, all of which support higher-order thinking skills. Enhanced computational thinking contributes to higher programming self-efficacy and learning motivation among students (Yilmaz & Yilmaz, 2023).

From a theoretical perspective, Papert's Constructionism (1980) supports the use of programming tools (like Scratch) as a medium for learners to construct knowledge through designing and building computational artifacts. This process aligns with Bloom's revised taxonomy, situating CT at the upper levels of cognitive engagement—analyzing, evaluating, and creating.

2-2. Programming Logic

Programming logic refers to understanding the structural elements of code, such as sequence, conditionals, and iteration. It facilitates the application of CT in practice. Grover and Pea (2013) assert that programming logic is foundational for developing computational literacy. It helps reduce extraneous cognitive load (Sweller, 1988) by offering a structured framework for solving complex problems.

In educational settings, programming logic instruction often serves as a gateway to deeper computational learning. When taught using visual programming environments, learners can bypass syntax-related frustration and focus on the logic and flow of programs—an approach consistent with Cognitive Load Theory.

2-3. Learning Motivation

Learning motivation is influenced by internal and external factors that determine a student's willingness to engage with content. Self-Determination Theory (Deci & Ryan, 1985) distinguishes between intrinsic motivation (engagement due to interest) and extrinsic motivation (driven by external rewards or pressures). Previous studies suggest that effective programming curricula by emphasizing autonomy, challenge, and relevance. Students'

intrinsic programming motivation is positively associated with their computational thinking proficiency (Fagerlund et al., 2022).

Additionally, Flow Theory (Csikszentmihalyi, 1990) provides a lens through which motivation in programming tasks can be understood. When students are fully immersed in challenging but achievable programming tasks, they are more likely to experience a "flow" state, leading to deeper engagement and persistence.

2-4. Learning Satisfaction

Learning satisfaction reflects a student's overall appraisal of their learning experience and is strongly associated with their perception of teaching quality, curriculum relevance, and personal achievement. According to Keller's ARCS Model (1987)—which focuses on Attention, Relevance, Confidence, and Satisfaction—courses that incorporate real-world relevance and offer scaffolded success are more likely to produce satisfied learners.

In programming education, satisfaction can be increased by employing constructivist-aligned methods such as project-based learning, collaborative problem-solving, and immediate feedback—all of which support a positive learning climate.

2-5. Summary of Literature Review

Prior research and theoretical models converge on the view that computational thinking and programming logic instruction—when grounded in sound pedagogical practices—can yield cognitive, motivational, and affective benefits. Drawing from constructivist learning theory, cognitive load theory, and motivational psychology, this study hypothesizes that integrating computational thinking with programming logic in a junior high school context will positively influence students' motivation and satisfaction.

3. Research Methodology

3-1. Research Design

This study employed a quasi-experimental design with a pretest-posttest control group approach to investigate the effects of a computational thinking and programming logic curriculum on junior high school students' learning motivation and satisfaction. The quasi-experimental method is particularly suitable for educational contexts where randomized assignment is not feasible, allowing for practical implementation while maintaining internal validity (Cook & Campbell, 1979).

The research framework is grounded in constructivist learning theory, which posits that learners actively construct knowledge through experiences and reflection. The experimental curriculum was designed to align with constructivist principles by emphasizing active problem-solving, iterative design, and collaborative learning.

3-2. Ethical Considerations

This study adhered to standard ethical guidelines for educational research. Prior to data collection, ethical clearance was obtained from the institutional review board of the authors' affiliated university. All participants and their legal guardians were informed about the purpose, procedures, and voluntary nature of the study. Written informed consent was obtained from both students and their parents or guardians. Participants were assured that their responses would be kept confidential and used solely for research purposes. Anonymity was maintained throughout data collection, analysis, and reporting.

3-3. Participants

Participants were drawn from eighth-grade classes in a junior high school located in Miaoli County, Taiwan. A total of 56 students participated, with 28 assigned to the experimental group and 28 to the control group. To maintain instructional consistency and reduce contamination, intact classroom groups were used. The sampling approach was purposive, focusing on the accessibility and cooperation of the school site.

3-4. Research Instruments

To assess learning outcomes, two instruments were developed and validated:

1. Learning Motivation Scale

Adapted from Pintrich and De Groot (1990), the scale includes intrinsic and extrinsic motivational components, reflecting the Self-Determination Theory (Deci & Ryan, 1985). The revised version contains 20 items tailored to the context of programming education.

2. Learning Satisfaction Scale

This instrument measures satisfaction across multiple dimensions, including content relevance, instructional delivery, perceived learning, and affective responses. The scale comprises 18 items, developed based on existing models of learner satisfaction (Keller, 2010; Sun et al., 2008).

Both scales utilized a 5-point Likert response format, and expert review confirmed their content validity. A pilot study was conducted to ensure the reliability and internal consistency of both instruments, yielding Cronbach's alpha values above .85.

3-5. Teaching Intervention

The experimental group participated in a four-week curriculum centered on computational thinking and programming logic, incorporating constructivist and experiential learning strategies. Each weekly session lasted 45 minutes and included the following components:

- Introduction to computational thinking frameworks (Wing, 2006)

- Instruction on programming logic structures (sequence, conditionals, loops)
- Engagement with block-based programming environments such as Scratch
- Application of real-world problem-solving scenarios
- Collaborative projects to reinforce concepts through peer interaction

The control group continued with the standard technology education curriculum, focusing on general computer skills without explicit instruction in computational thinking or programming logic.

3-6. Data Collection and Analysis

Data were collected at two time points: before (pretest) and after (posttest) the intervention. Quantitative data were analyzed using SPSS 26.0, applying the following statistical techniques:

- Descriptive statistics to summarize sample characteristics
- Independent samples t-tests to compare post-intervention outcomes between groups
- Paired samples t-tests to analyze within-group pre-post changes
- Significance was determined at $\alpha = .05$

Effect sizes (Cohen's *d*) were calculated to determine the magnitude of observed effects, with benchmarks set at 0.2 (small), 0.5 (medium), and 0.8 (large).

4. Results and Discussion

4-1. Analysis of Learning Motivation

An independent samples t-test indicated that the experimental group ($M = 4.32$, $SD = 0.41$) scored significantly higher on the learning motivation scale compared to the control group ($M = 3.87$, $SD = 0.53$), $t(54) = 3.18$, $p < .01$, $d = 0.92$, representing a large effect.

These findings support the notion that integrating computational thinking principles enhances learners' motivation by fostering autonomy, competence, and relevance—three key constructs in Self-Determination Theory (Deci & Ryan, 1985). The use of visual, interactive tools reduced the intimidation often associated with text-based programming, thereby promoting a more engaging learning environment.

4-2. Analysis of Learning Satisfaction

The experimental group also reported significantly greater satisfaction ($M = 4.45$, $SD = 0.36$) than the control group ($M = 3.92$, $SD = 0.49$), $t(54) = 3.76$, $p < .01$, $d = 1.01$.

According to Keller's ARCS model (Attention, Relevance, Confidence, Satisfaction), instructional designs that incorporate real-life applications, gamified elements, and progressive challenge levels can substantially increase satisfaction. The results suggest that

the curriculum addressed students' psychological needs and enhanced their perception of learning value.

4-3. Discussion

The outcomes align with prior research demonstrating that computational thinking frameworks can improve both affective and cognitive learning outcomes (Grover & Pea, 2013; Brennan & Resnick, 2012). Students exposed to structured, visually-supported learning environments demonstrated more positive attitudes toward learning, consistent with Cognitive Load Theory (Sweller, 1988), which posits that reducing extraneous cognitive load enhances student engagement and retention.

Furthermore, the findings validate the constructivist premise that hands-on, authentic learning tasks foster deeper motivation and satisfaction. This has strong implications for the design of future technology education curricula. Project-based computational thinking instruction can re-engage apathetic students and improve their motivation and academic behavior (Díaz-Lauzurica & Moreno-Salinas, 2019).

These results are consistent with findings from Yilmaz and Yilmaz (2023), who observed a similar increase in student motivation following computational thinking instruction in middle school contexts. The alignment suggests that well-structured computational thinking curricula can have a broad positive impact on affective learning outcomes across diverse educational settings. However, the relatively short intervention period in this study may limit the extent to which long-term changes in motivation and satisfaction can be observed. Future studies employing longitudinal designs would be necessary to determine whether these gains are sustained over time and how they influence subsequent academic achievement or persistence in STEM-related fields.

5. Conclusions and Recommendations

5-1. Conclusions

Based on the empirical evidence, the study concludes that:

- A curriculum centered on computational thinking and programming logic significantly enhances learning motivation, likely due to increased relevance, perceived competence, and autonomy.
- Student learning satisfaction also improves under such instructional conditions, especially when interactive, problem-based methods are employed.
- Constructivist and self-determination-based instructional strategies are effective for promoting affective engagement in programming education.

5-2. Recommendations

For Curriculum Development

- Incorporate computational thinking as a core element in junior high school technology courses.
- Design curricula that emphasize problem-solving, logic, and creativity, rather than rote syntax memorization.

For Instructional Design

- Utilize visual programming environments, real-life problem contexts, and project-based learning to enhance student engagement.
- Embed formative assessments and reflection activities to deepen understanding.

For Teacher Training

- Provide professional development programs focused on computational thinking pedagogy and programming logic instruction.
- Encourage peer collaboration and community-based teaching resources to support curricular adoption.

三、For Future Research

- Expand the study to include diverse geographic regions and school types to enhance generalizability.
- Conduct longitudinal studies to examine the sustained effects of computational curricula on student motivation and achievement.
- Explore qualitative data such as interviews or classroom observations to gain insights into student experiences and instructional interactions.

References

- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice* (No. JRC104188). Joint Research Centre.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada.
- Choi, S., & Choi, J. (2023). The influence and relationship between computational thinking, learning motivation, attitude, and achievement of Code.org in K-12 programming education. *arXiv preprint*. <https://arxiv.org/abs/2412.14180>
- Chen, B. L. (2020). *Using project-based learning to enhance programming learning outcomes for elementary school students* (Unpublished master's thesis). National Taipei University of Education, Department of Computer Science.

- Chen, J. H. (2018). *A study on the impact of integrating information technology into teaching on the learning motivation and effectiveness of junior high school students* (Unpublished master's thesis). National Changhua University of Education, Institute of Education.
- Cook, T. D., & Campbell, D. T. (1979). *Quasi-experimentation: Design and analysis issues for field settings*. Houghton Mifflin.
- Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. Harper & Row.
- Díaz-Lauzurica, L., & Moreno-Salinas, D. (2019). Computational thinking and robotics: A teaching experience in compulsory secondary education with students with high degree of apathy and demotivation. *Sustainability*, 11(18), 5109. <https://doi.org/10.3390/su11185109>
- Fagerlund, J., Mikkonen, J., Mikkonen, T., & Lattu, M. (2022). Finnish teachers' and students' programming motivation and their role in teaching and learning computational thinking. *Frontiers in Education*, 7, 948783. <https://doi.org/10.3389/educ.2022.948783>
- 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>
- Keller, J. M. (2010). *Motivational design for learning and performance: The ARCS model approach*. Springer.
- Li, Z. Q. (2017). *A study on the learning satisfaction and effectiveness of high school students in a digital learning environment* (Unpublished master's thesis). National Kaohsiung Normal University, Department of Industrial Technology Education.
- Lin, Y. T., Tseng, Y. M., & Chien, S. Y. (2019). Enhancing computational thinking capability of preschool children by game-based smart toys. *Electronic Commerce Research and Applications*, 36, 100859. <https://doi.org/10.1016/j.elerap.2019.100859>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Piaget, J. (1970). *Science of education and the psychology of the child*. Orion Press.
- Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology*, 82(1), 33–40. <https://doi.org/10.1037/0022-0663.82.1.33>
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4
- Wang, S. F. (2021). *A study on the learning motivation and satisfaction of upper-grade elementary school students participating in programming courses* (Unpublished master's thesis). National Pingtung University, Institute of Educational Technology.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>

Wu, Y. X. (2021). *An action research on the design and implementation of computational thinking and programming logic courses for junior high school students* (Unpublished master's thesis). National Tainan University, Institute of Information Education.

Yilmaz, R. M., & Yilmaz, F. G. K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Education and Information Technologies*.
<https://doi.org/10.1007/s10639-023-11880-4>

Wu, J. H., & Chen, T. H. (2022). Combining Webduino programming with situated learning to promote computational thinking, motivation, and satisfaction among high school students. *Sustainability*, 14(17), 10894. <https://doi.org/10.3390/su141710894>

Appendix A

Learning Motivation and Satisfaction Scale for Computational Thinking and Programming Logic Courses

Instructions: Please indicate your level of agreement with each statement by checking the box that best represents your opinion.

Scale:

1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree

Basic Information

Gender: Male Female

Section 1: Learning Motivation

No. Item	1	2	3	4	5
1 I am interested in the computational thinking and programming logic course content.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 I find the course content engaging and stimulating.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 I believe computational thinking and programming logic are useful in real life.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Learning this course has been a refreshing experience.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5 I enjoy learning topics related to computational thinking and programming logic.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6 I am confident that I will perform well in the assessments for this course.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7 I am motivated to learn because I see this course as an opportunity for personal growth.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8 I feel I have learned a lot through this course.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Section 2: Learning Satisfaction

No. Item	1	2	3	4	5
9 I find the computational thinking and programming logic course to be effective.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10 The course materials provide useful and relevant content.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11 I believe I performed well in this course.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12 This course encouraged collaboration and peer discussion.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13 I enjoyed the computational thinking and programming logic course.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14 I am satisfied with the overall experience of the course.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Appendix B

Scoring Rubric

- **Total Score Range:** 14 items × 5 points = **70 points maximum**
- **Interpretation Guide:**
 - 56–70 points: **High motivation and satisfaction**
 - 42–55 points: **Moderate motivation and satisfaction**
 - 28–41 points: **Low motivation and satisfaction**
 - Below 28: **Very low; intervention may be required**

You may also break down scores by section to independently assess motivation (Items 1–8) and satisfaction (Items 9–14).

Appendix C

Teacher Reflection Log (Template)

Date	Lesson Topic	Student Engagement (1–5)	Notes on Participation & Interest	Teacher Reflections / Adjustments for Next Time
YYYY/MM/DD	e.g., Introduction to Loops in Scratch	<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	Most students were excited to create animations.	Will allow more time for experimentation next time.
YYYY/MM/DD				
YYYY/MM/DD				

Use this log weekly to reflect on student progress, engagement, and course design.