The Mediating Effect of Creativity on the Relationship Between Mathematic Achievement and Programming Self-Efficacy

Jun Liu[1], Meng Sun[1], Yue Dong[2], Fei Xu[3], Xue Sun[1] and Yan Zhou[4]*

[1] College of Education, Capital Normal University, Beijing, China, [2] Beijing No. 35 High School, Beijing, China, [3] College of Teacher Education, Capital Normal University, Beijing, China, [4] Xinyu School Affiliated to Beijing Normal University, Xinyu, China

Purpose: This study aimed to explore the relationship between mathematic achievement and programming self-efficacy, and adopt a mediation model to verify the mediating role of creativity on the relationship between mathematic achievement and programming self-efficacy.

Methods: A total of 950 upper-secondary school students were surveyed using their math test scores, the Kirton Adaption-Innovation and the Programmed Self-Efficacy Scale. SPSS-26 was used for descriptive statistical analysis and correlation analysis of related variables. The PROCESS plugin was used to test the mediating effect of creativity.

Results: (1) Mathematic achievement has a positive effect on programming self-efficacy, mathematic achievement is positively related to creativity, and creativity also has a positive influence on programming self-efficacy. (2) Creativity has a mediating effect on the relationship between mathematic achievement and programming self-efficacy.

Conclusion: The results revealed that mathematic achievement affected programming self-efficacy directly and also indirectly through creativity. This provided certain ideas for the development of programming education for teenagers. Since students' mathematics learning and creativity are related to programming learning, it is necessary to pay attention to the integration of the disciplines of programming education and mathematics. Further, the cultivation of innovative thinking is also critical to facilitate programming learning.

Keywords: programming self-efficacy, mathematic achievement, creativity, mediating effect, upper-secondary school students

INTRODUCTION

An increasing number of countries and regions have been attaching great importance to programming education for young people in recent years. In the beginning, most programming students were IT-related professionals who learned programming for their own work purpose. However, the LOGO language was introduced in the field of education which made programming

easier for children to learn (Feurzeig et al., 2011). At present, with the development of artificial intelligence (AI), programming has become a significant course for students. Meanwhile, various countries have put forward many specific requirements for programming learning in primary and secondary schools. According to the iDREAMS project, designed by the National Science Foundation, using a Scalable Game Design in the regular school curriculum can improve computer science education (Repenning and Ioannidou, 2008). Moreover, in 2014, the European Commission officially launched Code Week in Europe, with it then receiving widespread attention and continuing to this day (Europe Commission, 2014). In Australia and Turkey, programming education has become an important course in primary and secondary levels (StartupSmart, 2015; Arslan and Tanel, 2020). In recent years, China has also introduced certain educational policies to promote programming education for teenagers in primary and secondary schools. For instance, in 2019, the Ministry of Education of China emphasized that China will popularize programming education at primary and secondary schools (Ministry of Education of the People's Republic of China, 2019). Meanwhile, several extracurricular programming classes and adaptive programming languages sprang up since the programming learning of teenagers has aroused the concern of society, parents and school (Sun and Zhou, 2019). Hence, it is clear that programming education has attracted significant attention of many countries.

However, several difficulties and challenges emerged as programming education continues to spread among younger students. First, the complexity of programming itself makes some students have a low programming self-efficacy (Davidsson et al., 2010; Hongwarittorrn and Krairit, 2010), which is an important factor affecting the success of students' programming learning (Altun and Mazman, 2012; Korkmaz, 2012; Tsai, 2019). For most students, learning programming is both complex and challenging, except for those who are interested in and talented at it (Cheng et al., 2013; Akinola, 2015; Kucuk and Sisman, 2017). Altun also pointed out the improvement of programming achievement not only depends on programming cognitive skills and prior knowledge, but also depends on the improvement of programming self-efficacy (Altun and Mazman, 2012). Thus, to improve students' programming self-efficacy is one of the challenges for programming education. Second, because of abstract thinking and logical reasoning, programming may cause learners great distress which puts forward higher demand for learners' mathematic achievements (Jenkins, 2002; Garner, 2009; Yukselturk and Altiok, 2017). Many researchers pointed out that math is the foundation of programming learning (Ramalingam and Wiedenbeck, 1998; Rubio et al., 2013). Another study found that students with mathematical thinking, logic and knowledge are more likely to achieve success in programming (Tomai and Reilly, 2014). Therefore, mathematic achievement becomes an important factor affecting programming learning (White, 2003). Third, the goals of programming education for teenagers have changed in intelligent society, because it put forward higher requirements on the innovation ability of talents. Specifically, programming education for teenagers not only emphasizes code

writing and program functions, but also creative expression (Dufva, 2018). Creative programming works require students to master programming knowledge and develop innovative thinking in the process of coding (Soykan and Kanbul, 2018; Sandberg, 2019; Fragapane and Standl, 2021). Thus, it can be seen that creativity is an indispensable ability for students' successful programming.

Although scholars have not directly focused on the complex relationship among mathematics, creativity and programming, some related studies have proved that there may be a certain relationship between them. First, some researchers have explored the relationship between mathematics and programming. They discovered that students' mathematic achievement could be improved in programming (Friend et al., 2018; Serpe, 2019). However, other studies also found that mathematic learning can promote programming learning, which is an important factor for a programming beginner to learn programming skill (Ramalingam and Wiedenbeck, 1998). Similarly, Tomai and Reilly (2014) revealed that students with math preparation were more likely to successfully complete programming learning. Second, there have been a lot of studies on the relationship between creativity and programming. Most scholars indicated that students can cultivate creativity in programming learning (Kobsiripat, 2015; Noh and Lee, 2020), however, the influence of creativity on programming learning also needs attention. For example, a study by Pardamean et al. (2011) discovered that creativity is essential for learning computer programming. Other studies have also pointed out that programming education not only requires students' strong programming ability, but also requires them to have strong innovation ability (Zhang and Chen, 2016). Meanwhile, creativity was related to a person's degree of self-efficacy (Zhang Y. et al., 2018). In summary, there may be a potential influence among mathematic achievement, creativity and programming self-efficacy, but the specific relationship among the three is still unclear, which is worth exploring.

Thus, the purpose of this research is to verify the relationship between mathematic achievement and programming self-efficacy as well as the mediating effect of creativity in this relationship. Understanding the relationship between these three variables is conducive to clarifying the direction of adolescent programming education reform from a more comprehensive perspective, breaking through the current difficulties and challenges, and realizing the development of adolescent programming education.

LITERATURE REVIEW AND HYPOTHESES

Programming Self-Efficacy and Mathematic Achievement

Self-efficacy refers to an individual's expectations and judgments of their own specific behaviors or abilities to achieve a desired goal (Bandura, 1977). *Programming self-efficacy* refers to the confidence of students in their ability to learn and complete

programming tasks (Compeau and Higgins, 1995; Rohatgi et al., 2016). There are two descriptions of self-efficacy in programming education. First, there is computer self-efficacy, which refers to a person's judgments of their own ability to use a computer (Compeau and Higgins, 1995). Second, there is computer programming self-efficacy (CPSE). The Computer Programming Self-Efficacy Scale (CPSES) used to measure CPSE, as developed by Ramalingam and Wiedenbeck (1998), is one of the most frequently used tools for measuring programming self-efficacy. Related to this research, studies have focused on the measurement of students' programming self-efficacy and its influencing factors. One study found that using Visual Programming Language to intervene in students' programming lessons can reduce the difficulty of programming learning, while also improving their self-efficacy in this area (Tsai, 2019). Simultaneously, there is also a relationship between computer programming self-efficacy and programming learning (Abdunabi et al., 2019).

Academic achievement refers to the level of knowledge and skills acquired by students after learning the contents of a certain subject (Zimmerman, 1990; Zhou et al., 2006). According to this concept, it can be concluded that *mathematic achievement* refers to the level and skills of students after a period of mathematical learning. Students' mathematic achievement can be generally represented by their math scores in some key math exams, such as mid-term and final exams (Jacob, 2012; Yaratan and Kasapoğlu, 2012; Buzzai et al., 2020). According to the research, the change of students' math scores represents their growth in mathematic achievement (Scammacca et al., 2020).

There may be a connection between programming and mathematical learning. On the one hand, it has been proved by many studies that self-efficacy can affect academic performance (Ayotola and Adedeji, 2009; Affuso et al., 2017), but some studies also revealed that academic performance can predict self-efficacy (Heggestad and Kanfer, 2005; Mijung et al., 2018). As Hwang et al. (2016) found, there is an interactive relationship between self-efficacy and academic performance, that is, the impact of past academic performance on self-efficacy is greater than that of self-efficacy on academic performance. This suggests that it is feasible to predict programming self-efficacy with mathematic achievement. On the other hand, programming requires both logical reasoning and problem-solving abilities (Govender, 2007). These two are also important skills in the field of mathematics, which are directly reflected in the mathematic achievement of learners (Bocconi et al., 2018). A study by Mathews (2017) suggested that mathematical achievement would affect the success of programming learning. Another research confirmed that students' mathematic achievement is positively correlated with their levels in programming courses (White and Sivitanides, 2003). However, the predictive relationship between mathematic achievement and programming self-efficacy is unclear. Therefore, we propose the following hypothesis:

H1: Mathematic achievement has a positive effect on students' programming self-efficacy.

Creativity and Programming Self-Efficacy

Research on *creativity* can be traced back to Kirton's discussion about adapters and innovators (Kirton, 1976, 1978). In this theory, Kirton divides people into innovators and adaptors, according to the cognitive style adopted by them. Adaptors like to follow the rules, whereas innovators cannot stand routine and tend to be creative. At present, several researchers believe that creativity is a highly important skill (Pink, 2006; Van Harpen and Sriraman, 2013; Henriksen et al., 2019; Bicer et al., 2020). Simultaneously, studies have shown that a positive creative atmosphere contributes to students' cultivation of overall creative abilities (Liu et al., 2018).

The current international programming education for teenagers calls for attention to the influence of creativity on programming learning (Fragapane and Standl, 2021). Existing researches claimed there is a strong link between creativity and programming learning. Perez-Poch et al. (2016) discovered that high levels of creativity are associated with the excellence of learners in programming. Giannakos et al. (2013) proposed that developing creativity is an excellent way to promote and teach programming. Furtherly, Pardamean et al. (2011) revealed that it is very valuable for future research to explore the forecasting effect of creative thinking on students' programming education. Although many researchers have studied the relationship between creativity and programming education, the specific effect between creativity and programming self-efficacy is still not clear. Hence, we proposed a second hypothesis:

H2: Creativity has a positive effect on students' programming self-efficacy.

Mathematic Achievement and Creativity

Creativity is regarded as an important factor closely related to mathematic achievement, and many scholars attached importance to it. Previous works have demonstrated that mathematic education has an effect on students' creativity (Hwang et al., 2005; Livne and Milgram, 2006; Jeon et al., 2011; Kattou et al., 2013). Shen (2014) also indicated that students' creativity can be interpreted and enhanced in math learning. A survey of junior middle-school students also indicated that there is a general difference in scientific creation among students with poor, average and excellent mathematics scores (Jin, 2006). This means that students' mathematic achievement also has an impact on their creativity. Thus, this led us to propose a third hypothesis:

H3: Mathematic achievement has a positive effect on students' creativity.

Mediating Effect

The aforementioned studies showed that students' mathematic achievement is related to creativity, and creativity is also closely related to programming learning, which indicate that there may be a close relationship among them. The related research provides important enlightenment for us to study the relationship among the three. For instance, there have been studies

exploring the relationship between creativity, computer science and mathematics (Erdogan et al., 2008). Similarly, researchers have examined the relationship between the use of computer technology in interdisciplinary learning (STEM subjects) and creativity (Kuo et al., 2019). More importantly, as Kastl et al. (2017) proposed, students need mathematical knowledge to create programming works, as well as creative exposition. Pasini et al. (2017) also suggested that math learners' creative thinking has an impact on their programming performance. These studies provide clues for us to explore how mathematic achievement and creativity affect programming self-efficacy. Creativity may be a bridging factor between math learning and programming education. Hence, this study hypothesized the following:

H4: Creativity plays a mediating role between mathematic achievement's relationship with students' programming self-efficacy.

In view of this hypothesis, this study will build a model and verify the mediating effect (see **Figure 1**).

MATERIALS AND METHODS

Participants

The research participants were selected from a group of students with programming experience in high schools in Beijing, China. A total of 1,019 students from four different high schools took part in the survey. The participants who are ranging in age from 16 to 18, had one or two 45-min programming lessons a week at school. In addition, while taking into account the differences between the four schools, this study adopted a proportional random sampling method, with a 3:1 ratio of urban and suburban areas, and then selected four regions in Beijing, China.

On the day of data collection, students and their guardians provided written informed consent. students were told that their participation was voluntary and that there would be no consequences if they decided not to participate in the study. None of the participants were blind to the nature and contents of the experiment. Additionally,

there were professional teachers present, who helped students when they faced problems while filling in the questionnaire. This ensured that participants were explicitly involved in the study.

After removing any incomplete responses, the data came to 950 (93.23%) usable surveys. The statistical collation results are shown in **Table 1**. Of the 950 respondents, 504 (53.1%) were female and 446 (46.9%) were male. According to the school areas, 201 students were studying in the Xicheng (urban) District (21.16%), 105 were studying in the Haidian (urban) District (11.05%), 210 were studying in the Chaoyang (urban) District (22.11%), and 434 were studying in the Shunyi (suburban) District (45.68%).

Materials

The questionnaire used to collect data in this study contained three sections. In the first section, we collected demographic information, including participants' gender, age and certain family information. Moreover, with the consent of both teachers and students, we obtained students' recent mid-term math scores, including the specific scores and the levels. Considering that the experiment was carried out in different schools, math grades were based on their total scores and were divided into five levels. Scores ranging from 120–150, 105–119, 90–104, to 75–89 were classified as levels 5, 4, 3 and 2 respectively; scores under 75 indicated Level 1. In this way, students' mathematic achievements were outlined. In order to ensure the accuracy of the participant's recall of their math scores and grades, we actively communicated with teachers to ensure that the questionnaires were administered 1 week after the midterm exam. In addition, in order to ensure the reliability of the research data, the existing data were strictly screened and invalid and/or problematic data were removed.

The second section included the Programming Self-Efficacy Scale. Finally, the last section collected the Kirton Adaption-Innovation (KAI). The two scales used in this study were derived from their English versions and were translated using the back-translation method (Brislin, 1970). The back-translation method improves the accuracy of a given translation and involves the following process: a researcher translates the scale from English to Chinese, then a second researcher translates it back from the Chinese version and creates a reverse translated version. In the end, a third researcher compares and checks the three versions (the original, translated, and reverse translated versions) to ensure that there is equivalence between the original English and the translated Chinese versions.

The Programming Self-Efficacy Scale was initially developed by Kukul et al. (2017) to measure the programming self-efficacy
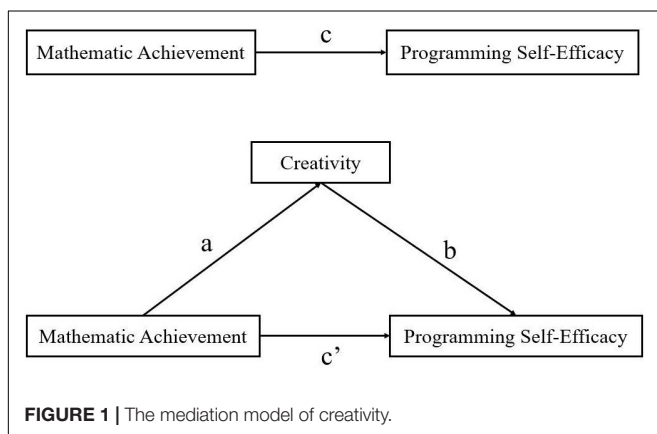

FIGURE 1 | The mediation model of creativity.

TABLE 1 | Descriptive statistical analysis for the observed variables.

Variables	Number of samples	Mean	SD
MA	950	3.100	1.760
PSE	950	3.237	1.128
Creativity	950	3.480	0.766

MA, mathematic achievement. PSE, programming self-efficacy.

of secondary school students. Afterward, this scale was adapted by Soykan and Kanbul (2018) to survey K12 students' coding self-efficacy. Therefore, this scale was used in this study to collect the computer programming self-efficacy of high school students. This scale consists of 31 items, measured on 5-point Likert-type scales, ranging from 1 point for "strongly disagree" to 5 points for "strongly agree". A score ranging from 1 to 2.49 means that one has low self-efficacy, with an average self-efficacy being measured at 2.50–3.49 points. Finally, achieving 3.50 points and higher reflects a high degree of self-efficacy. The Cronbach's alpha of the Programming Self-Efficacy Scale is 0.988, meaning that it has a good reliability. The sample items include: "When encountering a problem, I can distinguish whether the problem is suitable for programming," "I can break complex programming problems into small problems to solve," "I can choose the best solution to a programming problem."

The KAI developed by Kirton (1976) was used to measure students' creativity. Although KAI is primarily used to measure the style of creativity, which classifies learners as adaptors and innovators, several studies have reported a strong correlation between style of creativity and creativity level (Puccio and Chimento, 2001; Phelan and Young, 2003; Ee et al., 2007; Puccio et al., 2011). For example, the innovators generally have higher innovative thinking and creativity (Camacho-Minano and del Campo, 2017; Zhang H. et al., 2018). Therefore, KAI was used in this study to test students' creativity. The KAI consists of 32 items measured on a 5-point Likert-type scale, ranging from "strongly disagree" to "strongly agree." Finally, the average score is calculated and the higher the score, the higher the creativity. Cronbach's alpha of the KAI was 0.976, which indicated good reliability. The sample items include: "I have a lot of ideas when I encounter problems or things," "I think it is easier to create something new than to improve on something that is already there."

Data Analysis

SPSS-26 and the PROCESS plugin (version 3.3) were used to analyze the data. First, following a reliability and validity analysis, a descriptive analysis was carried out using SPSS. Second, a Pearson product-moment correlation analysis was used to check the relationships between mathematic achievement, programming self-efficacy and creativity, and a coefficient of variance inflation test was conducted. Finally, a mediation analysis, using the PROCESS plugin in SPSS, was conducted to explore the mediating effect of creativity, and to test our four hypotheses. The causal steps published by Baron and Kenny (1986) and the steps of the causal effect test summarized by Wen et al. (2004) were mainly used for testing the mediating effect. The method proposed by Baron and Kenny is popular for testing the mediating effect. However, some doubts about the causal steps arose after an increasing number of experts studied the method of the mediating effect test. Wen et al. (2004) verified several popular and effective methods and finally came up with a clear mediation effect testing method. His method was adopted in this study.

RESULTS

Descriptive Statistical Analysis and Correlation Analysis

Table 1 contains the descriptive analysis results of the students' mathematic achievement, programming self-efficacy, and creativity. As demonstrated in the table, the mean of mathematic achievement was 3.100 and the standard deviation was 1.760; the mean of programming self-efficacy was 3.237 and the standard deviation was 1.128. The mean of creativity was 3.480 and the standard deviation was 0.766.

Next, the correlation analysis among these three variables was conducted by calculating the Pearson product-moment correlation coefficient (see **Table 2**). First, mathematic achievement had a positive impact on programming self-efficacy, with a significant correlation ($r = 0.073$, $p < 0.05$). Second, a strong positive correlation was found between creativity and programming self-efficacy [$r = 0.746$, $p < 0.001$, VIF = 1.000 < 10 (there was no multicollinearity)]. Third, there was another significant positive correlation between mathematic achievement and creativity ($r = 0.084$, $p < 0.01$).

Mediation Analysis

The final hypothesis of this study involved examining the mediating effect of creativity. The PROCESS plugin (version 3.3) was used to analyze its mediating effect with mathematic achievement as the independent variable, programming self-efficacy as the dependent variable, and creativity as the mediating variable. The model (Model 4) was used to identify the role of creativity in the relationship between mathematic achievement and programming self-efficacy. The results (see **Table 3**) show that mathematic achievement significantly predicts programming self-efficacy ($B = 0.073$, $t = 2.259$, $p < 0.05$). In addition, mathematic achievement has a significant positive predictive effect on creativity ($B = 0.084$, $t = 2.599$, $p < 0.01$). Furthermore, creativity has a significant positive predictive effect on programming self-efficacy ($B = 0.746$, $t = 34.350$, $p < 0.001$). This means that, in the model (see **Figure 1**), there is a significant positive correlation in paths a–c. However, the direct influence of mathematic achievement on programming self-efficacy is not significant ($B = 0.011$, $t = 0.482$). We used the bias-corrected percentile Bootstrap method as proposed by Wen et al. (2004) to test the mediating effect. The Bootstrap test results indicated that the mediating effect of creativity had bootstrap confidence intervals (95%) with no zero between their lower and upper limits, while the direct effect of mathematic achievement on

TABLE 2 | Pearson correlations for the observed variables.

Variables	MA	PSE	Creativity
MA	1		
PSE	0.073**	1	
Creativity	0.084**	0.746**	1

MA, mathematic achievement; PSE, programming self-efficacy.
*$p < 0.05$; **$p < 0.01$.*

programming self-efficacy had bootstrap confidence intervals (95%) with zero between their lower and upper limits. This suggests that the model includes significant indirect effects, with the direct effect accounting for 14.33% and the indirect effect accounting for 85.67% (see **Table 4**).

DISCUSSION

Discussion of the Results

The results of this study support its hypotheses and are consistent with the findings of previous research.

Firstly, the study results support its first hypothesis in that they reveal a significantly positive correlation between mathematic achievement and programming self-efficacy. As shown in the "Results" section, there is a high or low correlation among these variables, as well as a relatively small effect size. This is due to the peculiar context. Researchers have indicated that computer science is largely a field of applied mathematical principles (Papert, 1972), and that there is a strong link between maths and programming (Sauter, 1986). Research has revealed that there is a connection between mathematics and programming, which has a lot to do with the nature of the subject itself. Mathematical ability has been considered as one of the necessary abilities for programming learning activities. The abilities of reasoning, logical thinking and basic computing required in programming learning all need a mathematical base. Mathematics not only represents a kind of ability, but also a way of thinking (Liu, 2019). This is in alignment with the study conducted by Kattou et al. (2013), which stated that "computer science is evolving along mathematical lines;" this statement shows how important mathematic abilities are for computer science students (Fajrina et al., 2020). Certain computer science educators have also explained this by outlining that mathematical thinking is a method of computer education (Mcmaster et al., 2010). This means that mathematics is a necessary ability for programming learning, because the inherent reasoning, logical thinking, and problem-solving abilities required by programming all rely on mathematics. Additionally, the positive correlation between mathematic and programming scores has also been verified (Razak and Ismail, 2018). This suggests that students who perform well in maths are more likely to have a higher programming self-efficacy. Therefore, many

TABLE 4 | Total, direct and indirect effects among the variables.

	Effect size	Boot SE	Boot CI lower limit	Boot CI upper limit	Relative effect size
Total effect	0.0586	0.0261	0.0078	0.1107	100.00%
Direct effect	0.0084	0.0181	−0.0268	0.0437	14.33%
Indirect effect	0.0502	0.0192	0.0125	0.088	85.67%

Boot SE, boot standard error; Boot CI, boot confidence interval; Lower limit and upper limit, the lower limit and upper limit of the 95% boot confidence interval; Relative Effect Size, direct effect size or indirect effect size divided by the total effect size.

researchers have developed programming courses that integrate mathematical thinking to ensure that students achieve a higher level of programming self-efficacy through the process of using their mathematical thinking skills (Soloway, 1993; Taylor et al., 2010).

Secondly, the results of this study support its second hypothesis that states that there is a significantly positive correlation between creativity and programming self-efficacy. First, (Kirton, 1976) analyzed the differences between innovator and adaptor thinking styles. Innovators tend to believe that novel talents are possessed by people with the "correct" form of creativity, and thus they tend to have a higher programming self-efficacy. This can be explained by the fact that innovators tend to maintain a higher programming self-efficacy when learning this skill. Creativity is crucial to computer learning (Liu et al., 2011) because students cannot learn computer programming without forming creative new ideas (Kobsiripat, 2015). Furthermore, one study revealed that high levels of creativity are associated with superior programming achievements (Perez-Poch et al., 2016).

Thirdly, the results of this study support its third hypothesis that states that there is a significant correlation between mathematic academic achievement and creativity. We can analyze the reasons for this conclusion from examining the characteristics of creativity. First, both innovators and adaptors are achievement-oriented in their learning, but they have different goals. Innovators are more pioneering and divergent in their goal setting, aiming at proficiency, while adaptors are more risk-averse and have clearer self-goals (Ee et al., 2007). Simultaneously, as mentioned earlier, mathematical learning requires more logical thinking and divergent abilities, which is consistent with the goals of innovators. Leikin (2007) also indicated, in their study, that students were more likely to generate innovative ideas from multiple perspectives when solving complex mathematical problems. Several math competitions are aimed at cultivating students' creativity in the process of applying mathematical knowledge (Kovari and Rajcsanyi-Molnar, 2020). It is generally believed that students with higher mathematic academic performance are more creative than those with lower performances in this area (Manchanda and Sood, 2012). Meanwhile, other researchers have demonstrated a significant positive correlation between math achievement and creativity. Therefore, students with higher mathematical achievements are more likely to be trailblazers and innovators.

TABLE 3 | Mediation analysis results for the observed variables.

Regression equation		Fitting indices				Significance	
Outcome variables	Predictor variables	R	R^2	F(df)	Â	T
Creativity	MA	0.084	0.007	6.755**	0.084	2.599**
PSE	Creativity	0.746	0.557	595.674***	0.746	34.350***
	MA				0.011	0.482
PSE	MA	0.073	0.005	5.103*	0.073	2.259*

MA, mathematic achievement; PSE, programming self-efficacy.
*$p < 0.05$; **$p < 0.01$; ***$p < 0.001$.*

Fourthly, the findings of this study also support its fourth hypothesis, which states that creativity plays a mediating role in the relationship between mathematic achievement and programming self-efficacy, which is consistent with the findings of prior research. First, students with higher mathematic achievements tend to be more creative, which is related to the flexibility and novelty inherent in mathematical thinking (Ayllón et al., 2016). Second, creativity is essential in learning computer programming. As such, there is a connection between creativity and the two variables analyzed in this study. Finally, we verified that the direct effect of mathematic achievement and programming self-efficacy was not significant, but that the effect is caused by creativity as a mediating variable. For example, students will constantly develop their creativity in the process of mathematical learning and will need to solve complex logical problems from multiple perspectives (Leikin, 2007). Simultaneously, programming is also a process of design creation and imagination (Romero et al., 2017). Learning computer programming can thus help students to understand mathematical concepts, as well as develop both logical thinking and creativity, thus enabling them to have a higher degree of programming self-efficacy (Park and Hong, 2009). Therefore, mathematic achievement, creativity, and programming self-efficacy are closely related.

Implications

In a theoretical sense, this study complements previous research which examined the relationship between mathematics, programming learning, and creativity. On one hand, this study comprehensively explored the positive influence between mathematic achievement and programming self-efficacy. Additionally, the study also tested the mediating effect of creativity through a mediation analysis. This provides theoretical support and inspiration for an integrated study design of mathematics and programming education in the future.

In a more practical sense, this study provides a new way of thinking for the interdisciplinary curriculum integration and teaching design of programming education. First, interesting programming languages should be fully utilized to stimulate K-12 students' interest, and the integration of mathematics and programming education should be strengthened. This can help cultivate students' logical analysis and problem-solving abilities, which are also needed in mathematic learning. Second, students' personalities and characteristics should be considered in programming education. Teachers should get to know students and improve their teaching skills from the perspective of those learning. Teachers provide different teaching resources and teaching strategies for students with different levels of creativity, so as to promote a better teaching effect. In programming education, teachers can help students' programming self-efficacy through innovative learning activities to cultivate students' confidence and interest in programming. The most important thing is that teachers should constantly explore the links between disciplines and programming education, strengthen the integration of disciplines and cultivate students' higher-order thinking in teaching practice.

Limitations and Directions for Future Research

This study has some limitations. First, the research participants of this study were all upper-secondary high-school students in Beijing, which is one of the regions with the fastest development of and superior conditions enabling programming education. This may influence the generalization performance of the findings. In future studies, students from other regions and learning stages should be involved. Second, this study utilized a cross-sectional design, meaning that long-term research is required in future. Third, the measurement of students' creativity in this study can adopt more diversified measurement tools. For example, use KAI for style differentiation of creativity, and combine with the Torrance Tests of Creative Thinking (TTCT; Torrance, 1974) to comprehensively measure students' creativity performance.

As for the future research, first of all, how to better improve the effect of programming education through the role of mathematics and creativity needs further research. Because this study results indicate that mathematic achievement and creativity can affect programming self-efficacy, and programming self-efficacy is crucial to programming education. Thus, how teachers and students actually use this relationship to improve students' programming self-efficacy and promote programming learning needs to be considered. Second, there may still be a number of factors that have not been considered in this study. In future studies, other possible variables, such as learning methods and interest in programming, should be studied in the area of adolescent programming. More importantly, with the new requirements in the development of education in the current era, programming and innovative education methods are gaining greater levels of importance in the field of teaching. In future, the methods to promote programming education and innovative development more effectively is a topic worth exploring.

CONCLUSION

In the present era of AI, programming has become a crucial skill closely related to people's life and survival. In recent years, a number of policies have promoted the popularization of programming education from a younger age. However, programming learning itself has certain difficulties, especially for young programming students. Therefore, it is extremely important to know how to effectively improve students' programming learning. As one of the important factors in programming learning, programming self-efficacy needs attention. Considering the possible effects of both mathematics and creativity on programming, this study established a model to explore the relationship among mathematic achievement, programming self-efficacy and creativity. To conclude, the correlation analysis suggests that mathematic achievement has a positive effect on programming self-efficacy. Further, the mediation analysis confirmed the mediating role of creativity in this relationship. Due to the mediation of creativity, students with higher achievements

in mathematics tend to have a higher programming self-efficacy. The results of this study provide insights for the future development of programming education for adolescents. Both teachers and students need to pay attention to the integration of subjects and the cultivation of higher order thinking, with the help of mathematical and innovative thinking to promote programming learning.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the Ethics Committee of Capital Normal University. Written informed consent to participate in this study was provided by the participants' legal guardian/next of kin.

AUTHOR CONTRIBUTIONS

JL and YZ designed the research. MS, YD, and FX performed the literature search and data analysis. JL, YZ, and XS reviewed and revised the manuscript. All the authors have commented on previous editions, read, and approved the final draft of the manuscript.

FUNDING

ACKNOWLEDGMENTS

REFERENCES

Abdunabi, R., Hbaci, I., and Ku, H.-Y. (2019). Towards enhancing programming self-efficacy perceptions among undergraduate information systems students. *J. Inform. Technol. Education-Res.* 18, 185–206. doi: 10.28945/4308

Affuso, G., Bacchini, D., and Miranda, M. C. (2017). The contribution of school-related parental monitoring, self-determination, and self-efficacy to academic achievement. *J. Educ. Res.* 110, 565–574. doi: 10.1080/00220671.2016.1149795

Akinola, S. O. (2015). Computer programming skill and gender difference: an empirical study. *J. Sci. Industrial Res.* 7, 1–9. doi: 10.5251/ajsir.2016.7.1.1.9

Altun, A., and Mazman, S. (2012). Developing computer programming self-efficacy scale. *J. Measurement Eval. Educ. Psychol.* 3, 297–308.

Arslan, K., and Tanel, Z. (2020). Analyzing the effects of Arduino applications on students' opinions, attitude and self-efficacy in programming class. *Educ. Inform. Technol.* 26, 1143–1163. doi: 10.1007/s10639-020-10290-10295

Ayllón, M. F., Gómez, I. A., and Ballesta-Claver, J. (2016). Mathematical thinking and creativity through mathematical problem posing and solving. *Propositos Representaciones* 4, 195–218. doi: 10.20511/pyr2016.v4n1.89

Ayotola, A., and Adedeji, T. (2009). The relationship between mathematics self-efficacy and achievement in mathematics. *Procedia-Soc. Behav. Sci.* 1, 953–957. doi: 10.1016/j.sbspro.2009.01.169

Bandura, A. (1977). Self-efficacy: toward a unifying theory of behavioral change. *Adv. Behav. Res. Therapy* 1, 139–161. doi: 10.1037//0033-295x.84.2.191

Baron, R. M., and Kenny, D. A. (1986). The moderator-mediator variable distinction in social psychological research: conceptual, strategic, and statistical considerations. *J. Pers. Soc. Psychol.* 51, 1173–1182. doi: 10.1037//0022-3514.51.6.1173

Bicer, A., Chamberlin, S., and Perihan, C. (2020). A meta-analysis of the relationship between mathematics achievement and creativity. *J. Creat. Behav.* 55, 569–590. doi: 10.1002/jocb.474

Bocconi, S., Chioccariello, A., and Earp, J. (2018). The Nordic approach to introducing computational thinking and programming in compulsory education. Report prepared for the Nordic@ BETT2018 New Delhi: Steering Group. doi: 10.17471/54007

Brislin, R. W. (1970). Back-translation for cross-cultural research. *J. Cross-Cultural Psychol.* 1, 185–216. doi: 10.1177/135910457000100301

Buzzai, C., Filippello, P., Puglisi, B., Mafodda, A. V., and Sorrenti, L. (2020). The relationship between mathematical achievement, mathematical anxiety, perfectionism and metacognitive abilities in Italian students. *Med. J. Clin. Psychol.* 8, 1–18. doi: 10.6092/2282-1619/mjcp-2595

Camacho-Minano, M. D., and del Campo, C. (2017). The role of creativity in entrepreneurship: an empirical study on business undergraduates. *Educ. Train.* 59, 672–688. doi: 10.1108/et-08-2016-2132

Cheng, C.-C., Huang, P. L., and Huang, K. H. (2013). Cooperative learning in lego robotics projects: exploring the impacts of group formation on interaction and achievement. *J. Networks* 8, 1529–1535.

Compeau, D. R., and Higgins, C. A. (1995). Computer self-efficacy: development of a measure and initial test. *MIS Quarterly* 19, 189–211. doi: 10.2307/249688

Davidsson, K., Larzon, L. K., and Ljunggren, K. (2010). *Self-efficacy in Programming Among STS Students*. Seattle, WA: Semantic Scholar

Dufva, T. (2018). *Art Education in the Post-digital Era-Experiential Construction of Knowledge through Creative Coding*. Finland: Aalto University.

Ee, J., Seng, T. O., and Kwang, N. A. (2007). Styles of creativity: adaptors and innovators in a Singapore context. *Asia Pacific Educ. Rev.* 8, 364–373. doi: 10.1007/BF03026466

Erdogan, Y., Aydin, E., and Kabaca, T. (2008). Exploring the psychological predictors of programming achievement. *J. Instructional Psychol.* 35, 264–270.

Europe Commission (2014). *CodeWeek*. Brussels: Europe Commission.

Fajrina, S., Lufri, L., and Ahda, Y. (2020). Science, technology, engineering, and mathematics (STEM) as a learning approach to improve 21st century skills: a review. *Int. J. Online Biomed. Eng.* 16, 95–104. doi: 10.3991/ijoe.v16i07.14101

Feurzeig, W., Papert, S. A., and Lawler, B. (2011). Programming-languages as a conceptual framework for teaching mathematics. *Interact. Learn. Environ.* 19, 487–501. doi: 10.1080/10494820903520040

Fragapane, V., and Standl, B. (2021). "Work in progress: creative coding and computer science education - from approach to concept," in *Proceedings of the IEEE Global Engineering Education Conference*, (Piscataway, NJ: IEEE).

Friend, M., Matthews, M., Winter, V., Love, B., Moisset, D., and Goodwin, I. (2018). "Bricklayer: elementary students learn math through programming and art," in *Proceedings of the 49th ACM SIGCSE Technical Symposium on Computer Science*

Education (SIGCSE), (New York, NY: ACM), 628–633. doi: 10.1145/3159450. 3159515

Garner, S. (2009). "Learning to program from Scratch," in *Proceedings of the 2009 9th IEEE International Conference on Advanced Learning Technologies*, (Piscataway, NJ: IEEE), 451–452. doi: 10.1109/ICALT.2009.50

Giannakos, M. N., Jaccheri, L., and Proto, R. (2013). *Teaching Computer Science to Young Children Through Creativity: Lessons Learned from the Case of Norway*: Raipur: CSERC

Govender, I. (2007). Experiences of learning and teaching: problem solving in computer programming. *African J. Res. Mathemat. Ence Technol. Educ.* 11, 39–50. doi: 10.1080/10288457.2007.10740620

Heggestad, E. D., and Kanfer, R. (2005). The predictive validity of self-efficacy in training performance: little more than past performance. *J. Exp. Psychology-Appl.* 11, 84–97. doi: 10.1037/1076-898x.11.2.84

Henriksen, D., Creely, E., and Henderson, M. (2019). Failing in creativity: the problem of policy and practice in Australia and the United States. *Kappa Delta Pi Record* 55, 4–10. doi: 10.1080/00228958.2019.1549429

Hongwarittorrn, N., and Krairit, D. (2010). "Effects of program visualization (jeliot3) on students' performance and attitudes towards java programming," in *Proceedings of the Spring 8th International Conference on Computing, Communication and Control Technologies*, (Orlando, FL).

Hwang, D. J., Lee, K. S., and Seo, J. J. (2005). Relationship between divergent thinking in mathematical and non-mathematical situations-based on the TTCT; Figural A and the MCPSAT. *J. Gifted/Talented Educ.* 15, 59–76. doi: 10.1177/1356262215590049

Hwang, M. H., Choi, H. C., Lee, A., Culver, J. D., and Hutchison, B. (2016). The relationship between self-efficacy and academic achievement: a 5-year panel analysis. *Asia-Pacific Educ. Res.* 25, 89–98. doi: 10.1007/s40299-015-0236-233

Jacob, S. M. (2012). Mathematical achievement and critical thinking skills in asynchronous discussion forums. *Procedia-Social Behav. Sci.* 31, 800–804. doi: 10.1016/j.sbspro.2011.12.144

Jenkins, T. (2002). *On the Difficulty of Learning to Program*. Woodhouse: University of Leeds.

Jeon, K. N., Moon, S. M., and French, B. (2011). Differential effects of divergent thinking, domain knowledge, and interest on creative performance in art and math. *Creat. Res. J.* 23, 60–71. doi: 10.1080/10400419.2011.545750

Jin, L. (2006). Investigation and research on science creativity of junior middle school students with different mathematics academic level. *Theory Pract. Educ.* 26, 39–41.

Kastl, P., Krisch, O., and Romeike, R. (2017). "3D printing as medium for motivation and creativity in computer science lessons," in *Proceedings of the International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, (Berlin: Springer International Publishing), doi: 10.1007/978-3-319-71483-7_3

Kattou, M., Kontoyianni, K., Pitta-Pantazi, D., and Christou, C. (2013). Connecting mathematical creativity to mathematical ability. *ZDM* 45, 167–181. doi: 10. 1007/s11858-012-0467-461

Kirton, M. (1976). Adaptors and innovators: a description and measure. *J. Appl. Psychol.* 61, 622–629. doi: 10.1037/0021-9010.61.5.622

Kirton, M. (1978). Have adaptors and innovators equal levels of creativity? *Psychol. Rep.* 42, 695–698. doi: 10.2466/pr0.1978.42.3.695

Kobsiripat, W. (2015). Effects of the media to promote the Scratch programming capabilities creativity of elementary school students. *Procedia - Soc. Behav. Sci.* 174, 227–232. doi: 10.1016/j.sbspro.2015.01.651

Korkmaz, Ö (2012). The impact of critical thinking and logico-mathematical intelligence on algorithmic design skills. *J. Educ. Comp. Res.* 46, 173–193. doi: 10.2190/EC.46.2.d

Kovari, A., and Rajcsanyi-Molnar, M. (2020). Mathability and creative problem solving in the MaTech math competition. *Acta Polytechnica Hungarica* 17, 147–161. doi: 10.12700/aph.17.2.2020.2.9

Kucuk, S., and Sisman, B. (2017). Behavioral patterns of elementary students and teachers in one-to-one robotics instruction. *Comp. Educ.* 111, 31–43. doi: 10. 1016/j.compedu.2017.04.002

Kukul, V., Gökçearslan, S., and Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: development, validation and reliability. *Eŏitim Teknolojisi Kuram ve Uygulama* 7, 158–158. doi: 10.17943/ETKU.288493

Kuo, H. C., Tseng, Y., and Yang, Y. C. (2019). Promoting college student's learning motivation and creativity through a STEM interdisciplinary PBL human-computer interaction system design and development course. *Think. Skills Creat.* 31, 1–10. doi: 10.1016/j.tsc.2018.09.001

Leikin, R. (2007). "Habits of mind associated with advanced mathematical thinking and solution spaces of mathematical tasks," in *Proceedings of the 5th Conference of the European Society for Research in Mathematics Education - CERME*, (Atlanta : Gerogia State University).

Liu, D. (2019). Study on influencing factors of programming course teaching. *Think Tank Era* 6, 170–175.

Liu, W.-S., Wu, Y.-T., and Wu, T.-T. (2018). "The study of creativity, creativity style, creativity climate applying creativity learning strategies - An example of engineering education," in *Innovative Technologies and Learning, ICITL 2018*, eds T. T. Wu, Y. M. Huang, R. Shadiev, L. Lin, and A. I. Starcic (Cham: Springer).

Liu, Y.-M., Xie, T.-Q., and Cheng, S. (2011). "Using creativity to boost computer science course learning in higher education. intelligent information technology application association," in *Proceedings of the 2011 International Conference on Applied Social Science (ICASS 2011 V1)*, (Spain).

Livne, N. L., and Milgram, R. M. (2006). Academic versus creative abilities in mathematics: two components of the same construct? *Creat. Res. J.* 18, 199–212. doi: 10.1207/s15326934crj1802_6

Manchanda, R., and Sood, R. (2012). A study of relationship between creativity and achievement of students in mathematics. *Int. J. Knowl. Res. Manag. E-Commer.* 2. Available online at: http://rgjournals.com/index.php/ijkrm/article/view/338/ 161

Mathews, D. K. (2017). *Predictors of Success in Learning Computer Programming*. Fountain Valley, CA: University of Rhode Island.

Mcmaster, K., Rague, B., and Anderson, N. (2010). "Integrating mathematical thinking, abstract thinking, and computational thinking," in *Proceedings of the 2010 IEEE Frontiers in Education Conference, S3G-1-S3G-6*, (Piscataway, NJ: IEEE).

Mijung, Y., Myunghee, P., Seokwon, Y., and Kyung-Sik, K. (2018). Structural relationship among factors influencing academic achievement by gender of high school students in Daegu -Focusing on the mediating effects of satisfaction of relationship with teachers, self-efficacy and class attitude. *Secondary Educ. Res.* 66, 195–227. doi: 10.25152/ser.2018.66.1.195

Ministry of Education of the People's Republic of China (2019). *Notice of the General Office of the Ministry of Education on Printing and Distributing the "Key Points of Education Informatization and Cybersecurity in 2019"*. Office OF the Ministry of Education: New Delhi

Noh, J., and Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educ. Technol. Res. Dev.* 68, 463–484. doi: 10.1007/s11423-019-09708-w

Papert, S. A. (1972). Teaching children to be mathematicians versus teaching about mathematics. *Int. J. Mathemat. Educ. Sci. Technol.* 3, 249–262. doi: 10.1080/ 0020739700030306

Pardamean, B., Evelin, E., and Honni, H. (2011). "The effect of Logo programming language for creativity and problem solving," in *Proceedings of the 10th Wseas International Conference on E-activities*, (New York, NY).

Park, K. M., and Hong, T. J. (2009). A study on object-oriented programming education for improving logical thinking ability of elementary school students. *J. Digital Contents Soc.* 10, 367–373.

Pasini, M., Solitro, U., Brondino, M., Burro, R., Raccanello, D., and Zorzi, M. (2017). "Psychology of programming: the role of creativity, empathy and systemizing," in *Proceedings of the 7th International Conference in Methodologies and Intelligent Systems for Technology Enhanced Learning (MIS4TEL)*, (Cham.: Springer).

Perez-Poch, A., Olmedo Torre, N., Sanchez, F., Salan, N., and Lopez, D. (2016). On the influence of creativity in basic programming learning in a first-year engineering course. *Int. J. Eng. Educ.* 32, 2302–2309.

Phelan, S., and Young, A. M. (2003). Understanding creativity in the workplace: an examination of individual styles and training in relation to creative confidence and creative self-leadership. *J. Creat. Behav.* 37, 266–281. doi: 10.1002/j.2162-6057.2003.tb00994.x

Pink, D. H. (2006). *A Whole New Mind : Why Right-Brainers will Rule the Future*. London: Penguin.

Puccio, G. J., and Chimento, M. (2001). Implicit theories of creativity: laypersons' perceptions of the creativity of adaptors and innovators. *Perceptual Motor Skills* 92(3 Pt 1), 675–681. doi: 10.2466/pms.2001.92.3.675

Puccio, G. J., Talbot, R. J., and Joniak, A. J. (2011). Examining creative performance in the workplace through a person-environment fit model. *J. Creat. Behav.* 34, 227–247. doi: 10.1002/j.2162-6057.2000.tb01213.x

Ramalingam, V., and Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *J. Educ. Comp. Res.* 19, 367–381. doi: 10.2190/C670-Y3C8-LTJ1-CT3P

Razak, M. R. B., and Ismail, N. Z. B. (2018). Influence of mathematics in programming subject. *AIP Conference Proc.* 1974:050011. doi: 10.1063/1.5041711

Repenning, A., and Ioannidou, A. (2008). Broadening participation through scalable game design. *ACM SIGCSE Bull.* 40, 305–309. doi: 10.1145/1352322.1352242

Rohatgi, A., Scherer, R., and Hatlevik, O. E. (2016). The role of ICT self-efficacy for students' ICT use and their achievement in a computer and information literacy test. *Comp. Educ.* 102, 103–116. doi: 10.1016/j.compedu.2016.08.001

Romero, M., Lepage, A., and Lille, B. (2017). Computational thinking development through creative programming in higher education. *Int. J. Educ. Technol. Higher Educ.* 14:42. doi: 10.1186/s41239-017-0080-z

Rubio, M., Hierro, C., and Pablo, Á (2013). "Using Arduino to enhance computer programming courses in science and engineering," in *Proceedings of the Edulearn*, (Spain: IATED Barcelona).

Sandberg, E. (2019). *Creative Coding on the web in p5. js: a Library where JavaScript Meets Processing.* Sweden: Blekinge Institute of Technology

Sauter, V. L. (1986). Predicting computer programming skill. *Comp. Educ.* 10, 299–302. doi: 10.1016/0360-1315(86)90031-x

Scammacca, N., Fall, A.-M., Capin, P., Roberts, G., and Swanson, E. (2020). Examining factors affecting reading and math growth and achievement gaps in grades 1-5: a cohort-sequential longitudinal approach. *J. Educ. Psychol.* 112:718. doi: 10.1037/edu0000400

Serpe, A. (2019). "Programming with scratchjr to create interactive maths - experiences in preschool," in *Proceedings of the 13th International Technology, Education and Development Conference (INTED), 2007-2016*, (Valencia: IATED Academy).

Shen, Y. (2014). *Elementary School Teachers' Interpretation and Promotion of Creativity in the Learning of Mathematics: A Grounded Theory Study.* Lincoln, NE: University of Nebraska - Lincoln.

Soloway, E. (1993). Should we teach students to program? *Commun. ACM* 36, 21–24. doi: 10.1145/163430.164061

Soykan, F., and Kanbul, S. (2018). Analysing K12 students' self-efficacy regarding coding education. *TEM J.* 7, 182–187. doi: 10.18421/TEM71-22

StartupSmart. (2015). *Coding to be Taught in Australian Schools.* New York, NY: Mashable.

Sun, L., and Zhou, D. (2019). Research status and action path on international children programming education. *Open Educ. Res.* 25, 23–35. doi: 10.13966/j.cnki.kfjyyj.2019.02.003

Taylor, M., Harlow, A., and Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Proc. Soc. Behav. Sci.* 8, 561–570. doi: 10.1016/j.sbspro.2010.12.078

Tomai, E., and Reilly, C. F. (2014). "The impact of math preparedness on introductory programming (CS1) success," in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, (New York, NY: ACM), 711–711.

Torrance, E. P. (1974). *The Torrance Tests of Creative Thinking-Norms-Technical Manual Research Edition-Verbal Tests, Forms A and B- Figural Tests, Forms A and B.* Princeton, NJ: Personnel Press.

Tsai, C. Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: the role of self-efficacy. *Comp. Hum. Behav.* 95, 224–232. doi: 10.1016/j.chb.2018.11.038

Van Harpen, X. Y., and Sriraman, H. B. (2013). Creativity and mathematical problem posing: an analysis of high school students' mathematical problem posing in China and the USA. *Educ. Stud. Mathemat.* 82, 201–221. doi: 10.1007/s10649-012-9419-9415

Wen, Z. L., Chang, L., Hau, K. T., and Liu, H. Y. (2004). Testing and application of the mediating effects. *Acta Psychol. Sinica* 36, 614–620. doi: 10.1007/BF02911031

White, G. (2003). Standardized mathematics scores as a prerequisite for a first programming course. *Mathemat. Comp. Educ.* 37, 96–104.

White, G., and Sivitanides, M. (2003). An empirical investigation of the relationship between success in mathematics and visual programming courses. *J. Inform. Systems Educ.* 14, 409–416.

Yaratan, H., and Kasapoğlu, L. (2012). Eighth grade students' attitude, anxiety, and achievement pertaining to mathematics lessons. *Proc. Soc. Behav. Sci.* 46, 162–171. doi: 10.1016/j.sbspro.2012.05.087

Yukselturk, E., and Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *Br. J. Educ. Technol.* 48, 789–801. doi: 10.1111/bjet.12453

Zhang, C., and Chen, G. (2016). "The reform of teaching mode for algorithm courses in computer specialty," in *Proceedings of the 2016 2nd International Conference on Education, Social Science, Management and Sports*, eds Z. Wei and D. Guo (Amsterdam: Atlantis Press).

Zhang, H., Wang, L., and Zhang, M. (2018). The relations among creative cognitive style, creative personality, and creative thinking. *Psychol. Behav. Res.* 16, 51–57.

Zhang, Y., Liu, H., Wang, M., and Qing, P. (2018). The impact of challenge stress and hindrance stress on employee creativity: the mediating role of self-efficacy and the moderating role of justice. *Acta Psychol. Sinica* 50, 450–461. doi: 10.3724/SP.J.1041.2018.00450

Zhou, Z., Li, M., and Zhao, D. (2006). Social competence and academic achievement in middle childhood: a cross-lagged regression analyses. *Psychol. Sci.* 29, 1071–1075. doi: 10.16719/j.cnki.1671-6981.2006.05.011

Zimmerman, B. J. (1990). Self-regulated learning and academic achievement: an overview. *Educ. Psychol.* 25, 3–17. doi: 10.1207/s15326985ep2501_2