Accepted Manuscript

A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education

Siu-Cheung Kong, Ming Ming Chiu, Ming Lai

PII: S0360-1315(18)30236-7

DOI: 10.1016/j.compedu.2018.08.026

Reference: CAE 3443

To appear in: *Computers & Education*

Received Date: 24 June 2018

Revised Date: 27 August 2018

Accepted Date: 30 August 2018

Please cite this article as: Kong S.-C., Chiu M.M. & Lai M., A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education, *Computers & Education* (2018), doi: 10.1016/j.compedu.2018.08.026.

Title

A Study of Primary School Students' Interest, Collaboration Attitude, and Programming Empowerment in Computational Thinking Education

Name and affiliation of Authors

Prof. Siu-Cheung KONG*

Professor of Department of Mathematics and Information Technology

Director of Centre for Learning, Teaching and Technology

The Education University of Hong Kong

10 Lo Ping Road, Tai Po, New Territories, Hong Kong.

Tel: +852 29487646 Fax: +852 29487046

E-mail address: sckong@eduhk.hk

*Corresponding Author

Prof. Siu-Cheung KONG*

Prof. Ming Ming CHIU

Chair Professor of Department of Special Education & Counselling

The Education University of Hong Kong

10 Lo Ping Road, Tai Po, New Territories, Hong Kong.

Tel: +852 29488602

E-mail address: mingchiu@eduhk.hk

Dr. Ming LAI

Post-doctoral Fellow of the Centre for Learning, Teaching and Technology

The Education University of Hong Kong

10 Lo Ping Road, Tai Po, New Territories, Hong Kong.

Tel: +852 29488025

E-mail address: mlai@eduhk.hk

A Study of Primary School Students' Interest, Collaboration Attitude, and Programming Empowerment in Computational Thinking Education

Abstract

Building on Seymour Papert's view of empowering students by mastering programming, this study conceptualized programming empowerment as consisting of four components: meaningfulness, impact, creative self-efficacy, and programming self-efficacy. A sample of 287 primary school students in grades four to six completed a corresponding survey. Confirmatory factor analysis validated the proposed components of the programming empowerment instrument. A structural equation model indicated that students with greater interest in programming perceived it as more meaningful, had greater impact, had greater creative self-efficacy, and had greater programming self-efficacy. Also, students with attitudes toward collaboration that were more positive than others had greater creative self-efficacy. Boys showed more interest in programming than girls did. Students in higher grade levels than others viewed programming as less meaningful and had lower programming self-efficacy. These results support future studies that evaluate the impacts of interest-driven computational thinking and programming curricula with ample collaboration opportunities.

Keywords: collaboration attitude; computational thinking education; programming empowerment; student interest; self-efficacy

1. Introduction

Seymour Papert founded the idea of computational thinking and highly valued its power of empowerment, arguing that "[we] can give children unprecedented power to invent and carry on exciting projects by providing them with access to computers, with a suitably clear and intelligent programming language and with peripheral devices capable of producing on-line-real-time action" (Papert, 1972: p.245). However, Papert did not provide an instrument to measure children's empowerment and did not identify its antecedents beyond provision of access to computers. In this study, we define programming empowerment and test an explanatory model of it.

Guided by Thomas and Velthouse (1990), this study defines programming empowerment as a person's perceived autonomy and competence to use computational thinking effectively. Computational thinking (CT) "is the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts" (Shute, Sun & Asbell-Clarke, 2017, p. 151), and can be cultivated with engagement in programming activities (Lye & Koh, 2014). We propose that programming empowerment comprises four components: meaningfulness, impact, creative self-efficacy, and programming self-efficacy. Furthermore, we test whether student interest in programming (e.g., Weber & Patterson, 2000), attitude towards collaboration (Denner, Werner, Campe, & Ortiz, 2014; Lewis, 2011) and demographics such as gender and grade level are related to programming empowerment. As students have different degrees of interest in computers and they often work together in class, we address the following research questions: (a) do students with greater interest in computers have greater programming empowerment? (b) do students with more positive attitudes toward collaboration have greater programming empowerment?

2. Conceptual Framework of the Study

2.1. Importance of computational thinking in K-12 education

Wing (2006) argued that CT is an essential skill for everyone, like reading, writing, and arithmetic competencies. Indeed, many countries have incorporated CT into their K-12 education (Angeli et al., 2016; Grover & Pea, 2013; Voogt, Fisser, Good, Mishra, & Yadav, 2015; Webb et al., 2017). CT is a way to think and act, via skills such as decomposition, abstraction, generalization, algorithmic design, debugging, and iteration (Shute et al., 2017). Block-based programming environments such as App Inventor (Wolber, Abelson, Spertus, & Looney, 2015) and Scratch (Resnick et al., 2009) can support young learners cultivate CT during programming activities by enabling them to focus more on the problem solving process via dragging and dropping blocks, as they learn. Brennan and Resnick (2012) proposed a framework that conceptualized CT with three dimensions: 1) CT concepts, which involves learning of programming constructs (sequence, loops, conditionals, etc.); 2) CT practices, which involves uses of CT concepts (reusing and remixing, iterative and incremental, and testing and debugging, etc.); and 3) CT perspectives, which involve students' understandings of themselves, the technological world, and their relationships with one another. As few studies examine CT perspectives (Lye & Koh, 2014), this study does so by defining and modeling the components of students' programming empowerment.

2.2. Programming empowerment and its components

Originating from the notion of liberating the oppressed through education (Freire, 1973), empowerment emphasizes giving people control over their lives and issues of concern to them (Page & Czuba, 1999; Rappaport, 1987). While some researchers view empowerment as a single component (self-determination, Sprague & Hayes, 2000; competence, Breton, 1994; self-confidence, Larson, Walker, & Pearce, 2005), others define multiple components of empowerment (e.g., Becker, Kovach, & Gronseth, 2004; Spreitzer, Kizilos, & Nason, 1997). In particular, Thomas and Velthouse (1990) view empowerment as consisting of four components: meaningfulness, impact, self-determination/choice, and competence (see also Hur, 2006; Spreitzer, 1995). In the context of learning, Frymier, Shulman, and Houser (1996) operationalized these four components to examine learner empowerment, with the identification of three factors: meaningfulness, impact, and competence, while the factor of self-determination/choice did not emerge (see also Kirk, Lewis, Brown, Karibo, & Park, 2016; You, 2016). Frymier et al. (1996) explained that in a classroom context, students usually have little power to determine the activities to be conducted, which teachers typically control (unlike work contexts, in which employees might have more control over their activities, Spreitzer et al., 1997).

Although primary school students often cannot choose their programming activities, they can approach them either routinely by following instructions or creatively with novel approaches (e.g., Deschryver & Yadav, 2015; Voogt et al., 2015). Papert (1972) argued that as children master programming skills, they are empowered to be creative. Brennan and Resnick (2012) also emphasized that through the development of CT, students could express themselves creatively. Following this line of thought, this study proposes to examine students' perceived creativity rather than self-determination/choice. Hence, this study proposes that programming empowerment occurs –people feel empowered –when they (a) view the purpose of a programming task as meaningful, (b) see impact from completing the task(s), (c) believe they can creatively complete them, and (d) perceive their competence to complete them.

Meaningfulness is a person's perceived value of a task's purpose based on his or her ideals or standards (Hur, 2016; Schiefele, 1998; Wang & Lee, 2009; Weber, Martin, & Cayanus, 2005). When a person views a task as meaningful, he or she is more likely to start working on it, to exert more effort to complete it, to persist despite obstacles, and to succeed than otherwise (Spreitzer, 1995; Thomas & Velthouse, 1990). Applied to a learning context, students who perceive a learning task as meaningful are more likely than others to feel empowered (Frymier et al., 1996). In developing the popular block-based programming environment, Scratch, Resnick et al. (2009) highlighted the importance of allowing learners to create personally meaningful programming projects.

Impact is the degree to which "accomplishment of a task is perceived to make a difference in the scheme of things" (Frymier et al., 1996: p.183-184). A person who believes that completion of a task has greater impact will have more internal motivation and is more likely to exert effort to complete it (Frymier et al., 1996; Spreitzer et al., 1997). In this vein, computational thinking education has the potential to transform students from being simply consumers of technology to becoming producers of technology (Angeli et al., 2016), which in turn can empower them to make an impact on both the technological world and on people's lives (Brennan & Resnick, 2012).

Creative self-efficacy is a student's belief that he or she can produce novel ideas and solutions (Brennan & Resnick, 2012). When faced with an obstacle on a task, a person with greater creative self-efficacy is more likely to try different ideas and methods to complete the task, and hence is more likely to succeed (Paulus & Brown, 2003). Computational thinking education can allow the cultivation of student's creativity (Deschryver & Yadav, 2015; Voogt et al., 2015), and empower them to become producers of technology via the creation of programming artifacts (Angeli et al., 2016). Past studies indicate that a suitable programming environment can foster students' creative self-efficacy (Burke, 2012).

Programming self-efficacy is a belief that one has the necessary skills and abilities to perform a programming task well (Frymier et al., 1996; Gist, 1987; Hur, 2006). When students have greater self-efficacy, they have greater confidence in their competence to do a task, and are more likely to both start doing it and to continue working on it until completion (Chiu & Klassen, 2010). Students who believe they can successfully complete a programming task are more likely to feel empowered (Frymier et al., 1996). Also, successful interventions to teach computer programming often increase students' programming self-efficacy (e.g., Denner et al., 2014; Wolz, Stone, Pearson, Pulimood, & Switzer, 2011).

Lastly, all four empowerment components, meaningfulness, impact, creative self-efficacy, and programming self-efficacy, might increase the likelihood of successful task completion. In turn, successful task completion can validate perceived empowerment and further increase it. To summarize, the proposed model of programming empowerment contains four components, namely, meaningfulness, impact, creative self-efficacy, and programming self-efficacy. It is related to a person's self-understanding with reference to programming. And through the creation of programming artifacts, students can potentially impact the world and others' lives (Brennan & Resnick, 2012). Next, we consider antecedents of students' programming empowerment.

2.3. Interest, collaboration attitude, demographics, and programming empowerment

2.3.1 Interest in programming and programming empowerment

CT curricula implementation often highlights how its novelty can stimulate student interest, and how students benefit form sharing computers and collaborating; therefore, this study focuses on two primary antecedents of programming empowerment: interest and collaboration attitude. Students with interest in programming or positive attitudes toward collaboration might have greater programming empowerment. Akin to Hidi (2006), this study defines interest in programming as a psychological state in which a person shows greater attention, concentration, positive feelings or relatively enduring predisposition toward programming activities.

A student with greater interest than others in a domain such as programming views it as meaningful, beyond simply an instrumental means toward positive outcomes; instead, he or she is curious about it and explores it to make discoveries about its impacts and consequences (*self-determination theory*, Ryan & Deci, 2017). Moreover, such a student tends to have a greater internal locus of causality than other students (Dimmock, Jackson, Podlog & Magaraggia, 2013), so he or she views the task as easier (Gattiker, 1992), is willing to spend more time on it (Ryan & Deci, 2017), develops greater competence at it (Venkatesh & Speier, 2000), and perceives greater programming self-efficacy (*technology acceptance theory*, Venkatesh, 2000). Faced with obstacles or task difficulties, such a student is more likely than others to view it as a challenge, underestimate the difficulties, and persevere to find creative solutions (Ryan & Deci, 2017; Senko & Harackiewicz, 2005). In short, a student with more interest in programming than others is more likely to view it as meaningful, learn more about its impacts, have greater creative self-efficacy, and have greater programming self-efficacy – all of which suggests greater programming empowerment (Weber & Patterson, 2000; Weber et al., 2005).

H-1a. Students with greater interest in programming than others view it as more meaningful.

H-1b. Students with greater interest in programming than others believe it has greater impact.

H-1c. Students with greater interest in programming than others have greater creative self-efficacy.

H-1d. Students with greater interest in programming than others have greater programming self-efficacy.

2.3.2 Collaboration attitude and programming empowerment

Collaboration attitude might be linked to empowerment via creative self-efficacy or programming self-efficacy. Akin to Tseng, Wang, Hu and Sun (2009), this study defines attitude toward collaborative programming as a person's orientation or feelings toward cooperative programming activities with peers. Past studies show that having students work together in pairs or small groups on a computer programming activity (typically with one computer) can improve students' programming performance and confidence (e.g., McDowell, Werner, Bullock, & Fernald, 2006; Sawyer & Goldman, 2010). Students with better collaboration attitudes are more likely to recognize shared goals and shared benefits from working with others (Johnson & Johnson, 2009). Such students might exert more effort when working with others, develop better collaborative skills, and collaborate more effectively to solve difficult problems creatively. Hence, we hypothesize that students with better attitudes toward collaboration have greater programming empowerment in two ways: creative self-efficacy and programming self-efficacy.

H-2a. Students with better attitudes toward collaborative programming than others have greater creative self-efficacy.

H-2b. Students with better attitudes toward collaborative programming than others have greater programming self-efficacy.

2.3.3 Age and programming empowerment

Past studies suggest that demographics are also linked to empowerment. Older students often view school subjects as less meaningful and feel less competent than younger students. Compared to younger students, older students face more decontextualized instruction, which many find less relevant or useful in their daily lives; as a result, they often view school subjects as less meaningful (Lepper, Corpus, & Iyengar, 2005). Furthermore, as students age, they become more aware of their peers and their relative abilities (Dweck, 2000), which tends to reduce their domain-specific attitudes, overall school-related attitudes, and achievement motivation (Chiu & Klassen, 2010). Older students' perceived competence may also fall due to uncertainties resulting from less personalized instruction, and perceptions of increased academic pressures, rather than an objective fall in actual ability or performance (Pajares & Cheong, 2003). Hence, older students are expected to view computer programming as less meaningful and to perceive less competence at it, compared to younger students.

H-3a. Students in later grades view programming as less meaningful.

H-3b. Students in later grades have less programming self-efficacy.

2.3.4 Gender and programming empowerment

Compared to girls, boys are more interested in computer programming, in part due to popular media stereotypes. The popular media stereotype of male computer specialists serves as role models for boys and anti-role models for girls, thereby encouraging interest in computer programming among boys and discouraging it among girls (Master, Cheryan, & Meltzoff, 2016). Indeed, girls in a classroom with stereotypical computer science objects (e.g., Star Trek poster, video games) showed less interest in computer science than girls in a classroom without those objects (Cheryan, Plaut, Davies, & Steele, 2009). Hence, we expect boys to show more interest in computer programming than girls do.

H-4. Boys are more interested than girls are in programming.

3. Method of Study

This study is part of a larger project that aims to promote CT education among primary schools. In designing the CT curriculum for senior primary students, we emphasize the development of CT concepts, CT practices, and CT perspectives (Brennan & Resnick, 2012), and consider programming empowerment as an important CT perspective for students to develop. In this curriculum, students were told and asked to apply CT and programming concepts and practices to address community problems in their final projects at the end of each school year after taking this course. This study focuses on the development of an instrument for assessing programming empowerment. After designing and validating the survey instrument via experts, the study collected survey responses from senior primary students. Then, this study conducted confirmatory factor analyses of the programming empowerment components (meaningfulness, impact, creative self-efficacy and programming self-efficacy), interest in programming, and attitude toward collaboration. Afterwards, we tested our hypotheses with a structural equation model.

3.1. Participants and procedures

We asked 323 senior primary school students who participated in the project to complete an online survey in 30 minutes. 138 (42.7%) of them indicated that they had programming experience before taking part in this project. Of the 323 surveys received, 36 (11%) of them had missing responses and were excluded from our analysis via listwise deletion (Joreskog & Sorbom, 2015). In the resulting sample of 287 students (49.8% boys and 50.2% girls), 106 students were primary fourth graders, 86 students were primary fifth graders, and 95 were primary sixth graders.

3.2. Variables

The research staff of this study developed the items for each construct of this instrument: meaningfulness, impact, creative self-efficacy, programming self-efficacy, interest in programming, and attitude toward collaboration. Then, school teachers and experts in computer science, teacher development, developing programming environments, and educational assessment, reviewed and revised the items to ensure content validity (see survey items in Table 1). Students were asked to indicate their level of agreement with each item on a 5-point Likert scale (1= Strongly disagree; 5 = Strongly agree). To maximize measurement reliability, some subscales had more items; in fact, the number of items in a subscale does not affect the relations among constructs in a structural equation model.

[Insert Table 1 about here]

Demographic variables considered in this study include gender and grade level. *Girl* has a value of one for a girl and a value of zero for a boy. *Grade* indicates the grade level of each student, with possible values of four, five, and six.

3.3. Analysis

3.3.1. Confirmatory factor analysis

The study conducted a confirmatory factor analysis (CFA) with maximum likelihood estimation on each construct with Amos 24. χ^2 (df), CFI, TLI, RMSEA were used as the fit indices for the measurement model of the programming empowerment construct. As the self-report items yielded categorical data, we analyzed their polychoric correlation matrix using maximum likelihood estimation. A ratio of χ^2 over its degrees of freedom (df) or χ^2 / df that is less than 2 or a non-significant χ^2 suggests a good fit (Joreskog & Sorbom, 2015). Furthermore, a Comparative Fit Index (CFI) and a Tucker-Lewis Index (TLI) greater than .90 suggest a good fit, and greater than .95 indicate an excellent fit (Bentler, 1990). A root mean square error of approximation (RMSEA) less than .08 indicates a good fit, and less than .06 indicates an excellent fit (Hu & Bentler, 1999).

3.3.2. Measurement invariance across gender and grades

To ensure that the factor structure of programming empowerment was robust, we conducted multiple group analyses using Amos 24 to test whether the assumptions of measurement invariances across gender and grades were valid. First, we examined the least restrictive baseline model (configural model), which only specified the factor structure, with all parameters freely estimated across gender, based on the goodness of fit indices stated above. To test whether metric and scalar invariances hold across gender, we (a) constrained the boys' factor loadings to be equal to the girls' corresponding factor loadings (metric model), and (b) also constrained the intercept estimates to be equal, along with the boys' and girls' factor loadings (scalar model). As the three models (configural, metric, and scalar) are nested models, we tested metric and scalar invariances against the baseline model of configural invariance: a significant increase in χ^2 for the metric or scalar models would indicate that the factors are not invariant across gender. Similarly, we tested whether the assumptions of measurement (configural, metric, and scalar) invariances were valid across grade levels (four, five, and six). According to Kline (2016), scalar invariance indicates strong invariance across groups.

3.3.3. Structural equation modeling

To test the above hypotheses, we conducted structural equation modeling (SEM) with Amos 24 (Arbuckle, 2016). As only omission of significant variables causes omitted variable bias (Kennedy, 2008), non-significant paths were then removed from the SEM. The SEM's goodness of fit was evaluated via the χ^2 (df), CFI, TLI, and RMSEA criteria described above (Hu & Bentler, 1999).

4. Results

4.1. Confirmatory factor analyses

CFA of the programming empowerment items showed that one creative self-efficacy item, "There are many different ways to solve a problem using programming" did not have a significant factor loading, so it was removed (see Table 1). The remaining programming empowerment items showed a good fit to a four-factor structure of meaningfulness, impact, creative self-efficacy and programming self-efficacy ($\chi^2(84) = 164.103$, $p < .001$; CFI = .975; TLI = .969; and RMSEA = .058). Moreover, all factor loadings ranged from .73 to .90, which confirmed the convergent validity of the construct of programming empowerment (see Figure 1).

[Insert Figure 1 about here]

We also compared the four-factor model of programming empowerment with the one-factor model with all 15 items loaded on one single factor (see Figure 2). The goodness of fit indices indicated that the four-factor model fits the data better than the one-factor model (χ^2/df: 1.954 < 6.130; CFI: .975 > .855; TLI: .969 > .831; RMSEA: .058 < .134).

[Insert Figure 2 about here]

CFA of the students' interest in programming items showed that the hypothesized single factor model fit the data well ($\chi^2(2) = 4.262$, $p = .119$; CFI = .997; TLI = .990; and RMSEA = .063), and the factor loadings ranged from .76 to .90 (see Figure 3). CFA of the students' attitude towards collaborative programming also showed that the single factor model was acceptable ($\chi^2(2) = 5.493$, $p = .064$; CFI = .988; TLI = .963; and RMSEA = .078), with factor loadings ranging from .41 to .82 (see Figure 4).

[Insert Figure 3 about here]

[Insert Figure 4 about here]

4.2. Measurement invariance across gender and grades

We then tested the measurement invariance of the four-factor model of programming empowerment across gender and grades using multiple group analyses. The baseline configural, metric, and scalar invariance models all showed good fits to the data (see Table 2). Furthermore, nested model comparisons showed no significant differences (metric invariance: $\Delta\chi^2(11) = 9.362$, $p = .589$; scalar invariance: $\Delta\chi^2(26) = 26.252$, $p = .449$), supporting metric and scalar invariances and hence strong invariance across gender (Kline, 2016).

[Insert Table 2 about here]

Similarly, we tested measurement invariance of the four-factor model of programming empowerment across grades (four, five, and six) with configural, metric, and scalar invariance models. As shown in Table 3, the goodness of fit of each model was acceptable. Furthermore, nested model comparisons showed no significant differences (metric invariance: $\Delta\chi^2(11) = 15.152$, $p = .176$; scalar invariance: $\Delta\chi^2(26) = 25.906$, $p = .468$), supporting metric and scalar invariances and hence strong invariance across grades (Kline, 2016).

[Insert Table 3 about here]

4.3. Structural equation model

Next, we tested our hypotheses regarding gender, grade, programming interest, attitude toward collaborative programming and programming empowerment with an SEM, as shown in Figure 5. The indices indicated that the hypothesized model fits the data well ($\chi^2(269) = 551.864$, $p < .001$; CFI = .939; TLI = .926; and RMSEA = .061). The results supported hypotheses H-1a, b, c and d, showing that students with greater interest in programming viewed it as more meaningful, viewed it as having more impact, had greater creative self-efficacy, and had greater programming self-efficacy.

Furthermore, the SEM results supported hypothesis H-2a but not H-2b. Students with better attitudes toward collaboration had more creative self-efficacy but not more programming self-efficacy.

The results also supported the demographic hypotheses. Students in later grades viewed programming as less meaningful (H-3a) and had less programming self-efficacy (H-3b). Lastly, boys showed more interest in programming than girls did (H-4).

[Insert Figure 5 about here]

5. Discussion

Confirmatory factor analysis confirmed the design of the 15-item instrument for assessing students' programming empowerment in the context of K-12 education, which consists of four components, meaningfulness, impact, creative self-efficacy, and programming self-efficacy. This study tests further whether programming interest, attitude toward collaborative programming, gender and grade level were related to the four components of programming empowerment among 287 senior primary school students. After the confirmatory factor analyses supported each of the six factors, a structural equation model supported eight of the nine hypotheses. Students with greater interest in programming viewed it as more meaningful, believed that it had more impact, had more creative self-efficacy, and had greater programming self-efficacy. Furthermore, students with better attitudes toward collaboration had greater creative self-efficacy but not more programming self-efficacy. Students in later grades perceived programming as less meaningful and had less programming self-efficacy, consistent with past studies (Pajares & Cheong, 2003). Lastly, boys indicated more interest in programming than girls, also consistent with past studies (Master at al., 2016).

5.1 Interest in programming and programming empowerment

Students' interest in programming was related to all four factors of programming empowerment. This result is in line with the literature on empowerment (Weber & Patterson, 2000; Weber et al., 2005), *self-determination* (Ryan & Deci, 2017) and *technology acceptance* (Venkatesh, 2000) and suggests that interest is a crucial factor affecting different aspects of programming empowerment. This result also suggests the importance of a student interest-driven approach to the design and implementation of learning activities in CT education and testing whether these activities engage student interests (Kong, 2016). If students are not interested in the programming activities, fostering their programming empowerment with respect to meaningfulness, impact, creative self-efficacy, or programming self-efficacy becomes much more difficult.

5.2 Collaboration attitude and programming empowerment

Students with better attitudes toward collaboration had greater creative self-efficacy but not more programming self-efficacy. As collaboration can help students generate ideas to solve problems that they cannot do alone (*zone of proximal development*, Vygotsky, 2011), they might view collaboration positively as a means to enhance creativity for solving challenging programming problems, which is in line with findings in the learning sciences that creativity is a collective achievement rather than an individual gift (e.g., Sawyer, 2017). The result of this study implies that suitable incorporation of collaboration in teachers' design of activities in learning programming can empower students to be more creative.

The short-term and dramatic collaboration benefit of creativity for solving challenging programming problems might overshadow its longer-term, gradual enhancement of individual programming self-efficacy, which might account for the non-significant link between collaboration attitude and programming self-efficacy. Unlike a short-term, striking creative solution, improved competence can be gradual and unrecognized until subsequent tasks, so students might not appreciate how collaboration can enhance individual competence (*internalization*, Vygotsky, 2011). The results might differ for students who learn regularly from collaborative programming activities (see e.g., Bishop-Clark, Courte, & Howard, 2006; Denner et al., 2014; Lewis, 2011; Zarb & Hughes, 2015). This result also suggests testing whether teacher highlighting of how specific collaborative efforts improve each member's individual competences can improve students' collaboration attitudes and their programming self-efficacy.

5.3 Age and programming empowerment

In addition, senior grade students tended to perceive programming as less meaningful compared to junior grade students. Possibly, senior grade students might not see programming as connected to their daily lives (Lepper et al., 2005), which suggests the importance of designing learning and teaching activities that allow students to connect programming to their everyday lives to help them appreciate programming as useful and meaningful.

Students in later grades also perceived themselves as less competent at programming compared to students in earlier grades. As discussed in the conceptual framework section of the study, such a fall does not necessarily indicate an objective decrease in actual capability, but might stem from increased academic pressures and uncertainties because of less personalized instruction (Chiu & Klassen, 2010; Pajares & Cheong, 2003). These results also suggest a moderate difficulty level in the curriculum design to facilitate successful solution of challenging programming problems to support students' programming self-efficacy.

5.4 Gender and programming empowerment

The results related to gender suggest that more effort is needed to attract girls to engage in programming activities, as girls indicated less interest in programming than boys did. More recently, studies have examined interventions aiming to engage girls more in programming and explore their programming processes and outcomes (e.g., Denner, Werner, & Ortiz, 2012; Seneviratne, 2017). Successful strategies for engaging girls, such as highlighting female role models in programming and using classroom discussion as a programming pedagogy (e.g., Seneviratne, 2017) could be embedded in the design of learning activities to enhance girls' interest in programming. Programming tasks such as those related to storytelling (Holt, 2011) might also help to enhance the interests of girls. Future studies can test whether these strategies enhance girls' programming empowerment. When guiding students to work on programming activities, teachers might also need to pay special attention to girls' engagement levels.

6. Limitations and Future Studies

The limitations of this study include its sample, cross-sectional data, few contextual variables, and omission of other beliefs of students. This sample only includes students from a few primary schools with students in grades four through six, and is not a representative sample of Hong Kong, let alone students in other regions. Future studies can examine students from more primary schools in Hong Kong and other regions, students at other ages, and use representative samples. Furthermore, this data is cross-sectional, which limits causal inferences. Further studies can include longitudinal data to make stronger inferences. Also, this study collected only grade level data, which limits testing of moderation effects. Future studies can collect more data from other contextual variables such as information about teacher, school and family, all of which might influence beliefs of students. Specifically, instructional designs, such as the final project for students to apply CT and programming concepts and practices to address community problems might affect students' programming empowerment, interest in programming, and attitude toward collaboration. Also, the beliefs of students in this study such as creative self-efficacy and programming self-efficacy might be linked to their other beliefs (Psycharis & Kallia, 2017), which can be explored in future studies..

7. Conclusion

A 15-item instrument for assessing students' programming empowerment in the context of K-12 education was developed and validated. Responses from 287 senior primary school students confirmed the conceptualization of the construct of programming empowerment as consisting of four components: 1) meaningfulness, 2) impact, 3) creative self-efficacy, and 4) programming self-efficacy.

The structural equation model supported our explanatory model. Students with more interest in programming considered it as more meaningful, believed that it had more impact, and believed that they had more creative self-efficacy and programming self-efficacy. Furthermore, students with better collaboration attitudes had more creative self-efficacy but not more programming self-efficacy. Students in senior grades perceived programming as less meaningful and themselves as less competent at programming compared to students in junior grades. Also, boys showed more interest in programming than girls did.

These results suggest the importance of designing an interest-driven computational thinking curriculum, with a moderate difficulty level, and collaboration opportunities to enhance students' creative self-efficacy. In addition, teachers might need to pay attention to the engagement level of girls, and employ strategies to enhance their interest in programming. Future studies can evaluate whether such an interest-driven curriculum with opportunities to collaborate enhances students' programming empowerment.

References

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework- Implications for Teacher Knowledge. *Educational Technology & Society, 19*(3), 47–57.

Arbuckle, J. L. (2016). *IBM SPSS AMOS 24 User's Guide*. Armonk, NY: Amos Development Corporation.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review, 84*, 191-215.

Becker, J., Kovach, A.C., & Gronseth, D.L. (2004). Individual empowerment: How community health workers operationalize self-determination, self-sufficiency, and decision-making abilities of low-income mothers. *Journal of Community Psychology, 32*(3), 327–342.

Bentler, P. M. (1990). Comparative fit indexes in structural models. *Psychological Bulletin*, *107*, 238–246.

Bishop-Clark, K., Courte, J., & Howard, E. (2006). Programming in pairs with Alice to improve confidence, enjoyment, and achievement. *Journal of Educational Computing Research, 34*(2), 213–228.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *2012 Annual Meeting of the American Educational Research Association (AERA'12),* Canada.

Breton, M. (1994). Relating competence-promotion and empowerment. *Journal of Progressive Human Services, 5*, 27–44.

Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *Journal of Media Literacy Education*, *4*(2), 121–135.

Cheryan, S., Plaut, V. C., Davies, P. G., & Steele, C. M. (2009). Ambient belonging: How stereotypical cues impact gender participation in computer science. *Journal of Personality and Social Psychology, 97*(6), 1045-1060.

Chiu, M. M., & Klassen, R. M. (2010). Relations of mathematics self-concept and its calibration with mathematics achievement. *Learning and Instruction*, *20*, 2-17.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education, 58*(1), 240–249.

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education, 46*(3), 277-296.

Deschryver, M. D., & Yadav, A. (2015). Creative and computational thinking in the context of new literacies: working with teachers to scaffold complex technology-mediated approaches to teaching and learning. *Journal of Technology and Teacher Education, 23*(3), 411–431.

Dimmock, J., Jackson, B., Podlog, L., & Magaraggia, C. (2013). The effect of variety expectations on interest, enjoyment, and locus of causality in exercise. *Motivation and Emotion*, *37*(1), 146-153.

Dweck, C. S. (2000). *Self-theories: Their role in motivation, personality, and development*. Psychology Press.

Freire, P. (1973). *Education for critical consciousness*. New York: Continuum Publishing Company.

Frymier, A. B., Shulman, G., & Houser, M. (1996). The development of a learner empowerment measure. *Communication Education*, *45*, 181–199.

Gattiker, U. E. (1992). Computer skills acquisition: A review and future directions for research. *Journal of Management, 18*(3), 547-574.

Gist, M. E. (1987). Self-efficacy: Implication for organizational behavior and human resource management. *Academy of Management Review*, *12*, 472-485.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43.

Hidi, S. (2006). Interest: A unique motivational variable**.** *Educational Research Review, 1*(2), 69-82

Holt, L. (2011). Creating Digital Stories with Scratch to Promote Computational Thinking. In M. Koehler & P. Mishra (Eds.), *Proceedings of SITE 2011--Society for Information Technology & Teacher Education International Conference* (pp. 31-32).

Nashville, Tennessee, USA: Association for the Advancement of Computing in Education (AACE).

Hu, L. T., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journ*al, *6*(1), 1-55.

Hur, M. H. (2006). Empowerment in terms of theoretical perspectives: exploring a typology of the process and components across disciplines. *Journal of Community Psychology, 34*(5), 523-540.

Hur, M. H. (2016). Empowering the elderly population through ICT-based activities: An empirical study of older adults in Korea. *Information Technology & People*, *29*(2), 318-333.

Johnson, D. W., & Johnson, R. T. (2009). An educational psychology success story: Social interdependence theory and cooperative learning. *Educational Researcher*, *38*(5), 365-379.

Joreskog, K., & Sorbom, D. (2015). *LISREL 9.2*. New York: Scientific Software International.

Kennedy, P. (2008). *A guide to econometrics (6th ed.)*. Malden, MA: Blackwell.

Kirk, C. M., Lewis, R. K., Brown, K., Karibo, B., & Park, E. (2016). The power of student empowerment: Measuring classroom predictors and individual indicators. *The Journal of Educational Research, 109*(6), 589-595.

Kline, R. B. (2016). *Principles and practice of structural equation modeling (4th ed.)*. New York, NY: Guilford Press.

Kong, S. C. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education, 3*(4), 377-394.

Larson, R., Walker, K., & Pearce, N. (2005). A comparison of youth-driven and adult-driven youth programs: Balancing inputs from youth and adults. *Journal of Community Psychology, 33*(1), 57–74.

Lepper, M. R., Corpus, J. H., & Iyengar, S. S. (2005). Intrinsic and extrinsic motivational orientations in the classroom: Age differences and academic correlates. *Journal of Educational Psychology*, *97*(2), 184-196.

Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students? *Computer Science Education, 21*(2), 105-134.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51-61.

Master, A., Cheryan, S., & Meltzoff, A. N. (2016). Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. *Journal of Educational Psychology*, *108*(3), 424.

McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, *49*(8), 90–95.

Page, N., & Czuba, C. E. (1999). Empowerment: What is it? *Journal of Extension*, *37*(5), 24–32.

Pajares, F., & Cheong, Y. F. (2003). Achievement goal orientations in writing. *International Journal of Educational Research*, *39*, 437-455.

Papert, S. (1972). Teaching children thinking. *Innovations in Education & Training International*, *9*(5), 245-255.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Paulus, P. B., & Brown, V. (2003). Ideational creativity in groups. In P. B. Paulus, & B. A. Nijstad (Eds.), *Group creativity* (pp. 110-136). New York: Oxford University Press.

Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, *45*(5), 583–602.

Rappaport, J. (1987). Terms of empowerment/exemplars of prevention: Toward a theory for community psychology. *American Journal of Community Psychology*, *15*, 121-148.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60-67.

Ryan, R. M., & Deci, E. L. (2017). *Self-determination theory*. New York, NY: The Guilford Press.

Sawyer, R. K. (2017). *Group genius: The creative power of collaboration*. New York, NY: Basic Books.

Sawyer, R. K., & Goldman, K. J. (2010). Collaborative learning of computer science concepts. In K. Littleton & C. Howe (Eds.), *Educational dialogues: Understanding and promoting productive interaction* (pp. 323–345). London, UK: Routledge.

Schiefele, U. (1998). The role of interest in motivation and learning. In S. Messick & J. Mo Collis (Eds.), *Intelligence and personality: Bridging the gap in theory and measurement*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Seneviratne, O. (2017). Making computer science attractive to high school girls with computational thinking approaches: A case study. In P. Rich & C. Hodges (Eds), *Emerging research, practice, and policy on computational thinking* (pp. 21-32). Cham, Switzerland: Springer.

Senko, C., & Harackiewicz, J. M. (2005). Achievement goals, task performance, and interest: Why perceived goal difficulty matters. *Personality and Social Psychology Bulletin*, *31*(12), 1739-1753.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142-158.

Sprague, J., & Hayes, I. (2000). Self-determination and empowerment: A feminist standpoint analysis of talk about disability. *American Journal of Community Psychology, 28*(5), 671–695.

Spreitzer, G. M. (1995). Psychological empowerment in the workplace: Dimensions, measurement, and validation. *Academy of Management Journal, 38*(5)1442-1465.

Spreitzer, G. M., Kizilos, M. A., & Nason, S. W. (1997). A dimensional analysis of the relationship between psychological empowerment and effectiveness, satisfaction, and strain. *Journal of Management, 23*(5), 679–704.

Thomas, K., & Velthouse, B. (1990). Cognitive elements of empowerment: An interpretive model intrinsic task motivation. *Academy of Management Review, 15*, 666–681.

Tseng, H., Wang, C. H., Ku, H. Y., & Sun, L. (2009). Key factors in online collaboration and their relationship to teamwork satisfaction. *Quarterly Review of Distance Education*, *10*(2), 195.

Venkatesh, V. (2000). Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. *Information Systems Research*, *11*(4), 342-365.

Venkatesh, V., & Speier, C. (2000). Creating an effective training environment for enhancing telework. *International Journal of Human-Computer Studies*, *52*(6), 991-1005.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 1-14.

Vygotsky, L. S. (2011). The dynamics of the schoolchild's mental development in relation to teaching and learning. *Journal of Cognitive Education and Psychology, 10*(2), 198-211.

Wang G. & Lee, P. D. (2009). Psychological Empowerment and Job Satisfaction: An Analysis of Interactive Effects. *Group & Organizational Management, 34*, 271-296.

Webb M., Davis N., Bell T., Katz Y. J., Reynolds N., Chambers D. P., et al. (2017). Computer science in K-12 school curricula of the 2lst century: why, what and when? *Education and Information Technologies, 22*(2), 445-468.

Weber, K., & Patterson, B. R. (2000). Student interest, empowerment and motivation. *Communication Research Reports*, *17*, 22-29.

Weber, K., Martin, M. M., & Cayanus, J. L. (2005). Student interest: A two-study re-examination of the concept. *Communication Quarterly*, *53*(1), 71–86.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2015). *App Inventor 2: Create your own Android apps*. Sebastopol, CA: O'Reilly.

Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (2011). Computational thinking and expository writing in the middle school. *ACM Transactions on Computing Education*, *11*(2), 1-22.

You, J. W. (2016). The relationship among college students' psychological capital, learning empowerment, and engagement. *Learning and Individual Differences, 49*, 17-24.

Zarb, M., & Hughes, J. (2015). Breaking the communication barrier: Guidelines to aid communication within pair programming. *Computer Science Education, 25*(2), 120-151.

Figures

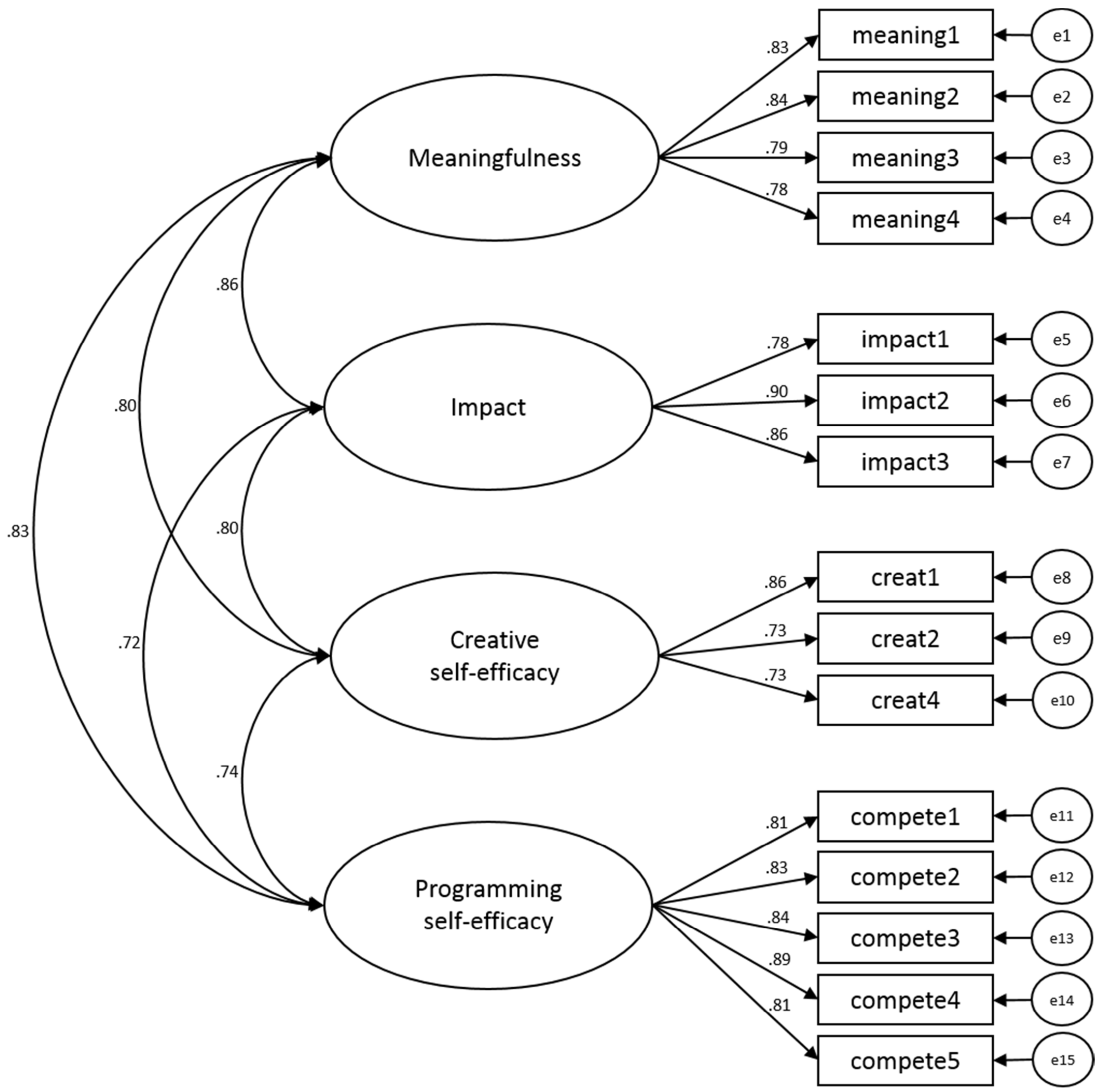

Figure 1. Confirmatory factor analysis of the four components of programming empowerment with Cronbach's α's for Meaningfulness (.883), Impact (.878), Creative self-efficacy (.822), Programming self-efficacy (.919); $\chi^2(84) = 164.103$, $p < .001$; CFI = .975; TLI = .969; and RMSEA = .058

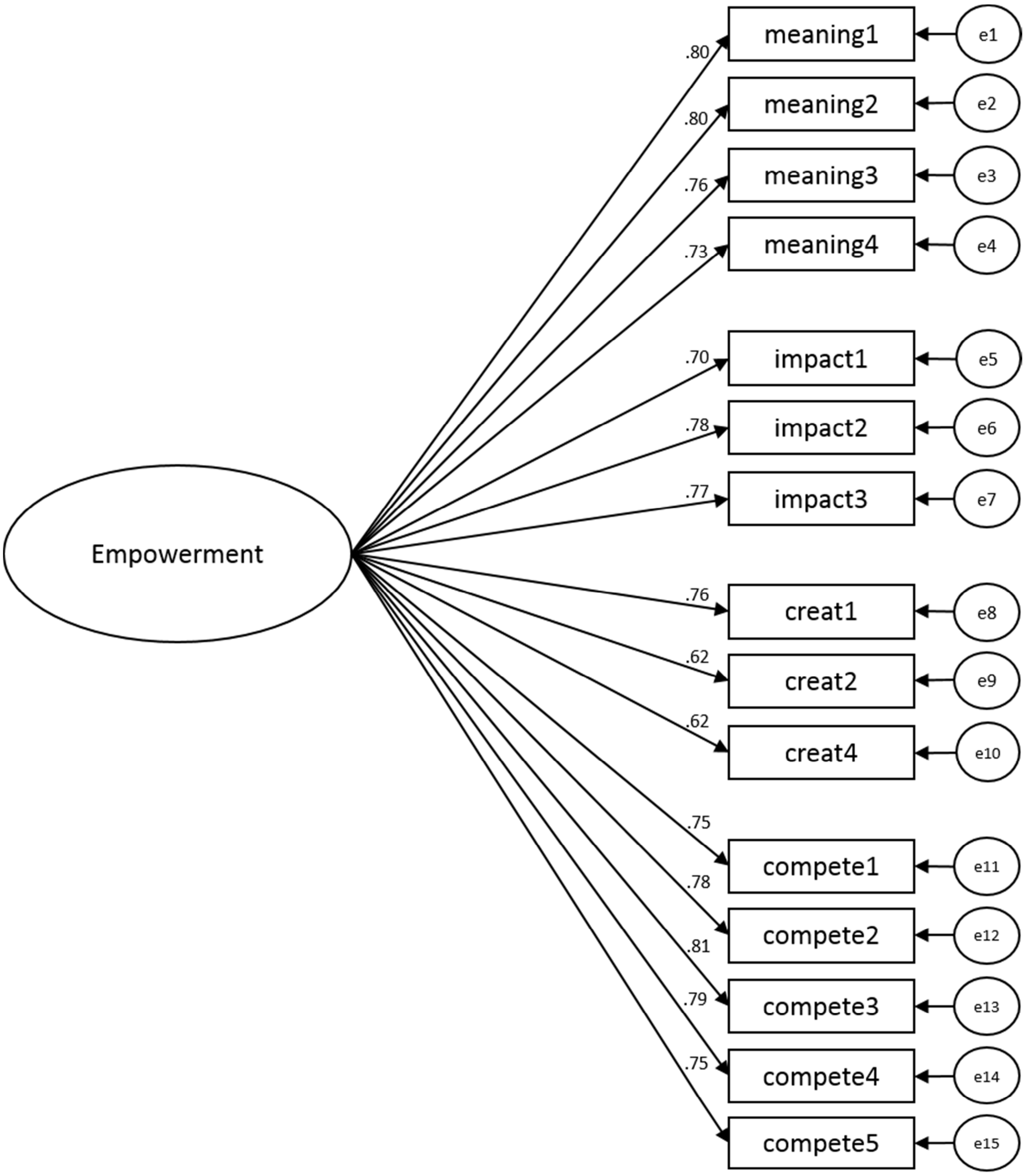

Figure 2. Factor structure of a one-factor model of programming empowerment with fit indices: χ^2 (90) = 551.720; χ^2/df = 6.130; CFI = .855; TLI = .831; RMSEA = .134

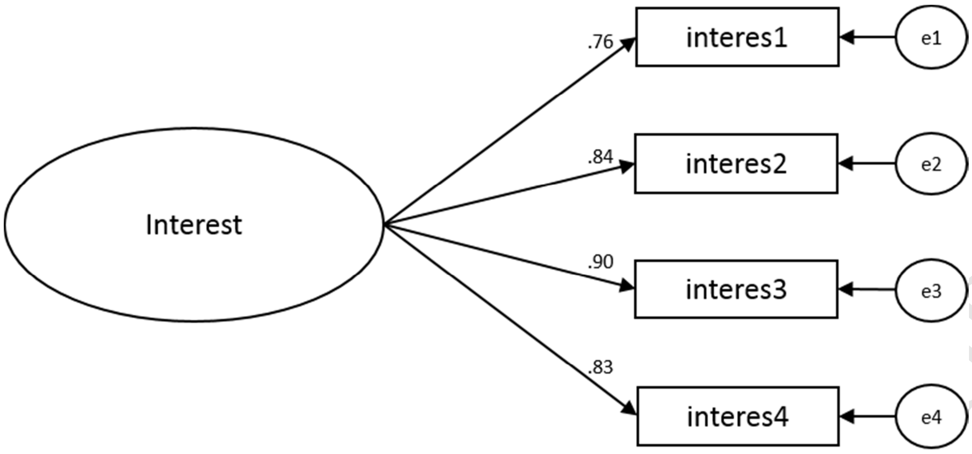

Figure 3. Factor structure of student interest in programming with Cronbach's α of .895; $\chi^2(2) = 4.262$, p =.119; CFI = .997; TLI = .990; and RMSEA = .063

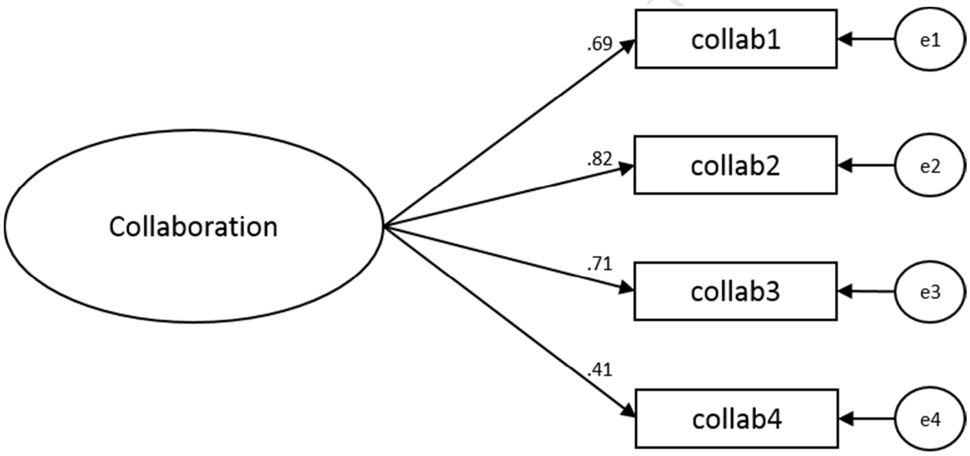

Figure 4. Factor structure of students' attitude towards collaboration in programming with Cronbach's α of .742; $\chi^2(2) = 5.493$, p = .064; CFI = .988; TLI = .963; and RMSEA = .078

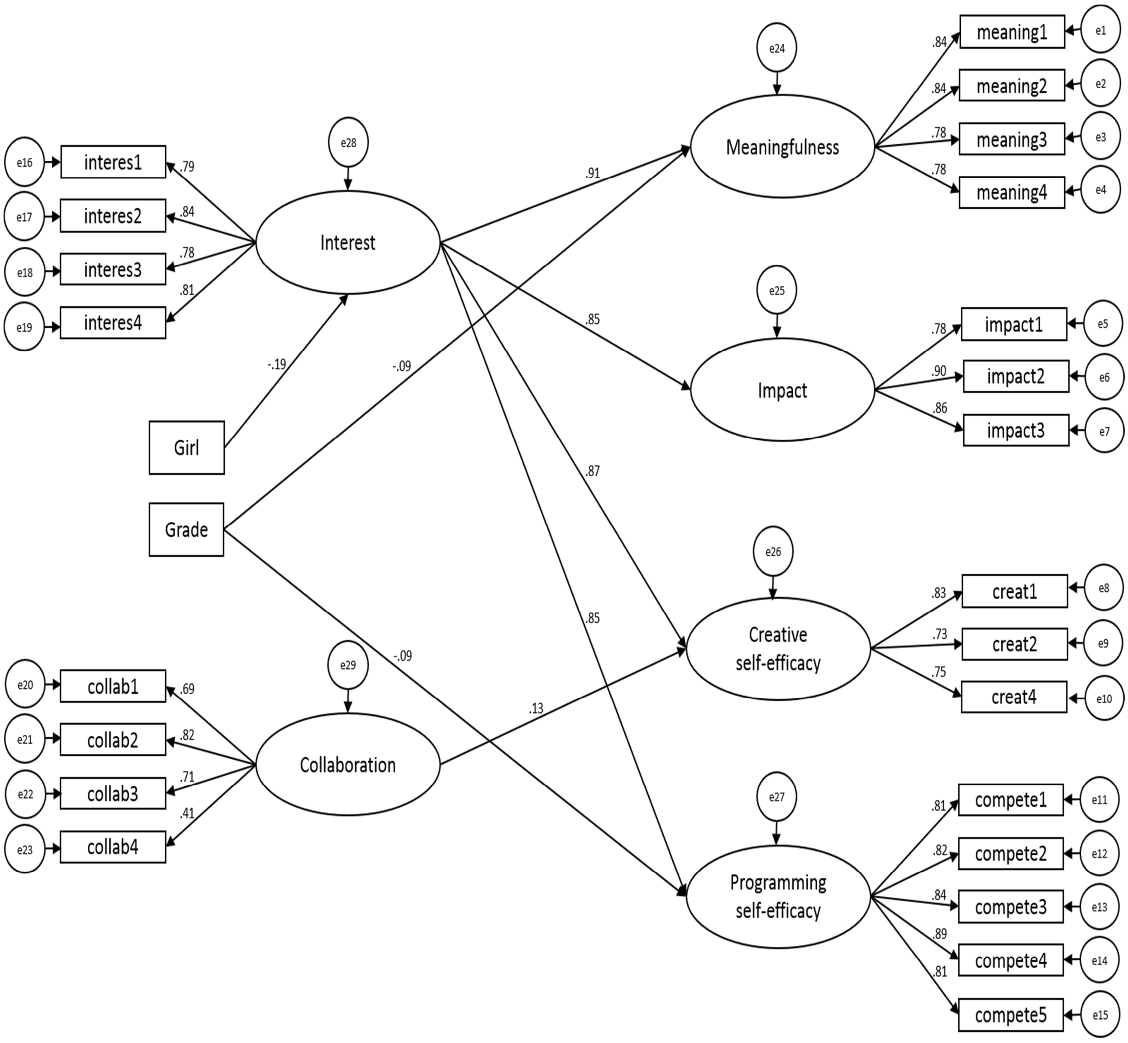

Figure 5. Structural equation model on programming empowerment and students' interest, attitude toward collaboration in programming, grade level and gender; $\chi^2(269) = 551.864$, $p < .001$; CFI = .939; TLI = .926; and RMSEA = .061

Tables

Table 1. Survey items of the instrument in the area of meaningfulness, impact, creative self-efficacy, programming self-efficacy, interest in programming and attitude toward collaboration in programming

Items
Meaningfulness
Programming is useful to me.
Programming will help me achieve my goals.
I want to become good at programming.
Programming is important to me.
Impact
I want to use programming to help solve problems in the world.
I want to use programming to make people's lives better.
I can use programming to make daily life easier.
Creative self-efficacy
I would like to design things using programming.
Computer programmers are creative.
There are many different ways to solve a problem using programming. (deleted due to insignificant factor loading)
It is important to be creative when you are programming.
Programming self-efficacy
I can learn how to program.
I am good at programming.
I think of myself as someone who can program.
I have the skills to program.
I have confidence in my ability to program.
Interest in programming
Programming is interesting.
I am curious about the content of programming.
I think the content of programming is fun.
I am very attracted to computer programming activities.
Attitude toward Collaboration in Programming
I like to program with others.
I finish things faster when I program with others.
I have better ideas when I program with others.
People ask me for help with computers a lot.

Table 2. Testing of measurement invariance across gender

	χ2	df	χ2/df	p	CFI	TLI	RMSEA
Configural invariance (baseline model) (Model 1)	286.946	168	1.708	<.001	.957	.946	.054
Metric invariance (Model 2)	296.309	179	1.655	<.001	.958	.950	.052
Scalar invariance (Model 3)	313.199	194	1.614	<.001	.957	.954	.050

Table 3. Testing of measurement invariance across grades

	χ2	df	χ2/df	p	CFI	TLI	RMSEA
Configural invariance (baseline model) (Model 1)	595.589	303	1.966	<.001	.913	.910	.058
Metric invariance (Model 2)	610.741	314	1.945	<.001	.912	.912	.058
Scalar invariance (Model 3)	621.496	329	1.889	<.001	.913	.917	.056

Acknowledgment

The authors would like to acknowledge the funding support of this Coolthink@JC project from the Hong Kong Jockey Club Charities Trust. The authors would also like to take this opportunity to acknowledge Professor Hal Abelson for his inspiration of the term programming empowerment that triggered this research study.

A Study of Primary School Students' Interest, Collaboration Attitude, and Programming Empowerment in Computational Thinking Education

Highlights

- Empowerment consists of meaningfulness, impact, creative and programming self-efficacy.
- The study validated the programming empowerment instrument.
- Students with greater interest in programming perceive greater empowerment.
- Students with positive collaboration attitude have greater creative self-efficacy.
- Students in senior grades viewed programming as less meaningful than junior grades.