International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
http://ijcjournal.org/

Developing and Applying Online Basic Programming Tools to School Students in a Developing Country

Dr. Mona Nabil Demaidi[a]*, Dr. Manar Qamhieh[b]

[a,b]Computer Engineering Department, An-Najah National University, Nablus, Palestine
[a]Email: m.demaidi@najah.edu
[b]Email: m.qamhieh@najah.edu

Abstract

Programming helps school students develop problem-solving, critical thinking, and logical reasoning skills. It also helps students develop skills at multiple levels of Bloom's taxonomy, including remembering, understanding, applying, analyzing, evaluating, and creating. Several programming tools have been developed to teach coding concepts to school students of different age groups. However, these tools are not well integrated into the Palestinian curriculum due to the inefficient teaching methods in programming. Programming activities focus only on remembering and understanding basic concepts. Therefore, this research focused on studying the importance of applying online programming tools on school students' application and analytical levels, and examining students' perception towards integrating online programming tools in the educational curriculum. A pre/ post experimental design was carried out on 84 children aged 14 years old in four schools in Palestine. The results revealed that 41.16% of students had an increase in their application and analytical levels. The results also revealed that students agreed on integrating online programming tools in schools and that programming helped them improve their problem-solving skills.

Keywords: Computer programming; children; application level; analysis level; Bloom's taxonomy.

1. Introduction

Learning how to program is an important skill for children in both developing and developed countries [1,2]. It helps them understand how computers work and enables them to create and modify their own software and websites. There are many resources available for children to learn programming, including online courses, textbooks, and online programming tools [3,4]. Many of these resources are designed specifically for school students and use fun, interactive methods to teach programming concepts. In developed countries, programming education is more widely available and is often taught in schools as part of the standard curriculum [3]. Many schools offer after-school coding clubs or have dedicated computer science classes [5]. In developing countries, access to programming education is more limited due to a lack of resources and infrastructure [6].

----

* Corresponding author.

Overall, basic programming is an important skill for school students to learn, regardless of where they live. It helps them develop critical thinking and problem-solving skills and prepares them for a future in which technology will play a central role. School students could achieve a high level of cognitive thinking in the form of computational thinking and imagination [7-9].

Cognitive thinking is explained and measured through Bloom's taxonomy, which categorizes learning objectives into six levels [10]:

- Remembering: recalling previously learned information.
- Understanding: comprehending the meaning of the material.
- Applying: using the information in a new context.
- Analyzing: breaking down the material into parts and understanding their relationships.
- Evaluating: making judgments about the value or quality of the material.
- Creating: using the information to generate something new or original.

According to Bloom's taxonomy, teaching basic programming addresses several levels of the taxonomy, but mostly the analysis level [11, 12]. Usually, programmers analyze the problem by breaking it into sub-problems with valid relationships between them [13]. Then, solving each sub-problem gradually can be applied to Bloom's hierarchy from the analysis level to the knowledge level [14]. Higher levels of Bloom's taxonomy such as synthesis and evaluation levels can be achieved in advanced programming [14].

In Palestine, basic programming courses are taught to children in schools starting in 5th grade using several programming online tools such as App-Inventor, embedded programming (Arduino Uno), and Visual Basic [15]. However, programming at schools is taught in a conventional way. Activities focus only on remembering and understanding basic concepts, such as what a variable is or how to use a loop. No activities focus on Bloom's application, analysis, evaluation, and creation levels. Students do not design and build their own programs or use programming to solve real-world problems.

This paper aims to examine the importance of improving programming courses at schools to integrate online programming tools and address Bloom's application and analysis levels. The paper is organized as follows: Section 2 presents the literature review which illustrates similar research and explains the importance of this research. Section 3 explains the experiment carried out to answer the research question. Section 4 illustrates the results and discussion. Finally, Section 5 concludes the paper with a summary of the results and future work.

2. Literature Review

Teaching programming is an effective way to engage students in higher-order thinking and problem-solving activities, as it involves a range of skills that align with different levels of Bloom's taxonomy [16]. At the lower levels of Bloom's taxonomy, programming can involve skills such as remembering and understanding the syntax and structure of programming languages, as well as the concepts and principles underlying them. This involves activities such as reviewing and practicing basic programming constructs, such as loops and control structures, and learning about data types and variables. As students progress through the curriculum, they can engage in higher-level thinking skills such as analyzing and evaluating their code and the algorithms they are using. This might involve debugging and troubleshooting their code, as well as identifying and correcting errors or bugs. At the highest level of Bloom's taxonomy, programming can involve creating and designing new programs and applications. This might involve synthesizing and integrating various programming concepts and techniques in order to solve more complex problems, or developing new algorithms and approaches to solving problems [16].

There is a growing body of research that suggests that programming has a positive impact on children's cognitive development. Studies have shown that programming improves problem-solving skills, critical thinking, and logical reasoning abilities in children [17 - 20]. One study found that children who participated in a programming course showed significant improvement in their problem-solving skills compared to a control group [17]. The study included 66 high school students and the experiment studied the improvement in their reasoning skills after participating in a certain programming course. Another study found that programming activities can help children develop logical reasoning skills, as they must plan and execute steps in a specific order to complete a task [18]. The study targeted 60 students in 11^{th} and 12^{th} grade to measure the effect of computer programming on the various reasoning skills.

In developed countries, programming is often taught as a subject in schools, either as part of the regular curriculum or as an elective. Many schools offer computer science or programming classes at the secondary or post-secondary level. Some schools also offer extracurricular programming clubs or camps to provide additional opportunities for students to learn programming [8].

In developing countries, the availability and quality of programming education may vary significantly. Some schools may not offer programming classes at all, or the classes may not be up-to-date with current technologies and practices. Additionally, there may be a lack of trained teachers or resources to support programming education [6].

Overall, the research suggests that programming has a positive impact on children's cognitive development, particularly in the areas of problem-solving, critical thinking, and logical reasoning. It also helps children develop skills at multiple levels of Bloom's taxonomy. However, more research is needed to fully understand the effect of integrating online programming tools on school students' cognitive development in developing countries. This research aims to study the effect of online programming tools on students in Palestine as a case of a developing country, and this analysis is believed to be the first in Palestine.

3. Methodology

Learning programming at Bloom’s application level helps school students develop problem-solving skills by giving them the ability to write code to solve specific problems. They can learn to break down a problem into smaller parts, think logically, and use their knowledge of programming concepts to create a solution. This can be an important skill in many areas, including math, science, and even daily life. At the analysis level, programming can help children develop critical thinking skills by encouraging them to break down complex problems into smaller pieces and understand how the different parts work together. They can learn to analyze code, identify patterns and relationships, and debug problems. These skills can be useful in a variety of contexts, including schoolwork, career development, and personal projects.

This section consists of the research question, experimental set-up, and participants.

3.1. Research question

This research aims to examine the effect of online programming tools on school students' application and analysis levels.

3.2. Experimental Setup

A pre/ post experimental design was carried out in 2022 on students who voluntarily participated in a programming competition called "Think like a Programmer". The competition is organized on a yearly basis in Palestine to introduce students of different age groups and disciplines to programming. The competition is based on the concept of Blockly online Programming tool [21] (see Figure 1), which promotes the idea of programming, especially for those who have no coding background, including school students. The competition challenges students to apply programming concepts using a set of predefined blocks. The blocks help students apply programming concepts using drag and drop blocks.

The competition provided students with 11 programming problems developed by four experts and was distributed as follows: four easy, three medium, two hard, and two advanced problems. The problems focused on using a predefined set of blocks in Blockly to draw different geometric shapes. For example, students are asked to draw a square similar to a predefined square (in location, width, size and rotation). Blockly has no "draw a square" instruction, therefore, the student needs to find a starting point, the length of the square, and then the sequence of instructions required to draw the square.

Prior to the competition, students were provided with a comprehensive tutorial to ensure that they understand how to use Blockly. The experiment took place at the schools' computer laboratories. Each student has a dedicated computer to participate in the competition for two hours and the environment was very quiet to ensure that the students have high concentration.

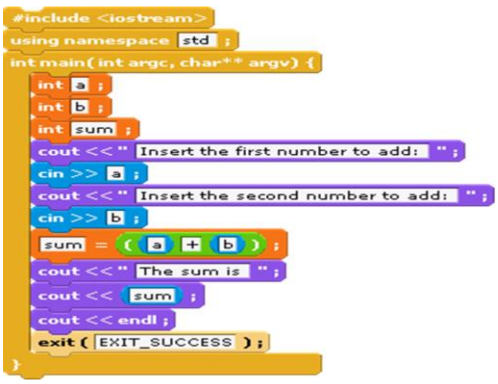

Figure 1: Example of Blockly programming.

3.2.1. Assessment tests

Pre and post assessment tests which consist of five questions were used in the experiment. The question aimed to examine the effect of online programming tools on students' application and analysis levels. The first question aimed to assess students' imagination and their ability to explore alternative solutions and perspectives. The question assessed students' ability to form a square from several parts and to understand the relationship between the parts (see Figure. 2).

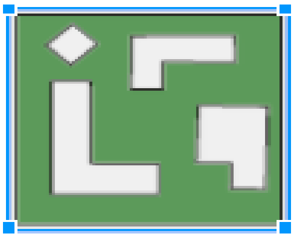

Figure 2: "Forming a square" question.

The second question in the assessment test asked students to solve the Rubik's cube. Rubik's Cube incorporates different levels of Bloom's Taxonomy into their learning [22]. It aimed to assess if students can practice applying what they have learned by attempting to solve the Rubik's Cube on their own, and if they can analyze the different patterns and algorithms they have learned to determine which ones work best for them. The question also aimed to assess if students could provide an "out of the box" solution such as; disassembling and then reassembling the cube correctly, or solving the cube by applying conventional rule-based steps.

The third question (see Figure 3) focused on following a pattern using parallel thinking. Parallel thinking addresses the application and analysis levels of Bloom's Taxonomy as it involves breaking a problem down into smaller, more manageable pieces, working on them simultaneously, and applying and analyzing the information and skills students have learned to solve the problem.

Figure 3: “Continue the pattern" question.

The fourth question focused on using personality and self-reflection in conjunction with the application and analysis levels in Bloom's Taxonomy. It assessed how students' unique characteristics and experiences can inform their approach to solving problems and understanding concepts, and apply and analyze the knowledge and skills they have learned to come up with effective solutions. The question asked students to select a shape that represents them and to state the reason for choosing the shape (see Figure 4). This question examined if students changed their chosen shape after participating in the competition, as students were provided with a set of complicated geometry shapes which they were asked to draw using Blockly blocks (see Section 3.2).

Changing the shape to a more complex one reflects the students' perspective toward complex shapes formed in the competition.

Figure 4: "Shape representation" question.

The fifth question focused on a detail-oriented approach in conjunction with the application and analysis levels in Bloom's Taxonomy. It aims to assess students' ability to pay close attention to the specific details of a task or problem, and apply and analyze the knowledge and skills students have learned to come up with accurate and efficient solutions. The question provided to students aimed to assess students' ability to analyze the question's stem template. The question stem was: "I have 5 apples, and i take away 3. How many apples do i have?". The trick in the question is using (I) and (i). The correct answer of the question is three.

Students' scores in the assessment tests were normalized to obtain a score out of 10 and were measured as follows:

- Zero is given if the student changed their answer from correct to incorrect.
- One if the student did not change their incorrect answer from pre-test to the post-test.
- Two if the student changed their answer from incorrect to an answer close to the correct answer
- Three if the student answered the question correctly in the post-test.

3.2.2. Questionnaires

Two questionnaires were used. The pre questionnaire consisted of seven questions that collected students' personal information, cities, background knowledge and opinions about programming. The post questionnaire consisted of four questions. The questions examined if students enjoyed the competition and are willing to participate in any upcoming competitions. It also examined if students agree on including online programming tools in schools and if programming helped them improve their problem-solving skills.

3.3. Participants

In 2022, 84 school students aged 14 years old from four schools in Palestine volunteered to participate in this study. Table 1 illustrates the schools, cities, gender, and the number of students from each school. 96.4% of students learned at least one programming language such as; Visual Basic, or used one online programming tool such as; drag and drop applications (Blockly, Scratch, and/ or App-inventor). A high percentage is expected, as Visual Basic (a high-level language) is mandatory in the Palestinian educational curriculum.

Table 1: List of Palestinian schools that participated in the experiment.

School	City	Gender	Number of students
Rosary Sisters High School	Jerusalem	Females	27
Tala'e Al-Amal School	Nablus	Females	18
Islamic Secondary School	Nablus	Males	25
Qura'an Academic School	Nablus	Males	14

4. Results and discussion

This section presents the results which address the research question illustrated in Section 3.1.

4.1. Effects of programming on students' application and analytical levels

As mentioned in Section 3.2.1, students participated in the experiment were provided with pre/ post assessment tests which consisted of five questions. Figure 5 shows that 35.7% of students had an increase in their application and analytical levels, considering that 21.4% of students answered the question correctly in the post-test, and 14.3% of students changed their answers from wrong in the pre-test to an answer close to right in the post-test. On the other hand, 60.7% of students who wrongly answered the pre-test had no change in their application and analysis levels.

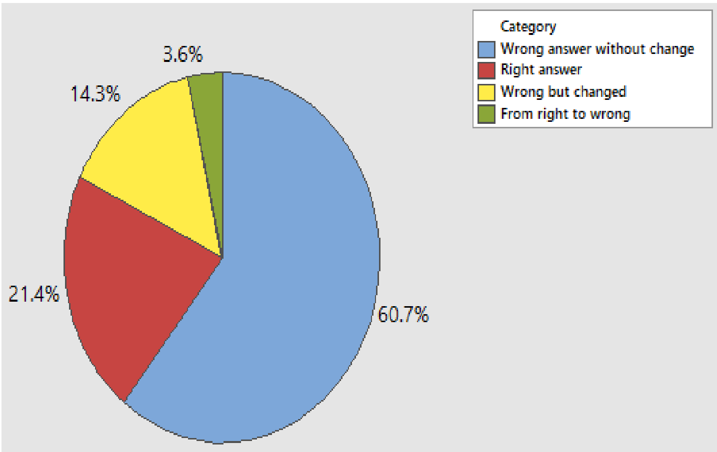

Figure 5: Students' answer to the first question.

Students had similar progress in the second question. Figure 6 shows that 32.1% of students had an increase in their application and analytical levels. On the other hand, 39.3% provided a conventional answer and 28.6% did not provide an answer.

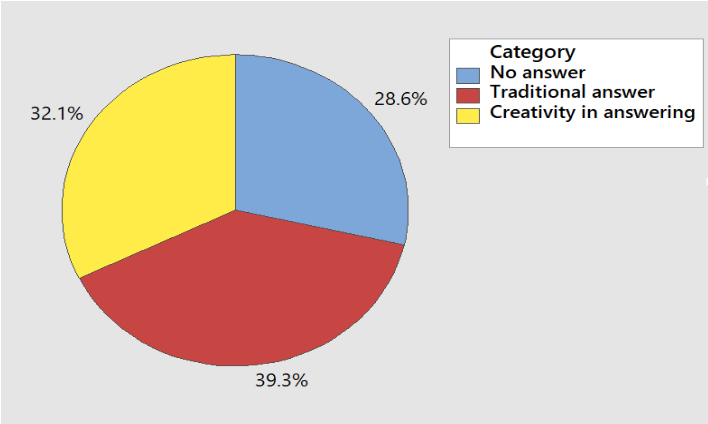

Figure 6: Students' answer to Rubik's cube question.

The pattern question was answered correctly by a higher percentage of students compared to the first and second questions. Figure 7 shows that 82.1% of students had an increase in their application and analysis levels, since 75.0% answered the question correctly in the post-test, and 7.1% of students changed their answers from wrong in the pre-test to an answer close to right in the post-test. On the other hand, 6.0% of students changed their answers from right to wrong, and 11.9% of students did not change their answers.

Similar to previous questions the fourth question showed that 26.2% of students had an increase in their application and analytical levels. The method used for analyzing this question was based on the reason students provided for selecting the shape. If a student changed the selected shape with a reasonable justification, the teachers considered it an increase in the application and analytical levels.

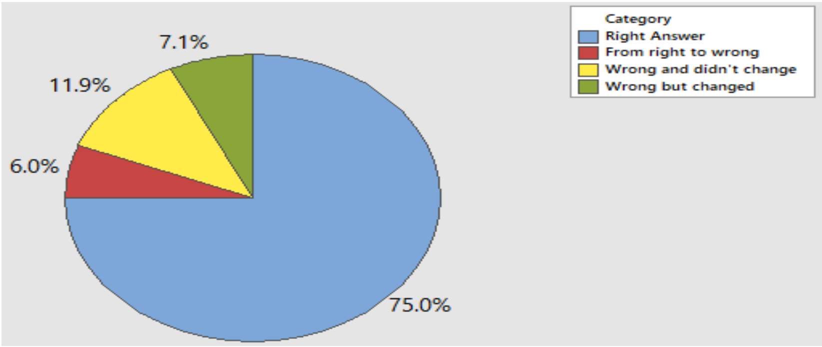

Figure 7: "Continue the pattern" question analysis.

The fifth question which focused on a detail-oriented approach also showed that 7.1% of students answered the question correctly in the post-test and 22.6% changed their answers from wrong in the pre-test to an answer close to right in the post-test. This indicates that 29.7% of students had an increase in their analytical cognitive level (see Figure 8).

On average, 41.16% of students had an increase in their application and analytical levels based on pre-test and post-test results.

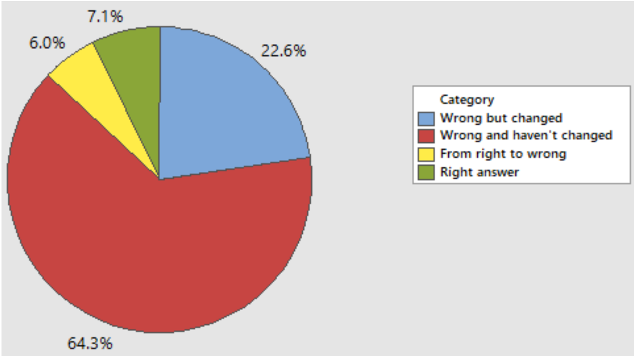

Figure 8: Result analysis of detail-oriented question.

The increase/decrease of students' scores in the five questions between the pre-test and post-test had a total score of 10. To study the effect of students' programming background knowledge on students' scores, the one-way analysis of variance test (ANOVA) was used. Figure 9 shows that students' programming background knowledge was categorized as follows:

- Zero indicates no programming background knowledge
- One indicates that students have background knowledge in drag and drop programming tools.
- Two indicates that students have background knowledge in high-level programming languages.
- Three indicates that students have background knowledge in both high-level programming languages and drag and drop programming tools.

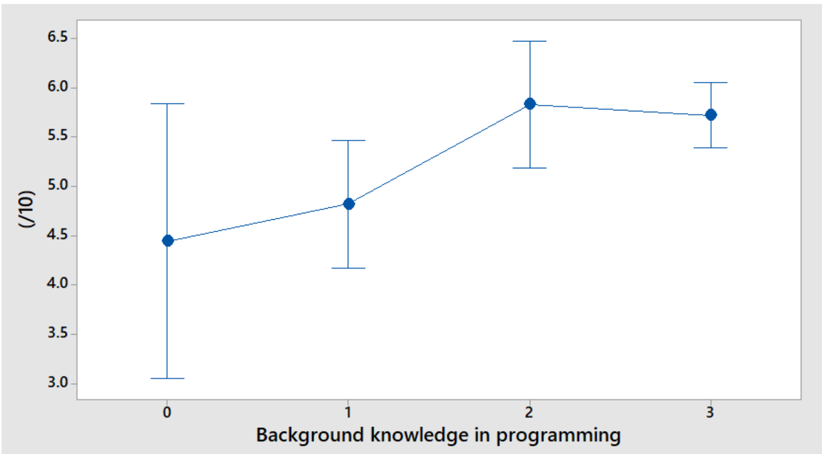

Figure 9: Scores of students with different background knowledge.

The results revealed that there is a significant difference between students' scores based on their background knowledge (%CI=95%, F-Value =2.16, P-Value =0.02, degrees of freedom (DF) = 3). Figure 9 shows that students who had background knowledge in high level programming and students who have background knowledge in both high level languages and drag and drop programming tools had the highest increase in their application and analysis levels.

4.2. Students' opinions of programming

The results revealed that 91.7% of students enjoyed the experience and are willing to repeat the competition (see Figure 10). In addition, the results revealed that 94.1% of students agreed on integrating online programming tools in schools (see Figure 11), and 56.0% of students agreed that programming helped them improve their problem solving skills (see Figure 12).

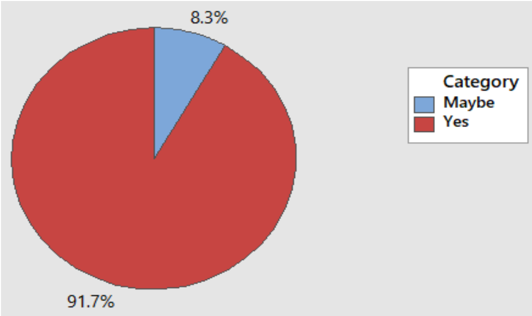

Figure 10: Would you participate in any upcoming competitions?

The results above indicate that students are willing to participate in programming competitions, and to attain more knowledge in programming in general. In addition, the results showed that students agree that programming helps them increase their motivation which consequently improves their learning productivity [23, 24].

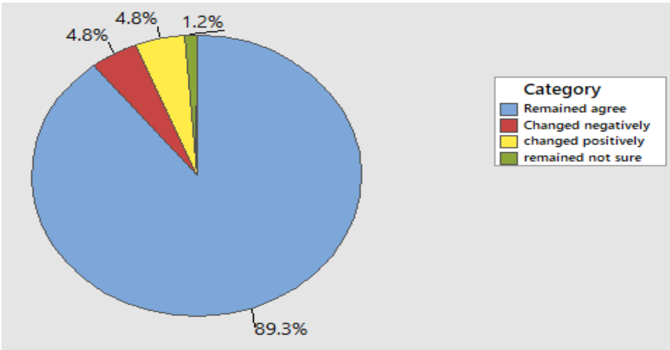

Figure 11: Do you agree on integrating online programming tools in schools?

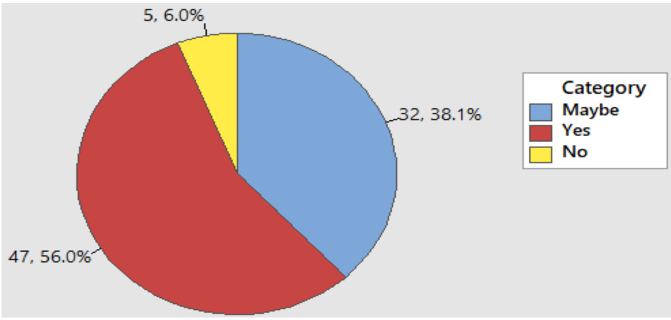

Figure 12: Does programming help you improve your problem solving skills?

5. Conclusion

This research studied the effect of online programming tools on school students' application and analytical levels. A pre/post experimental design was applied on 84 students at four schools in Palestine. The results revealed that 41.16% of students had an increase in their application and analytical levels. The results also revealed that students who had background knowledge in high level programming and students who have background knowledge in both high level languages and drag and drop programming tools had the highest increase in their application and analysis levels. The paper also showed that students agreed on including online programming tools in schools and that programming helped them improve their problem- solving skills.

In the future, this research will be carried out across a wider range of students, and schools in Palestine. In addition, programming hackathons will be organized across high schools in Palestine.

References

[1] Evgeniy Aleksandrovich Danchikov, Natalia Alekseevna Prodanova, Yulia Niko-laevna Kovalenko, and Tatiana Grigorievna Bondarenko. "Using different approaches to organizing distance learning during the covid-19 pandemic: opportunities and disadvantages." *Linguistics and Culture Review*, 5(S1):587–595, 2021.

[2] Marina Umaschi Bers. "Coding and computational thinking in early childhood: The impact of scratchjr in europe". *European Journal of STEM Education*, 3(3):8, 2018.

[3] Peter J Rich, Samuel F Browning, McKay Perkins, Timothy Shoop, Emily Yoshikawa, and Olga M Belikov. "Coding in k-8: International trends in teaching elementary/primary computing". *TechTrends*, 63(3):311–329, 2019.

[4] Diana Pérez-Marín, Raquel Hijón-Neira, Adrián Bacelo, and Celeste Pizarro. "Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children?" *Computers in Human Behavior*, 105:105849, 2020.

[5] Pauliina Tuomi, Jari Multisilta, Petri Saarikoski, and Jaakko Suominen. "Coding skills as a success factor for a society". *Education and Information Technologies*, 23(1):419–434, 2018.

[6] Idongesit Eteng, Sylvia Akpotuzor, Solomon O Akinola, and Iwinosa Agbon-lahor. "A review on effective approach to teaching computer programming to undergraduates in developing countries". *Scientific African*, page e01240, 2022.

[7] Xanthippi Tsortanidou, Thanasis Daradoumis, and Elena Barberá. "A k-6 computational thinking curricular framework: pedagogical implications for teaching practice". *Interactive Learning Environments*, pages 1–21, 2021.

[8] Dan Crow. “Computational thinking, the university of edinburgh, school of informatics” : https://www.inf.ed.ac.uk/research/programmes/comp-think/, Dec. 22, 2022.

[9] Emily Relkin, Laura E de Ruiter, and Marina Umaschi Bers. “Learning to code and the acquisition of computational thinking by young children”. *Computers & education*, 169:104222, 2021.

[10] Benjamin Samuel Bloom. “Taxonomy of educational objectives: The classification of educational goals: Cognitive Domain”. *Longman*, 1956.

[11] Sónia Rolland Sobral. “Bloom’s taxonomy to improve teaching-learning in introduction to programming”. 2021.

[12] James Zhang, Casey Wong, Nasser Giacaman, and Andrew Luxton-Reilly. “Automated classification of computing education questions using bloom’s taxonomy”. In *Australasian Computing Education Conference*, pages 58–65, 2021.

[13] Abdelghani Babori. “A didactic study of an algorithmic and programming mooc: Learning strategies adopted by students and their difficulties”. *International Journal of Emerging Technologies in Learning*, 17(19), 2022.

[14] Zahid Ullah, Adidah Lajis, Mona Jamjoom, Abdulrahman Altalhi, and Farrukh Saleem. “Bloom’s taxonomy: A beneficial tool for learning and assessing students’ competency levels in computer programming using empirical analysis”. *Computer Applications in Engineering Education*, 28(6):1628–1640, 2020.

[15] Abdelrahman Mohammad Abu Sarah. “Designing programmed-based educational activities to develop procedural knowledge and motivation towards learning mathematics among eleventh grade technology-stream students in palestine”. *Journal of Al-Quds Open University for Educational & Psychological Research & Studies*, 12(33):4, 2021.

[16] Cynthia C Selby. “Relationships: computational thinking, pedagogy of programming, and bloom’s taxonomy”. In *Proceedings of the workshop in primary and secondary computing education*, pages 80–87, 2015.

[17] Sarantos Psycharis and Maria Kallia. “The effects of computer programming on high school students’ reasoning skills and mathematical self-efficacy and problem solving”. *Instructional Science*, 45(5):583–602, 2017.

[18] Robert W Fox and Michael E Farmer. “The effect of computer programming education on the reasoning skills of high school students”. In *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)*, page 1. Citeseer, 2011.

[19] Robert H Seidman. “The effects of learning a computer programming language on the logical reasoning of school children”. 1981.

[20] Gavriel Salomon and David N Perkins. “Transfer of cognitive skills from programming: When and how?” *Journal of educational computing research*, 3(2):149–169, 1987.

[21] Stefano Federici. “minic++: a minimal, drag and drop environment to teach c/c++”: http://scratched.gse.harvard.edu/sites/default/files/minicprogram.png, Dec. 22, 2022.

[22] Bangkit Nugroho, Riyadi Riyadi, and Sri Subanti. “Profile of higher order thinking skill based on mathematical connection levels in industrial revolution 4.0”. In *Proceedings of the 2nd International Conference on Education, ICE* 2019, 27-28. September 2019, Universitas Muhammadiyah Purworejo, Indonesia, 2020.

[23] Ramli Bakar. “The effect of learning motivation on student’s productive competencies in vocational high school, west sumatra”. *International Journal of Asian Social Science*, 4(6):722–732, 2014.

[24] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. “Active learning increases student performance in science, engineering, and mathematics”. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415, 2014.