BUEFAD | 2022, Volume 11, Issue 2, 351-363
Bartın University Journal of Faculty of Education
dergipark.org.tr/buefad
DOI: 10.14686/buefad.991448

Pair Programming Experiences of Prospective Information Technologies Teachers

Ebru ALBAYRAK[a], Elif POLAT[b*]

a Dr., Sakarya University, Turkey, (https://orcid.org/0000-0003-1327-9576), ealbayrak@sakarya.edu.tr

b* Dr., Istanbul University-Cerrahpasa, Turkey, (https://orcid.org/0000-0002-6086-9002), *elif.polat@iuc.edu.tr

Research Article	Received: **05.09.2021**	Revised: **10.11.2021**	Accepted: **10.11.2021**

ABSTRACT

This study aims to reveal the experiences of undergraduate students regarding a pair programming method used in their programming course. Qualitative and quantitative methods were used to collect data for the study. The pair programming method required students to work in pairs throughout the semester. The participants of the study consist of 22 sophomores from computer education and instructional technologies department enrolled on the programming languages course. Collaboration Experiences, Team Member Evaluation, and Self-Assessment forms and a semi-structured interview form were used to collect data. The findings indicate that all the students were positive about the course. According to the students, collaboration within pairs was carried out successfully. At the end of the course, students stated that the lessons were sufficient for programming, and they achieved a good performance. In addition, the students were satisfied with the collaboration of their partner and the opportunities they had to improve their communication skills thorough pair programming. In addition, students emphasized that pair programming reduced the instructor's workload. However, a few students also stated that the process had some limitations. The findings of the study will be useful particularly for instructors while designing programming education.

Keywords: Computer Education and Instructional Technologies, CEIT, prospective information technologies teachers, pair programming, programming education, collaborative learning

Bilişim Teknolojileri Öğretmen Adaylarının Eşli Programlama Deneyimleri

ÖZ

Bu çalışma, üniversite öğrencilerinin, programlama dersinde kullanılan eşli programlama yöntemine ilişkin görüş ve deneyimlerini ortaya çıkarmayı amaçlamıştır. Çalışmada veri toplamak için nitel ve nicel yöntemler kullanılmıştır. Eşli programlama yöntemi ile öğrenciler dönem boyunca ikili olarak çalışmışlardır. Araştırmanın örneklemini eğitim fakültesi lisans programı olan programlama dilleri dersine kayıtlı 22 ikinci sınıf bilgisayar ve öğretim teknolojileri eğitimi (BÖTE) öğrencisi oluşturmaktadır. Verileri toplamak için "İşbirliği Deneyimleri", "Takım Üyesi Değerlendirme" ve "Öz Değerlendirme" formları ile yarı yapılandırılmış görüşme formu kullanılmıştır. Bulgular, tüm öğrencilerin ders hakkında olumlu görüşe sahip olduklarını göstermektedir. Öğrencilere göre işbirliği bu yöntemle başarıyla gerçekleştirilmiştir. Ders sonunda kodlama konusunda yeterli olduklarını ve iyi bir performans elde ettiklerini belirtmişlerdir. Bunun yanında öğrenciler, takım arkadaşlarının işbirliğinden memnun kalmışlar ve eşli programlama ile iletişim becerilerini geliştirme imkanı da bulmuşlardır. Ayrıca bu yöntemle eğitmenin iş yükünün de azaldığını belirtmişlerdir. Ancak az sayıda öğrenci sürecin bazı sınırlılıklarının olduğunu da belirmiştir. Çalışmanın bulguları özellikle programlama öğretiminin tasarlanmasında öğretim elemanları için faydalı olacaktır.

Anahtar kelimeler: Bilgisayar ve Öğretim Teknolojileri Eğitimi, BÖTE, bilişim teknolojileri öğretmen adayları, eşli programlama, programlama eğitimi, işbirlikli öğrenme

To cite this article in APA Style:
Albayrak, E. & Polat, E. (2022). Pair programming experiences of prospective information technologies teachers. *Bartın University Journal of Faculty of Education, 11*(2) 351-363. https://doi.org/10.14686/buefad.991448

1 | INTRODUCTION

In the 21st century, knowledge of computer science has become a key skill for adapting to the digital world (McManus & Costello, 2019). Therefore, today's education systems need to provide students with this skill (Demir & Seferoglu, 2021). Learning programming is a fundamental step in computer science education as it involves skills such as algorithmic thinking, problem solving, and designing, all of which are necessary for the future workforce (Witherspoon et al., 2016). However, for decades, students have failed to succeed in this area (Raigoza, 2017). Novice programmers in particular have difficulty learning the fundamentals, struggle to succeed, and often finally give up (Denner et al., 2021; Falloon, 2016). This is because programming includes syntax and how to set up logical structures (Korhonen & Malmi, 2000), requiring novice learners to develop advanced skills and learn syntax at the same time. This makes the learning process difficult and confusing for learners (Koulouri et al., 2014). A constructivist approach would be useful for learners to help them establish connections between the knowledge and skills they learn during the learning process. Learner-centered programming education needs to be provided, especially for novice learners, in which they make progress based on their own knowledge gained during the process. The research shows that programming needs both willingness and programming knowledge (Denner et al., 2021). It would be beneficial for learners to be supported in tasks that require them to use high-level skills such as analysis, constructing algorithms and coding in order to finally succeed (Falloon, 2016; Hwang et al., 2012). With collaborative learning, learners have the opportunity to better develop their problem solving skills and cognitive abilities by both receiving support and expressing their own ideas (Bernard & Bachu, 2015; Kuhn, 2015). Thus, programming education supported by collaborative learning is quite successful and increases learning performance (Preston, 2005). In addition, while learner self-confidence tends to be low in traditional programming education (Bravo et al., 2005), it increases in collaborative environments, helps them become more successful (Hwang et al., 2012) and enables learners to reach their goals in a shorter time by focusing more deeply (Williams et al., 2003).

For teaching programming, pair programming based on collaborative working is important in addressing learners' needs (Cao & Xu, 2005). With pair programming, students collaborate each other and solve the problems with their partner's support (Liebenberg et al., 2012). Two learners work on one computer, design a programming strategy, and develop software (Hanks et al., 2011; Williams et al., 2003). In this way, with two learners working on the same task rather than one, they arrive at more productive ideas and progress better (Beck & Gamma, 2000). In addition, the programming education is conducted in a planned way, and each individual fulfils their tasks in accordance with the role assigned to them. Within the pairs, each learner has different responsibilities (Campe et al., 2020): driver and navigator. The driver develops code while the navigator observes the driver, identifies any problems, develops a strategy, and comes up with solutions for any errors (Williams et al., 2000). Through the process, the pairs swap responsibilities so that both partners become involved in the brainstorming process (Tsan et al., 2020; Wei et al., 2021). In this way, a more efficient teaching environment is created that doesn't focus primarily on syntax and coding (Satratzemi et al., 2021). Sherriff (2017) highlighted the importance of pair programming in his study because this method improves retention and programming confidence, ensures better performance on exams and projects. In this study, Sherriff developed a form to examine in-depth, emphasizing that the attitudes of the students as an important variable in the "Computer Science 1" course using the pair programming method with 500 students. This form is also used in the present study. Braught et al. (2011) used this method to improve the programming skills of high school students and found that in pairs, students are more successful and more likely to complete the process than those who do programming on their own. In addition, as the learners make fewer mistakes in this process (Zacharis, 2011), the method enables them to be more effective and productive (Hannay et al., 2009). In their research with university students, McDowell et al. (2006) concluded that with pair programming, students continue to program for a longer period of time. In addition, pair programming supports learners socially as learners have the opportunity to improve their communication skills (Faja, 2011) and increase their interaction (Howard, 2006).

Research has shown that university students experience many problems in introductory programming courses. As we have seen, these courses, provided by many departments, are important for students' future careers (Witherspoon et al., 2016). However, despite the difficulties many students face, universities have continued to provide traditional educational methods, only giving importance to programming and syntax (Iqbal Malik, 2016; Vihavainen et al., 2011). In such an environment, learners have difficulty turning what they have learnt into

experience using programming structures and developing strategy; they often lack practical knowledge and fail to fully grasp the logic of programming (Bruhn & Burton, 2003).

The pair programming method, which is presented as a solution to this problem, makes the programming learning process more effective (Zhong et al., 2016). Dongo et al. (2016) tried a lab experiment including advanced programming with undergraduate students enrolled on Management Information Systems (MIS). The research showed that the students were more confident and successful. Another study observed that university students were more motivated by this method, resulting in them completing the process more successfully (Mentz et al., 2008). The method also helped university students develop reasoning skills (Othman et al., 2019) and they enjoyed the course more (Cliburn, 2003). Vasconcelos and Kim (2020) carried out a similar study teaching programming in teams with preservice teachers. The research concluded that the students learned programming better by brainstorming together. What is more, as Cliburn (2003) states, the burden on instructors decreased.

Despite the research referred to above, all aspects of the effects of pair programming still need to be investigated. Some studies have highlighted the limitations of pair programming. Balijepally et al. (2009) carried out a pair programming method with entry level university students on an information systems course. The research found that the less experienced partners made progress, but the more expert partner failed to make progress. In addition, a few studies have observed further success with this method (Chigona & Pollock, 2008; Hanks et al., 2004). While the effects of pair programming are mostly beneficial, some studies found that expectations were not fully reached. As mentioned above, university students have serious difficulties in learning programming. Although this issue has been studied for many years, there are still failures (Hanks et al., 2011). Pair programming, on the other hand, provides students with more confidence (Braught et al., 2011; Werner et al., 2004), increases their satisfaction and communication levels (Dongo et al., 2016), increases their high-level learning (Othman et al., 2019) and it is seen as an important method in programming education. Therefore, pair programming requires further research and the benefits and limitations of applying such a method need to be explored more extensively. This study will make an important contribution to the literature by revealing university students' in-depth evaluation of their own and their partner’s performance during the process.

RESEARCH QUESTIONS

The aim of the research is to reveal the experiences of university students in the pair programming process and seeks answers to the following research questions:

1. What is the collaborative experience of students in the pair programming process?
2. How do students evaluate themselves in the pair programming process?
3. How did students evaluate their partners in the pair programming process?

2 | METHOD

The aim of this study is to reveal the experiences of undergraduate students regarding the pair programming method. Creswell (2012) suggests that it would be better to use both quantitative and qualitative methods together in understanding the research problem. Therefore, this study includes both quantitative and qualitative means was used in the study to reveal the opinions and experiences of students. While the quantitative part included forms the qualitative part of the study used semi-structured interviews. The pair programming method was used in a programming course in which students worked in pair for a semester. Students completed assessment forms after each lesson. The courses were taught face to face in a computer lab for 14 weeks. The course is a basic programming course and the topics are algorithm, flowchart, basic concepts, variables, conditional expressions, operators, loop, array, methods. C# was used as the programming language. Google classroom was used for out-of-class sharing, submitting homework, and for questions and answers. Lecture notes, activities (exercises), and lecture presentations were shared each week. In addition, one-on-one interviews were conducted at the end of the semester for in-depth analysis.

PARTICIPANTS

Purposeful sampling was used in the study. The study involved 22 sophomore students of the Faculty of Education Computer Education and Instructional Technology Department on the Programming Languages 1 course.

13 of the participants of the study were female and 9 were male. The distribution of their ages was 21. A total of 7 students, 2 males and 5 females, participated in the qualitative part of the study. All were taking the course for the first time and all were computer literate. Pairs were formed at the beginning of the semester. Students are allowed to form their own groups so that they could choose to be with the friends they felt able to work with as it is not motivating to work with someone they would choose not to. This allowed them to work in a pair with a friend with whom they work generally work with during the course. Evidence of the benefit of this way of working is presented in Choi (2015)'s study. The author states that female and male students have different characteristics. Because they express themselves differently, females in particular prefer being in same gender pairs. In addition, the study emphasizes that females stated that they felt more comfortable and were able to communicate more openly when working with another female.

Two students worked collaboratively at one computer. One student was the driver and the other the navigator, so, for example, while the driver did program writing or coding, the navigator was involved in strategic thinking and guiding. The course is given by an instructor in a natural course environment. This instructor provided feedback and guidance to the students in all necessary situations and walked around monitoring the students during the lesson and the practice. The content of the course included basic concepts about computers and software, algorithm, flow charts, variables, constants, arithmetic operators, logical operators, control structure, if/else, switch/case, loops, break-continue, while, do-while, randomize, arrays, methods (functions), parameters, return (returning a value) and activities about them. Activities, from easy to difficult, are presented to reinforce these topics. Number puzzles, simple games, quiz preparation, automations are examples of activities.

INSTRUMENTS

Collaboration Experiences, Self-Assessment, and Team Member Evaluation forms were used in the research. The questions on the forms were adapted from Sherriff's (2017) study. The necessary permissions were obtained. In addition, a semi-structured interview form was prepared by the researchers for one-on-one interviews. Three educational technology expert opinions were taken for the forms and interview questions, and a pilot application was carried out. After the forms were finalized, they were distributed to the students.

Collaboration Experiences form: This consists of a total of 15 questions, two of which are open-ended and 13 of which are 5-point Likert-type questions, for example, "When I work with my partner, I feel responsible for his or her success. It saves time working with my friend while doing homework, I prefer to work with my friend on homework." Cronbach's Alpha was calculated as .79 which is appropriate.

Team Member Evaluation form: This consists of a total of 10 questions, one of which is open-ended and 9 of which are multiple-choice. For example, "Does the student make an effort to collaborate in the pair? Evaluate how compatible you and your partner are." Cronbach's Alpha was calculated as .82 which is appropriate.

Self-Assessment form: This form consists of a total of 11 questions, one of which is open-ended and 10 which are multiple-choice questions. For example, "When you have homework, when do you start? When you are working on a programming assignment, what do you think about? Did you study enough for your lesson?" Cronbach's Alpha was calculated as .81 which is appropriate.

Semi-structured Interview form: This form consists of 7 open-ended questions. For example, "What are your positive/negative experiences with pair programming? How was your communication with your partner during the pair programming process? Did you feel like a part of the pair during the process?"

DATA COLLECTION AND ANALYSIS

Participants were informed about the purpose of the study, the study instruments, and that it was voluntary involvement. The data collection process included the participants filling out the relevant forms and conducting one-on-one interviews. Participants filled the forms over a period of 9 weeks. It took approximately 30 minutes for the participants to fill out three forms. At the end of the term, semi-structured interviews lasting approximately 20-25 minutes were held with each participant.

Data analysis was done using SPSS 21 and Excel. Descriptive statistics were used. In addition, content analysis was applied to the qualitative data. During this process, the data obtained from the interviews, which is the first

stage of content analysis, was first categorized under certain codes. In the second stage, six themes were created covering these codes. The resulting codes and themes were tabulated. The frequency of the codes was added in Table 4. In order to ensure reliability of the study, two researchers analyzed the data obtained (Merriam, 2015). A hundred percent agreement was achieved between the researchers. In addition, the opinions of two field experts were taken of the codes and themes. After revisions, the final table was obtained (see Table 4). In addition, some notable student statements are included. The male and female participants were coded as M1, M2, F1, F2, F3, F4, F5.

3 | RESULTS

The aim of the research is to reveal the experiences of university students in the pair programming process. In this context, answers to the research questions were examined.

RQ1: COLLABORATIVE EXPERIENCES

As can be seen in Table 1, both the weekly means and the total means of the items are above 2, showing that students can work collaboratively. Although the total means of the scale vary according to the weeks, all means are above 2 and the average of the total means is above 3 (M=3.31, SD=0.33). Considering the total means of the items, Item 1 (M=2.93, SD=0.73) scored the lowest mean while Item 7 (M=3.95, SD=0.51) scored the highest mean. Accordingly, it can be said that the students do not feel responsible for the success of their friends and they get a lot of help from their friends while solving a difficult problem. 4, 8, 9th weeks have relatively low means. It can be said that they had difficulty in working collaboratively because they had difficulties in the problems in those weeks.

Table 1. Collaborative Experiences of Students

	Means of Weeks									Total	
Items	1	2	3	4	5	6	7	8	9	M	SD
1. When working with my partner, I feel responsible for her/his success.	3.64	3.20	3.22	2.00	3.00	3.67	3.60	2.00	2.00	2.93	0.73
2. Working with my friend while doing homework saves time.	3.88	3.50	3.67	2.00	3.67	3.33	3.75	2.00	2.00	3.09	0.83
3. I prefer to work on homework with my friend.	3.75	3.29	3.56	2.50	4.00	3.50	3.60	2.00	2.50	3.19	0.68
4. I prefer to work alone on large projects.	2.64	3.00	3.00	3.50	3.00	3.00	3.00	3.50	3.50	3.13	0.30
5. I learn more while working by myself.	3.25	3.57	3.56	4.00	3.33	3.25	3.80	4.00	4.00	3.64	0.32
6. I am more organized when doing homework with others.	3.81	3.33	3.67	3.00	3.67	3.50	4.00	1.50	3.00	3.28	0.75
7. When solving a difficult problem, I seek advice from my friends.	4.38	4.14	4.22	3.50	4.33	4.50	4.00	3.00	3.50	3.95	0.51
8. If I was paired with my friend, I could avoid coding bugs.	3.40	3.20	3.56	2.50	4.00	3.75	3.25	2.50	2.50	3.18	0.57
9. I have a tendency to procrastinate when working on my own.	3.44	3.13	3.33	2.50	2.00	4.00	3.60	2.50	2.50	3.00	0.65
10. If I had a choice, I would always work alone.	2.56	3.14	3.22	4.00	2.67	2.75	3.40	3.50	4.00	3.25	0.53
11. I get new ideas about problem solving from my friends.	4.13	4.07	4.00	2.50	4.33	4.25	4.00	3.50	2.50	3.70	0.72
12. If I am matched to work with a friend and this person is slow, I tell the course lecturer.	3.56	3.36	3.78	2.50	3.00	3.00	4.00	3.50	2.50	3.24	0.53
13. When I explain my logic to my partner, I sometimes find faults in my thinking.	3.13	3.33	3.78	3.50	3.50	3.75	3.40	3.00	3.50	3.43	0.26
Total	3.50	3.41	3.58	2.92	3.42	3.56	3.65	2.81	2.92	3.31	0.33

RQ2: Self-Assessment of Students

It can be seen from Table 2, more than half of the students (n=16, 72.7%) start their homework in the middle of the allocated time. 63.6 percent of the students (n=14) think that they are slower than others in programming. 77.3 percent (n=17) stated that they had no difficulty in starting to solve the programming problem and that they found computer science topics interesting. While almost all of them (n=21, 95.5%) stated that they liked to code, almost all of them (n=20, 90.9%) stated that they knew they would find the answer somewhere when faced with a problem.

Table 2. Students' Self-assessment

Questions	Answers	n	%
When you have homework, when do you start?	Too early	5	22.7
	Too late	1	4.5
	In the middle	16	72.7
When working on a programming assignment, what do you think about?	I am faster than others at solving programming tasks.	8	36.4
	It takes me longer to do programming tasks than my classmates.	14	63.6
When working on a programming problem:	I have no trouble starting to solve the problem.	17	77.3
	I don't know where to start solving new programming problems.	5	22.7
Regarding computer science topics I've seen so far:	Not all computer science topics I've seen so far have piqued my interest.	5	22.7
	I've found all computer science topics interesting and intriguing so far.	17	77.3
Regarding coding:	I hate coding.	1	4.5
	I love coding.	21	95.5
Did you study enough for your lesson?	I got the best grade I could get	12	54.5
	I only worked one-on-one	10	45.5
When helping others with computer science, you:	I know I can help others.	14	63.6
	I feel like I don't know enough to help others.	8	36.4
After the exam with computer science	I think I did badly.	8	36.4
	I think I'm doing very well	14	63.6
When I encounter an obstacle while thinking	My mind wanders to other things.	2	9.1
	I know I will find the answer somewhere.	20	90.9

RQ3: Experiences with Team Member

As can be seen from Table 3, it is seen that the feedbacks of the students about their teammates are mostly positive. Half of the students (n=11, 50%) stated that their teammates always attend the meeting, they always inform if they cannot attend the meeting, and they put a lot of effort into the homework given before the meeting. All of the students stated that their teammates usually or always both try to do what they can do and make an effort to cooperate.

Table 3. Student Experiences with their Team Member

	Never		Rarely		Sometimes		Usually		Always	
	n	%	n	%	n	%	n	%	n	%
Did the student attend pair meetings?	1	4.5	1	4.5	2	9.1	7	31.8	11	50
Did the student inform their partner if they were unable to attend a meeting or fulfil a responsibility?	2	9.1	-	-	3	13.6	6	27.3	11	50
Did the student put in a serious effort on the assignment given before the pair meetings?	1	4.5	-	-	5	22.7	5	22.7	11	50
Did the student try to do what they can do in pair meetings?	-	-	-	-	-	-	7	31.8	15	68.2
Did the student make an effort to collaborate in the pair?	-	-	-	-	-	-	6	27.3	16	72.7

In addition to quantitative data, the opinions of the students regarding the paired programming method applied in programming education were taken through interviews. The qualitative data was converted into specific codes and these codes were categorized under the relevant themes. Then, the tabulation process was done in a systematic way. The relevant data is in Table 4.

Table 4. Students' Opinions of Pair Programming

Theme	Code	f
Collaboration	Complementing each other in missing points	7
	Providing support	2
	Solving the problem together	4
	Discovering new solutions with a different perspective	2
	Brainstorming	4
Academic Support	More efficient	3
	Faster learning and solving problems	2
	Facilitating the solution process by talking about each other deficiencies	1
	Retention of information	2
	Rising academic performance	4
	Keeping active	2
	Decreasing anxiety	1
	Rising motivation	3
	Rising self-confidence	2
Communication	Good	7
	Improving communication skills	2
	Increasing communication with the instructor	4
	Decreasing the need for the instructor	1
	No change in communication with the instructor	1
	Increased communication with partner	7
	Increased understanding of the partner	3
	Started paying attention to ideas	1
	Felt their opinions mattered	1
	Partners understand each other clearly	2
	Felt like a part of the pair	7
Other	Feeling inadequate in the course subjects that both peers do not know	1
	Prolongation of the process due to disagreement	1
I prefer to work in pairs		7
I would like to benefit from pair-programming in the future		7

All of the students in the study had a positive approach toward pair programming. The codes generated from the research data were grouped under the themes of collaboration, academic support, communication, other, and preferring to pair-programming today and in the future. Under collaboration, the students stated that they provided the opportunity to complement each other in missing points (f:7). Secondly, it was stated that an environment of solving the problem together (f:4) and brainstorming (f:4) was formed. These statements were followed by providing support (f:4) and discovering new solutions with a different perspective (f:4). Noteworthy statements on the topic are as follows.

F1: "… *The exchange of information we made with each other during the learning process, the fact that we offered different solutions to each other for problem solving, helped me to gain different perspectives on problems and solutions in programming languages, and when I took into account my friend's suggestions for solutions, it*

helped me to say, "yes, it could be solved this way, I was writing the codes in a longer way" etc. and to create new suggestions for solutions."

F2: "*Since we were paired, we understood the problems better in the form of question and answer. We got instant feedback by asking each other questions. Since we were pairs, we had the opportunity to brainstorm. This helped us to express ourselves more easily. In this way, it enabled us to solve the problems that we encountered later more easily and from different perspectives*."

These statements show that the students obtained new solutions together by working collaboratively and benefiting from the brainstorming method and that this process works in a healthy way.

Another issue that students focused on was the academic support provided by the process. Here, students stated that their academic performance increased the most (f:4). In addition, students stated that their motivation (f:3) and the efficiency of the lesson increased (f:3). In addition, they learned and created solutions faster (f: 2), their learning was permanent (f: 2), the lesson was kept active (f: 2), their self-confidence increased (f: 2), the problem-solving process became easier by talking about each other deficiencies (f:1), and their anxiety reduced (f:1). Example statements on these issues are as follows;

F1: "*By talking, you can obtain different perspectives and solutions that your friend has built in his/her mind. In this process, I think that I expressed myself well and expressed my own solution proposals clearly*."

F5: "*We can get a different experience that is completed faster and easier and I can get many solutions.*"

M1: "*When you work with a friend, he/she tells you your weaknesses and you correct them. In this regard, you also gain self-confidence*."

M2: "*Knowing that I am not alone and my friend's thinking about the solution increases my motivation.*"

Communication in pair programming is another theme that the students mentioned. According to the data, all of the students stated that good communication was provided, their communication with their partner increased, and they felt themselves as a part of the pair (f:7). In addition, they stated that their communication with the instructor increased (f:4) and their understanding of their partner improved (f:3). The improvement of students' communication skills (f:2), understanding each other clearly with the partner (f:2), the decrease in the need for instructors (f:1), starting to give importance to ideas (f:1), and feeling that their ideas are important (f:1) were seen as other benefits of the pair-programming method. One student stated that his communication with the instructor did not change. Noteworthy statements on the topic are as follows:

F5: "*It allowed me to consult my friends and teachers more easily in problem sentences in the programs given to me later on. I started to evaluate from their point of view. In this way, I was able to obtain many different solutions*."

F2: "*My communication skills increased and I started to be more understanding and helpful*."

In addition, the students also stated limitations related to the method. These topics are grouped under the "other" heading. These are feeling inadequate in the course subjects that both peers do not know (f:1) and the prolongation of the process due to disagreement between peers (f:1). The following is an example statement: M2: "*Differences of opinion cause loss of time and motivation*". Finally, all students stated that they prefer to work with pairs in a programming lesson and they want to benefit from pair programming in their future careers (f:7).

As a result, with pair programming, students achieved better programming performance by working collaboratively and producing new and faster solutions to problems together. In this way, their motivation may increase during the process, which would help them feel more self-confident. In addition, their communication with their partners and instructor improved. Very few problems were experienced. At the end of this whole process, the students stated that they preferred pair programming and that they intend to use it in the future. In this respect, the process has been found to be beneficial.

4 | DISCUSSION AND CONCLUSIONS

In this study, students formed pairs within the scope of pair programming and under the guidance of an instructor. The pairs developed programs by working on the same project. According to the findings, the students felt that they could successfully work in a collaborative way (Satratzemi et al., 2021). Students did not feel responsible for the success of their partner. In this respect, it would be beneficial to provide an environment where students can empathize with each other.

Secondly, students were asked to evaluate themselves regarding the process. All of the students evaluated themselves positively and found themselves competent in programming. Since learning and interaction are closely related (Yildiz Durak, 2018), interaction is thought to be effective in achieving course objectives. Of the topics they studied, they felt they were most sufficient in coding. However, the number of students who had the impression that their partner progresses faster in programming or solving problems than they did is very high.

Another area of investigation in this research was students' experiences with their partner. The results show that students were satisfied with their partner. However, students also stated that there were problems in participating in pair meetings. To ensure active participation in pair meetings, instructors should meet with them from time to time to both motivate them both and control the process. The research shows that the students were prepared for pair work before the pair meetings and that they made an effort in this regard. In addition, students stated that their partner made an effort to collaborate. This finding supports the conclusion that they can successfully carry out the pair programming process.

The research also shows that students thought they achieved better academic performance (Dongo et al., 2016) and produced new and faster solutions to problems with their partners (Beck & Gamma, 2000). In addition, while solving the problems together, they completed each other's missing information. Owing to their partners, the students gained a different perspective and they were able to make their knowledge permanent through more effective learning. Also, they remained active in the course and their motivation for programming increased (Mentz et al., 2008). The students in the pair-programming course conducted by McDowell et al. (2006) continued programming for longer periods of time. It is possible that this is due to an increase in student motivation.

Another finding was that during pair programming, students were able to benefit from each other's ideas and improve their productivity skills by brainstorming (Tsan et al., 2020; Wei et al., 2021). In addition, the anxiety they felt at the beginning of the process decreased and their self-confidence increased. These findings are in line with the results of another study, Dongo et al. (2016), with university students, which observed that students are more successful and are more self-confident in pair programming. Information about the communication processes of the students was also obtained. According to the findings, students' communication with both their partners and instructor improved. In addition, their understanding of their partner increased and they developed in valuing ideas mutually. In this way, they were able to see themselves as part of the pair. Similarly, Faja (2011) states that learners found an environment to improve their communication skills and gained support through social media. In this way, learners can be involved in more interaction (Howard, 2006). On the other hand, a few students mentioned some limitations. One limitation was that the students could not progress when both partners failed to produce a solution. Another was that students sometimes disagreed about the solution to a problem. In their studies, Chigona and Pollock (2008) and Hanks et al. (2004) found that higher success was not achieved in pair programming as expected. This may be due to the limitations of the research or other problems. For a better process, the instructors should encourage their students to consult them in case of problems. In addition, pair programming was seen to reduce instructors' workload (Cliburn, 2003). This would enable instructors to allocate more time to students with problems. Finally, the students stated that they were satisfied with pair programming, want it to be applied again, and may use this method in the future when appropriate.

At the end of the process, the students are motivated, have self-confidence, communicate well, and, therefore, developed their programming skills. One of the biggest problems experienced by individuals in programming education in general is that many tasks need to be completed and that students often fail and leave the course (Denner et al., 2021; Falloon, 2016; Koulouri et al., 2014; Raigoza, 2017). The findings of this research suggest that pair programming may be an effective alternative teaching method in programming education in solving many of the existing problems. In literature, there are a few studies that resulted further success with this method

(Chigona & Pollock, 2008; Hanks et al., 2004). The study revealed university students' evaluation of their own and their partner's performance elaborately. In future research, this method can be tested with a larger group of students. In addition, different possible effects of pair programming can be examined in terms of individual differences such as personality type and learning preference. Some students may progress faster than their partners, and this may not positively affect the other partner. To address this, the instructor may monitor the pairs to identify those students having difficulties and offer further guidance. In addition, extracurricular activities should be done to create team spirit in order to adapt to the process. This study and studies in the literature have also shown that negative situations can occur, albeit in small numbers. Observing larger groups by taking into account their individual characteristics may help to address any deficiencies by better clarifying the problems that arise in pair programming. Many teaching approaches used compatible with programming education in the field. In future research, pair programming can be compared with different teaching approaches such as cooperative learning (Garcia, 2021), programming in the context of digital stories (Burke, 2012; Zha et al., 2020), the use of tools such as Scratch (Bean et al., 2015; Burke, 2012; Kalelioglu, & Gülbahar, 2014).

STATEMENTS OF PUBLICATION ETHICS

Istanbul University - Cerrahpasa's ethical committee were approved the data collection procedures and the study was recruited by following the ethical standards with 15153 ID.

RESEARCHERS' CONTRIBUTION RATE

Author 1: Conceptualization; Literature study; Investigation; Reporting; Original drafting, reviewing and editing

Author 2: Conceptualization; Methodology; Running the project; Original drafting, reviewing and editing

CONFLICT OF INTEREST

Authors declare no competing interests.

REFERENCES

Balijepally, V. G., Mahapatra, R. K., Nerur, S., & Price, K. H. (2009). Are two heads better than one for software development? The productivity paradox of pair programming. *MIS Quarterly*, *33*(1), 91–118. https://doi.org/10.2307/20650280

Bean, N., Weese, J., Feldhausen, R., & Bell, R. S. (2015, October). Starting from scratch: Developing a pre-service teacher training program in computational thinking. In *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1-8). IEEE. https://doi.org/10.1109/FIE.2015.7344237

Beck, K., & Gamma, E. (2000). *Extreme programming explained: embrace change*. Addison-Wesley professional. https://books.google.com.tr/books?hl=tr&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=Extreme+programming+explained:+embrace+change&ots=jbxErnfZts&sig=C75q_KZo5lNfH45BOx4l-cZwxRg&redir_esc=y#v=onepage&q=Extreme%20programming%20explained%3A%20embrace%20change&f=false

Bernard, M., & Bachu, E. (2015). Enhancing the metacognitive skill of novice programmers through collaborative learning. In *Metacognition: Fundaments, Applications, and Trends* (pp. 277-298). Springer, Cham. https://doi.org/10.1007/978-3-319-11062-2_11

Braught, G., Wahls, T., & Eby, L. M. (2011). The case for pair programming in the computer science classroom. *ACM Transactions on Computing Education (TOCE)*, *11*(1), 1-21. https://doi.org/10.1145/1921607.1921609

Bravo, C., Marcelino, M. J., Gomes, A. J., Esteves, M., & Mendes, A. J. (2005). Integrating educational tools for collaborative computer programming learning. *The International Journal of Universal Computer Science*,

11(9), 1505-1517. http://jucs.org/jucs_11_9/integrating_educational_tools_for/jucs_11_9_1505_1517_cbravo.pdf

Bruhn, R. E., & Burton, P. J. (2003). An approach to teaching Java using computers. *ACM SIGCSE Bulletin*, *35*(4), 94-99. https://doi.org/10.1145/960492.960537

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. https://files.eric.ed.gov/fulltext/EJ985683.pdf

Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: Variations in interactions and behaviors. *Computer Science Education*, *30*(1), 22-46. https://doi.org/10.1080/08993408.2019.1648119

Cao, L., & Xu, P. (2005). Activity patterns of pair programming. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* 3-5 January (pp. 88a-88a). IEEE. https://doi.org/10.1109/HICSS.2005.66

Chigona, W., & Pollock, M. (2008). Pair programming for information systems students new to programming: Students' experiences and teachers' challenges. In *PICMET'08-2008 Portland International Conference on Management of Engineering & Technology* 27-31 July (pp. 1587-1594). IEEE. https://doi.org/10.1109/PICMET.2008.4599777

Choi, K. S. (2015). A comparative analysis of different gender pair combinations in pair programming. *Behaviour & Information Technology*, *34*(8), 825-837. https://doi.org/10.1080/0144929X.2014.937460

Creswell, J. W. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research* (4th ed.). Boston, MA: Pearson.

Cliburn, D. C. (2003). Experiences with pair programming at a small college. *Journal of Computing Sciences in Colleges*, *19*(1), 20–29. https://dl.acm.org/doi/abs/10.5555/948737.948741

Demir, Ö., & Seferoglu, S. S. (2021). A Comparison of solo and pair programming in terms of flow experience, coding quality, and coding achievement. *Journal of Educational Computing Research*, *58*(8), 1448-1466. https://doi.org/10.1177/0735633120949788

Denner, J., Green, E., & Campe, S. (2021). Learning to program in middle school: How pair programming helps and hinders intrepid exploration. *Journal of the Learning Sciences*, 1-35. https://doi.org/10.1080/10508406.2021.1939028

Dongo, T., Reed, A. H., & O'Hara, M. (2016). Exploring pair programming benefits for MIS majors. *Journal of Information Technology Education: Innovations in Practice*, *15*, 223-239. http://www.informingscience.org/Publications/3625

Faja, S. (2011). Pair programming as a team-based learning activity: A review of research. *Issues in Information Systems*, *XII*, (2), 207–216. https://doi.org/10.48009/2_iis_2011_207-216

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. *Journal of Computer Assisted Learning*, *32*(6), 576-593. https://doi.org/10.1111/jcal.12155

Garcia, M. B. (2021). Cooperative learning in computer programming: A quasi-experimental evaluation of Jigsaw teaching strategy with novice programmers. *Education and Information Technologies*, *26*, 1-18. https://doi.org/10.1007/s10639-021-10502-6

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: A literature review. *Computer Science Education*, *21*(2), 135-173. https://doi.org/10.1080/08993408.2011.579808

Hanks, B., McDowell, C., Draper, D., & Krnjajic, M. (2004). *Program quality with pair programming in CS*. In Proceedings of the 9th annual SIGCSE conference on innovation and technology computer science education June 28-30 (pp. 176–180). https://doi.org/10.1145/1007996.1008043

Hannay, J. E., Dybå, T., Arisholm, E., & Sjøberg, D. I. K. (2009). The effectiveness of pair programming: A meta-analysis. *Information and Software Technology*, *51*(7), 1110–1122. https://doi.org/10.1016/j.infsof.2009.02.001

Howard, E. V. (2006). Attitudes on using pair-programming. *Journal of Educational Technology Systems*, *35*(1), 89–103. https://doi.org/10.2190/5K87-58W8-G07M-2811

Hwang, W. Y., Shadiev, R., Wang, C. Y., & Huang, Z. H. (2012). A pilot study of collaborative programming learning behavior and its relationship with students' learning performance. *Computers & Education*, *58*(4), 1267–1281. https://doi.org/10.1016/j.compedu.2011.12.009

Lqbal Malik, S. (2016). *Role of ADRI model in teaching and assessing novice programmers*. Doctorate dissertation, Deakin University. https://dro.deakin.edu.au/view/DU:30088862

Kalelioglu, F., & Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: a discussion from learners' perspective. *Informatics in Education*, *13*(1), 33-50. https://www.ceeol.com/search/article-detail?id=123695

Korhonen, A., & Malmi, L. (2000, July). Algorithm simulation with automatic assessment. In *Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education* (pp. 160-163). https://doi.org/10.1145/343048.343157

Koulouri, T., Lauria, S., & Macredie, R. D. (2014). Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Transactions on Computing Education (TOCE)*, *14*(4), 1-28. https://doi.org/10.1145/2662412

Kuhn, D. (2015). Thinking together and alone. *Educational Researcher*, *44*(1), 46-53. https://doi.org/10.3102/0013189X15569530

Liebenberg, J., Mentz, E., & Breed, B. (2012). Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT). *Computer Science Education*, *22*(3), 219-236. https://doi.org/10.1080/08993408.2012.713180

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, *49*(8), 90–95. https://doi.org/10.1145/1145287.1145293

McManus, J. W., & Costello, P. J. (2019). Project based learning in computer science: A student and research advisor's perspective. *Journal of Computing Sciences in Colleges*, *34*(3), 38-46. http://www.ccsc.org/publications/journals/JCSC_34_3_complete.pdf#page=39

Mentz, E., Van der Walt, J. L., & Goosen, L. (2008). The effect of incorporating collaborative learning principles in pair programming for student teachers. *Computer Science Education*, *18*(4), 247-260. https://doi.org/10.1080/08993400802461396

Merriam, S. B. (2015). Qualitative research: Designing, implementing, and publishing a study. In Victor X. Wang (Ed.), *Handbook of Research on Scholarly Publishing and Research Methods* (pp. 125-140). Hershey, PA: IGI Global. https://doi.org/10.4018/978-1-4666-7409-7.ch007

Othman, M., Rosmani, A. F., Fauzi, S. S. M., & Mazlan, U. H. (2019). The impact of pair programming on students' logical thinking:_A case study on higher academic institution. *Social and Management Research Journal*, *16*(1), 85-98. https://doi.org/10.24191/smrj.v16i1.6085

Preston, D. (2005). Pair programming as a model of collaborative learning: A review of the research. *Journal of Computing Sciences in colleges*, *20*(4), 39-45. https://dl.acm.org/doi/abs/10.5555/1047846.1047852

Raigoza, J. (2017). A study of students' progress through introductory computer science programming courses. In *2017 IEEE Frontiers in Education Conference (FIE)* 18-21 October (pp. 1-7). https://doi.org/10.1109/FIE.2017.8190559

Satratzemi, M., Xinogalos, S., Tsompanoudi, D., & Karamitopoulos, L. (2021). A two-year evaluation of distributed pair programming assignments by undergraduate students. *Research on E-Learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives*, 35-57. Switzerland: Springer Nature. https://doi.org/10.1007/978-3-030-64363-8_3

Sherriff, (2017). Pair Programming in the Classroom**.** http://www.cs.virginia.edu/tapestry/years/2017/resources/day-2/2017-tapestry-day-2-pair-programming.pdf

Tsan, J., Vandenberg, J., Zakaria, Z., Wiggins, J. B., Webber, A. R., Bradbury, A., Lynch, C., Wiebe, E., & Boyer, K. E. (2020). A comparison of two pair programming configurations for upper elementary students. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* March 11 – 14 (pp. 346-352). https://doi.org/10.1145/3328778.3366941

Vasconcelos, L., & Kim, C. (2020). Preparing preservice teachers to use block-based coding in scientific modeling lessons. *Instructional Science*, *48*(6), 765-797. https://doi.org/10.1007/s11251-020-09527-0

Vihavainen, A., Paksula, M., & Luukkainen, M. (2011). Extreme apprenticeship method in teaching programming for beginners. In *Proceedings of the 42nd ACM technical symposium on Computer science education* March 9 – 12 (pp. 93-98). https://doi.org/10.1145/1953163.1953196

Wei, X., Lin, L., Meng, N., Tan, W., & Kong, S. C. (2021). The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. *Computers & Education*, *160*, 104023. https://doi.org/10.1016/j.compedu.2020.104023

Werner, L. L., Hanks, B., & McDowell, C. (2004). Pair-programming helps female computer science students. *Journal on Educational Resources in Computing (JERIC)*, *4*(1), 4-es. https://doi.org/10.1145/1060071.1060075

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE software*, *17*(4), 19-25. https://doi.org/10.1109/52.854064

Williams, L., McDowell, C., Nagappan, N., Fernald, J., & Werner, L. (2003). Building pair programming knowledge through a family of experiments. In *2003 International Symposium on Empirical Software Engineering*, 30 September-1 October (pp. 143-152). IEEE. https://doi.org/10.1109/ISESE.2003.1237973

Witherspoon, E. B., Schunn, C. D., Higashi, R. M., & Baehr, E. C. (2016). Gender, interest, and prior experience shape opportunities to learn programming in robotics competitions. *International Journal of STEM Education*, *3*(1), 1-12. https://doi.org/10.1186/s40594-016-0052-1

Yildiz Durak, H. (2018). Digital story design activities used for teaching programming effect on learning of programming concepts, programming self-efficacy, and participation and analysis of student experiences. *Journal of Computer Assisted Learning*, *34*(6), 740-752. https://doi.org/10.1111/jcal.12281

Zacharis, N. Z. (2011). Measuring the effects of virtual pair programming in an introductory programming java course. *IEEE Transactions on Education*, *54*(1), 168–170. https://doi.org/10.1109/TE.2010.2048328

Zha, S., Jin, Y., Moore, P., & Gaston, J. (2020). Hopscotch into coding: introducing pre-service teachers computational thinking. *TechTrends*, *64*(1), 17-28. https://doi.org/10.1007/s11528-019-00423-0

Zhong, B., Wang, Q., & Chen, J. (2016). The impact of social factors on pair programming in a primary school. *Computers in Human Behavior*, *64*, 423-431. https://doi.org/10.1016/j.chb.2016.07.017