American Journal of Qualitative Research
2024, Vol. 8 No. 2, pp. 116-135
https://doi.org/10.29333/ajqr/14396

AMERICAN JOURNAL OF QUALITATIVE RESEARCH
ISSN: 2576-2141

Examining the Foreign Language Writing Experiences of Computer Engineering Students

Hakan Bayezit[1]
Turkish National Defense University, Türkiye

Tufan Adiguzel
Ozyegin University, Türkiye

ABSTRACT
Identifying the relation between the processes of programming and foreign language writing may lead to new directions for programming language and natural language focused instructional design. The present qualitative case study supported by quantitative data investigated foreign language writing experiences of computer engineering students taking an object-oriented programming course. Forty-five sophomores learning programming and academic English simultaneously in a foundation university in Ankara, Turkey, were selected purposefully for the case study. There were two data sources (students' opinions and documents) and three data collection tools (a semi-structured interview, a short diary, and a composition. In terms of the findings of the research, four themes were obtained; however, only the use of metacognitive strategies will be explained in detail due to the length of the study. Participants stated that they feel the positive effects of programming experience on the use of self-evaluation strategy and that there are similarities and differences between the processes of programming and foreign language writing. Participants' views on the effect of programming on foreign language writing did not differ according to their writing and programming performance scores. Participants stated that programming experience may have an effect on the use of metacognitive language learning strategies in the writing process. Upon analyzing participants' comments, it is understood that programming experience does not hinder the use of metacognitive strategies but has a role in supporting and reinforcing their use. It is suggested that multiple case studies be done for similar views on the effects of programming and that each finding be proven by quantitative studies.

KEYWORDS: programming, foreign language writing, metacognitive strategies.

The skills of the workforce and individuals in the 21st century have diverged from the skills needed in the previous century in parallel with the developments in information and communication technologies (Anderson, 2008; Dede, 2010). Examining 12 different frameworks on what skills are needed in the 21st century, Binkley et al. (2012) classified these skills into four themes: ways of thinking (creativity and innovation, critical thinking, problem-solving and decision-making; learning to learn, and metacognition), ways of working (communication and collaboration), tools for working (information literacy, information and communication technology [ICT] literacy), and living in the world (local and global citizenship, life and career, personal and social responsibility). According to Silva (2009), who said that

[1] Corresponding Author: An Instructor in the Department of Foreign Languages at the Army NCO Vocational HE School, Turkish National Defense University, Türkiye. E-Mail: hbayezit@msu.edu.tr; hakanbayezit@gmail.com

skills such as critical thinking, analytical thinking and creative thinking from 21st century skills do not belong to the new century, the main emphasis here should be on what can be done with this knowledge. However, the business community continues to express that they have difficulty finding educated individuals who can do the work they offer society today (Van Damme, 2017). As a result, while the workforce decreases in the work that can be done by programming computers, the need for work requiring thinking and communication increases. Also, there is a need for individuals who can leave a strategy that does not work while solving problems, who have a strategy repertoire and can choose the strategy group suitable for the next problem, and have mastered metacognitive strategies (Levy & Murnane, 2004). In this context, it is foreseen that communication in a common foreign language will continue to be important with the introduction of technology in the globalized world, and the importance of programming in the context of managing information will further increase. The importance of computer programming is noted in those words of Resnick et al. (2009):

> *Many young people are very comfortable sending text messages, playing online games, and browsing the Web. But does that really make them fluent with new technologies? Though they interact with digital media all the time, few are able to create their own games, animations, or simulations. It's as if they can read but not write.* (p. 62)

Just as "listening and speaking skills" in the first language learning process or "reading and writing skills" in literacy are learned together, learning computer programming as well as using word processing programming is said to be necessary in order to emphasize the distinction between individuals who produce and consume in today's digital world:

> *In the emerging, highly programmed landscape ahead, you will either create the software or you will be the software. It's really that simple: Program, or be programmed. Choose the former, and you gain access to the control panel of civilization.* (Rushkoff, 2010, p. 7)

Today, platforms such as Scratch, CodeHS and Code.org are trying to give young students basic programming skills on the basis of the idea that programming is a means of thinking and producing just like reading and writing. Also, individuals learning programming from an early age can take part in automation-based economics and management. Knowing what is behind the programming and algorithm-based services that drive daily life and solving individual daily problems is only possible with programming that takes place in a cognitive process (Bers, 2019). For example, the CodeHS platform offers entry-level courses for Python, Java, and SQL, and school-age children are also taught basic level programming in many countries (Duncan et al., 2014; Howland & Good, 2015). In the programming process, once a need arises, requests are forwarded to the programmer. Then, design, coding, testing, debugging, and maintenance steps are applied. In the model preferred, the transitions between these steps can be more flexible and adaptable to the changes according to the characteristics of the project (Dooley, 2011).

Writing, another thinking process, is a cognitive activity, and the cognitive performance can be likened to the computer's processing of information (Hyland, 2016). Writing is an activity in which an author gathers intellectual resources, discovers and organizes one's ideas in order to solve a problem by treating the task of writing as a problem. Developed by Flower and Hayes (1981, p. 369), the writing process, which successful authors carry out using self-repeating strategies in a nonlinear line, consists of three modules: "the task environment (rhetorical problem and text produced so far), the writer's long-term memory and the writing processes (planning, translating and reviewing which are under the control of a monitor)." This

model remains the most accepted approach by teachers (Atkinson, 2018; Ferris & Hedgcock, 2013; Hyland, 2019). According to Oxford (2011), each step in the process is actually a language-learning strategy.

When the steps involved in the programming process (Dooley, 2011) are compared with the ones in the cognitive writing process (Hyland, 2003), it is seen that the steps of the two processes are similar. Barnett and Ceci (2002) emphasized that the degree to which different fields of knowledge have common elements would affect the transfer. They also attributed the difficulty of interdisciplinary transfer to the fact that students are not being proficient in seeing the similarities between the two fields. Therefore, the transfer potential can be considered to exist due to the overlapping steps in the programming and writing processes. Schuster et al. (2020) stated that cognitive strategies are field-specific strategies, and their transfer to another field is limited. However, since metacognitive strategies are general strategies that manage the learning process (Donker et al.; Schraw, 2001), it may be possible to transfer them to various learning tasks and new contexts just as a metacognitive strategy has an effect on both reading novels and experimental work.

Programming has traditionally been interpreted with problem-solving skills, and the steps of planning and dividing a problem into parts in the programming process have been mainly focused and studied. However, little is known about experimental studies on the relationship between programming and other cognitive skills (especially language skills), and those studies found in the literature are limited to metaphorical descriptions and comparisons in terms of syntax and semantics (Bers, 2018; Fedorenko et al., 2019; Hermans & Aldewereld, 2017; Taylor & Paine, 1993; Strawhacker & Bers, 2019). When the introduction of programming at all levels of education, the importance given to programming and foreign language learning by organizations such as Horizon Report (Adams Becker et al., 2017), International Society for Technology in Education (ISTE Standards, 2008) and Partnership for 21st Century Learning (P21, 2013), the importance of natural languages influencing each other, the similarity studies between programming languages and natural languages, researches on the comparison of programming and natural language writing processes are taken into consideration, it has been found out there is not sufficient literature on what kind of effects programming learning can have on foreign language learning (Akcay et al., 2018).

Especially on the basis of the idea that writing in natural language and program development in programming language consist of overlapping steps similar to each other but produce different products, the possible effects of programming process and programming languages, having a popular place at different levels in educational environments, on learning foreign languages can be discovered. This could give a new direction to programming language and natural language-oriented teaching designs. For example, Kay (1998), emphasizing that writing is similar to the program development process, suggests that computer science students can carry out the writing process more easily based on these similarities: organization, compliance with certain rules, iterative process, etc.

In this context, taking the opinions of the participants to understand whether the programming experience has a transfer effect (Perkins & Salomon, 1992) on the foreign language writing process, such as the adaptation of the central control strategy used in chess to military operations due to the similarities seen between the two processes, may lead to giving meaning to situations and experiences and then testing more general inferences (Frankel & Devers, 2000) where there is not enough in-depth research in this field or may lead to the generalization of findings into theory (Creswell, 2014) within situations in the study environment through future multi-case studies. Based on the similarity of the steps in foreign language writing and programming processes, exploring the effects of learning programming within the scope of 21st-century skills on English as a foreign language can guide new approaches and strategies in instructional designs. Therefore, the aim of this study is to reveal the opinions of computer engineering sophomores who are learning programming on their

writing experiences in a foreign language. It concentrated on addressing two research questions: The first research question attempted to explore the thoughts of computer engineering students about whether they see the effect of their programming experience on foreign language writing. With the second research question, it was examined whether there was a difference between the views of the students in the upper and lower groups, especially in the profile that emerged in terms of the writing and programming performances of the participants.

Methods

This study employed qualitative research method supported by quantitative data to identify foreign language writing experiences of computer engineering students. Since there is no in-depth study on the experiences of programming and foreign language learners in the literature, the basic idea of this study is to choose a case and explain how the case depicts the problem (Creswell, 2017/2016). The sequence suggested by Yin (2009) was followed in the planning of the stages of the case study. After the data collection and analysis process were carried out, instead of making statistical generalizations to the universe in order to interpret the results, both detailed and concrete evidence was found in the light of the in-depth findings obtained from the data sources. An interview form was used as the main data collection tool in this study (Yildirim & Simsek, 2008). The diary was used as the second data collection tool in line with Creswell's (2017/2016) recommendation of "collecting data from multiple data sources to provide an in-depth perspective on the subject matter" (p. 268). In addition, one writing paper for each participant was examined in five different sub-dimensions (content, organization, language use, punctuation, and reviewing strategy) in order to illustrate the foreign language writing profiles of the participants. Yildirim and Simsek (2008) state that checking data by comparing results obtained through data triangulation will contribute to the reliability and validity of the research. Table 1 displays the data sources, data collection instruments, and data analysis aligned with the research questions.

Table 1
Research Design

Research Questions	Data Sources	Data Collection Instruments	Data Analysis
(1) What are the opinions of computer engineering students who take programming courses on their foreign language writing experiences?	Participants' opinions	Interview form Short diary	Content analysis
(2) Do students' views on the effect of programming on the writing process change according to the academic performance of writing and programming? If so, in what direction does it change?	Participants' opinions Composition	Interview form Short diary Writing rubric	Content analysis Descriptive and Inferential analysis

Participants

Computer engineering sophomores studying at a foundation university in Ankara took part in the study as participants (N = 82) while taking the third programming course and the third academic English course in the first semester of the 2019-2020 Academic Year. In terms of the sampling method, purposive sampling "allows for in-depth research by selecting

information-rich cases depending on the purpose of the study" (Buyukozturk et al., 2016, p. 90). It is also suggested that a group that can give the best answer to the research questions should be consciously chosen during data collection with data collection tools such as interviews, observations, and document analysis (Creswell, 1998; Miles & Huberman, 1994). Therefore, it is accepted that especially computer engineering students at the university can have the deepest knowledge about programming rather than other departments' students who take some programming courses. Since plagiarism was detected in the composition of two students at the end of the semester by the course teachers and four students did not submit their compositions, 76 (24 female, 52 male) students were included in the study. 45 students agreed and 31 students did not agree to be interviewed. In terms of English learning time, it was determined that the majority of the students (N = 76) who were contacted and the majority of the students (n = 45) who were interviewed had been learning English for more than eight years. English writing and programming course scores of the participants can be seen in Table 2.

Table 2
Number of Participants Across by English and Programming Course Scores

Score Range	Participants Contacted		Participants Interviewed	
	English	Programming	English	Programming
90-100	16	10	7	4
85-89	12	7	8	4
80-84	15	9	8	7
75-79	6	14	4	11
70-74	13	11	9	6
65-69	13	7	9	3
60-64	1	6	-	2
50-59	-	9	-	5
Below 50	-	3	-	3
Total	76	76	45	45

Data Collection

In the process of collecting the data needed for the research; views of the participants were collected with a semi-structured interview form and a short diary. In addition, each participant wrote one composition.

Interview Form

As a result of the readings made on the literature, fourteen interview questions were prepared in line with the proposition assuming that learning programming may have an effect on writing and English in a foreign language. While preparing for the interview, the researcher wanted the opinions of nine computer engineers working in a defense industry company (with three groups of three people at different times), two computer engineers working in different companies, an electronics engineer who took programming courses, two computer and instructional technologies teachers and two English teachers. In the pilot interview conducted with five computer engineering students, the explanations and examples given by the students to the questions enabled the researcher to develop a positive opinion in terms of the understandability of the questions. To illustrate interview questions, questions three and four can be seen below:

- How do you think what you do during the analysis phase of the program development process affects your writing in a foreign language?
- What effects do you think doing research during the program development process has on your writing assignment?

Short Diary

The researcher requested participants to keep a diary of their thoughts about the possible effects of learning programming on the foreign language writing process. This could be used to verify information obtained from other sources (Yin, 2009). Participants were given the steps of foreign language writing process (Hyland, 2003) and they tried to explain their thoughts on each step. For example, participants considered the revising step and wrote an opinion on the matter.

Writing Rubric

One composition written by participants in a nine-week period during the semester (the first draft version and the final version of the article) was received from the department management by covering the names and assigning a nickname to each. It was assumed that students applied the revision strategy if a statistically significant difference was obtained between the first draft and the final draft essay scores. To examine the first and the final draft versions of the participants' compositions, the rubric for the academic writing course taught in the Department of Modern Languages of Middle East Technical University was preferred. Compositions were evaluated on content, organization, language use, and punctuation.

Data Analysis

The data obtained from the interview and short diary was analyzed within the same method. First, the interview recordings were converted into text format in the form of a Microsoft Word document so the researcher could internalize the study by "embedding in the data" and reach the data at least twice before the analysis (Patton, 2018/2001, p. 441). MAXQDA software was preferred in this study since qualitative analysis software provides convenience compared to paper-pencil coding (Miles & Huberman, 2019/1994, p. 65; Saldaña, 2019/2015, p. 36). Then, data analysis principles proposed by Miles and Huberman (1994, 2019/1994) and Saldaña (2019/2015) were applied as a basic principle in data analysis. According to Miles and Huberman (1994), the data analysis method consists of three steps: (a) reducing the data obtained with data collection tools (extraction, summarizing, and transforming), (b) visualizing the data (matrix, graph, and tables), (c) conclusion and confirmation (citing blocks from field notes and benefiting from other research results in the literature). The two-stage coding process consists of "first level coding" (summing data pieces) and "pattern coding" (grouping summaries under themes) (Miles & Huberman, 2019/1994, p. 69).

To examine the first and the final draft versions of the participants' compositions, two different raters scored each composition independently of each other in terms of content, language use, organization, and punctuation. The researcher took the opinions of a third rater in cases where there were serious differences between the scores and then used the technique of converging the scores of the two raters by using the discussion method between the raters. A statistically significant difference between the first and the last draft composition scores of the participants was investigated with the *t*-test developed for dependent groups, and comments were made on the use of the students' reviewing strategy. Then, the themes obtained from the

interview and diary and the scores obtained from the composition examination were evaluated in a comparative way.

In this study, data diversification (Creswell, 2017/2016) using different data sources, participant confirmation by presenting the researcher's themes to the participants and asking whether the themes are an accurate indicator of the participants' feelings (findings were confirmed by one-on-one interviews with two participants), the accuracy of the research obtained from the external audits (an English teacher with a Ph.D. in educational sciences and an educational technologist with a Ph.D.), and finally intensive and rich descriptive methods were applied with a detailed description of the setting, people, and themes. It has been determined that all the activities performed during the nine-week writing assignment were applied in the same way for the participants. It is emphasized that the writing task should give the type of writing to be measured in the question and take into account the scoring criteria for the sub-components specified in the structure and that the raters should carry out the scoring process by considering the aforementioned criteria (Weigle, 2007). Raters preferred the consensus method (Trace et al., 2016) in which they had the opportunity to share common values, digest different perspectives, and make sense of the rubric and performance together so differences greater than one point were considered discordant. Two raters revised the scores of the compositions that were deemed incompatible, and they updated their scores until they reached an agreement. In the points where an agreement could not be reached, the discussion was made again by taking the third rater's score as a reference, and the scores were finalized. To conclude, the back-translation method (Brislin, 1970) was used for translating interview transcripts used in this study into English: "Two bilinguals are employed, one translating from the source to the target language, the second blindly translating back from the target to the source" (p. 186). The researchers then checked if the two versions were identical, then used the English translation of the participants' views in this study.

Findings

As a result of the qualitative data analysis, four themes were obtained: the use of metacognitive strategies, the use of cognitive strategies, the similarities between the two processes, and the differences between the two processes. Participants stated that their programming experience might lead to their choice of using learning strategies in foreign language learning. In addition, it was determined that the participants found similarities and differences between program development and foreign language writing processes. The terms in O'Malley and Chamot's (1990) classification of language learning strategies were used for the category (self-management, selective attention, self-evaluation, etc.) and the theme (metacognitive and cognitive strategies) names obtained from the codes reached in the data analysis. It is noteworthy that the number of categories and the frequency of expressing those categories for the use of metacognitive strategies ($n = 6$; $f = 495$) were considerably higher than the use of cognitive strategies ($n = 2$; $f = 35$). In the study, similarities in nine areas ($f = 167$) and differences in four areas ($f = 40$) between programming and foreign language writing processes were determined. In this article, only the use of metacognitive strategies will be explained in detail. Upon analyzing the transcripts of the interviews and short diaries, six categories were determined under the metacognition theme.

Self-Evaluation

Self-evaluation is the evaluation of language production (OMalley & Chamot, 1990; Oxford, 2011). Most participants expressed the opinion that having the habit of reviewing during the programming process and the necessity of using the programming language correctly

may have an impact on the self-evaluation strategy on the writing side (n = 35; f = 129) as one can see it in the following statement:

> *Since the results of my mistakes can put me in a dead end in programming, I started to skim through everything I wrote over time. As a result, I see that I give similar reactions while writing in a foreign language.* (Participant 1)

Planning

Planning is setting goals (understanding the purpose of the task) and organizational planning (planning how the task will be accomplished) (O'Malley & Chamot, 1990). When the views of the participants that algorithmic thinking can affect the systematic progress in the writing process are examined, it is seen that the planning category is expressed by two-thirds of the participants (n = 30; f = 93). One of the views is as follows:

> *Knowing the algorithm allows us to plan better. It's more efficient. Although planning is taught in English classes, I mean efficiency, knowing algorithms allows us to generate more ideas and produce better ideas. After learning the algorithm, I think that I find more practical, faster and more productive things while writing the outline.* (Participant 34)

Self-Management

Self-management is regulating the conditions necessary for the successful performance of the task and directing the use of language by focusing on what is known (O'Malley & Chamot, 1990). The codes for writing shorter, giving importance to avoiding redundancy and clarity, were combined under the category of self-management. Particularly, it was stated by two-thirds of the participants that ensuring the efficiency of the algorithm in the programming process may have an effect on writing shorter, giving importance to avoiding redundancy and clarity (n = 29; f = 88). Ensuring the efficiency of the algorithm can affect the emphasis on clarity (not giving place to unnecessary expressions) in foreign language writing:

> *Likewise, when writing in English or any other language, instead of going into unnecessary details, exceeding the word limit to make the article look full and writing in word salads, programming really affected me to write something shorter and more concise, result-oriented.* (Participant 42)

Selective Attention

Selective attention is focusing on the information that will enable the task to be performed (O'Malley & Chamot, 1990). More than half of the participants emphasized the selective attention strategy (n = 27; f = 91). It has been expressed that programming may have an effect on trying to write error-free in a foreign language, the importance of punctuation in programming may help to pay attention to punctuation in the process of foreign language writing, and the compiler's showing the error may have an effect on not being able to see the mistakes in the foreign language writing process:

> *Inevitably, you become someone who looks at the fine details while doing other things. So you have to scrutinize it a little bit. Because for someone who is used to that error-free writing situation, I think it is necessary to be very careful when writing in English, in the same way, etc. It affected me quite a bit about it.* (Participant 40)

Problem Identification

Problem identification is to find the place in the task that requires a solution or an aspect of the task that prevents it from being completed successfully (O'Malley & Chamot, 1990). Almost half of the participants expressed the opinion that having algorithmic thinking skills in the programming process may have an effect on analyzing the subject on the writing side (n = 19; f = 42):

> *We get used to it that way because they teach us the logic of it on the computer. For the following problems, these may be English or other department courses. Not focusing on a single solution, I go step by step and do research. There can be more than one solution to a problem.* (Participant 31)

Self-Monitoring

Self-monitoring is controlling the use of language or understanding it during the task, checking how the plan works, and the issues that were considered before during the task (O'Malley & Chamot, 1990). Nearly half of the participants expressed opinions that adhering to the roadmap and paying attention to error-free progress during the programming process may have an effect on foreign language writing by performing a self-monitoring strategy (n = 19; f = 52):

> *There is an effect from programming: we can say that it's a habit that comes from there, because we have to monitor exactly the same way in programming. I have to be mindful of that on the writing side as well. If I'm talking about a topic, I have to go back and write the rest of that topic clearly. I am monitoring so as not to deviate from the topic.* (Participant 23)

As for the quantitative data, all the participants who were contacted and the details of the scores and composition score means obtained after examining the composition of each of the 45 interviewees are presented in Table 3. When evaluated in terms of averages, there is no remarkable difference between the averages of content, language use, organization, and punctuation. Also, the participants are in all score ranges.

In order to reveal the accuracy of the use of the reviewing strategy, which is one of the categories under the self-evaluation theme, it was analyzed whether there was a statistically significant difference between the first draft and the final draft composition scores with the Kolmogorov-Smirnov Test. Composition scores showed a normal distribution (p = 0.142). According to the t-test result developed for the dependent groups, the final draft composition average scores ($\bar{X}$ = 72.11) differed significantly compared to the first draft composition average scores ($\bar{X}$ = 63.55) (t = -10,742, p = ,0001).

Table 3
Composition Score Mean

Participants	n	Criteria				
		Content (30 pts)	Language Use (30 pts)	Organization (25 pts)	Punctuation (15 pts)	Mean (100 pts)
Participants Contacted	76	21,97	18,58	20,03	12,76	73,35
Participants Interviewed	45	21,77	18,22	19,94	12,16	72,11

The minimum score the participants get from their compositions is 47.50, and the maximum score is 100. For the determination of the lower group, seven participants (P7, P20, P24, P31, P33, P37, and P44) with a mean score ($\bar{X}$ = 72.11) less than one standard deviation (*SD* = 13) below the score (59) and to determine the upper group, seven participants (P3, P10, P18, P26, P27, P29, and P30) whose mean score is one standard deviation higher than 85 were selected. It was investigated whether there was a difference in the qualitative data obtained from the participants in the lower (low composition success score) and upper (high composition success score) groups. The theme of the use of metacognitive strategies with the highest frequency of 495 out of 737 codings obtained as a result of the qualitative data analysis was reviewed in terms of the codes expressed by the participants in the upper group and the lower group and the frequency of expression of the codes. When the opinions of the participants in the upper and lower groups regarding the use of metacognitive strategy were examined in terms of composition success score, the frequency of the codes mentioned by the two groups did not reveal a pattern. When the views of the upper and lower groups on the use of metacognitive strategies were examined in terms of the programming success scores of the participants, the frequency of the codes mentioned by the two groups did not reveal a pattern, either. However, when the use of metacognitive strategies in the upper and lower groups, which were formed according to the composition and programming performance scores of the participants, was examined in more detail, some remarkable trends were identified (Table 4).

Table 4
Use of Metacognitive Strategies in Upper and Lower Groups According to Composition and Programming Performance Scores

Strategy	Composition		Programming	
	Upper Group (f)	Lower Group (f)	Upper Group (f)	Lower Group (f)
Self-management	12	15	12	10
Planning	16	9	20	14
Selective attention (positive effect)	16	16	20	10
Selective attention (negative effect)	6	-	6	-
Self-evaluation	16	16	14	18
Self-monitoring	16	4	6	4
Problem identification	4	10	4	10

- Participants in composition lower group mentioned less about the impact of programming on the use of planning strategy compared to other groups.
- Participants in composition and programming lower group did not express an opinion that compiler's showing error (selective attention) had a negative effect on the foreign language writing process.
- Participants in the upper group of composition talked more about the effect of programming on the use of self-monitoring strategy than the other groups.

Discussion

The category names discovered as a result of the analysis of the data obtained on the possible effects of the programming experiences of the participants on the foreign language writing process generally overlapped with the metacognitive strategies for foreign language learning in O'Malley and Chamot's (1990) classification. Metacognitive strategies, which Oxford (1990, 2011) said serve as the director of the play and, twenty-one years later, as the conductor, can be used to manage the use of cognitive strategies. Although metacognitive strategies have not yet been extensively studied in the field of programming (Campbell et al., 2016; Loksa & Ko, 2016; Prather et al., 2020), there are various classifications for strategies in foreign language learning (O'Malley & Chamot, 1990; Oxford, 1990, 2011). In this context, it might be important for participants with programming experience to talk about metacognitive strategies in their statements about the writing process in a foreign language.

Approaching from a cognitive point of view, it can be thought that the findings in this study support the literature: First, metacognitive strategies (planning, monitoring, and evaluation) are used in the solution process of the programming problem (Havenga, 2011). Second, sub-processes in the foreign language writing process are accepted as language learning strategies (Oxford, 2011). Last, programming and writing processes are similar (Bers, 2018; Détienne, 2002; Hassenfeld & Bers, 2020; Hassenfeld et al., 2020; Helm, 1988; Kaufman, 1988; Pennington & Grabowski, 1990).

Although three metacognitive strategies (planning, monitoring, and evaluation) that regulate cognition can be applied to different learning contexts and tasks once learned (Schuster et al. 2020), some of the codes obtained from the findings showed that it does not always mean that a positive transfer will take place in the programming and foreign language writing processes. In this study, when the codes under the metacognitive strategy categories are re-evaluated, it is thought that the programming experience has both positive and negative effects on the use of metacognitive strategies in the foreign language writing process.

Positive Effects of Programming Experience

Self-Evaluation

Participants expressed the opinion that having the habit of reviewing the programming process and the necessity of using the programming language correctly may have an impact on the self-evaluation strategy on the side of foreign language writing. They emphasized that they gained the habit of going back to the beginning and reviewing when the composition was finished and acknowledged that they learned this strategy in the English class, but gained consciousness with the effect of programming.

In addition, the participants explained that they received feedback automatically thanks to the compiler's display of the error and stated that they needed a similar application because of the effects of this experience in programming on the writing process in a foreign language. Likewise, Hermans and Aldewereld (2017) state that "testing and debugging steps" in programming and "revision and correction processes" in foreign language writing are similar.

Regarding the self-evaluation strategy, the final draft composition scores of the participants according to the t-test result developed for dependent groups differed significantly from the first draft composition scores (t = -10,742, p = .0001). As a result, participants with programming experience may have gained awareness of using the self-evaluation strategy in the foreign language writing process; however, whether the self-evaluation strategy in the programming process raises awareness and is transferred to the writing process needs to be proven.

Planning

Participants expressed the opinion that algorithmic thinking in the programming process could lead to systematic progress in the foreign language writing process. Regarding algorithmic thinking, the participants said that they proceeded step by step by dividing the problem into parts, brainstormed and created a draft during the programming process. About the effects of this experience in programming on the foreign language writing process, the participants stated that they were able to organize ideas and create drafts effectively, they could progress more easily and faster than their friends studying in other departments, and they did not give enough importance to planning before they started to learn computer programming.

When the opinions and composition scores of the participants were compared, 15 participants did not mention any code in the context of algorithmic thinking and planning. Those participants' mean score for organization of composition was 19.66, and their overall score mean was 70.6 out of 100. For the rest of the participants, the mean score for organization of composition was 20, the overall score average was 72.83 out of 100. The fact that the participants did not mention this strategy during the interview may indicate that two groups with extreme values did not occur when their average scores were taken into account.

The conclusion reached regarding the planning strategy is that the participants with programming experience may have gained awareness of using the planning strategy in the writing process. There is a high level of planning in both processes (Hermans & Aldewereld, 2017). Dale and Lewis (2016) state that the strategies of asking questions, finding similarities, and dividing a problem into smaller parts lead a person to plan, and the equivalent of this plan in programming is known as an algorithm. In this context, it is understood from the literature that there is a similarity between drafting in the writing process and creating an algorithm, and it can be assumed that the participants are aware of this similarity.

Self-Management

Under the title of self-management strategy, the participants expressed their opinions about writing shorter (explained in the negative effect section), avoiding redundancy, and maintaining clarity in the writing process. According to the participants, ensuring the efficiency of the algorithm on the programming side may help give importance to avoiding redundancy. Also, the effort of providing algorithm efficiency and writing a comprehensible program can affect the emphasis on clarity on the writing side.

Regarding the efficiency of the algorithm, the participants mentioned the importance of ensuring the readability of the language while keeping the lines of code lower for the program to run fast. Saying that one of the fundamental issues in computer science is performance, King et al. (2004) state that algorithm efficiency is important for many applications, from life-supporting applications to automatic cash machines, and among the foremost goals of software engineering are those that exceed or meet customer expectations. Regarding the effects of this programming experience on foreign language writing, the participants stated that they avoided repetitions, tended not to go into detail, tried not to include unnecessary expressions, and acted consciously so that the program and the composition would be understandable by the readers.

Commands given in programming languages should be clear. Therefore, the absence of unnecessary expressions in the program provides fewer semantic dilemmas in defining the problem; however, there is a decrease in the narrative power of the program (Liu & Wu, 2018). The researcher investigated the issues of avoiding unnecessary expressions and repetitions in the participants' opinions during the composition analysis, and no contradiction was detected. The conclusion reached regarding the self-management strategy is that the participants with programming experience were aware of attaching importance to avoiding redundancy and maintaining clarity in the writing process. Since an author-oriented method is used in the process model approach, a detailed explanation of the features of the text has not been found in the literature. Therefore, it would be better to make a more comprehensive comparison by including other approaches (structural, functional, expressive, content, and genre) based on foreign language writing (Hyland, 2019; Manchón et al., 2007).

Selective Attention

The participants mentioned the selective attention metacognitive strategy by expressing the opinion that programming does not accept mistakes in the programming process and that the importance of punctuation may have an effect on trying to write without mistakes and paying attention to punctuation in foreign language writing. There are statements in the literature that adherence to syntax is an absolute necessity in the programming process, that the code does not work in the slightest error, and that the time allocated for correction causes a waste of time when an error occurs later on (Percival, 2017; Schneider & Gersting, 2018). Regarding the effects of this experience in programming on foreign language writing, the participants expressed their opinions about trying to write error-free, raising awareness about finding and correcting mistakes while checking the text, trying to write in a format suitable for not only grammatical compatibility but also the genre of writing. The conclusion reached regarding the selective attention strategy is that the participants with programming experience may be inclined to use the selective attention strategy consciously at the point of obeying the rules in the foreign language writing process.

Problem Identification

Participants expressed the opinion that having algorithmic thinking skills in the programming process may have an effect on analyzing the subject on the writing side. The code for analyzing the subject is included under the problem identification category, which is a metacognitive learning strategy. Regarding algorithmic thinking skills, the participants said that defining the problem is the most important step in the program development process and that they moved on to algorithm design after dealing with all aspects of the problem and understanding it deeply. In the classifications of programming skills, planning knowledge is explicitly included in Robins et al.'s (2003) classification, while it is implicitly stated in other classifications (Bayman & Mayer, 1988; Du Boulay, 1986; Linn & Dalbey, 1985). The conclusion reached regarding the problem identification strategy is that participants with programming experience may tend to make a detailed analysis to understand the subject during the writing process.

Self-Monitoring

Participants expressed their opinions that adhering to the roadmap in the programming process and paying attention to error-free progress can have an effect on foreign language writing by performing self-monitoring. The participants said that they had the discipline of progress by adhering to the flow chart in the programming process. They also claimed they

checked the coding at regular intervals so the process would be completed with fewer mistakes. As stated by Schneider and Gersting (2018), it is recommended that the testing process be carried out at the end of each module, when the modules are connected to each other, and when any changes are made. Otherwise, it may become difficult to deal with the errors detected as a result of debugging at the end of the process. Regarding the effects of this experience in programming on foreign language writing, the participants stated that they consciously did the writing process in accordance with the plan, going back and checking it in terms of grammar and meaning, as they gained a habit in programming. The conclusion reached regarding the self-monitoring strategy is that the participants may continue their self-management habits during the writing process.

Negative Effects of Programming Experience

Self-Management – Writing Shorter

It was stated by 13 participants that ensuring algorithm efficiency in the programming process is similar to the tendency to write shorter in a foreign language: "In fact, I prefer to write clearly and descriptively, and the number of words in my compositions will be less than that of friends in other sections" (P13). Those participants' composition scores were analyzed, and the mean content score in this group (19.8/25) was lower than the mean content score of the other participants (22.57/25) who were not in this group. However, it is thought that the effect of programming on the text length produced in foreign language writing can be better revealed after comparing the foreign language writing processes of the students studying in other departments and the participants studying in computer engineering.

Selective Attention – Not Seeing the Mistakes

Paying attention to punctuation in foreign language writing due to the importance of punctuation in programming, trying to write without mistakes in a foreign language because programming does not accept mistakes, and applying self-management strategy in foreign language writing because of adhering to the roadmap in programming was examined whether there was a negative effect of programming on writing: it was determined that the participants gave positive opinions. However, five participants stated that they had problems in seeing mistakes in a foreign language as one of them said:

> *If we think of 'compile error' grammatically, this somewhat blunts finding errors; because it is something that pushes people to be lazy. I rarely look for grammatical errors in the composition; I don't have to find mistakes all the time.* (P16)

The mean grammar score of the participants in this group (22.5/25) is higher than that of the other participants (17.68/25) who are not in this group. Since the compiler shows the error, more than half of the participants reported that they felt the need for an automatic correction tool, thinking that instant feedback would be more useful on the side of writing in a foreign language in another code pair. In order to understand the real reason behind this need, a think-aloud protocol can provide evidence because both the positive opinion in other codes in the selective attention category (trying to write without mistakes and paying attention to punctuation) and the statements of laziness against mistakes seem to contradict each other.

Conclusion and Implications

The conclusion reached in this study is that the experiences of the participants in the programming process may have guided the strategies they preferred in the foreign language writing process, and when it is assumed and accepted that the aforementioned influence is the first step of strategy teaching (Rubin et al., 2007), the participants may have accomplished their writing task more consciously and effectively. When the statements of the participants were examined, it was understood that the programming experience did not have a hindering effect on metacognitive writing strategies, on the contrary, it played a supportive and reinforcing role.

The opinions of the computer engineering sophomores on the possible effects of learning programming on foreign language writing revealed that they thought there were similarities and differences between programming and foreign language writing. It is well known that there are various classifications on the use of language learning strategies but studies about the relationships between programming and metacognitive strategies have started in recent years and no theory has yet been formed. Computer engineering is a new discipline compared with foreign language teaching, but because there have been studies on programming education since the 1980s, there is a lack of a comprehensive framework on the relationship between programming and metacognition (Pea & Kurland, 1984; Scherer, 2016).

The findings in this study were obtained from students studying at a foundation university in Ankara within the scope of the case study and do not include any generalizations. The fact that participants found two processes similar to each other and the same result is already in the literature showed that the awareness of the students was real. It is thought that participant views on the effects of programming need to be expanded with multiple case studies, and the metacognitive strategies that participants express in different age groups had better be researched with experimental studies as Mayer (2001) did for each principle in multimedia learning design.

Nunan (1997) stated that a student must go through five stages in order to become autonomous in the learning process: awareness, involvement, intervention, creation and transcendence. The process, which starts with students being aware of the educational goals and content offered to them, ends with determining strategies, making choices, and finally reaching the level where they can research and produce by making a connection between what is learned in the classroom and the world outside the classroom. If it is achieved to investigate whether awareness levels up and there is a statistical relationship between learning programming and using metacognitive strategies and develop a framework for programming learning strategies, then it will be possible to believe a higher quality education process will be experienced by the stakeholders of the educational environment by reflecting the obtained results to the instructional designs for programming education at the primary and secondary education level.

As a result, new needs arise depending on the developments in technology and educational psychology, and research trends can change direction in order to meet these needs. The general conclusion reached by the researchers in this study is that there is need for a well-established bridge among theory, design and practice in order to solve real problems in the journey of improving learning. Also, the use of technology (programming) to increase the quality of education with a human-based approach can bring a true transformation.

Funding Details

This research and study were not funded by any agency.

Disclosure Statement

This research contains no financial interest or benefit from the direct applications of the research.

References

Adams Becker, S., Cummins, M., Davis, A., Freeman, A., Hall Giesinger, C., & Ananthanarayanan, V. (2017). *NMC horizon report: 2017 Higher education edition*. The New Media Consortium.

Akcay, N., Avcı, H., Gungor, A., & Adiguzel, T. (2018). The relationship between computer programming and English language skills. *Proceedings of the Constructionism 2018 Conference* (pp. 782–785). Vilnius, Lithuania.

Anderson, R. E. (2008). Implications of the information and knowledge society for education. In J. Voogt, & G. Knezek (Eds.), *International handbook of information technology in primary and secondary education* (pp. 5–22). Springer Science.

Atkinson, D. (2018). Theory in second language writing. In *The TESOL Encyclopedia of English language teaching* (pp. 1–6). Wiley.

Barnett, S. M., & Ceci, S. J. (2002). When and where do we apply what we learn? A taxonomy for far transfer. *Psychological Bulletin*, *128*(4), 612–637. https://psycnet.apa.org/doi/10.1037/0033-2909.128.4.612

Bayman, P., & Mayer, R. E. (1988). Using conceptual models to teach BASIC computer programming. *Journal of Educational Psychology*, *80*(3), 291–298. https://doi.org/10.1037/0022-0663.80.3.291

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, *6*(4), 499–528.

Bers, M.U. (2018). *Coding as a playground*. Routledge.

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, M. (2012). Defining twenty-first century skills. In P. Griffin, B. McGaw, & E. Care (Eds.), *Assessment and teaching of 21st century skills* (pp. 17–66). Springer, Dordrecht.

Brislin, R. W. (1970). Back-translation for cross-cultural research. *Journal of Cross-Cultural Psychology*, *1*(3), 185–216.

Buyukozturk, S., Cakmak, E.K., Akgun, O. E., Karadeniz, S., & Demirel, F. (2016). *Bilimsel arastırma yontemleri* [Scientific research methods]. Pegem Akademi.

Campbell, J., Horton, D., & Craig, M. (2016). Factors for success in online CS1. *Proceedings of the 2016 ACM conference on innovation and technology in computer science education* (pp. 320–325). https://doi.org/10.1145/2899415.2899457

Carr, V., Jones, M., & Wei, B. (2020). Interdisciplinary computing: Applied computing for behavioral and social sciences. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)* (pp. 400–406). Association for Computing Machinery, New York, USA.

Cohen, A. D. (1998). *Strategies in learning and using a second language*. Longman.

Creswell, J. W. (1998). *Qualitative inquiry and research design: Choosing among five traditions.* SAGE Publications.

Creswell, J. W. (2013). *Qualitative inquiry and research design: Choosing among five approaches* (3rd ed.). SAGE Publications.

Creswell, J. W. (2014). *Research design: Qualitative, quantitative, and mixed methods approaches*. SAGE Publications.

Creswell, J.W. (2017). *Nitel araştırmacılar için 30 temel beceri.* [30 essential skills for the qualitative researcher]. (H. Özcan, Trans.). Anı Yayıncılık (Original work published 2016)

Dale, N. B., & Lewis, J. (2016). *Computer science illuminated*. Jones & Bartlett Learning.

Dede, C. (2010). Comparing frameworks for 21st century skills. In J. Bellanca, & R. Brandt (Eds.), *21st century skills: Rethinking how students learn* (pp. 51–76). Solution Tree Press.

Détienne, F. (2002). *Software design: Cognitive aspects*. Springer Science & Business Media.

Donker, A. S., De Boer, H., Kostons, D., Van Ewijk, C. D., & van der Werf, M. P. (2014). Effectiveness of learning strategy instruction on academic performance: A meta-analysis. *Educational Research Review*, *11*, 1–26.

Dooley, J. (2011). *Software development and professional practice*. Apress.

Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, *2*(1), 57–73. https://doi.org/10.2190/3LFX-9RRF-67T8-UVK9

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn to code? *Proceedings of the 9th workshop in primary and secondary computing education* (pp. 60–69). Berlin, Germany: ACM.

Fedorenko, E., Ivanova, A., Dhamala, R., & Bers, M. U. (2019). The language of programming: A cognitive perspective. *Trends in Cognitive Sciences*, *23*(7), 525–528. https://doi.org/10.1016/j.tics.2019.04.010

Ferris, D. R., & Hedgcock, J. (2013). *Teaching L2 composition: Purpose, process, and practice*. Routledge.

Flower, L., & Hayes, J. R. (1981). A cognitive process theory of writing. *College Composition and Communication*, *32*(4), 365–387. https://doi.org/10.2307/356600

Frankel, R., & Devers, K. (2000). Study design in qualitative research-1: Developing questions and assessing resource needs. *Education for Health*, *13*(2), 251–261. https://doi.org/10.1080/13576280050074534

Goldfine, R. A. (2004). *Computer science majors' perception of the overlapping cognitive structures between computer programming and English composition* [Unpublished doctoral dissertation]. Georgia State University, College of Arts and Sciences, Atlanta.

Hassenfeld, Z. R., & Bers, M. U. (2020). Debugging the writing process: Lessons from a comparison of students' coding and writing practices. *The Reading Teacher*, *73*(6), 735–746. https://doi.org/10.1002/trtr.1885

Hassenfeld, Z. R., Govind, M., De Ruiter, L. E., & Bers, M. U. (2020). If you can program you can write: Learning introductory programming across literacy levels. *Journal of Information Technology Education*, *19*, 65–85. https://doi.org/10.28945/4509

Havenga, H. M. (2011). Problem-solving processes in computer programming: A case study. *Southern African Computer Lecturers' Association (SACLA) Conference Proceedings* (pp. 91–99). http://www.sacla.org.za/wp-content/uploads/2020/03/Proceedings-SACLA-2012.pdf

Helm, L. B. (1988). From programming to technical writing. *IEEE Transactions on Professional Communication*, *31*(4), 175–176. https://doi.org/10.1109/47.9220

Hermans, F., & Aldewereld, M. (2017). Programming is writing is programming. *Companion to the first International Conference on the Art, Science and Engineering of Programming (Programming '17), ACM,* (pp. 1–8). https://doi.org/10.1145/3079368.3079413

Howland, K., & Good., J. (2015). Learning to communicate computationally with Flip: A bi-modal programming language for game creation. *Computers & Education*, *80*, 224–240. https://doi.org/10.1016/j.compedu.2014.08.014

Hsiao, T. Y., & Oxford, R. L. (2002). Comparing theories of language learning strategies: A confirmatory factor analysis. *The Modern Language Journal*, *86*(3), 368–383. https://doi.org/10.1111/1540-4781.00155

Hyland, K. (2003). *Second language writing*. Cambridge University Press.

Hyland, K. (2016). Methods and methodologies in second language writing research. *System, 59*, 116–125. https://doi.org/10.1016/j.system.2016.05.002

Hyland, K. (2019). *Second language writing* (2nd ed.). Cambridge University Press.

ISTE Standards. (2008). *ISTE standards for students*. https://www.iste.org/standards/for-students

Kaufman, J. (1988). Technical writing and computer programming. *IEEE Transactions on Professional Communication*, *31*(4), 171–174. https://doi.org/10.1109/47.9219

Kay D. G. (1998). Computer scientists can teach writing: An upper division course for computer science majors. *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education (SIGCSE '98), ACM*, (pp. 117–120). https://doi.org/10.1145/274790.273174

King, R. E., Roggio, R. F., & Killinger, J. G. (2004). Programming skills for a changing world: Back to the basics. *J. Comput. Sci. Coll., 19*(3), 249–257.

Korkmaz, Ö., & Erdoğmuş, F. U. (2019). Programlama öğretiminde yönelimler [Trends in teching programming]. In Ö. Korkmaz (Ed.), *Programlama öğretimi yaklaşımları* (pp. 215–231). Nobel Akademik Yayıncılık.

Levy, F., & Murnane, R. J. (2004). *The new division of labor: How computers change work and pay*. Princeton University Press.

Linn, M. C., & Dalbey, J. (1985). Cognitive consequences of programming instruction: Instruction, access, and ability. *Educational Psychologist*, *20*(4), 191–206. https://doi.org/10.1207/s15326985ep2004_4

Liu, X., & Wu, D. (2018). From natural language to programming language. In S. Goschnick (Ed.), *Innovative methods, user-friendly tools, coding, and design approaches in people-oriented programming* (pp. 110–130). IGI Global.

Loksa, D., & Ko, A. J. (2016). The role of self-regulation in programming problem solving process and success. *Proceedings of the 2016 ACM conference on international computing education research* (pp. 83–91). https://faculty.washington.edu/ajko/papers/Loksa2016SelfRegulation.pdf

Manchón, R. M., Roca de Larios, J., & Murphy, L. (2007). A review of writing strategies: Focus on conceptualizations and impact of first language. In A. D. Cohen, & E. Macaro (Eds.), *Language learner strategies: Thirty years of research and practice* (pp. 229–250). Oxford University Press.

Mayer, R. E. (2001). *Multimedia learning*. Cambridge University Press.

Mayer, R. E. (2015). On the need for research evidence to guide the design of computer games for learning. *Educational Psychologist*, *50*(4), 349–353. https://doi.org/10.1080/00461520.2015.1133307

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. SAGE Publications.

Miles, M. B., & Huberman, A. M. (2019). *Nitel veri analizi* [Qualitative data analysis]. (S. A. Altun, & A. Ersoy, Trans.). Pegem Akademi. (Original work published 1994)

Nunan, D. (1997). Designing and adapting materials to encourage learner autonomy. In P. Benson & P. Voller (Eds.), *Autonomy and independence in language learning* (pp. 192–203). Longman.

O'Malley, J. M., & Chamot, A. U. (1990). *Learning strategies in second language acquisition*. Cambridge University Press.

Oxford, R. L. (1990). *Language Learning Strategies: What Every Teacher Should Know*. Heinle & Heinle Publishers.

Oxford, R. L. (2011). *Teaching and researching language learning strategies*. Pearson Longman.

Partnership for 21st Century Skills. (2013). *Framework for 21st century learning*. http://www.p21.org/about-us/p21-framework

Patton, M. Q. (2018). *Nitel araştırma ve değerlendirme yöntemleri* [Qualitative research and evaluation methods]. (M. Bütün, & S. B. Demir, Trans.). Pegem Akademi. (Original work published 2001)

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology*, *2*(2), 137–168. https://doi.org/10.1016/0732-118X(84)90018-7

Pennington, N., & Grabowski, B. (1990). The tasks of programming. In J. M. Hoc, T. R. G. Green, R. Samurçay, & D. J. Gilmore (Eds.), *Psychology of programming* (pp. 45–62). Academic Press.

Percival, R. (2017). *Confident coding: Master the fundamentals of code and supercharge your career*.: Kogan Page.

Perkins, D. N., & Salomon, G. (1992). Transfer of learning. In T. Husén & T. N. Postlethwaite (Eds.), *The international encyclopedia of education* (2nd ed., pp. 425–441). Pergamon. Polio, C. (2017). *Teaching second language writing*. Routledge.

Prather, J., Becker, B. A., Craig, M., Denny, P., Loksa, D., & Margulieux, L. (2020). What do we think we think we are doing? Metacognition and self-regulation in programming. *Proceedings of the 2020 ACM Conference on International Computing Education Research* (pp. 2–13). https://doi.org/10.1145/3372782.3406263

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). *Scratch: Programming for all. Communications of the ACM, 52*(11), 60–67.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), 137–172. https://doi.org/10.1076/csed.13.2.137.14200

Rubin, J., Chamot, A. U., Harris, V., & Anderson, N. J. (2007). Intervening in the use of strategies. In A. D. Cohen, & E. Macaro (Eds.), *Language learner strategies: 30 years of research and practice* (pp. 141–160). Oxford University Press.

Rushkoff, D. (2010). *Program or be programmed*: *Ten commands for a digital age*. Or Books.

Saldaña, J. (2019). *Nitel araştırmacılar için kodlama el kitabı* [The coding manual for qualitative researchers]. (A. Tüfekci Akcan, & S. N. Şad, Trans.). Pegem Akademi. (Original work published 2015)

Scherer, R. (2016). Learning from the past: The need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in Psychology*, *7*, Article 1390. https://doi.org/10.3389/fpsyg.2016.01390

Schneider, G. M., & Gersting, J. (2018). *Invitation to computer science*. Cengage Learning.

Schraw, G. (2001). Promoting general metacognitive awareness. In H. J. Hartman (Ed.), Metacognition in learning and instruction: Neuropsychology and cognition (Vol. 19, pp. 3–16). Springer, Dordrecht. https://doi.org/10.1007/978-94-017-2243-8_1

Schuster, C., Stebner, F., Leutner, D., & Wirth, J. (2020). Transfer of metacognitive skills in self-regulated learning: An experimental training study. *Metacognition and Learning*, *15*(3), 455–477. https://doi.org/10.1007/s11409-020-09237-5

Silva, E. (2009). Measuring skills for 21st-century learning. *The Phi Delta Kappan*, *90*(9), 630–634. https://doi.org/10.1177/003172170909000905

Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*, *67*(3), 541–575. https://doi.org/10.1007/s11423-018-9622-x

Taylor, H., & Paine, K. P. (1993). An inter-disciplinary approach to the development of writing skills in computer science students. *Proc. of the Twenty-Fourth ACM Special Interest Group on Computer Science Education (SIGCSE) Technical Symposium on Computer Science Education* (pp. 274–278). Indianapolis, Indiana. NY: ACM Press.

Trace, J., Meier, V., & Janssen, G. (2016). "I can see that": Developing shared rubric category interpretations through score negotiation. *Assessing Writing*, *30*, 32–43. https://doi.org/10.1016/j.asw.2016.08.001

Van Damme, D. (2017). *Does the world need people who understand? Does the world need people who understand problems, or who can solve them?* https://medium.com/@OECD/does-the-world-need-people-who-understand-problems-or-who-can-solve-them-b34b7e75af76

Weigle, S. C. (2007). Teaching writing teachers about assessment. *Journal of Second Language Writing*, *16*(3), 194–209. https://doi.org/10.1016/j.jslw.2007.07.004

Yildirim, A., & Simsek, H. (2008). *Sosyal bilimlerde nitel araştırma yöntemleri* [Qualitative research methods in the social sciences]. Seckin Yayincilik.

Yin, R. K. (2009). How to do better case studies. In *The SAGE handbook of applied social research methods* (2 ed., pp. 254–282). SAGE Publications. https://doi.org/10.4135/9781483348858

Notes on Contributor

Hakan Bayezit completed his B.A in English Language Teaching at Hacettepe University and his M.A. in Teaching Turkish as a Foreign Language at 9 Eylul University. His doctoral degree is in Educational Technology from Bahcesehir University. He has 24 years of experience in foreign language teaching.

Tufan Adiguzel is a Professor in the Sectoral Education department and the founding director of the Office for Learning and Teaching Enhancement at Ozyegin University. He holds a Ph.D. in Educational Psychology (specialization in Educational Technology) from Texas A&M University, USA. He has many national and international projects, articles, papers, and book studies as well as been teaching a variety of undergraduate and graduate level courses on technology design, development, and acceptance in the fields of Computer Education, Science, Mathematics and Special Education.

ORCID

Hakan Bayezit, https://orcid.org/0000-0001-6220-9190
Tufan Adiguzel, https://orcid.org/0000-0001-6232-1246