

Learning programming online: Influences of various types of feedback on programming performances

Ünal Çakıroğlu¹

Samet Atabas²

Doğukan Sarıyalçınkaya³

İbrahim Enes Öner¹

¹Trabzon University, Faculty of Education

²Avrasya University, Vocational School

³Ondokuz Mayıs University, Vezirköprü Vocational School

DOI: [10.21585/ijcses.v3i3.57](https://doi.org/10.21585/ijcses.v3i3.57)

Abstract

This article draws on a one-semester online programming course to determine the relationships between the types of feedback that students received and their programming performances. An explorative study was designed including both quantitative and qualitative data. Participants were 15 second-year students enrolled at an Information Technology department of a public university who received weekly programming problems as projects. The instructor provided various types of feedback (corrective, confirmatory, explanatory, diagnostic and expanded feedback) to the students via GitHub. The results showed that; when the students who received corrective and explanatory feedback more, the programming performance scores were high. A strong positive relationship existed between a total amount of feedback with programming performance scores, also a number of times that students received explanatory feedback and programming performance scores. In addition, while the positive correlations existed between programming performance scores and the amount of all types of feedback, there was a weak correlation between the number of times that students received expanded feedback and programming performance scores. The difficulty of the programming projects, the needs of the participants and the affordances of the online learning environment played key role in the students' preference for receiving feedback type. Recommendations for future research and practice are included.

Keywords: types of feedback, online programming course, programming performances

1. Introduction

In recent years, the increasing demand for programming has led many institutions to utilize online settings for delivering programming courses. Despite the advance in the tools for programming instruction, teaching programming remains a big challenge for instructors and programming courses are still considered complex and often resulting in low retention (Barr & Guzdial, 2015; Sáez-López, Román-González, & Vázquez-Cano, 2016). Especially, in online programming courses, repetitive failures during lessons may lead students to lose enthusiasm and interest (Law, Lee, & Yu, 2010). In this sense, students often need feedback from the instructors to perform programming tasks (Eng, Ibrahim, & Shamsuddin, 2015). Through ongoing monitoring of online learning, instructors can observe students' progress and guide them via feedback. Effective feedback can support students to progressively identify their strengths and weaknesses and refine their understanding (Gikandi, Morrow, & Davis, 2011).

Researchers in online learning argues that online learning has distinct pedagogical demands owing to the asynchronous nature of interactivity between the teacher and learners (Naidu, 2007). Thus, although online feedback have the potentials to enhance online learning outcomes, the process of using feedback online may be somewhat different from the face-to-face settings. The instructors giving and the students' receiving process

may be affected by various factors of the online learning components. This can also affect the preferences of various types of feedback. Thus, a question may come into mind: "Does use of different types of feedback affect the learning outcomes differently?" Although feedback in teaching online programming have a positive effect on outcomes, the relationship between the feedback types used and the programming performances during the instructional process need to be investigated further.

1.1. Feedback in Online Programming

In online learning, the responsibility of learning is shifted from the instructor to the student (Xia & Liitiäinen, 2017). Researchers argued that traditional pedagogical practices that do not fit online classrooms (Baran, Correia, & Thompson, 2011). In this line, ongoing support for scaffolding is suggested in online learning, and the learning process can be facilitated through well-designed feedback (Ludwig-Hardman & Dunlap, 2003). According to Hattie and Timperley (2007), the feedback has been shown to hold great potential for student learning in higher education. The instructor may increase student motivation to learn by facilitating self-directed studies with well-organized instructions and timely feedback (Kop, 2011; Xia, 2015). Feedback enhances learning by informing the user about learners' performance, correcting the user, explaining the correct answers, evaluating, motivating the user, rewarding the user or attracting attention (Lawless & Pellegrino, 2007).

Creating pedagogically sound feedback is a complex task. Feedback content, learning tasks, characteristics of the students, frequency of the feedback and timing may affect to contributions of the feedback to the learning outcomes (Kluger & DeNisi, 1996; Shute, 2008). In this regard, divergent views exist about what to include in feedback messages, and even in the amount of information to reach the pedagogical objectives (Campos, et al., 2012). Instructors should have prior knowledge about students' readiness, the sources of the mistakes to provide the appropriate type of feedback to meet the students' needs.

1.2. Types of Feedback

In online learning giving and receiving online feedback is somewhat different than the traditional feedback, because learners have more time to deal with feedback in the tasks (Pyke & Sherlock, 2010). Considering the online learning settings, Vasilyeva et al. (2007) classified the functions of online feedback as no feedback, simple verification feedback, correct response feedback, elaborated feedback and try again feedback.

Schimmel (1983) classified types of feedback as explanatory, corrective, diagnostic, confirmatory and expanded.

Explanatory feedback gives information to the students about their learning outcomes including why the answer is wrong or correct.

Corrective feedback provides the correct answer which is given along with the confirmatory feedback through the student's learning outcomes.

Diagnostic feedback includes information about what the student should work to correct the wrong answer and how it should work.

Confirmatory feedback provides information about the results of learning whether it is true or false.

Expanded feedback enables the student to expand students' existing knowledge by helping them to build relationships between prior knowledge and new knowledge.

Research conducted about online feedback includes different perspectives. For instance, Gikandia and Marrow (2016) used Moodle and found that peer formative feedback was useful in understanding how the students learned actively and contributed to peer review. Cakiroglu et al. (2016) studied the effects of instant feedback in an online programming course and observed that the online course cultivated students' programming skills. In another study Horne et al. (2018) indicated that it is beneficial for students to receive feedback on a regular basis, but that they do not need to use excessive feedback to get a positive impact. While prior work provides compelling evidence that feedback is one of the keys to successful programming, it leaves several open questions about the effect of types of feedback. Following conclusions from the previous studies, we aim at gaining an insight into the nature of feedback types provided in online programming instruction.

In an online learning environment, various tools can be used for giving feedback. LMSs are common settings used to deliver online settings. Also, some specialized web-based tools are used in online learning. For instance, GitHub is originally developed as a tracking software development and collaboration during the software development process. GitHub provides educational tools such as GitHub Classroom and GitHub Education. Thanks to these tools GitHub fulfills some requirements of LMS, such as asynchronous communication, immediate feedback, content delivering and evaluation. The Fork provides to copy the repositories belonging to someone else. Anyone in the GitHub group can develop projects and receive visual feedback with graphs of all commits and fork made.

1.3. Need for Study

Like all learning models, online learning has some inherent problems, especially in the areas of isolation, support, technology, discipline and feedback (Mishra & Koehler, 2006). It is valuable for instructors to acknowledge the importance of students' needs and expectations in formulating their feedback (Hyland, 2010; Jara & Mellar, 2010). Although previous research has investigated widespread use of feedback in various courses, the conditions under which types of feedback foster learning online programming remain unclear. Thus, the findings of this study will be useful to guide for instructors to provide appropriate feedback types for the context.

1.4. Purpose of the Study

Given the importance of feedback for learning programming and the increasing trend of online programming courses, this study seeks to examine which types of feedback enhance students' success in the given tasks in the online training environment. Particularly, Github as an asynchronous programming teaching platform is used.

In line with the overall purpose of the study, the following research questions were directed:

- How did different types of feedback effect on programming performance scores?
- How did students explain the contribution of various feedback types to the online programming learning process?

2. Method

In order to address the research questions, an explorative study was designed. One semester online course was carried out through synchronous online techniques. Both quantitative and qualitative data were used gathered to understand the relationships between the types of feedback and programming performances. The instructor gave one project per week for 9 weeks through GitHub, and students worked projects on Github by receiving feedback within this platform.

GitHub was developed to track the development of projects, share them with other developers and to provide a common development environment. The affordances of Github in the collaboration and feedback facilities directed programmers to use it for programming instruction (Gunnarsson, et al., 2017). GitHub fulfills a lot of features of an LMS, such as asynchronous communication, immediate feedback, transmitting content and evaluate students. For instance, Fork provides to copy the repositories belonging to someone else. Visual feedback provides the users with graphs of all commits and fork made. So that anyone to follow the project during its development.

2.1. Participants

In this study, participants were 15 (12 male, 3 female) students (between 20-29 age) enrolled in a programming department of a vocational high school in Turkey. The study was carried out in Javascript class in the 4th semester of the academic year 2017–2018.

2.2. Process

The course was delivered via both synchronous online sessions via Adobe Connect 3 hours per week. In these sessions, the instructor generally delivered presentations and sample programming codes and provided discussions about the codes. Also, the instructor asked students to work on 9 programming projects during a semester of 14 weeks. Projects covering conceptual and strategic knowledge of programming were presented starting from easy to difficult projects. Projects include tasks about the programming knowledge that is provided by the instructor during the week. Students were required to complete the online tasks in their own learning time at home. The instructor gives the projects on GitHub and students sent the answers including design of the form, the question, the answer screenshots about the answer and detailed explanations are provided by the GitHub Repo. The projects are briefly presented in Table1.

Table 1. Weekly programming projects

Projects	Programming Concepts	Project topics
P1	Basic variable and transaction	Priority of operations
P2	Basic loop and comparison	Divisibility question
P3	Inner loop	Multiplication table of numbers
P4	Conditions (if-else)	Odd or even numbers
P5	Usage of function	Environment, area, volume
P6	Logical testing and loop	Calculation of multiplication
P7	Arrays	Sorting the numbers
P8	Operators	Rules about triangles
P9	Number guessing game	Random numbers

The instructor examined the programming codes and provided feedback to the students' programming codes on the GitHub Commit tool. Using the repository of GitHub students were allowed to submit their programming codes for the projects. Students could follow the feedback for all projects. In the feedback, the instructor provided conceptual and logical information about programming. Also, the structure of Javascript and the web page design was also included in the feedback. The instructor was able to follow students' progression in the tasks via GitHub. Using their own accounts in the Fork tool, students provided their codes and their responses for the feedback including how they benefited from feedback. At the end of the deadline for each project, students submitted the final version of programming codes and shared them with other students via GitHub.

The answers for the programming projects were shared by the instructor at the end of the project deadline. A view from the project including the student response and the instructor feedback is presented in Figure 1.

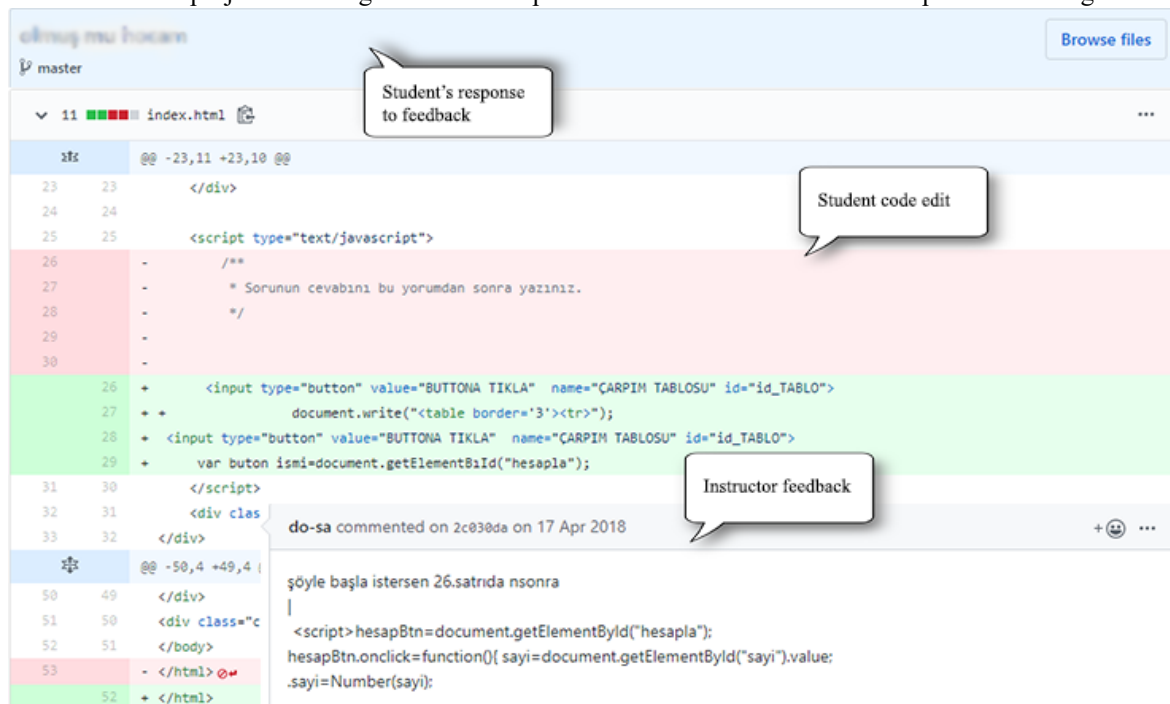


Figure 1. A view from the journey of a feedback

2.3. Data Collection Tools and Analysis

Both quantitative and qualitative data were used to acquire a better understanding of the influences of feedback on students' programming performances. Data Collection tools include Github logs were analyzed as quantitative data collection tools, also rubric and question form were used to help explain the quantitative results from Github data.

GitHub logs: GitHub logs included quantitative data (amount and time) about students' answers for the project tasks and instructor feedback. The feedback is given and the students' responses for the feedback (their revisions regarding the feedback or explanations about the feedback) were provided through GitHub. The logs were analyzed considering the amount of types of feedback. Two researchers determined the types of feedback on the projects.

Rubrics: An evaluation rubric was created for examining students' programming codes in order to determine their performances. Rubrics include an evaluation protocol for three types of programming knowledge (syntactic knowledge, conceptual and strategic knowledge) (McGill, & Volet, 1997) in their tasks. Two experts in online learning and programming instruction reviewed the rubric items and the scales for content validity. Three sections (syntax, conceptual, strategic) were grouped in three levels of evaluation as regarding the total scores of 0-15. The T- scores of the total scores from the rubric was taken as a performance score. Two researchers first assigned the scores for the tasks on the rubric individually, and then they discussed the students' codes each other until they come to reach an agreement on the score of students for the task. The well-defined score categories assisted in maintaining consistent scoring when rating. Spearman Correlation tests were used to determine the relationships between the amount of each type of feedback used during the instructional process.

According to Schimmel (1988), feedback differs according to the amount of information its solution to the wrong answer (except for the last step). In this study, feedback given by teachers in the learning-teaching process was handled according to Schimmel's classification of feedback types that provide information to students and based on the amount of this information. Answers such as true, false, yes, no are confirmative feedback. The corrective feedback provides the correct response to the student. For example, "Wrong answer, Turkey's capital is Ankara." There are various forms of use of descriptive feedback. It can be presented in stages by showing the content to the student before giving the wrong answer. The diagnostic feedback aims to correct the student's erroneous mental modeling process. Therefore, the answer has more function than just correcting it. The expanded feedback was considered feedback directing the student to find the result own.

Open-Ended Question Form: The form includes open-ended questions about feedback, difficulties in the process and effects of the feedback on the learning progress, and closed-ended questions for demographics. The form was given to the participants as a result of the training and the form was completed themselves. The form included such as "What were the challenges you faced during the process?", "How did the comments written to you on GitHub affect your learning of programming?". "What kind of feedback did you benefited most? Why did you think that you benefited more?"

3. Results

The programming performances were evaluated by scoring the students' responses to the tasks in the weekly projects through rubrics. Students' perspectives in the question form were used to explain the contributions of the feedback to the programming performance.

3.1. Relationships between Feedback Students Received and Programming Performances

In order to determine the relationships between the feedback types and the programming performances, first, we defined the relationships descriptively on the basis of all students and all projects. Then we provided statistical results to understand this relationship.

3.1.1. Amount of Feedback and Programming Performances

The amount of feedback was differentiated in the projects due to the students' needs. We considered the amount of feedback as the indicators of feedback use. Totally 292 posts were provided by students through 547 feedback in the types of corrective, confirmatory, explanatory, diagnostic and expanded feedback. The programming performance scores and the total amount of feedback were presented together in Figure 2.

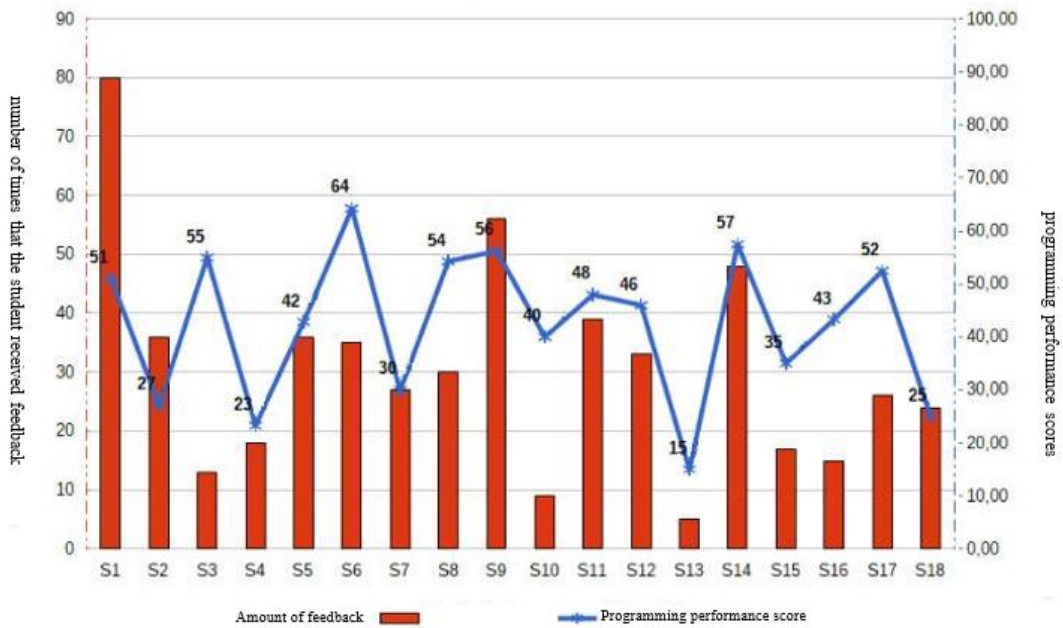


Figure 2. Relations between the amount of feedback (per student) and programming performances

In the question forms students also expressed that they were highly satisfied with receiving feedback as part of learning programming. Some of the participants indicated that feedback was helpful, reminding them that they were guiding them in correcting their mistakes. In this sense, S10 stated that: *“The comments my teacher provided helped me to define the correct programming statements. This was a good way that teacher’s comments directed us when we try to program.”*

3.1.2. Types of Feedback and Programming Performances

Totally 547 feedback including 157 corrective, 53 confirmatory, 204 explanatory, 100 diagnostic and 33 expanded feedback were used in 9 projects. The average of the feedback for each project is 60,7. Figure 3 shows the total amount of feedback and average programming performance scores. While students' average scores were 51, they received an average of 42 feedback for each project. It was found that, the student received a lot of feedback, the programming performance scores were generally high. As a matter of fact, three students (S9, S14) who received the much amount feedback got the highest scores. It was worth noting that the average score of some students was higher than the other students even though they receive a low amount of feedback. For instance, S10 and S3 received a low amount of feedback, but their programming performance scores were 40, 55 respectively. Likewise, S6 who got the highest average score (64,29) from the projects, received averagely 35 feedback from all projects. It was seen that 61% of the students had a score above the average, while others were at the interval of 15 to 40. At this point, in order to determine whether the types of feedback were influenced by the programming performances much, a weekly analysis of the feedback was provided. Descriptively, the results based on the projects were presented in Figure 3.

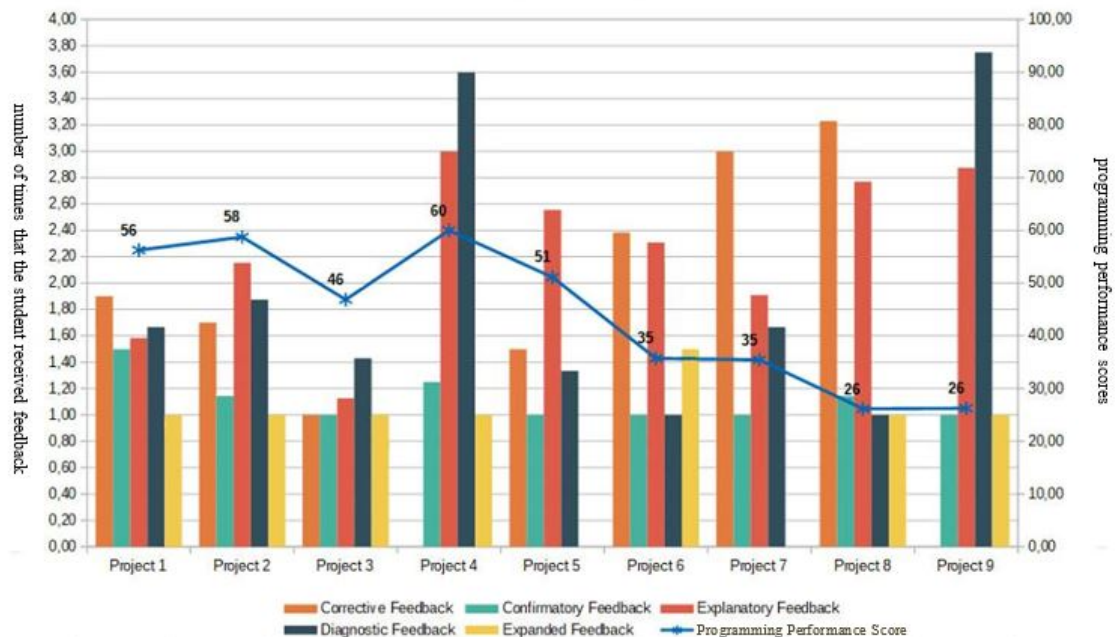


Figure 3. Relations between the number of times that students received feedback (per project) and programming performance scores

Figure 3 shows that the most frequently used feedback types were explanatory, corrective and diagnostic feedback. While more amount of diagnostic feedback was given in Project 4 and Project 9, also more amount of explanatory feedback was provided in Project 7 and Project 8. Also, the instructor provided an average amount of the corrective feedback that the student received was less for each project. Although the amount of feedback use increases towards the end of the implementation, it seems that the average of the scores of the students' answers decreased gradually.

3.1.3. Correlations between the Amount of Feedback and Programming Performances

In addition, statistically, Table 2 shows the relationship between the amount of feedback that the students received and programming performances. If the sample size is less than 30, data distribution is not normal or heterogeneous data structures, the Spearman correlation analysis method is used (Bishara, & Hittner, 2012). As can be seen from the table, the Spearman Product Moment Correlation analysis, which was conducted to determine the relationship between the rubric scores obtained from each project and the total amount of feedback in each project. Statistically, the correlation coefficient was interpreted with its value as below: Exactly -1. A perfect (negative) linear relationship, -0.70. A strong (negative) linear relationship, -0.50. A moderate (negative) relationship, -0.30. A weak (negative) linear relationship, 0. No linear relationship, +0.30. A weak (positive) linear relationship, +0.50. A moderate (positive) relationship, +0.70. A strong (positive) linear relationship, Exactly +1. A perfect (positive) linear relationship.

The results indicate that higher programming performance scores positively correlate with more feedback (gathered from the rubric) ($p < .05$, $r = 0.884$). The correlation between the programming performance scores obtained from each project and the total amount of corrective feedback in each project correlates moderate positive at $p < .05$ level ($r = 0.615$). The higher programming performance scores also correlates with more confirmatory feedback moderate positively ($r = 0.675$). Also the programming performance scores and the amount of explanatory feedback that the students received showed a moderate positive correlation ($r = 0.771$). Moderate positive correlation also exists between the number of items of diagnostic feedback and programming performance scores ($r = 0.519$). Weak positive correlation exists only between high programming performance scores and more expanded feedback ($r = 0.261$).

Table 2. Correlations between number of times that the students received feedback and programming performances

		The number of times that the students received							
<i>Correlations</i> Spearman's rho		total	corrective feedback	confirmative feedback	explanatory feedback	diagnostic feedback	expanded feedback	programming performance scores	
The number of times that the students received	Total	CC Sig. (2-tailed) N	1,000 162	,692** 162	,678** 162	,911** 162	,585** 162	,329** 162	,884** 162
	Corrective feedback	CC Sig. (2-tailed) N	,692** 162	1,000 162	,444** 162	,643** 162	,002 162	,010 162	,615** 162
	Confirmative feedback	CC Sig. (2-tailed) N	,678** 162	,444** 162	1,000 162	,485** 162	,275** 162	,143 162	,675** 162
	Explanatory feedback	CC Sig. (2-tailed) N	,911** 162	,643** 162	,485** 162	1,000 162	,476** 162	,177* 162	,771** 162
	Diagnostic feedback	CC Sig. (2-tailed) N	,585** 162	,002 162	,275** 162	,476** 162	1,000 162	,391** 162	,519** 162
	Expanded feedback	CC Sig. (2-tailed) N	,329** 162	,010 162	,143 162	,177* 162	,391** 162	1,000 162	,261** 162
	Programming performance scores	CC Sig. (2-tailed) N	,884** 162	,615** 162	,675** 162	,771** 162	,519** 162	,261** 162	1,000 162

**Correlation is significant at the 0.01 level (2-tailed) CC: Correlation Coefficient

* Correlation is significant at the 0.05 level (2-tailed)

3.2. Contributions of Various Feedback to the Programming Learning Process

Considering the types of feedback, most of the students evaluated the corrective feedback helpful in their problem-solving process. In this sense, S1 stated: "I think the feedback facilitated my work because they showed me my mistakes and how can I correct them". Another student S2 also emphasized the motivation effect of corrective feedback as "Sometimes I forgot how to use the statements and I could not continue on writing the code, at that time, instructor comments made me see my mistakes and continue writing". Related with the tasks in project 6 and project 8 (requiring extra programming conceptual knowledge about the structure of the language), a few numbers of students emphasized the contribution of the expanded feedback. For instance, according to S4, the instructor suggested them various kinds of resources; and taking advantage of these, they could investigate some structural and logical knowledge to write effective and correct codes. Similarly, S5 explained that instructor feedback for searching other sources helped them not only writing correct codes but also learn why to use such code pieces or functions. In this regard, S6 stated that various examples from various programming web sites which the instructor suggested were very helpful. Some of the students also indicated the effectiveness of the diagnostic feedback. They identified that this kind of feedback was able to make them aware of the mistakes in the programs. In this line, S10 stated: "Sometimes I have little mistakes in the codes but I cannot find where it is. Thanks to instructors' comments that he shows us where the mistake is. Sometimes I consider it as syntax error but the instructor shows that it was about the programming logical structure.". Besides, most of the participants claimed that feedback was needed to be more explanatory.

It was observed that the first responses to the tasks (codes for the problems) were including more mistakes, and students needed more descriptive feedback, namely including some explanations about the way for solutions. Some of them addressed that, they needed some basic examples directing them to define the ways of solutions for the weekly problems. Thus, they considered that more explanatory tips would have been more helpful than corrective feedback. In addition, students' perspectives showed that in some complex tasks more than one type of feedback was required to be given together. Surprisingly, depending on the students' prior programming knowledge, when the prior knowledge is low, the expanded feedback was not sufficient for their needs and they need explanatory feedback within the expanded one.

On the other hand, students' perspectives showed that the complexity of the problems was a crucial factor for using a diversity of feedback. For instance, S6 expressed that after starting the coding in complex problems she often could not continue, because she did not understand the tasks that the instructor ask them to do. According to S6, expanded feedback was not enough solely, so the instructor also added corrective feedback for some tasks.

4. Discussion and Conclusions

4.1. Relationships between Types of Feedback and Programming Performances

Hyland (2001) pointed out that online learners receive feedback from the online instructor is a useful means of communication between the students and the instructor. In this study, we focused on the contributions of using different types of feedback to the programming performances. The results indicated that various feedback in online programming instruction have a positive role in facilitating students' learning programming. In accordance with the findings, another research project addressed that during the programming process, students need to be aware of their mistakes by efficient feedback (HatziaPOSTOLOU & PARASKAKIS, 2010). Although some students' log data confirm the idea that a positive correlation exists between more feedback and programming performance scores, it was not supported by the data of a few numbers of students. Statistically, a strong correlation ($r=0.884$) was found between the rubric scores and the total amount of feedback. Although students in higher programming performance scores did not confirm that higher score correlates with more feedback (Niessen, Meijer, & Tendeiro, 2016); the adverse relationship was confirmed that is when the amount of the feedback received was less, the programming performance scores are also low. It was also found that the amount of feedback for each project was quite more in difficult projects. One reason for the might be that the nature of programming tasks influences the amount of feedback. Independent from types of feedback, the relationship between total number of times that the student received feedback and the programming performance scores was positive. The results were consistent with the study that the coding success of first-year students who received programming lessons through getting feedback during the course was high (Benachour & Edwards, 2009). The present study is also consistent with the findings of Kyrilov (2017) who examined the effectiveness of the feedback in programming exercises with the help of code texts and forms. As a result, feedback received by the students contributed positively to their programming performances. Conversely, inconsistent with the results of the present study, (Heo & Chow (2005) investigated the effectiveness of feedback on students' programming performances and showed that there was no benefit of feedback in online learning.

In this study, some types of feedback were given more than the others. It was observed that the frequency of use of feedback types was in the following order: explanatory, corrective, diagnostic, expanded and confirmatory in this study. One reason for the highest use of explanatory feedback may be the way of the interaction between students and instructors on the GitHub. As a matter of fact, the instructor could improve explanations in each of the posts when the participants require it. One other reason may be students' previous learning experiences. Because some of the students were accounted to be informed about their responses and took the explanations into consideration during the learning process.

4.2. Contributions of the Feedback to the Learning Process

According to Orsmond & Merry (2011), active use of feedback does not seem to be the primary choice for many students. In this study, the expanded feedback was used less because using this type of feedback does not cover enough information for the students to provide solutions for the programming projects. While expanded feedback includes useful knowledge for the given problems, sometimes students need more time to distill the required knowledge. Another obstacle to using this kind of feedback may be the lack of know-how to use this feedback. In this regard, researchers argue that, when the students could not apply strategies for using feedback for their tasks, they gave up using that feedback. A problem, however, is that in order to be effective, feedback should not only be delivered appropriately, but it should also be taken into consideration

by students (Nicol & Macfarlane-Dick, 2006). Therefore, students sometimes needed explanatory feedback in order to use the feedback to infer the conceptual knowledge required for their projects.

Overall, one reason for the positive effect of feedback on the programming performances may be the voluntariness of user. In this study, students did not have to get support from feedback. Accordingly, they did not provide extra effort to think about what they can do with the feedback. Similarly, Zaini (2018) addressed that when students considered the feedback compulsory and in turn altered their voice, they cannot benefit from the feedback as expected. The findings from open-ended questions reflect that the way of using feedback in this study provided a motivational influence on continuing to write programming codes and correcting the mistakes during the coding journey. One reason for this might be that; some of the students' considered that the feedback provided on time when they had problems while writing codes and the projects were not interrupted. Accordingly, their interest increased to continue writing the code.

On the other hand, the difficulty of the projects might be another factor influenced the feedback type given to the students. Because the students' responses to the difficult projects were not the same as those of the easy projects. In this study, the projects were given in an easy to difficult manner. In the easy projects, the expanded feedback was given less and the confirmatory feedback was given more. This was because most of the participants could do easy projects without referring to feedback.

The confirmation that syntax or some basic conceptual knowledge generally met students' needs in which they did not need more coding clues in the easy projects at the beginning of the instructional process. When most of the students could provide correct solutions for the projects, the instructor could not diagnose accurately the type of feedback that the students need. In this sense, the diagnostic feedback was not used more. In the advanced projects, the instructor provided more diagnostic feedback and expanded feedback. In the first five projects, the high mean programming performance scores somewhat positively correlates with more feedback, but the relationships in the subsequent projects which include difficult tasks projects are not clear. In the first five projects, the instructor asked the participants to provide solutions by themselves, thus the feedback was mostly corrective and explanatory. In the last five projects, the instructor and the participants interacted more than those of the first five projects. The instructor provided expanded feedback with the explanatory feedback together when it was not enough for the participants to understand the tips. At this time, sometimes he fulfilled shortcomings in students' code pieces to help them.

In this study, students' perspectives about the feedback in the projects showed that they preferred the feedback mostly when they were useful for them. It is remarkable that students mostly prefer specific and individualized feedback. Similarly, Walker (2009) found that if the students were engaged in one particular assignment, they want feedback directly appealing to them. However, providing feedback for all students individually in an online programming course is a time-consuming process (Duffy & Kirkley, 2004). In this sense, the instructors' role had a crucial influence in this study to provide feedback as soon as possible and to post responses for students' questions. It can be thought that the nature of the feedback provided a key role in its preference. For instance, when the feedback becomes more explanatory, it was considered a general comment all students and it becomes a model for similar codes. Cho and MacArthur (2010) also suggest providing this kind of feedback can result in positive outcomes.

Some of the students emphasized that using GitHub was positively contributed to the students' receiving feedback. In this sense, GitHub has some useful features for asynchronous communication, immediate feedback, transmitting content and it supports to evaluate students. When the instructor and students entered the GitHub at the same time, the amount of feedback increased in the current study. Because, the students rely on to be responded by the instructor, and they feel flexible to continue to the post to the instructor like a conversation on GitHub. This finding is in parallel with the idea of some researchers that many students prefer quick feedback (Rae & Cochrane, 2008). One can infer from the results that, when the feedback giving period is short, the students should be careful to keep in mind what the need was and how does the feedback fulfill the need.

Overall, the study concluded that the most frequently used feedback type was explanatory and corrective feedback. The Spearman Rho correlation proofs this foresight that the highest correlation between the feedback types and the programming performance scores was the explanatory, and corrective feedback. Even if the explanatory feedback was provided mostly in this study, it is difficult to match the feedback type and programming performances clearly. Because there are several factors influencing the feedback use such as the nature of projects, the features of the GitHub environment and the role of instructor. The results indicated that various types of feedback provided positive effects on the online instructional process of programming. One more contribution of this study is that; online programming process sometimes requires combining various types of feedback.

This study has certain limitations that should be enumerated to create opportunities for future research. First, the study utilized a small size of students, the selection and size of the study subjects may limit the generalization of the results. For generalization, future studies to investigate feedback with a broader sample. The data were also interpreted through affordances of GitHub, that different environments for feedback interactions were warranted.

5. Implications for Future Research

In online programming instructions, providing useful feedback is somewhat challenging and time-consuming for the instructor. Considering the affordances of the online platform and the nature of feedback, cautions should be taken when deciding to provide feedback for certain types of programming tasks. Instructors should deal with the affordances of the online learning environment and also should have experiences about learners' needs. In this study, the effect of the feedback was considered through the programming performances on the tasks. A further study may explain relationships between the nature of the tasks, the type of skills or knowledge required to task completion and the nature of the feedback provided for the tasks.

Consequently, this study addressed using various types of feedback in online instruction via GitHub. The results are hoped to guide instructors and instructional designers who may wish to organize courses through the various use of online feedback.

References

- Baker, W., & Hansen Bricker, R. (2010). The effects of direct and indirect speech acts on native English and ESL speakers' perception of teacher written feedback. *System*, 38(1), 75–84.
- Baldwin, L. P., & Kuljis, J. (2000). Visualisation techniques for learning and teaching programming. *Journal of Computing and Information Technology*, 8(4), 285–291.
- Baran, E., Correia, A. P., & Thompson, A. (2011). Transforming online teaching practice: critical analysis of the literature on the roles and competencies of online teachers. *Distance Education*, 32(3), 421–439.
- Barr, V., & Guzdial, M. (2015). Advice on teaching CS, and the learnability of programming languages. *Communications of the ACM*, 58(3), 8–9.
- Benachour, P., & Edwards, R. (2009). Animation and interactive programming: A practical approach. *Electronic Notes in Theoretical Computer Science*, 224, 133–140.
- Campos, D. S., Mendes, A. J., Marcelino, M. J., Ferreira, D. J., & Alves, L. M. (2012). *A multinational case study on using diverse feedback types applied to introductory programming learning*. Frontiers in Education Conference proceedings (FIE 2012) (pp. 594–599). Seattle, Washington.
- Cho, K., & MacArthur, C. (2010). Student revision with peer and expert reviewing. *Learning and Instruction*, 20(4), 328–338.
- Duffy, T. M., & Kirkley, J. R. (2004). *Learning theory and pedagogy applied in distance learning: The case of Cardean University*. Learner-Centered Theory and Practice in Distance Education: Cases from Higher Education, 107–141.
- Eng, T. H., Ibrahim, A. F., & Shamsuddin, N. E. (2015). Students' perception: Student Feedback Online (SuFO) in higher education. *Procedia - Social and Behavioral Sciences*, 167, 109–116.
- Ferris, D. R. (1995). Student reactions to teacher response in multiple-draft composition classrooms. *TESOL Quarterly*, 29(1), 33–53.
- Fini, A. (2009). The technological dimension of a massive open online course: The case of the CCK08 course tools. *The International Review of Research in Open and Distributed Learning*, 10(5).
- Gikandi, J. W., & Morrow, D. (2016). Designing and implementing peer formative feedback within online learning environments. *Technology, Pedagogy and Education*, 25(2), 153–170.
- Gikandi, J. W., Morrow, D., & Davis, N. E. (2011). Online formative assessment in higher education: A review of the literature. *Computers & Education*, 57(4), 2333–2351.
- Gunnarsson, S., Larsson, P., Månsson, S., Mårtensson, E., & Sönnnerup, J. (2017). *Enhancing student engagement using GitHub as an educational tool*. Lund University. Retrieved from <https://www.lunduniversity.lu.se/lup/publication/>
- Hatziaopoulou, T., & Paraskakis, I. (2010). Enhancing the impact of formative feedback on student learning through an online feedback system. *Electronic Journal of e-Learning*, 8(2), 111–122.
- Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1), 81–112.
- Heo, M., & Chow, A. (2005). The impact of computer augmented online learning and assessment tool. *Educational Technology & Society*, 8, 113–125.
- Huxham, M. (2007). Fast and effective feedback: are model answers the answer? *Assessment & Evaluation in Higher Education*, 32(6), 601–611.

- Hyland, F. (2001). Providing effective support: Investigating feedback to distance language learners. *Open Learning: The Journal of Open, Distance and e-Learning*, 16(3), 233–247.
- Hyland, F. (2010). Future directions in feedback on second language writing: Overview and research agenda. *International Journal of English Studies*, 10(2), 171–182.
- Jara, M., & Mellar, H. (2010). Quality enhancement for e-learning courses: The role of student feedback. *Computers & Education*, 54(3), 709–714.
- Jonsson, A. (2013). Facilitating productive use of feedback in higher education. *Active Learning in Higher Education*, 14(1), 63–76.
- Kluger, A. N., & DeNisi, A. (1996). The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological Bulletin*, 119(2), 254–284.
- Kop, R. (2011). The challenges to connectivist learning on open online networks: Learning experiences during a massive open online course. *The International Review of Research in Open and Distributed Learning*, 12(3), 19–38.
- Kyrilov, A. (2017). *Using case-based reasoning to improve the quality of feedback provided by automated assessment systems for programming exercises*. (Unpublished doctoral dissertation). UC Merced.
- Law, K. M. Y., Lee, V. C. S., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education*, 55(1), 218–228.
- Lawless, K. A., & Pellegrino, J. W. (2007). Professional development in integrating technology into teaching and learning: knowns, unknowns, and ways to pursue better questions and answers. *Review of Educational Research*, 77(4), 575–614.
- Lipnevich, A. A., & Smith, J. K. (2009). Effects of differential feedback on students' examination performance. *Journal of Experimental Psychology: Applied*, 15(4), 319–333.
- Ludwig-Hardman, S., & Dunlap, J. C. (2003). Learner Support Services for Online Students: Scaffolding for success. *International Review of Research in Open and Distance Learning*, 4(1).
- Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017–1054.
- McGill, T. J., & Volet, S. E. (1997). A conceptual framework for analyzing students' knowledge of programming. *Journal of research on Computing in Education*, 29(3), 276–297.
- Naidu, S. (2007). *Instructional design for optimal learning*. In G. M. Moore (Ed.), *Handbook of distance education*. Mahwah, NJ: Lawrence Erlbaum Associates. 247–258
- Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2), 199–218.
- Niessen, A. S. M., Meijer, R. R., & Tendeiro, J. N. (2016). Predicting performance in higher education using proximal predictors. *PLoS ONE*, 11(4).
- Orsmond, P., & Merry, S. (2011). Feedback alignment: effective and ineffective links between tutors' and students' understanding of coursework feedback. *Assessment & Evaluation in Higher Education*, 36(2), 125–136.
- Pitts, S. E. (2005). 'Testing, testing...': How do students use written feedback? *Active Learning in Higher Education*, 6(3), 218–229.
- Pyke, J. G., & Sherlock, J. J. (2010). A closer look at instructor-student feedback online: A case study analysis of the types and frequency. *Journal of Online Learning and Teaching*, 6(1), 110–121.
- Rae, A. M., & Cochrane, D. K. (2008). Listening to students: How to make written assessment feedback useful. *Active Learning in Higher Education*, 9(3), 217–230.
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129–141.
- Schimmel, B. J. (1983). *A meta-Analysis of feedback to learners in computerized and programmed instruction*. Paper presented at the AREA, Montreal.
- Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1), 153–189.
- Singh, R., Gulwani, S., & Solar-Lezama, A. (2013). Automated feedback generation for introductory programming assignments. *ACM SIGPLAN Notices*, 48(6), 15–26.
- Van Horne, S., Curran, M., Smith, A., VanBuren, J., Zahrieh, D., Larsen, R., & Miller, R. (2018). Facilitating student success in introductory chemistry with feedback in an online platform. *Technology, Knowledge and Learning*, 23(1), 21–40.
- Vasilyeva, E., Puuronen, S., Pechenizkiy, M., & Rasanen, P. (2007). Feedback adaptation in web-based learning systems. *International Journal of Continuing Engineering Education and Life Long Learning*, 17(4–5), 337–357.
- Vujošević-Janičić, M., & Tošić, D. (2008). The role of programming paradigms in the first programming courses. *The Teaching of Mathematics*, 11(2), 63–83.

- Walker, M. (2009). An investigation into written comments on assignments: Do students find them usable?. *Assessment & Evaluation in Higher Education*, 34(1), 67–78.
- Weaver, M. R. (2006). Do students value feedback? Student perceptions of tutors' written responses. *Assessment & Evaluation in Higher Education*, 31(3), 379-394.
- Xia, B. S. (2015). Benefit and cost analysis of massive open online courses: Pedagogical implications on higher education. *International Journal of Cyber Behavior, Psychology and Learning (IJCBL)*, 5(3), 47-55.
- Xia, B. S., & Liitiäinen, E. (2017). Student performance in computing education: an empirical analysis of online learning in programming education environments. *European Journal of Engineering Education*, 42(6), 1025–1037.
- Zaini, A. (2018). Word processors as monarchs: Computer-generated feedback can exercise power over and influence EAL learners' identity representations. *Computers & Education*, 120, 112–126.