COMPARISON OF NOVICE PROGRAMMERS' PERFORMANCES: Blended Versus Face-To-Face

Asist. Prof. Dr. Unal CAKIROGLU,
Department of Computer Education and Instructional Technology
Karadeniz Technical University,
Fatih Faculty of Education, Trabzon,TURKEY

ABSTRACT

This study investigated the effect of blended learning on novices' understandings of the introductory programming. A quasi-experimental design with participants of preservice computer and instructional technologies teachers, one control group (CG, N =64) and one experimental group (EG, N=61) who received the course 11 weeks. While face-to-face courses were taught face-to-face in classroom and in lab, blended courses were conducted in synchronous and asynchronous settings and also in lab sessions. The pretest, posttest and delayed tests were used to collect data. The participants in two groups were separated into three categories (poor, average and good) according to the pretest results.

The results of the study showed that blended and face-to-face courses have statistically similar effects on academic achievements among the three categories. However the delayed test results showed that, face-to-face courses were more effective on permanence than blended courses. Thus, considering the main goal of the introductory programming courses as enhancing students in the second category progress into the third, the organization of blended courses had to be revisited. Nevertheless, this study supported the idea that "Crucial challenges may exist on teaching some subjects via blended learning, which include intensive cognitive processes and some new approaches are needed to enhance permanence".

Keywords: Blended learning, face-to-face learning, introductory programming, teacher training

INTRODUCTION

The online learning encourages learners and tutors to reassess their roles in the learning process (Light, Nesbitt, Light, & White, 2000). Because new teaching methods due to the flexibility may present new opportunities. In this context, Tabor (2007) addresses that blended learning, an alternative teaching way, uses a combination of traditional face-to-face contact with on-line learning. Picciano (2006) made a more acceptable definition for blended learning as a method of instruction that combines online learning with face-to-face activities that are integrated in a planned, pedagogically valuable way and where some of the face-to-face time is replaced by online activities. By using the flexibility of learning environments higher education institutions are using blended instruction to improve pedagogy.

Also the cost-effectiveness is important for preferring this method. One type of blending is first; students complete activities online by sharing common knowledge base then they the face-to-face meetings are organized by the instructor. Generally content can be supplemented and enriched with exercises and problem solving activities during face-to-face time. Also the face-to-face time can be used to learn the topics at a deeper level by instructor and peer support. Aycock, Garnham, & Kaleta (2002) recommend another type of blending that is organized by teaching the course content during class time and allowing students to think critically and discuss their views about the material through online activities.

Considering the advantages of blended learning, higher education institutions have started to give courses with this method. The implementation of blending learning courses have been reported in education and computer science that the introductory programming course is also included (Rovai & Jordan, 2004).

On the other hand Dagdilelis, Satratzemi & Evangelidis (2004) put forward that novice programmers come up cognitive obstacles that make it difficult to understand the construction of programs. So the problems and difficulties associated with the learning of introductory programming remain to be investigated. Learning introductory programming includes the learning of theory and practical skills in a useful learning environment. Hadjerrouit (2007) noticed that, in programming courses, the application of modern technologies has considerable importance towards blended learning.

In this case, in programming courses using the blending method may improve student participation, and understanding as well as to encourage them begin more active on learning which can be particularly difficult in large-sized courses.

About blending programming; Ersoy (2003) founded that the students' perceptions about online instruction and online instructor were positive, but blended collaborative learning perceptions were neutral. Hadjerrouit (2007) conducted on a blended learning model in Java programming at the introductory level. The blended environment had a positive impact on the students' learning of Java programming as a result of well-organized and easy accessible information. In another study, more than 600 students in two higher education institutions used the blended learning environment and the results indicate a generally positive evaluation of the main components of this method (Massoud, Iqbal, Stockley, & Noureldin, 2011). Some other research studies found that online resources are highly important but not sufficient to learning programming (Bliuc, Goodyear, & Ellis, 2007; Stacey & Gerbic, 2007). Hadjerrouit (2007) points out that face-to-face learning is required to support online courses because the teachers not only convey subject information to the students, but act as facilitators and guides of learning on face-to-face learning. In addition, student collaboration is useful when the more competent students helped the ones who faced difficulties in accomplishing their programming tasks.

The study results above, reveals the superiority of face-to-face learning in some cases and of blended learning in some other cases. The idea of online presentation of a certain part of introductory programming, which has started to become widespread in recent years, needs to be verified by various evidences considering the nature of the course and the features of the learner.

Regarding a need as this, this study is planned to be a comparative study which will present experimental evidences. In the context of this study, the effects of introductory programming courses carried out in face-to-face and blended settings; on academic achievements have been compared. Within this framework, in this study the features of blended and face-to-face learning settings' contributions to different level students' programming languages knowledge were put forward.

METHOD

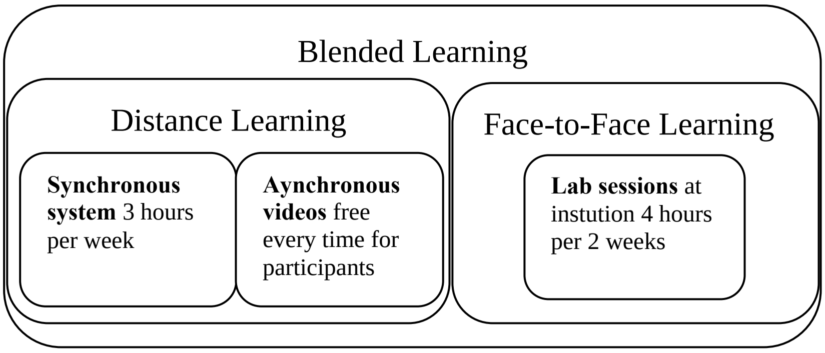
Design and Sample

In the study, a quasi-experimental design was administered which the naturally occurring groups are considered. The study utilized a pretest-posttest nonequivalent control group design. The sample for the study consisted of 125 sophomore students (60 girls, 65 boys, and age range 18–20 years). One control group (CG) (with 64 students, 26 girls, 28 boys) and one experimental group (EG) (with 61 students, 24 girls, 27 boys) were participated in the study.

Participants in EG and CG have received Information technologies in education course in previous year.
In this course they recognized the basic software and hardware components. So they have some introductory knowledge about programming components like what variable, memory, algorithm and etc are.

Process

The process in CG was conducted during course time through 5 lessons a week; 3 hours in the classroom and 2 hours in the lab, for 11 weeks. Both of the classroom and lab sessions were face-to-face. The EG students were in different locations and they received the courses by using online environment synchronously for 3 hours per week. Combining the remaining 2 hours required to be treated weekly in two weeks, they are treated in lab as 4 hours. When they appear in lab, students studied on programming applications and tasks with the help of their teachers. Since, blended courses are known as hybrid or mixed courses where a portion of the traditional face-to-face instruction is replaced by synchronous online learning, 3 hours of the course is delivered from distance learning (DL) environment as online learning in this study. Thus, the process in EG can be considered as blended learning. Blended learning environment in this study is summarized in Figure1.


Blended Learning
Distance Learning
Face-to-Face Learning
Synchronous system 3 hours per week
Aynchronous videos free every time for participants
Lab sessions at instution 4 hours per 2 weeks

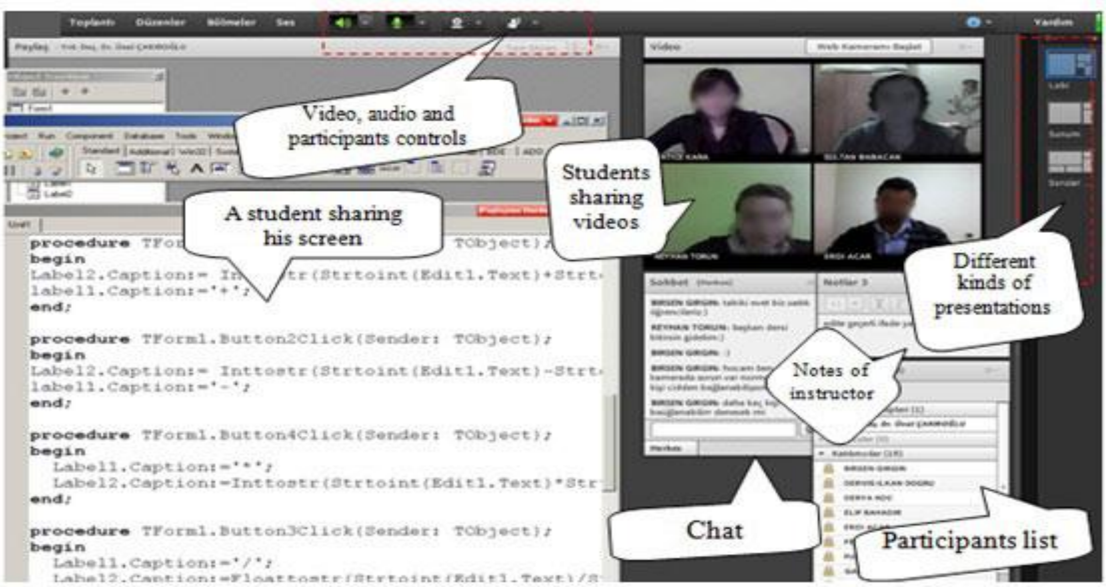
Both of the groups were informed about programming language, program design and problem solving in the courses.

The courses included programming language constructs, algorithm development, interface design, general programming concepts and problem solving activities. CG received traditional instruction, which included teacher explanations, demonstrations, and basic problem solving examples. They used text books and teacher notes as references. In the lab sessions they studied the examples and worked on their homework. EG students received the same subjects in DL environment both synchronous and a synchronous. The synchronous setting supports three main dimensions of the course which are; Meeting, communication and content sharing. Table 1 presents the details for synchronous setting.

Table: 1
Features of synchronous setting

Basic Components of Synchronous System	Context of the components
Meeting organization	Meeting organization and recording of meetings Attendance in the talks, teleconferences
Communication in the course	Sharing web cam video Attendance in virtual class Imaging participant list or changing it, conducting survey on the participants
Content sharing in the class	Sharing the desktop, documents, presentations Notes, chat, questions/answers and surveys Sharing white board Taking notes during the meeting

The synchronous system supports almost all of the features together at the same time due to the internet connection bandwidth. Figure 2 illustrates a view from a lesson in synchronous setting using some features together.



```
procedure TFor                              TObject);
begin
Label2.Caption:= In      tr(Strtoint(Edit1.Text)+Strt
label1.Caption:='+';
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
Label2.Caption:= Inttostr(Strtoint(Edit1.Text)-Strt
label1.Caption:='-';
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
  Label1.Caption:='*';
  Label2.Caption:=Inttostr(Strtoint(Edit1.Text)*Str
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  Label1.Caption:='/';
  Label2.Caption:=Floattostr(Strtoint(Edit1.Text)/S
```

Figure: 2
Features of synchronous setting.

The courses were carried out in synchronous setting in which qualifications about video, audio, chat, whiteboards, online surveys and online settings have been prospered. Also the video records of the courses were presented in the system free for participants to use asynchronously. Figure 3 illustrates a view from the video records.

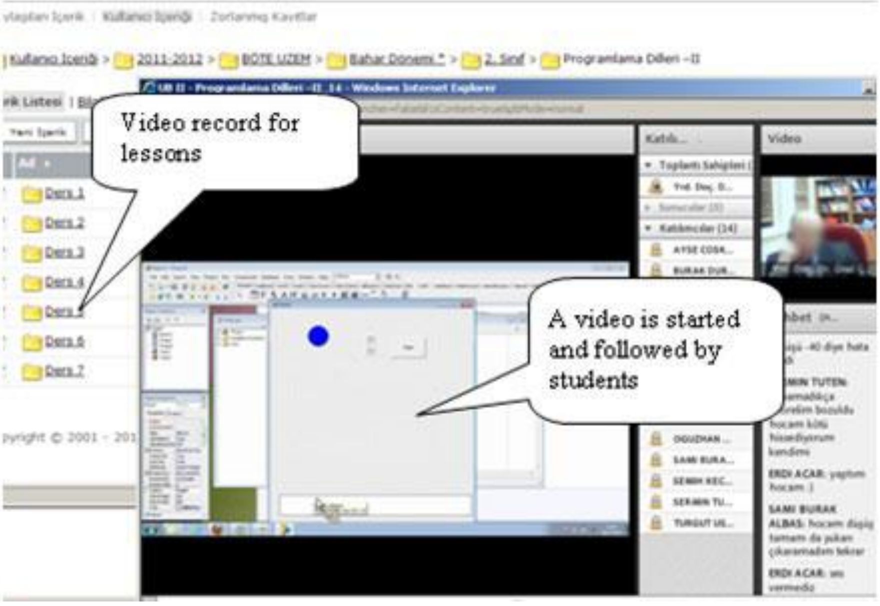

Kullanıcı İçeriği > 2011-2012 > BÖTE UZEM > Bahar Dönemi > 2. Sınıf > Programlama Dilleri –II
Video record for lessons
Ders 1
Ders 2
Ders 3
Ders 4
Ders 5
Ders 6
Ders 7
Video
A video is started and followed by students

Figure: 3
Teaching in synchronous setting.

Most of the students have gathered experience about the usage of synchronous setting in previous year, thus the instructor did not have to introduce the system. The participants were notified about programming language qualifications program design and problem solving in the courses. The schedule is illustrated in Table2.

Table: 2
The schedule for two groups

Week	Topics
1	Structural programming, algorithms
2	Algorithms
3	Introducing programming language environment: first program, compilation, syntax errors
4	Variables, data types, memory, arithmetic expressions, conditional statements
5	Loops (for, while, repeat)
6	Arrays, character arrays and string processing
7	Sub programs: Functions, procedures, parameter passing
8	Sorting, merging algorithms
9	Recursion, problem solving
10	Problem solving, sample codes
11	Problem solving, sample codes

The intervention in EG and the process in CG is summarized in Table3.

Table: 3
The process in CG and EG.

Control Group	Experimental Group
Instructor	**Instructor**
Presented program codes by projector.	Presented program codes by synchronous system.
Gave examples from textbook.	Used both textbook and internet for examples.
Gave copies of expert codes and designs of some programs.	Explained and discussed problem solutions.
Explained and discussed problem solutions.	
Walk around all students to check their design and codes in lab sessions.	Walked around all students to check their design and codes in face-to-face lab sessions.
Students	**Students**
Studied on weekly homework or projects.	Received subjects by using synchronous system features.
Used whiteboard to show how the code proceeds.	Received information by asynchronous videos recorded by system.
	Studied on weekly homework or projects.

Within each lesson, there were exercises both in EG and CG has a suitable number of examples, and each example had as a starting point code that embodies the concepts introduced in the lesson. In this way, after receiving theoretical knowledge, students were allowed to develop practices on simple codes.

Instruments

The study began with conducting three open-ended questions to both groups of students as pretest with the aim of determining whether the readiness's of the groups were similar or not. The pretest included items about basic programming language features, basic concepts in programming and a basic problem.

Since the participants in both group come from similar educational and socio-economic backgrounds, the distributions of students' graduate high schools were heterogeneous in the groups. Because some of them might have more experience in programming who graduated from vocational high schools and some of them were from traditional high schools that have less experience in programming. Thus, the aim of the pretest was to determine students' levels at programming which is important for revealing the impact of blended environment to students who has different readiness. Posttest was constructed for measuring the programming knowledge of students, which is related with the course outcomes and common programming concepts. The posttest included four items related to the three main dimensions for programming knowledge in which "programming knowledge evaluation model" is used. Bayman & Mayer (1988) address these three dimensions as syntactic knowledge, programming structure knowledge and strategic knowledge.

Two experienced programming instructors and two expert programmers examined both pretest and posttest for content validity and inter rater reliability technique is conducted for validation. The raters first assigned the points for the items individually, and then they discussed each other until they come to exact agreement on each item on the scale. It can be concluded that they share a common understanding of the criteria on the scale. The correlation coefficient among experts was 0.88.

Data Analysis

Dunican's (2002) categories for novice programmers are used in the study which include three categories shown in Table 4.

Table: 4
Categories of novice programmers

Categories of novice programmers	Abilities
Poor	do not understand the basic concepts
Average	understand basic concepts, it is possible if the teachers use effective teaching approaches
Good	easily grasp the nature of programming concepts

Generally introductory programming courses fails on enhancing poor students to average level but some evidences are recorded that research studies are succeeded to enhance students in the second category progress into the third, and also the good students into more good (Barg et al, 2000; Hagan & Macdonald; 2000; Williams & Kessler, 2000). Thus the pretest results are used to determine the programming knowledge level of students before intervention. Considering that the students had not received any programming language course at university level, they were assessed as 0-30: poor; 31-50: average; 51- 100: good. Results of posttest were utilized for determining the developments of the students EG and CG groups, in these three categories.

The experts (2 programming instructors) reviewed the papers as per the criteria developed by instructor and them together. The averages between the experts' scores are taken into consideration for each item of the test. The pretest and posttest results for both groups have been analyzed statistically. The data acquired from pretest and posttests provide the conditions of parametric tests. Therefore, paired samples t-test is carried out to define if there is a significant difference among the grades of pretest and posttest for each category students in both two groups. Similarly, according to the results of posttest; independent sample t-test is conducted if there is a significant difference between two groups. In the analysis of pretest and posttest, the total score of each student in categories of two groups as well as the mean score of each group were computed.

RESULTS AND DISCUSSION

In this section, students' pretest and posttest scores in CG and EG are compared statistically and the features of the settings and relations between academic performances are discussed. As a result of the pretest, the number of poor, average and good students and their average scores are shown in Table 5.

Table: 5
Distribution of poor, average and good students

Group	Poor	Average	Good	Average-Good	Total
CG	32	20	12	32	64
EG	31	21	9	30	61

The EG and CG students' averages of scores taken from pretest and posttest have been presented in Table 6.

Table: 6
The results of pretest and posttest scores of students in CG and EG

Test	Groups	N	M	SD	t	P
Pretest	CG	64	33.06	17.31	-0.931	0.353
	EG	61	30.16	19.69		
Posttest	CG	64	45.75	22.038	-0.488	0.626
	EG	61	47.46	19.56		

The mean scores of the pretest were similar between the CG (M=33.06; SD=17.31) and EG (M=30.16; SD=19.69). According to pretest results owing to the data, t(123) = -0.931 and p>0.05 there was no significant difference among the previous programming knowledge of the groups. So at the beginning, the pretest result reflects the similar backgrounds in both groups.

A statistically significant difference between the mean posttest scores of the CG (M=45.75; SD=22.038) and EG (M=47.46; SD=19.56) groups was not found (t(123)=-0.488, p=0.626).

This result shows that there was no significant difference on academic performances among the face-to-face and blended learning groups' students. Even if the pretest measures basic knowledge about programming, the posttest measures the academic achievements related to the introductory programming course objectives. It is found that while there is no significant difference between the groups' programming knowledge levels at pretest also no significant difference is recognized according to the posttest results.

These results show that students in both settings can enhance academic performances regardless of their categories according to their previous situation. The pretest and posttest paired samples t-test analysis results of CG and EG "poor" students are illustrated Table 7.

Table: 7
Pretest and posttest results for CG and EG "poor" students

Group	Tests	N	M	SD	df	t	p
CG	Pretest	32	12.94	8.14	31	-10.55	0,000
	Posttest	32	29.33	13.44			
EG	Pretest	31	18.35	9.76	30	-11.60	0,000
	Posttest	31	34.52	15.03			

Statistically, the number of the students in both groups is not sufficient for comparisons (Number of students in "average" group for CG is 20 and for EG is 21, number of the students in "good" group for CG is 12 and for EG is 9); so statistical interpretations were carried out by combining the students in two groups and named as "average-good". The pretest and posttest paired samples t-test analysis results of CG and EG "average-good" students are illustrated Table: 8.

Table: 8
Pretest and posttest results for CG and EG "average-good" students

Group	Tests	N	M	SD	df	t	p
CG	Pretest	32	46.92	8.14	31	-9.29	0,000
	Posttest	32	61.73	13.44			
EG	Pretest	30	46.9	9.62	35	-9.976	0,000
	Posttest	30	59.66	15.01			

When pretest and posttest scores of poor, average and good students in both groups were compared, it was observed that the averages of CG poor students increased from 12.94 to 29.33 and the averages of EG poor students increased from 18.35 to 34.52 and these increases were statistically significant.

Similarly, averages of CG average-good students increased from 46.92 to 61.73, and averages of EG average-good students increased from 46.9 to 59.66. It was determined that there was significant difference in increases of both groups.

In this context, it is clear that blended courses can make contribution to teaching the basic components of programming such as algorithm development, programming language constructs, interface design and general programming concepts as much as face-to-face can. In this matter, the effect of synchronous system in learning setting is in case which is similar setting to the one in face-to-face courses.

Thus, it is expressed by different researchers that blended learning settings are able to create a system to support seven fundamental principles required for good practice in Chickering & Gamson (1987) higher education. In these seven principles, components such as student-faculty contact, cooperation among students, active learning, and feedback for students, time management and diverse ways of learning are included.

Berk (2009) highlights blended learning is supposed to encourage students to share their ideas and opinions through discussions, collaborative exercises, social media networks and digital storytelling forums and blogs.

Synchronous system making way for discussions mentioned in blended settings played an important role. With the organization of synchronous system meeting, design of programming codes and with the organization of development and processing stage meeting the presentation of the subject were made for students. Again in this meeting setting, students made discussions on code pieces.

Communication channels on system (visual, vocal, written chat) and instructor's feedback presentations to students enabled them to make arrangements on the projects and present them. All these activities contributed to the fulfillment of expectations from instructor and students in the context of active learning. Besides, with sharing of screen images by students and instructor and whiteboard applications, active learning settings similar to face-to-face courses could be created in the context of synchronous applications of blended setting.The pretests and posttests results of EG and CG's "poor" students' were used to conduct an independent test and the results are shown in Table: 9.

Table: 9
The results of pretest and posttest scores in CG and EG for "poor" students

Test	Groups	N	M	SD	t	P
Pretest	CG	32	12.94	8.14	-1.081	0.076
	EG	31	18.35	9.76		
Posttest	CG	32	29.33	13.44	-1.52	0.132
	EG	31	34.52	15.03		

It was observed that there was no significant difference between pretest and posttest score averages of poor students. Table 10 illustrates whether there is significant difference between EG and CG groups' "average-good" students' scores.

Table: 10
The results of pretest and posttest scores in CG and EG for "average-good" students

Test	Groups	N	M	SD	t	P
Pretest	CG	32	46.92	8.14	-.01	0.992
	EG	30	46.9	9.62		
Posttest	CG	32	61.73	13.44	.563	0.575
	EG	30	59.66	15.01		

While no significant difference was observed in comparison between CG and EG poor students, absence of a significant difference in comparison between EG-CG average-poor students (t(60)=0.563, p=0.575) indicated that EG settings could contribute to average-poor students as much as face-to-face could. All of these comparison results can be interpreted in the way that blended settings are as practical as face-to-face settings for learning programming. That is supported by the idea of Kenney & Newcombe (2011) as blended or hybrid instruction uses a combination of face-to-face and online learning activities and has been found to increase understanding, interaction, and involvement in the learning process. As well as the enhanced features of system, the ability of instructor to use these features might also have an important role.

Both the developments of scores of "average students" and "poor students" from pretest to posttest are illustrated in Figure 6. When assessing the average scores of CG and EG students considering only the "average students", it was observed that there were 20 CG students in this level and that pretest CG (M= 39.39), posttest CG (M=53.04). While it was observed that there were 21 students in experiment group, mean values were found as pretest EG (M=42.67), posttest EG(M=53.38). When assessing average scores of only "good students" of CG and EG, it was observed that in control groups there were 12 students at this level and pretest CG (M=59.29), posttest CG (M=76.00) and there were 9 students in experiment groups with mean values as pretest EG (M=58.18), posttest EG (M=78.22). As is seen, developments of both "average" and "good" students in two groups are very close to each other according to average values. This reflects that learning programming course in face-to-face or blended way provides similar benefits for students at the same level.

On the other hand the same tests were also administered as delayed test three months after the posttest in order to discuss the permanence of the programming knowledge after intervention. It is assumed that the time interval between the conducting of the posttest, and delayed test is sufficient for the students to forget the test items. Thus, the effect of blended learning settings on permanence was revealed. In Table 11 independent test results of "poor" students, in Table 12 independent test results of "average-good" students among EG and CG are shown. These tests were conducted considering their scores in delayed tests.

Table: 11
The results of delayed test in CG and EG for "poor" students

Category	Test	Groups	N	M	SD	t	P
Poor	Delayed test	CG	32	24.38	11.21	-1.946	0.056
		EG	31	30.23	13.85		
Average-good	Delayed test	CG	32	56.37	12.94	2.225	0.029
		EG	30	49.27	14.30		

In delayed test conducted on "poor" students, significant difference was not observed at 0.05 level between CG and EG. But a significant difference was observed in favor of CG in "average-good students" delayed test. However, in permanence issue no significant difference was observed between "poor" students of both groups. As s summarize; the pretest, posttest and delayed test average scores are shown in Figure 6.

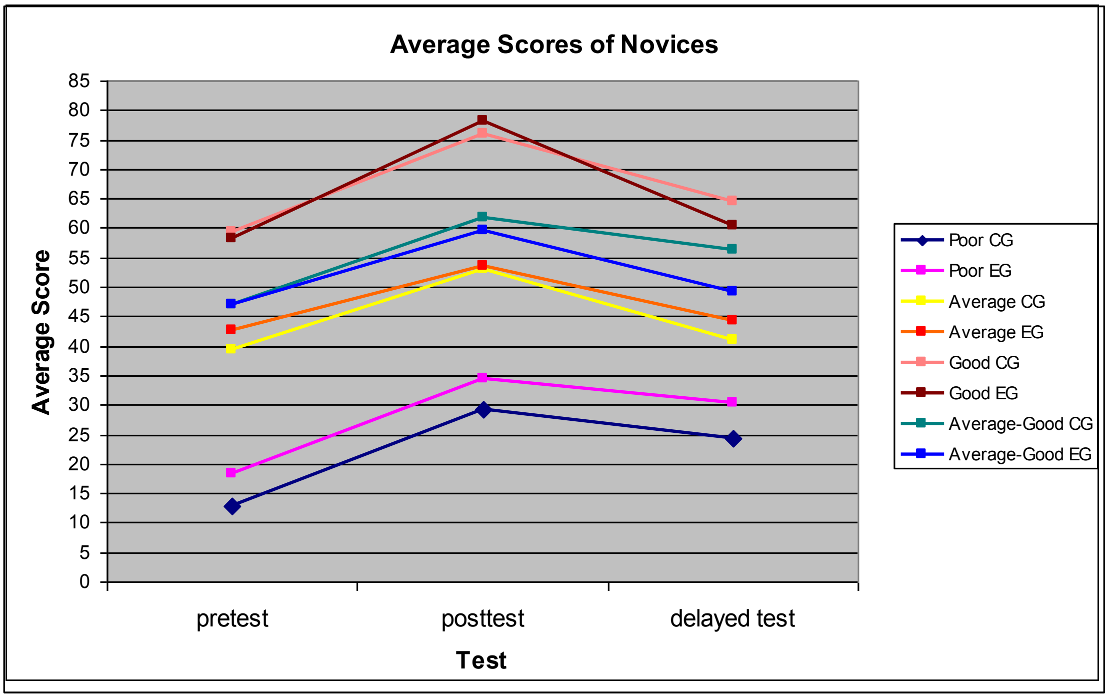


Figure: 6
Trends in categories of EG, CG from pretest to posttest and delayed test

Occurrence of a significant difference in favor of CG between permanence of academic performances of "average-good" students is one of the important findings of the study. This finding indicates that blended settings, an important part of which is treated in distance learning way, are not as effective as face-to-face on permanence. Herein, the reason of considering "average-good" students' results from the idea that introductory programming languages course provides an important development on these student categories (Dunican, 2002). On the other hand, course videos given as asynchronous via web, have been watched many times by the students in blended setting. Even though these videos are considered to be a good chance for remembering the problem solving methods in future, it was interesting to see that EG students had low performance in delayed test.

Students' easier communication with instructor in face-to-face courses and the high level cooperation among them could have contributed to permanence. However, it can be considered that instructor's physical existence in face-to-face setting provides a different support for student which is different from blended setting. Because, when instructor is approachable, the student may consider her/him more dependable and realistic (King, 2002). For an instructor it remains a presumption that he/she has conveyed the information effectively and that the student has the background knowledge that is essential to understand the presented. From this point of view, instructor support and guidance was this kind of scaffolding to enhance abstract subjects. Thus, Jonassen, Beissner, & Yacci (1993) defined scaffolding as any help intentionally given to students so that they could fulfill a task. As programming course is a course where cognitive processing is complex, the teaching this course in blended setting may cause some difficulties in understanding certain abstract subjects (memory, allocation, data structures) or in permanence. At this point students may need face-to-face supports.

Brown (2003) stated that blended learning supports all the benefits of e-learning including cost reductions, time efficiency and location convenience for the learner as well as the essential one-on-one personal understanding and motivation that face-to-face instructions presents. At this point, Figure 6 indicates that with this study "average-good students" are supported apart from their low performance in permanence. Some educators believe that problem solving in introductory programming requires a radical change from traditional instructional methods to constructivist learning environments and situated learning (Ben-Ari, 2001).

Since, programming is generally considered as a skill that students need to acquire through an active construction process. The results of this study may provide important evidences enabling constructivist learning settings to occur to some extend in blended settings. Of course, the absence of any differences, except form permanence, between CG and EG groups could have been resulted from the features of the students and the instructor. Hence, Tabor (2007) reported that students who disliked the blended format mentioned problems with finding materials, receiving less instructor feedback, and perceiving the course content to be too advanced for independent learning. In a programming instruction research study few education professors have taken the initiative to incorporate more online components into their courses as blended learning. The common properties of these faculty members were for the most part self-motivated and did not receive any compensation or workload reduction (Kenney & Newcombe, 2011).

In this study, it can be thought that the blended instruction of programming languages course with institution organization but with instructor initiative, and the occurrence of a necessity and a prepared schedule had an effective role in realizing the integration of face-to-face and DL environment. Another research in programming illustrates blended courses offer students all the necessary levels of knowledge in programming and gives satisfactory results in knowledge tests. In blended courses, students may have an advantage as repeating each course any number of times, which is not the case with traditional courses (Djenic, Krneta, & Mitic, 2011).

The DL environments can make students mature, assertive, self-disciplined, and independent (Buchanan, 2000). Also students may be high motivated and possess well-developed self-directed learning skills by DL environments (Carlson & Repman, 2000) in order to reach effective DL.

As is seen the advantages of DL and face-to-face learning should be taken into consideration. Vaughan (2007) highlights effective integration of DL and face-to-face learning creates environments that are highly conducive to student learning. In this context; Kaleta, Skibba, & Joosten (2007) pointed out that students did not like blended instruction if they perceived a poor integration between the face-to-face and the DL components or if they felt the DL components merely increased the course workload.

CONCLUSIONS

When the pretest and posttest scores of both group students in "average-good" students were examined, it was seen that the performance of students in both groups increased from pretest to posttest. The percentages of the scores in pretest are between 4 and 70% for EG group and between 2 and 68% for CG group, these scores are between 4 and 100% for EG group and between 8 and 100% for CG group in posttest. The calculations show the enhanced improvement in the EG and CG group students. On the other hand, the scores of the delayed test are between 4 and 78% for EG group and between 4 and 80% for CG group.

The study noticed an increase at both groups EG and CG. Since the literature emphasizes that introductory programming courses focuses on "average" and "good" students, the results of these category of students present crucial evidences. The pretest measures basic programming knowledge and the posttest measures the achievement about the programming concepts and problem solving studied during study and it is seen that while there is no significant difference on pretest scores. Also the intervention did not effect on posttests significantly, but significant difference is recognized according to the delayed test results of "average-good" students of EG and CG in favor of CG. This result indicated that in courses such as introductory programming here cognitive processing is complex in blended learning, there may occur certain difficulties in permanent learning of certain abstract subjects and students may need face-to-face support.

The academic performances of students in blended environment were as good as face-to-face students. This was because; incorporating the advantages of traditional and modern methodologies and DL technologies by shaping the programming courses combining online and offline components.

It should be noted that in permanence issue the limited social interaction is very important in keeping blended setting limited.

Thus, Frank, Reich & Humphreys (2002) stated the importance of social interaction in the learning environment. In their study, they investigated DL students and concluded that one disadvantage in DL was its lack of eye-to-eye interaction between both teacher-student and student-student.

As a result considering its advantages and cost effectiveness, blended learning can be utilized in programming languages course to teach in classes with high capacity (Dodero, Fernández, & Sanz, 2003). But transforming a traditional course into a blended one is not an easy process and requires instructor to take a different perspective on presenting the course.

RECOMMENDATIONS

The findings reveal that both the design features of the synchronous, asynchronous DL environments, students' views include the items which can effect to the academic achievements and permanent learning should be taken into consideration in order to design blended learning environments. The experiences of this study showed that, blended programming should be required the following:

- **The activities and resources should be user friendly and easy to use.**
- **Proper planning is needed to increase academic achievements depending on the systems with DL and for the students to perform the expectations.**
- **The instructor should adopt his/her roles in the application and to carry out the roles well.**
- **Encouraging students to continue on this new way of studying**
- **Observing feedbacks of the students and enhancing instructor support.**

This study addresses that; face-to-face learning is more permanent then blended learning for learning programming. Some new ways are needed to provide permanence for courses including intensive cognitive activities in blended settings. One way to realize this might be presenting information by coding it with dual coding Smart & Cappel (2006) in DL setting. Again, the development of a complete programming knowledge may be delivered by interactive simulations, bearing students' needs in mind as much as possible.

On the other hand; it is important to construct equilibrium between DL and face-to-face environments.

Thus in blending programming lab sessions become very important. Deficient information on DL must be completed in lab sessions. In order for consolidation in synchronous settings and for various solutions, specific activities are needed for lab sessions. There are not any fixed ratios of synchronous and asynchronous in blended learning that may be considered to be correct or incorrect. The main aim is to advance the learning experience by using a blend of face-to-face and DL environments.

However, for courses such as introductory programming, DL courses are expected to have quality to enable interaction as much as possible.

Finally, as Nichols (2003) emphasizes just providing educators with a mix of face-to-face learning and information technologies, may not present the desired effect, if the underlying blended learning model does not rely on learning theories. Currently, however, blended learning solutions to the teaching programming are still not enough. Thus there is more to be investigated on blending introductory programming, especially the cognitive processes during learning programming.

BIODATA and CONTAC ADDRESSES of the AUTHOR

Dr. Unal CAKIROĞLU is currently an assistant proffesor at the Department of Computer and Instructional Technologies at Karadeniz Technical University. His academical qualifications were in Computer Engineering (B. Sc.), in Computer Engineering (M.S) and Technology of Education (PhD.). He teaches computer programming, operation systems, and other IT related courses. His research interests include e-learning applications, social networking in education, technology integration and distance education.

Asist. Prof. Dr. Unal CAKIROGLU,
Department of Computer Education and Instructional Technology
Karadeniz Technical University, Fatih Faculty of Education,Trabzon,TURKEY
Phone: +90 4623777113
Email: cakiroglu@ktu.edu.tr

REFERENCES

Aycock, A., Garnham, C., & Kaleta, R. (2002). Lessons Learned from the Hybrid Course Project. *Teaching with Technology Today*, 8(6), 1-6.

Barg, M., Fekete, A., Greening, T., Hollands, O., Kay, J., Kingston, J. H., & Crawford, K. (2000). Problem-based learning for foundation computer science courses, *Computer Science Education*, 10, 2, 109-128.

Bayman, P., & Mayer, R.E. (1988). Using conceptual models to teach basic computer programming. *Journal of Educational Psychology*, 80, 291-298.

Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45–73.

Berk, R. (2009), Teaching strategies for the net generation. *Transformative Dialogues,Teaching & Learning Journal,* 3(2).

Bliuc, A., Goodyear, P., & Ellis, R. (2007). Research focus and methodological choices in studies into students' experiences of blended learning. *Internet and Higher Education*, 10, 231-244.

Brown, R. (2003). Blending learning: Rich experiences from a rich picture. *Training and Development in Australia*, 30 (3), 14-17.

Buchanan, E. A. (2000). Assessment Measures: Pre-tests for Successful Distance Teaching and Learning? *Journal of Distance Learning Administration*, 2 (4).

Carlson, R. & Repman, J. (2000). Chalk dust from the virtual classroom: Building that human touch into your web-based course. *WebNet Journal*, 2(3), 9-11.

Chickering, A. W., & Gamson, Z. F. (1987). Seven principles for good practice in undergraduate education. *AAHE Bulletin*, 39(7), 3–7.

Dagdilelis, V., Satratzemi, M., & Evangelidis, G. (2004). Introducing secondary education to algorithms and programming. *Education and Information Technologies*, 9 (2), 159-173.

Dodero, J.M., Fernández, C. & Sanz, D. (2003). An experience on students' participation in blended vs. online styles of learning. *SIGCSE Bulletin*, 35(4), 39–42.

Dunican, E. (2002). Making the analogy: Alternative delivery techniques for first year programming courses. Proceedings from the 14th Workshop of the Psychology of Programming Interest Group, Brunel University, 89-99.

Ersoy, H. (2003). Blending Online Instruction With Traditional Instruction in The Programming Language Course: A Case Study, Doctoral Dissertation, The Middle East Technical University, Ankara.

Frank, M., Reich, N., & Humphreys, K.(2003), Respecting the human needs of students in the development of e-learning, *Computers and Education*, 40 (1),57-70.

Hadjerrouit, S. (2007). A blended learning model in java programming: A design-based research approach, Proceedings Computer Scince IT Education Conference, 283–308.

Hagan, D.L. & Macdonald, I.D.H. (2000). A collaborative project to improve teaching and learning in first year programming. *Australasian Journal of Engineering Education*. 9,1.

Jonassen, D., Beissner, K., & Yacci, M. (1993). Structural Knowledge, Erlbaum, Hillsdale, NJ.

Kaleta, R., Skibba, K., & Joosten, T. (2007). Discovering, Designing and Delivering Hybrid Courses. In:Picciano, A. and Dziuban, C. (Eds.), Blended Learning: Research Perspectives, Needham, MA: Sloan Center for Online Education (SCOLE), 111-143.

Kenney, J. & Newcombe, E. (2011). Adopting a Blended Learning Approach: Challenges Encountered and Lessons Learned in an Action Research Study, *Journal of Asynchronous Learning Networks,* 15 (1), 8-19.

King, K. P. (2002). Identifying success in online teacher education and professional development. *Internet and Higher Education*, 5(3), 231-240.

Light, P., Nesbitt, E., Light, V., & White, S. (2000). Variety is the spice of life: Student use of CMC in the context of campus based study. *Computers and Education*, 34 (3–4), 252-267.

Massoud, A., Iqbal, U., Stockley, D. & Noureldin,A. (2011). Using Blended Learning to Foster Education in a Contemporary Classroom, *Transformative Dialogues: Teaching & Learning Journal*, 5 (2),1-11.

Nichols, M. (2003). A Theory for ELearning. *Educational Technology and Society*, 6(2), 1 10.

Picciano, A.G. (2006).Blended Learning: Implications for Growth and Access. *Journal of Asynchronous Learning Networks*, 10(3), 95-102.

Rovai, A. P. & Jordan, H. M. (2004). Blended learning and sense of community: A comparative analysis with traditional and fully on-line graduate courses. *The International Review of Research in Open and Distance Learning*, 5(2)

S. Djenic, R. Krneta, and J. Mitic. (2011). Blended Learning of Programming in the Internet Age, *Ieee Transactions on Education*, 54 (2), 247-254.

Smart, K., & Cappel, J. (2006). Students' Perceptions of Online Learning: A Comparative Study. *Journal of Information Technology Education*, 5, 201-219.

Stacey, E., & Gerbic, P. (2007). Teaching for blended learning - Research perspectives from on-campus and distance students. *Educational and Information Technologies*, 12(3), 165-174.

Tabor, S. (2007). Narrowing the distance: Implementing a hybrid learning model for information security education. *The Quarterly Review of Distance Education*, 8(1), 47–57.

Vaughan, N. (2007). Perspectives on Blended Learning in Higher Education. *International Journal on Elearning*, 6 (1), 81-94.

Williams, L. A. , & Kessler, R. R. (2000). Effects of 'Pair-pressure' and 'Pair-learning' on Software Engineering Education, *The 13th Conference on Software Engineering Education and Conference,* IEEE Computer Society, Austin, TX, 59 - 65.