THE EFFECT OF FLIPPED CLASSROOM AS A TEACHING STRATEGY ON UNDERGRADUATE STUDENTS' SELF-EFFICACY, ENGAGEMENT AND ATTITUDE IN A COMPUTER PROGRAMMING COURSE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CEMİL YURDAGÜL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER EDUCATION AND INSTRUCTIONAL TECHNOLOGY

SEPTEMBER 2018

Approval of the thesis:

THE EFFECT OF FLIPPED CLASSROOM AS A TEACHING STRATEGY ON UNDERGRADUATE STUDENTS' SELF-EFFICACY, ENGAGEMENT AND ATTITUDE IN A COMPUTER PROGRAMMING COURSE

submitted by **CEMİL YURDAGÜL** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Education and Instructional Technology Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Soner Yıldırım
Head of Department, **Comp. Edu. and Inst. Tech.** ________________

Assist. Prof. Dr. S. Tuğba Tokel
Supervisor, **Comp. Edu. and Inst. Tech., METU** ________________

Examining Committee Members:

Assist. Prof. Dr. İsmail Yıldız
Comp. Edu. and Inst. Tech., Kastamonu University ________________

Assist. Prof. Dr. S. Tuğba Tokel
Comp. Edu. and Inst. Tech., METU ________________

Prof. Dr. Soner Yıldırım
Computer Edu. and Inst. Tech. Dept., METU ________________

Assoc. Prof. Dr. Ayfer Alper
Computer Edu. and Inst. Tech. Dept., Ankara University ________________

Assoc. Prof. Dr. Ömer Delialioğlu
Computer Edu. and Inst. Tech. Dept., METU ________________

Date: 07.09.2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Cemil Yurdagül

Signature :

ABSTRACT

THE EFFECT OF FLIPPED CLASSROOM AS A TEACHING STRATEGY ON UNDERGRADUATE STUDENTS' SELF-EFFICACY, ENGAGEMENT AND ATTITUDE IN A COMPUTER PROGRAMMING COURSE

Yurdagül, Cemil

Ph.D., Department of Computer Education and Instructional Technology

Supervisor: Assist. Prof. Dr. S. Tuğba Tokel

September 2018, 152 pages

Learning programming skills is of crucial importance in today's world. However, there are some difficulties faced in teaching these skills. Flipped classroom has been used as a teaching approach to overcome difficulties in teaching programming skills. Flipped classroom approach provided students with a practice-involved learning setting where they were exposed to previously prepared video course content prior to attending the classroom element of the course. The purpose of this study was to investigate the effect of the use of the Flipped Classroom Approach on students' self-efficacy, engagement, and attitudes in a computer programming course. In this regard, an Introduction to Programming course was designed in which students were taught by the Traditional Classroom Approach during the first five weeks, followed by the Flipped Classroom Approach for the subsequent five weeks. The sample of the study consisted of 35 university students who participated in an introductory programming course. A mixed-methods research design was adopted in the study. Students' scores on self-efficacy in programming, general classroom engagement, and attitudes toward programming languages questionnaires were analyzed in order to determine the effect of the Flipped Classroom Approach. Students' subjective thoughts and views about the use of the Flipped Classroom Approach derived from semi-structured interviews were analyzed so as to gain comprehensive understanding of the Flipped Classroom Approach.

The results of the quantitative data analysis revealed that the flipped classroom positively affected students' self-efficacy in programming within the dimension of Programming in Complex Programming tasks, General Classroom Engagement in the dimension of Behavioral and Emotional Engagement, and Attitudes Toward Programming Languages in the dimension of Self-confidence. The results of the qualitative data analysis indicated that the students were satisfied with the use of the Flipped Classroom Approach in their programming course. Furthermore, the students considered the flipped classroom as an advantageous approach in terms of allowing them to be better prepared for the course to be conducted, giving them the opportunity to go back to look at course content through viewing course videos, and through involving them in active classroom participation.

Keywords: Flipped Classroom, Programming Education, Self-efficacy, Classroom Engagement, Attitude

ÖZ

PROGRAMLAMAYA GİRİŞ DERSİNDE BİR ÖĞRETİM STRATEJİSİ OLARAK KULLANILAN TERS-YÜZ EDİLMİŞ SINIF YAKLAŞIMININ, ÜNİVERSİTE ÖĞRENCİLERİNİN ÖZ YETERLİLİKLERİNE, KATILIMLARINA VE TUTUMLARINA OLAN ETKİSİ

Yurdagül, Cemil
Doktora, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü
Tez Danışmanı: Dr. Öğr. Üyesi. S. Tuğba Tokel

Eylül 2018, 152 sayfa

Günümüz dünyasında programlama becerilerinin öğrenilmesi büyük önem taşımaktadır. Ancak, bu becerilerin öğretilmesinde bazı zorluklar vardır. Ters-yüz edilmiş sınıf yaklaşımı farklı alanlarda giderek yaygınlaşan bir eğitim yaklaşımı olarak ortaya çıkmıştır. Bu yaklaşımda, dersin içeriği öğrencilere derse gelmeden önce video formatında verilir. Ders sürecinde ise çoğunlukla uygulamaya ağırlık veren bir ortam sağlanmaktadır. Bu çalışmanın amacı, ters-yüz edilmiş sınıf yaklaşımının programlama eğitiminde kullanılmasının, öğrencilerin öz-yeterlikleri, katılımları ve tutumları üzerine olan etkisini araştırmaktır. Bu amaç doğrultusunda, çalışmada, öğrencilere programlamaya giriş dersinin ilk 5 haftasında geleneksel sınıf yaklaşımı ile öğretim gerçekleştirilmiş, sonraki 5 haftasında ise, ters-yüz edilmiş sınıf yaklaşımı ile öğretim gerçekleştirilmiştir. Çalışmanın örneklemini, programlamaya giriş dersine katılan 35 üniversite öğrencisi oluşturmaktadır. Bu çalışmada karma araştırma yöntemi benimsenmiştir. Araştırmanın nicel aşamasında, ters-yüz edilmiş sınıf yaklaşımının, öğrencilerinin programlamaya karşı öz yeterliliklerine, genel ders katılımlarına ve programlama dillerine karşı tutumları açısından etkisi olup olmadığını anlayabilmek için yarı-deneysel tasarım kullanılmıştır. Çalışmanın nitel aşamasında ise, öğrencilerin ters-yüz edilmiş sınıf yaklaşımına karşı görüşlerini anlayabilmek için yarı yapılandırılmış görüşmeler gerçekleştirilmiştir.

Çalışmanın nicel verilerinin analizi sonucunda, ters-yüz edilmiş sınıf yaklaşımının öğrencilerin programlamaya karşı öz yeterliliklerine karmaşık programlama görevlerini gerçekleştirme boyutunda, genel ders katılımlarına davranışsal katılım ve duygusal katılım boyutunda ve programlama dillerine karşı tutumları açısından ise özgüven boyutunda olumlu etkisi olduğu ortaya çıkmıştır. Nitel verilerin analizi sonucunda ise, öğrencilerin ters-yüz edilmiş sınıf yaklaşımının programlama dersinde kullanılmasından memnun olduklarını göstermiştir. Ayrıca, öğrenciler, derslere daha hazır bir şekilde gelme, videolar aracılığıyla içerikleri tekrar etme ve sınıf içerisinde aktif olma konularında ters-yüz edilmiş sınıf yaklaşımının kendilerine büyük bir avantaj sağladığını ifade etmişlerdir.

Anahtar Sözcükler: Ters-Yüz Edilmiş Sınıf, Programlama Eğitimi, Öz Yeterlilik, Sınıf Katılımı, Tutum

TO EVERYONE WHO HAS SUPPORTED ME,

WITHOUT WHOM NONE OF MY SUCCESS WOULD HAVE BEEN POSSIBLE.

ACKNOWLEDGMENTS

Foremost, I would like to express my special appreciation and thanks to my advisor, Assist. Prof. Dr. S. Tuğba Tokel and thesis monitoring committee members, Assoc. Prof. Dr. Ayfer Alper and Assoc. Prof. Dr. Ömer Delialioğlu, for supporting and guiding me throughout this thesis.

I would like to thank my thesis defense committee members, Prof. Dr. Soner Yıldırım and Assist. Prof. Dr. İsmail Yıldız, for their insightful comments and encouragement.

My sincere thanks go to Erkan Hürnalı, who gave access to his classroom for the research of this thesis.

I am immensely indebted to my esteemed and devoted instructors for their contributions to my personal and professional development from my undergraduate days to the present.

I would like to thank my friends and colleagues, Zafer Kadırhan, Mustafa Sat, Saba Öz, and Enver Yılmaz, who never ceases to give me their feedback, cooperation and friendship.

My acknowledgment would be incomplete without thanking the biggest source of my strength, my family. I would like to thank my dear mother, Seher Yurdagül, my precious father, Rafet Yurdagül, and my brother, Ceyhun Yurdagül, for their continuous love, caring, support and sacrifice. I deeply thankful to my beloved wife Nevin Yurdagül for her understanding and support in this long journey.

The end of one journey is the beginning of the next. I would like to express my gratitude to everyone who has provided their contributions on this long journey.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

FC	Flipped Classroom
FCA	Flipped Classroom Approach
TC	Traditional Classroom
TCA	Traditional Classroom Approach
CPSES	Computer Programming Self Efficacy Scale
GCES	General Classroom Engagement Scale
ATCPLS	Attitudes Towards Computer Programming Languages Scale
CEIT	Computer Education and Instructional Technology

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

For the 21st century, the skills that students should gain include critical thinking, problem-solving, performing analysis and synthesis, working collaboratively, being innovative and productive, and accessing accurate and up-to-date information with ease (Yükseltürk & Altıok, 2015). One of the aims of gaining these skills is to enable students to find employment easier once their business life starts after completion of their formal education. When the employee characteristics required by today’s employers are examined, they indicate the aforementioned skills as those needed from employees in the 21st century. As a result of the 21st Century Student Profile (Milli Eğitim Bakanlığı [Ministry of National Education], 2011) which was research conducted by the Turkish Ministry of National Education (MoNE), the fact that students are not being adequately educated in order to become qualified personnel as required by today’s employment market is among the major problems stated in the report. In addition, teachers stated that the educational system could not provide students with sufficient critical thinking and problem-solving skills. The programming course is seen as means to enable students to acquire these required skills (Akpinar & Altun, 2014). For example, when students develop a program, they need to search for a solution to a problem and then put that solution into an algorithm. These problem solving and algorithm development applications contribute positively to the student’s ability to solve problems, and to analyze and synthesize them. Research shows that there is a positive relationship between programming skills and problem-solving skills (Tu & Johnson, 1990). In the research carried out to date, it has been shown that programming education contributes to students' other skills besides the ability to solve problems. For example, in a meta-analysis study conducted by Liao and Bright (1991), it was indicated that students could gain reasoning skills, logical thinking and planning skills, and general problem-solving skills through programming course activities. These studies have shown that individuals with increased programming skills can also increase other cognitive skills, and that increases in such skills also contribute to their success in other areas where such skills are vital.

Recently, the software sector, as a rapidly growing area, has been attracting attention in terms of employment provided by other sectors. In the past, computer-programs only used to include the software installed on personal computers, but today it has become much more than that. It is frequently shared on the news that the value of social media platforms is equal or even higher than the GDP of some countries. For instance, when the market value of WhatsApp was announced as US$19 billion in 2014, that figure was higher than the net worth of some of the top 20 companies listed on the Istanbul Stock Exchange (*known as Borsa Istanbul*) (WhatsApp Türkiye, 2014). Examining the contribution of the software sector based on the example of the United States, it can be noted that the economic value of the state of Texas, which is famous for its oil production, is US$1.8 trillion and California, which hosts the world famous Silicon Valley, contributes US$3.6 trillion to the U.S. economy (Gunes, 2015). This clearly shows the leading position of the software industry in the U.S. economy alone. Moreover, reports indicate that the software industry contributes US$1.17 trillion directly and US$564.4 billion indirectly to the U.S. economy as well as providing job opportunities for 2.9 million employees directly and 10.5 million people indirectly (BSA The Software Alliance, 2017). Similar to the United States, there are other countries such as Ireland, India, and Israel that make a huge contribution to their country's economy through the by exporting of computer software. It is estimated that the world software market is US$4.8 trillion and it is worth considering that if Turkey's software exports amounted to just 1% of this worldwide figure, that would amount to an export of US$50 billion (Gunes, 2015). When we consider that the pie is so big, even 1% of that share would be of great benefit in terms of working towards achieving Turkey's future development goals.

From the industry perspective, the prominence of programming skills is clearly noticeable. In the industrial sector, those hailing from generations skilled at programming adapt more easy to new technologies and are able to use them more immediately (Sener & Elevli, 2017). Hanus, Marulo, and Bauer (2017) pointed out that the new fields of study including robotics, artificial intelligence, nanotechnology, biotechnology, Internet of Things (IoT), 3D printing, and autonomous vehicles emerged with the introduction of Industry 4.0. Considering the basics of these fields, it seems that all of them fundamentally require skills in programming languages. It is also emphasized that the need for qualified staff to work should be increased in these fields (Yasad, 2009). It is critical that software developers, programmers, and analysts in the software industry are trained to be adequately qualified in order to increase the competitive power of the software industry (Demirer & Sak, 2016; Yasad, 2009). Taking a look at the skills required from manpower to work in this sector, employees should be well-educated individuals who possess analytical skills and the ability to write computer programs. In today's world, it is expected that programming skills across all fields will become more important than ever (Sayın & Seferoğlu, 2016).

It is often pointed out that students worldwide should learn programming at an early age. Countries such as India, Singapore, and Canada have been providing programming training in preschool programs. In Turkey, measures are being taken within the scope of this requirement, although not yet at the desired level (Akpinar & Altun, 2014). For example, since 2013, a software course has been taught as compulsory at some levels. In order for these individuals to be educated, Turkish education and training institutions bear tremendous responsibility. Especially in the future, in order to become important to Turkey it can be foreseen that departments, course areas, and training experts who will teach programming languages will need to provide generations with programming skills in order to meet the needs of industry and the software sector.

Although it is clearly understood that learning programming is important and that the introduction to programming course that involves the teaching of program writing skills is a key course in Turkey, students are facing a number of difficulties (de Jesus Gomes, Mendes, & Marcelino, 2015). Research shows that the course is perceived as complex and difficult (Helminen & Malmi, 2010; Siti Rosminah & Ahmad Zamzuri, 2012). An indicator that points to the problems in the course's learning is that the rate of dropout is high at 20-40% (Kinnunen & Malmi, 2008).

So, why are programming language courses perceived as so difficult to learn? When the reasons behind this are examined, it is seen that the reasons reported in the literature relate to the content of the course, the student's background, and the course teaching methods.

The first reason is due to the content of the course. One of the reasons stated in the literature is that some topics covered by the course are abstract and thereby difficult to understand (Ismail, Ngah, & Umar, 2010; Lahtinen, Ala-Mutka, & Järvinen, 2005). In some studies, these topics were named as "abstract methods, recursion, and pointers" (Lahtinen et al., 2005; Siti Rosminah & Ahmad Zamzuri, 2012). Hence, students face problems in comprehending these topics through abstract lectures. In order to improve this situation, studies on visualization methods are being conducted (Yukselturk & Altiok, 2017).

The second reason is that of the students' background. It is stated that students perceive this course as difficult because a lack of relevant background and that this affects their achievements and may be a reason for dropping out from the course. Studies in the literature show that students' lack of problem-solving, analytical thinking, logical deduction, and algorithm skills affects their achievements in this subject area (Ismail et al., 2010). Moreover, books and course materials aimed directly at teaching programming languages without focusing on improving and tackling these potential problems are also stated as reasons for failure (Ismail et al., 2010). It is known that the attitude of the student towards a course can be the cause of success or failure (Baser, 2013; Giannakos, Pappas, Jaccheri, & Sampson, 2017). While it is evident that course dropout rates decrease when students perceive the course as beneficial to themselves (Giannakos et al., 2017), having negative attitudes towards a course cannot be denied as being among the key reasons for course dropouts (Korkmaz & Altun, 2014).

The third reason seen for difficulties in learning the programming course is due to reasons related to the teaching methods (de Jesus Gomes et al., 2015). Since the course has an intensive level of content, it has also been pointed out that there is a problem associated with restricted time with the course (de Jesus Gomes et al., 2015; Siti Rosminah & Ahmad Zamzuri, 2012). In addition, when the courses are taught through direct lecturing, students' chances of putting theory into practice are lowered and therefore they cannot ultimately benefit from the course (Siti Rosminah & Ahmad Zamzuri, 2012). Students and teachers indicated that learning environments such as the laboratories in which they are able to practice are significant (Lahtinen et al., 2005; Siti Rosminah & Ahmad Zamzuri, 2012). Therefore, designing the course environments in a way that increases convenience to facilitate practice time may be a factor that would enhance students' success on the course (Lahtinen et al., 2005). They also stated that students understood the topics in the lessons, but when they encountered a problematic situation, they experienced difficulties in writing the actual code (Lahtinen et al., 2005). This means that they generally do not have problems with learning the basics, but that they face obstacles where they need to practice, develop algorithms, and write code (Özmen & Altun, 2014). All of these findings can be categorized under why practice should be afforded greater significance in the teaching of this course. Additionally, the lessons should be student-centered and that students should be given the chance to practice instead of being exposed to purely direct lectures throughout the course. However, another obstacle to the realization of all these solutions is the courses' time limitation (de Jesus Gomes et al., 2015). For this reason, researchers are seeking a more efficient and student-centered teaching methodology with the emphasis placed on increased practice.

When we consider the literature on programming education, it can be seen that different teaching methods and tools have been used in order to solve problems in programming classes. Algorithm visualization tools, robotic programming tools, web-assisted education, computer-aided education, educational games, gamification and pair programming are just some of these methods to be found in the literature. In the current study, the Flipped Classroom, which has emerged as a relatively new teaching approach, has a structure that changes the way courses are taught. It is an approach that suggests transferring what takes place in the traditional classroom to outside of the classroom and vice versa. While the learning part of the traditional approaches takes place in the classroom, homework is then set in order to reinforce what has been learned in the classroom. Considering the advantages of the Flipped Classroom Approach (Bergmann & Sams, 2012); that it makes classes more student-centered (O'Flaherty & Phillips, 2015), makes time spent in the classroom more efficient (Estes, Ingram, & Liu, 2016), and gives students the chance to learn on their own, it brings up the question as to whether or not this approach would prove useful in the teaching of computer programming. The purpose of the current study is to seek answers to this fundamental question.

1.2 Purpose of the Study

This study aims to examine the Flipped Classroom Approach (FCA) and Traditional Classroom Approach (TCA) by comparing the mean scores of undergraduate students' computer programming self-efficacy, classroom engagement, and attitudes toward computer programming on a computer programming course taught within each respective environment. In addition, it also aims to reveal the opinions of the students who have experienced the Flipped Classroom Approach.

1.3 Significance of the Study

Studies have shown that general reasoning skills, logical thinking and planning skills, and general problem-solving skills, which are seen as 21st century skills, develop in proportion to programming skills. For this reason, increasing students' skills in programming courses would contribute positively to 21st century skills acquisition. The current study will contribute to the programming education literature in order to improve programming education by implementing a new teaching approach in the teaching of computer programming.

It is known that companies in the software industry contribute significantly to the economies of many countries. In terms of achieving sufficient export figures in line with the development targets of Turkey, it is of higher priority to advance the software industry (Gunes, 2015). Looking at the characteristics of manpower requirements to work in the software sector, they need to be individuals in possession of analytical thinking and program development skills. The basic level course, in which the beginner level objectives are taught at universities, is the Introduction to Programming course. However, the course is perceived as difficult and complex by the students (Helminen & Malmi, 2010; Siti Rosminah & Ahmad Zamzuri, 2012). The reason for this is related to the course itself having an abstract structure by its very nature. Additionally, another reason is that students lack the necessary pre-course knowledge, problem-solving skills, and mathematical background (Ismail et al., 2010). Problems are also associated with the teaching instruction and approach to the course (de Jesus Gomes et al., 2015). The need for replacing the teacher-centric approach of traditional methods with the newer student-centered method has been boldly underlined in studies regarding the teaching of computer programming. The current study aims to contribute to the literature by researching the effects of the advantages of the Flipped Classroom Approach including student-centeredness, active participation, and the more efficient use of classroom time on teaching programming (Bergmann & Sams, 2012).

With regards to the flipped classroom, there are numerous quantitative studies to be found in the literature as well as suggestions about the need to research the qualitative aspect (Al-Ghamdi & Al-Bargi, 2017; Martin, 2015). The current study, through its qualitative and quantitative methodological approach, aims to contribute to the literature on the flipped classroom in this respect. Besides, there appears to be a lack of studies published on the effects of the flipped classroom on student attitude (Esperanza, Fabian, & Toto, 2016). Within the literature review of another study, it was suggested that research should also be conducted with regards to the student engagement dimension (O'Flaherty & Phillips, 2015). Therefore, the current study is aimed at making a contribution to the flipped classroom literature within the dimensions of student engagement and attitude.

1.4 Research Questions

In line with the aims of the study, the following research questions will be examined throughout the study;

1. Is there a significant difference in Computer Programming Self Efficacy mean scores between undergraduate students taught with the Flipped Classroom Approach and Traditional Instructional Approach?
    a. Is there a significant difference in Basic Programming component scores of the Computer Programming Self Efficacy Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    b. Is there a significant difference in Complex Programming component scores of the Computer Programming Self Efficacy Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

2. Is there a significant difference in General Class Engagement mean scores between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    a. Is there a significant difference in Cognitive Engagement component scores of the General Class Engagement Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    b. Is there a significant difference in Behavioral Engagement component scores of the General Class Engagement Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    c. Is there a significant difference in Emotional Engagement component scores of the General Class Engagement Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

3. Is there a significant difference in Attitudes toward Computer Programming Languages mean scores between undergraduate students taught with the Flipped classroom Approach and the Traditional Instructional Approach?
    a. Is there a significant difference in Importance component scores of the Attitudes toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

b. Is there a significant difference in Enjoyment component scores of the Attitudes toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
c. Is there a significant difference in Self-confidence component scores of the Attitudes toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
d. Is there a significant difference in Motivation component scores of the Attitudes toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

4. What are the students' experiences and opinions about the Flipped Classroom Approach as a teaching strategy?

1.5 Limitations

1. The study is limited to a ten-week duration.
2. The study is limited to participant groups.
3. The study is limited to convenience sampling method.
4. Validity is limited to the sincerity of the responses given by participants to questions in the research instruments.

1.6 Definition of Terms

This part includes the definitions of terms used throughout the study.

Programming: It is the process of taking an algorithm and encoding it into a notation, a programming language, so that it can be executed by a computer (Miller & Ranum, n.d.).

Flipped Classroom: It is an innovative teaching approach in which the basic topics are learned before attending class by means of video or similar multimedia technologies, and the allocation of classroom time to undertake practice.

Traditional Classroom: It is the teaching approach in which topics are mainly covered in the classroom, and homework is given to put the theory into practice.

Student-centered Learning: The focus of the teaching is directed away from the teacher to the students, with the aim of this type of learning including student autonomy and independence ("Student-centred learning", n.d.).

Self-efficacy: It refers to a reflective process in which an individual develops a belief in one's ability to perform tasks and adopt change (Bandura, 1986).

Attitude: It can be defined as generally a positive or negative emotional disposition (Haladyna, Shaughnessy, & Shaughnessy, 1983; McLeod, 1992).

Classroom Engagement: It is defined as "constructive, enthusiastic, willing, emotionally positive and cognitively focused participation with learning activities in the classroom" (Skinner & Pitzer, 2012, p. 22).

1.7 Conclusion

In this part, the importance of 21st century skills and its relation to the programming course in Turkey was discussed and the importance of the programming course mentioned. Next, the advantages of the Flipped Classroom Approach in teaching computer programming were included. In the following part, studies conducted on programming education, difficulties in teaching, and flipped classroom in the literature are discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter mainly focuses on programming education. Then, studies on the difficulties in teaching and learning programming languages are discussed. Additionally, the definition of Flipped Classroom (*hereinafter referred to as FC*) and its development in recent times are analyzed. Finally, what benefits FC can provide to the teaching environment and the summaries of recent studies on FC are included.

2.2 Programming

Programs are part of software that are designed for the execution of a specific purpose (Colkesen, 2017). Software, on the other hand, refers to the files created for various purposes on computers or similar systems in order to execute any specific work using any programming language (Colkesen, 2017). For example, Notepad on a personal computer is enabled running through software, while code written to create that software is termed as being the program. This process of creating code snippets is referred to as programming.

In a general sense, even if computers come first to the mind for either programming or software, it can be thought of as teaching any electronic device, which is a processor in programming, how it should operate. These devices in every aspect of our life serve us by fulfilling what is taught to them, from smartphones to the refrigerators in our kitchens.

The software was categorized into five groups by Colkesen (2017) according to their purposes.

- Scientific & engineering software
- Business software
- Artificial intelligence software
- Image related software
- System software

This categorization shows that programming affects different areas of human life. The programming languages used while creating software can also be classified differently within themselves.

2.3 Programming Education

As the importance of programming education is understood, a consensus is forming on its inclusion within school curriculums from the primary school level and is increasing by the day. In the meta-analysis study conducted by Liao and Bright (1991), it was suggested that students can gain the skills of reasoning, logical thinking and planning skills, and general problem-solving skills through programming course activities. It is thought that programming education can contribute to what are termed 21st century skills when it is considered that programming education contributes to students' cognitive skills. Sayın and Seferoğlu (2016) presented a parallel idea by saying that programming was seen as part of reasoning and that it was accepted as one of the 21st century skills.

When the knowledge which needs to be taught in programming courses is considered, it can be divided into language-specific constructs and constructs common to programming languages. Furthermore, Bayman and Mayer (1988) distinguished three basic groups of knowledge that need to be learned. Those being syntactic knowledge, conceptual knowledge, and strategic knowledge.

Syntactic knowledge: It expresses that each programming language has its own syntax. For example, in some languages, a semicolon (“;”) is placed at the end of every code block, whilst in others it is not. It is the collection of information that contains language-specific rules that need to be learned.

Conceptual knowledge: This kind of information usually involves learning the common structures of a programming language. It contains conditional statements, loops and the definition of variables and their usage. Also it includes knowledge related to understanding how previously written code works.

Strategic knowledge: This type of information includes the ability to find solutions to problems through the application of syntactic and conceptual knowledge.

In traditional programming education, while there was programming instruction given in the laboratory and through classroom lecturing, new teaching methods started to be used in programming education with the development of new technologies and new teaching approaches. One frequently used method is the application of visualization tools, which are subdivided into algorithm visualization and program visualization (Çetin & Top, 2014). These tools are used to overcome difficulties in the abstract structure of programming. Algorithm visualization can visualize an algorithm, the code block. In program visualization, it can visually present the result of an entire program. Apart from these, other tools are included in programming education that can display written code as a 3D story.

Robotic programming tools are also used in programming education. For example, the Arduino may be an example of tools used for such a purpose. It has been suggested that concretization features such as in Arduino can be used in solving the difficulties caused by the abstract nature of programming languages (Ersoy, Madran, & Gülbahar, 2011). Therefore, students are able to understand what the code they write can physically do. In the past, robotic tools were used in programming education such as Lego Mindstrom kits. For example, in the work of Barnes (2002), it was applied within an introductory programming lesson and it was stated that it enables students to undertake physical programming without the need for specific hardware knowledge.

When the literature in programming education is examined, it is seen that computer-aided instruction, web-based instruction, and blended learning approaches have been utilized. Attempts have been made to teach programming by using multimedia teaching materials prepared for lessons in a computer environment. For example, a study conducted by Ünal and Bay (2009) can be presented as an example of computer-aided instruction. In the study, Java programming language teaching was performed using educational software that consisted of 10 parts developed with Authorware for a programming course. On the other hand, Cabi's (2004) study may be shown as a significant example of web-based instruction. In his study, a website was prepared for the teaching of Pascal, with exercises and question types used to interact with this site. As a result of the study, it was stated that web-supported teaching contributed to the learning of the students. In a study that employed the blended learning approach, Boyle, Bradley, Chalk, Jones, and Pickard (2003) aimed to increase the programming success of more than 600 students. The study's results showed that the student's rate of passing the course increased.

Educational games are among some of the methods used in programming education. Some games are intended to teach specific structures of programming, whilst others aim to teach only algorithms. When it is evaluated from the point of view of the coding language, it can be seen that some have their own coding language while others use languages from real life. In a study by Malliarakis, Satratzemi, and Xinogalos (2017), some educational games were selected based on certain criteria and then comparisons made between games that teach programming languages. These games were categorized in terms of objectives related to programming, and it was revealed that educational features of games such as storytelling, scaffolding and interactivity may be useful in terms of programming education.

In the literature, the pair programming method, which is used in program development, also takes its place in programming education (McDowell, Werner, Bullock, & Fernald, 2002). This method, in which two people work together in order to develop the same program, is used in programming courses as a collaborative approach to education. Additionally, gamification is included in programming education literature as a teaching strategy in programming courses (Kaila, Laakso, Rajala, & Kurvinen, 2018). Intelligent Tutoring System applications are also used in programming education. For example, a web-supported tutoring system was developed in a study conducted by Butz, Hua, and Maguire (2004) as a system which provides appropriate course material, learning targets and a reading order for students through artificial intelligence support. As a result of the study, it is understood that Bayesian network can be used in web intelligence.

2.4 Challenges in Programming Education

Although programming is seen as the most basic skill in the field of computer science, the learning of programming is seen as a very difficult skill (Bennedsen & Caspersen, 2007; Helminen & Malmi, 2010; Ismail et al., 2010). Additionally, whilst programming is a necessary and compulsory course in the curricula of science, mathematics, and engineering departments, it is still considered as being both challenging and complex (Siti Rosminah & Ahmad Zamzuri, 2012).

Various studies have been conducted in order to investigate the reasons behind this perceived difficult and complex structure. Siti Rosminah and Ahmad Zamzuri (2012) researched the problems and challenges faced by 105 engineering students of a Fundamental Programming course. The students stated that the most difficult topics for them were those that were abstract. Besides, they added that they had experienced difficulty in producing solutions when faced with problems. It was also stated that the students did not learn from an adequate number of examples in their lectures and that laboratory work was very useful in their learning of programming.

Ismail et al. (2010) suggested that the reasons students encounter difficulties in learning programming, based on the studies in the literature, were as follows:

- Lack of problem-solving skills
- Lack of analytical thinking skills
- Lack of logical and reasoning skills
- Lack of programming planning
- Lack of programming conceptual understanding
- Lack of algorithmic skills

Ismail et al. (2010) also stated that the books used in programming instruction focus only on programming languages, and that algorithm development skills are neglected. Moreover, researchers in the field suggest that passive teaching sessions make it difficult for students to mentally understand. When we look at the characteristics of students who are successful, it is seen that those students are motivated to participate in class discussions and that they further discuss with both their teachers and peers regarding any problems they encounter. They also suggested when and why metacognitive skills are needed for programming and that their lacking in students was a barrier to their learning of programming.

There is a widespread argument that programming courses have a high dropout rate and a high student failure rate (Bennedsen & Caspersen, 2007). In a study conducted by Bennedsen and Caspersen (2007) to investigate whether or not the high rate of failing was really the case, data were collected from different regions. It was observed that the percentage of failures from programming courses was between 20% and 40%. Furthermore, different studies were conducted to investigate the causes of these dropout rates, including one by Giannakos et al. (2017) with the participation of 344 computer science students. The researchers mainly focused on the factors and barriers that provide for student retention (intention to complete their studies). According to the results of the study, it was found that the degree of usefulness of the qualification being studied positively impacted on student retention. In addition, cognitive gains and supportive environmental elements had positive effects on a degree's usefulness, while non-cognitive gains prevented it. Another finding was that feelings (personal values) negatively influenced student retention. In general, the study's findings showed that cognitive gains, non-cognitive gains (negative impact), a supportive environment, the usefulness of the degree being studied for, and the students' personal values (e.g., lack of belongingness in the CS field) were perceived as significant for retention in studies about CS. In another study conducted about the problems faced by CS students in the programming courses, Kinnunen and Malmi (2008) focused specifically on programming courses and 459 students who passed the course and another 119 students who failed. As a result of the study, the subjects that were reported as difficult for the students were found to be inheritance and abstract classes. As a result of factor analysis, there are five reasons given for dropout which are course arrangements, difficulty to understand course topics, time management and preferences, lack of consequences of dropping out, and effect of other courses. Özmen and Altun (2014) conducted another study to understand the difficulties in learning programming. In this study, they conducted a qualitative study to investigate the difficulties students experienced in the programming process and to investigate the causes of failures. As a result of interviews held with 12 students, the difficulties students experienced in the programming process included programming knowledge (syntax, functions and its parameters, defining variables, decision structures and loops), programming skills, comprehension, and debugging of the program. As for the reasons for failure, students listed the lack of practice, lack of algorithms, and lack of knowledge.

It is stated that specialization in programming takes approximately 10 years (Winslow, 1996). This shows that programming skills can be developed through experience. It was also found that difficulties in learning programming can vary between novice and expert learners. Robins, Rountree, and Rountree (2003) conducted a study examining previous research on psychology and education fields through a literature review and analyzed the differences between expert and novice learners. Their study revealed that novice programmers have only surface knowledge about programming, with knowledge basically limited to programming “line by line” instead of a holistic and meaningful approach. In another study, conducted by Lahtinen et al. (2005), 559 students and 34 teachers were administered international surveys. The results aimed to reveal the difficulties in the learning and teaching of expert and novice basic programming. As a result, it was found that the most difficult part of programming was seen as “understanding how to design a program to solve a certain task.” The most difficult topics were found to be recursion, pointers, and references; and both students’ and teachers’ opinions agreed on this point. The teachers suggested that the labs and practice sessions were the more useful, while students found sessions in which they worked on their own to be more useful in terms of learning. From the novice learners’ perspective, it was noted that they had problems in practice rather than the learning of basic concepts. The research findings recommended that practicing in order to deeply learn a topic and learning by doing should be among the teaching objectives (Lahtinen et al., 2005).

In general, the reasons that students experience difficulty in programming lessons include the abstract structure of the topics in programming (Ismail et al., 2010; Lahtinen et al., 2005), students’ previously held ideas about the course (Giannakos et al., 2017), students’ previous level of mathematics, problem solving and metacognitive skills (Ismail et al., 2010; Siti Rosminah & Ahmad Zamzuri, 2012), and the teaching methods that are employed in lessons (Lahtinen et al., 2005; Siti Rosminah & Ahmad Zamzuri, 2012). In addition, studies showed that the subjects that students perceived as difficult were “abstract methods,” “recursion” and “pointers” (Lahtinen et al., 2005; Siti Rosminah & Ahmad Zamzuri, 2012).

2.5 Self-efficacy in Programming

Bandura proposed self-efficacy as “people’s judgments of their capabilities to organize and execute courses of action required to attain designated types of performances” (Bandura, 1986, p. 391). He also proposed that student self-efficacy beliefs were an effective indicator of student success. Askar and Davenport (2009) also pointed to self-efficacy playing an important role in the field of education. They suggested that self-efficacy was influenced by the performance of learners and that this performance may affect their future performance. Besides, the effects of self-efficacy on success were studied in terms of different fields such as mathematics and English, as well as programming courses, and it was underlined that it is an important construct in the teaching of programming. Also, Nilsen and Larsen (2011) pointed to low self-efficacy and motivation as the main reasons behind students’ lack of understanding of this course and the algorithmic structure in terms of programming lectures. Additionally, Lishinski, Yadav, Good, and Enbody (2016), in their study about self-efficacy and its effects, found that one of the most important factors that determine performance in the Computer Science field is self-efficacy. Altun and Mazman (2012) and Baser (2013) argued that when students have low self-efficacy at the beginning of a programming course, this can increase the likelihood that they will fail the course. Based on these findings, it can be inferred that one of the reasons of the high rates of dropout in programming courses could be low self-efficacy.

Having reviewed the literature, it is seen that self-efficacy is considered an important factor for success and performance in programming courses, and that different methods are used to realize improvements (Korkmaz, 2016; Yukselturk & Altiok, 2017). For example, Yukselturk and Altiok (2017) examined the effects of using Scratch in programming lessons on students’ self-efficacy and attitudes towards programming. In the study, 151 prospective teachers were selected as the sample, and a mixed-method research was applied. The study’s results revealed that the sub-dimension of self-efficacy in complex programming tasks showed a meaningful increase in the programming environment where prospective teachers were taught programming through Scratch. On the other hand, Korkmaz (2016) examined the effects of Scratch-inspired game-based activities on students' self-efficacy beliefs and levels of academic achievement. 49 engineering students were included as the sample in their quasi-experimental study. As a result of the study, it was found that the Scratch-based game activities did not have a significant effect on the students' self-efficacy.

In summary, the literature showed that self-efficacy has an effect on learning performance. Similarly, the studies on programming education also point to a possible relationship between self-efficacy and success. For these reasons, self-efficacy is regarded as a construct to be significantly considered and investigated in teaching programming.

2.6 Engagement in Programming

Engagement is defined as the involvement of the student with positive feelings in learning activities (Skinner & Belmont, 1993). In educational research, engagement of learners has become an important field of study (Gunuc & Kuzu, 2015). While the term "engagement" can mean different things, in the context of the current study it is considered as classroom engagement. As indicated in the literature, engagement is directly related to the achievement and critical thinking of students in their school lessons (Carini, Kuh, & Klein, 2006; Pintrich & De Groot, 1990). In addition, Fredricks, Blumenfeld, and Paris (2004) suggested that the increase in school engagement could be a solution for students' failing academic motivation and achievement. Engagement is discussed in the literature in different dimensions, and the most common sub-dimensions are behavioral engagement, emotional engagement, and cognitive engagement.

Behavioral engagement: When students mentally engage in the behavioral norms, the expected outcome would be active attendance and involvement in the classroom discussions, thereby demonstrating the constructive and positive behavior (Trowler, 2010, p. 7).

Emotional engagement: Students with emotional engagement tend to experience affective reactions including interest in the lessons, enjoyment, or a sense of belonging (Trowler, 2010, p. 7).

Cognitive engagement: When it comes to cognitive engagement, students feel more inclined to invest in their own learning, take responsibility to go beyond the required, and equip themselves in order to challenge (Trowler, 2010, p. 7).

As in the other disciplines in the field of education, it was investigated whether or not engagement is important in the success of learning computer programming. Su, Ding, and Lai (2017) studied student engagement in a social media-supported learning environment through learner analytic methods on a computer programming course. In their study conducted with 43 students, a positive relationship was revealed between the students' engagement and their achievement. In another study, Scott et al. (2015) proposed that programming was difficult to learn and observed that student engagement was insufficient. To increase this, they provided students with a learning environment based on robotics programming within an Introduction to Programming course. This action resulted in the higher engagement of those involved in the robotics-centered learning environment. Engagement is important in terms of the participation of students in lectures. Hew (2016) attempted to reveal the factors affecting the participation and engagement of students in Massive Open Online Course (MOOC) environments, which have recently started to became more widespread. Three top-ranking MOOCs were selected for their study and 965 participants were selected as the sample. The MOOCs were mostly related to computer programming lessons. From this perspective, the results of his research contributed to the literature in terms of increasing engagement in programming lessons. As a result, five factors were found that could increase engagement:

- problem-centric learning with clear expositions
- instructor accessibility and passion
- active learning
- peer interaction

- using helpful course resources

Lindberg and Laine (2018) suggested that it was important for students to engage in programming lessons. To achieve this, 32 students were given game-based instruction while another 32 students were provided with handouts for the same educational content. As a result of their study, while the retention was seen as the same across the two groups, the engagement levels of those students in the gaming group were proven to be higher. Studies have shown that student engagement can therefore be a factor for academic success.

2.7 Attitude in Programming

In general, attitudes are defined as being either positive or negative emotional disposition (Haladyna et al., 1983; McLeod, 1992). Educators in the field agree that promoting attitudes is of significantly high importance in education (Sundre, Barry, Gynnild, & Ostgard, 2012). The attitude factor, which has an important place in mathematics and science education, is among the factors that can affect success in terms of programming education. According to Korkmaz and Altun (2014), one of the reasons that lays behind the problems seen in programming education might be students' negative attitudes toward programming courses (Korkmaz & Altun, 2014). It is also proposed that attitude is among the factors that could affect the success of programming lessons (Baser, 2013; Korkmaz & Demir, 2012). In addition, Korkmaz and Altun (2014) suggested that many researchers in the literature have shown that attitudes toward the course or instructor have an influence on students' academic achievement, self-confidence perceptions, self-sufficiency, and satisfaction. Therefore, they pointed out that a positive attitude can be seen as a tool for overcoming an important obstacle in addressing difficulties in learning. Moreover, they indicated that a major problem in programming instruction may be negative student attitudes about programming. In a different study, Baser (2013) examined the relationship between students' achievement, attitudes, and gender in programming lessons. A total of 179 students taking a programming course were selected as the study's sample. As a result of the statistical analysis, it was found that there was a significant positive relationship between the achievement and attitudes of the students. In terms of gender, male students' attitudes were found to be higher than those of female students. These studies provide important evidence that attitude may have a positive effect on computer programming course achievement.

It is seen that attitude is an important factor for success in programming education that has been addressed in the literature. For example, Yukselturk and Altiok (2017) conducted a study with the participation of 151 teachers of Information Technology who delivered programming lessons. It was found that teacher candidates had decreased negative attitudes toward programming when using the learning environment developed by Scratch. On the other hand, in an experimental study by Çetin and Top (2014), 62 mechanical engineering students who participated in a programming course were selected as a sample group. During the course, a visualization tool called ACE was used as a learning tool. The result of the study revealed no significant attitude difference was found between the experimental and control groups with regards to programming. Similarly, Korkmaz (2016) found that Scratch-based game activities had no significant effect on students' attitudes. Moreover, Wang, Hwang, Liang, and Wang (2017) aimed to increase student achievement in programming lessons using an online-peer assessment-based system. In their quasi-experimental study, 166 high school students from four different classes participated as a sample. As a result of the study, the group with the peer assessment-based teaching strategy showed higher levels of programming knowledge, positive learning attitudes, and critical thinking awareness.

In summary, there are several studies in the literature that have shown attitude to be one of the significant factors for success in computer programming courses. On the other hand, it has been revealed that while some of the methods used in programming education have had a significant effect on students' attitudes, some methods have had no effect. For this reason, the current study aims to reinvestigate the effect of new teaching approaches on students' attitudes, which the literature stated was considered to be important and valuable in future studies.

2.8 Flipped Classroom

With the development of technology and the effects of changes in learning approaches, there are continuing variations in teaching methods and searching for teaching approaches that are appropriate for new generation students. Blended learning approach is one of them. In this approach, learning takes place both outside the school and inside the classroom environment. Zack, Fuselier, Graham-Squire, Lamb, and O'Hara (2015) concluded that blended learning is the tool and approach of combining online learning and traditional learning. The Flipped Classroom Approach (Lage, Platt, & Treglia, 2000; Sirakaya, 2017; Staker & Horn, 2012), which emerged as a derivation of this teaching approach, has begun to gain popularity in today's research and teaching literature. According to the classification of blended learning models, the Flipped Classroom Approach takes its place as in Figure 2.1 (Staker & Horn, 2012).

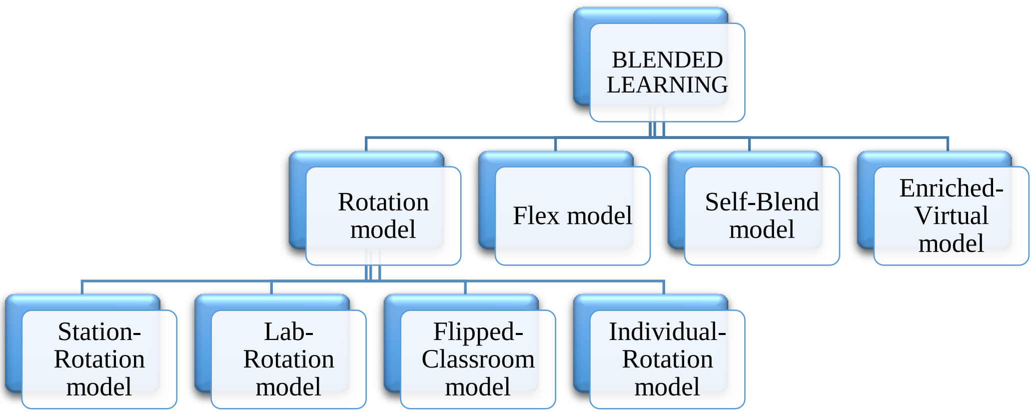


Figure 2.1 Classification of Blended Learning (Staker & Horn, 2012).

Although the Flipped Classroom Approach is not that new (Strayer, 2012), it has become increasingly popular among instructional strategies because of the increased availability of video technologies, the widespread use of open source teaching environments, and the use of video lessons. Although its first use in the 1990's was based on the use of "Inverted Classroom" in economics classes at Miami University (Lage et al., 2000), its spread began with two chemistry teachers, Jonathan Bergmann and Aaron Sams, taking video lessons for the students who missed the lesson and the students who saw this as a very good learning opportunity (Tucker, 2012). The chemistry teachers realized that these students became happy with this method and further contributed to the formation of the Flipped Classroom Approach, considering how to adapt this method to their subsequent lessons (Bergmann & Sams, 2012). Furthermore, the fact that Salman Kahn, the founder of Khan Academy, mentioned this approach and its use in his talks, has contributed to the further spread of this approach known as the Flipped Classroom (Sirakaya, 2017; Tucker, 2012).

While FC is perceived by many as the use of technology in the classroom, it is actually changing the pedagogy with the aid of technology (Bergmann & Sams, 2012). It is called "flipped classroom" because it is an approach that suggests transferring what is done in the traditional classroom management to outside of the classroom and vice versa. While the learning part of traditional approaches takes place in the classroom, homework assigned for outside the classroom is given in order to reinforce what has been learned within the classroom. This can cause students' difficulties as mostly such homework assignments tend to be more difficult than those learned within the classroom environment. FC is an approach that aims to reverse this method. Abeysekera and Dawson (2015) stated that there is no single definition of FC and described the characteristics of the approach with the following quotations from the literature:

- a change in use of classroom time;
- a change in use of out-of-class time;
- doing activities traditionally considered "homework" in class;
- doing activities traditionally considered as in-class work out of class;
- in-class activities that emphasize active learning, peer learning, problem-solving;
- pre-class activities;
- post-class activities, and;
- use of technology, especially video.

2.9 Benefits of Flipped Classroom

Bergmann and Sams (2012), who are widely regarded as the pioneers of the flipped classroom, shared their experience with the Flipped Classroom Approach in their book. They explained the following reasons in detail as to why FC should be used:

- Flipping speaks the language of today's students;
- Flipping helps busy students;
- Flipping helps struggling students;
- Flipping helps students of all abilities to excel;
- Flipping allows students to pause and rewind their teacher;
- Flipping increases student–teacher interaction;
- Flipping allows teachers to know their students better;
- Flipping increases student-student interaction;
- Flipping allows for real differentiation;
- Flipping changes classroom management;
- Flipping educates parents;
- Flipping makes your class transparent;
- Flipping is a great technique for absent teachers;
- Flipping can lead to the flipped-mastery program.

There are also other researchers who have discussed the benefits of the flipped classroom. For instance, O'Flaherty and Phillips (2015) suggested that FC motivates students to take on their own responsibilities and to learn at their own pace. In addition, it is inferred that students coming pre-prepared to class have higher levels of classroom attendance (McLaughlin et al., 2013). These point to FC being learner-centered and supports the learning of students through their own volition. Tune et al. (2015) stated that students in the flipped learning environment engage more in constructivist pedagogical content. Additionally, students become reflective and sophisticated in order to deeply understand and interpret what they learned in the FC. In this regard, flipped learning can be beneficial and functional in building confidence among gifted and talented learners (Siegle, 2014). As the activities of FC satisfy the specific demands of such students, it was effectively used to decrease the school dropout rates by securing trust and participation in the classroom. Thus, researchers with past experiences in the field including

Estes et al. (2016) and Lee (2018) suggest that flipped learning as a concept is essentially based on the need to optimize and maximize the time used in the classroom setting. Therefore, technology takes its place as a tool used in order to reach these objectives through the application of multimedia as the teaching medium (Bull, Ferster, & Kjellstrom, 2012). The conventional method of learning in the classroom involves students that are ready to learn new concepts and then leave the classroom in order to review and synthesize what they have learned, while the flipped method is different in terms of introducing new concepts. Hence, students have a general idea about the concept before they attend class and they come to the classroom prepared to participate in the in-class discussions by means of collaborative working and through interactive learning (Bergmann & Sams, 2012).

2.10 Research in Flipped Classroom

In this section, studies published in the literature on FC are summarized. First of all, the research in programming education is outlined and then the research the flipped classroom in other subject areas is addressed.

2.10.1 Programming Education and Self-efficacy

Studies about the flipped classroom as a new teaching approach began to emerge in order to examine its effect on students' computer programming self-efficacy. For instance, Souza and Rodrigues (2015) conducted an empirical study and compared the traditional classroom environment with the FC in terms of their effects on programming self-efficacy and academic performance in the context of an introduction to programming course. In their study, students in the flipped classroom group were found to have higher self-efficacy and performances when compared to those in the traditional classroom group. Accordingly, the researchers suggested that FC may be utilized as a potential way to increase programming self-efficacy and performance for courses on introduction to programming. In parallel, Özyurt and Özyurt (2018) investigated the effects of using the Flipped Classroom Approach in an Introduction to Programming course on software engineering students’ self-efficacy, success, and attitudes. A total of 46 students who attended the course took part in the study. From the results of the study, it was found that FC had a positive effect on the students’ success and self-efficacy. In another study, conducted by Amresh, Carberry, and Femiani (2013), the Flipped Classroom Approach was implemented in an introductory programming course to university students. The result of their study reported that FC improved students’ computing self-efficacy levels.

2.10.2 Programming Education and Engagement

When the focus was placed on studies undertaken with the flipped classroom, its effect on student engagement has been investigated with computer programming courses. For example, a study conducted by Turan (2016) on the Flipped Classroom Approach investigated the effect of student achievement and engagement on a computer programming and algorithm course. As a result of the study, the course attendance and programming lesson achievements were higher for students who were applied the FC approach when compared to those who had the traditional classroom approach. In another study, Pawelczak (2017) researched the effects of FC applied to the design of a programming course on university students’ exam results, and changes in their lesson preparations compared to those previously undertaken. The study results concluded that the students’ exam results increased, but only to a slight extent. They also observed that students were better prepared to attend FC lessons and that discussions on the topic were at a higher level. Moreover, in a study conducted by Puarungroj (2015), the Flipped Classroom Approach was used as a teaching strategy for an undergraduate programming course. The aim of the study was to investigate the perceptions according to FC, and the benefits that FC could provide in programming lessons. The study’s results showed that the students’ engagement in lectures increased, and that the problems that the students had previously encountered in their programming lessons were solved.

2.10.3 Programming Education and Attitude

Attitude is a factor that is investigated in programming education research. For example, Fetaji, Fetaji, Sukic, Gylcan, and Ebibi (2016) aimed to determine the benefits of FC to students in robotics programming lessons. The students were assessed for their attitudes, motivations, and effectiveness to this course. According to the researcher’s survey of 54 students, it was concluded that the motivation of the students was high. In the classroom environment, FCA has proved to be beneficial in terms of teacher-student relationship and time management. In another study, Tugun (2018) applied the FC method in computer science lessons for teaching algorithm and programming to 28 high school students. The results of the experiment revealed that the students showed a positive attitude towards the FC lessons. There was also a significant increase seen in the students’ technology self-efficacy levels, and the students showed a positive attitude towards using FC techniques. In the study conducted by Alhazbi (2017), students of Qatar University were applied FCA to their programming course. As a result of the study, it was concluded that the students’ attitudes toward the use of FC in their programming course were positive, and also that their performance improved. On the other hand, Özyurt and Özyurt (2018) conducted a study which analyzed the effects of Flipped Classroom Approach on student attitudes on a programming course. As a result of their study, it was found that FC had no effect on the students’ attitude towards programming.

2.10.4 Other Subjects

Having a general look at the recent studies conducted in the literature, it is seen that FC has been applied in a broad range of research, varying from secondary school level to graduate level. In recent studies, it can be seen that the achievement variable has been one of the most researched. These studies have mainly focused on the teaching of health (Bas-Sarmiento, Fernández-Gutiérrez, Baena-Baños, & Romero-Sánchez, 2017; Lewis, Chen, & Relan, 2017; Rui et al., 2017; Shiau et al., 2018), mathematics (Bhagat, Chang, & Chang, 2016; Foldnes, 2016; Lopes & Soares, 2018), statistics (Peterson, 2016), and English Language

(Ayçiçek & Yanpar Yelken, 2018; Ekmekci, 2017). These studies are analyzed in this section and summarized in Table 2.1.

Research on health education is one of the fields seen in studies related to FC. For example, Shiau et al. (2018) redesigned an epidemiology course with FCA. In their mixed-methods research of 150 students, exam scores, students' characteristics, and end-of-semester assessments were compared with the previous academic year, which was taught with the traditional teaching method. Although there were no statistically significant differences found in the examination scores or the students' assessments, positive opinions of the students about the course were revealed with regards to time management and the flexibility provided by FCA. In another study, Rui et al. (2017) conducted a research with the purpose of teaching Medical Diagnostic students with FC. In their study of 181 undergraduates, the students were compared based on post-instructional achievement tests. The results of the study showed that the achievements of those students taught with FC were significantly higher than those taught through traditional teaching. Furthermore, the majority of students who had experienced FC had positive attitudes toward it. In the study by Lewis et al. (2017), FC was applied in the training of surgery clerkship, with 136 students divided between control and experiment groups. In the study's results, there were no significant differences seen in the performance between the control and experiment groups (FC). Also, the students developed positive perceptions about FC, and their qualitative analysis showed evidence of student self-direction and more active and deeper in-class learning as perceived by the students. In another study by Bas-Sarmiento et al. (2017), FCA was applied in the empathy courses of nursing students. 48 participants were involved, and the results of the research found that pretest–posttest measures showed a significant increase in students' empathy performance.

Mathematics and statistics are seen as another field that comes to the fore in studies related to FC. In one example, a statistics course was studied in an experimental research by Peterson (2016), with the performances of students in statistics lessons compared from the perspectives of traditional teaching and FC. In the study involving 43 university students, the group receiving the FC showed a higher performance when compared to the control group who received traditional instruction. In an example of a mathematics course, Bhagat et al. (2016) studied the effects of FC-supported teaching environments on students' achievement and motivation. In addition, the impact on students at different achievement levels was also compared. Using a pretest–posttest quasi-experimental research method, 82 trigonometry course high school students were divided into two groups. It was observed that the group using FC significantly differed in terms of their success and motivation. In a study conducted by Foldnes (2016), the influence of FC on statistics and mathematics lessons was experimentally investigated. Performed in two stages, there was no meaningful difference seen between the FC and non-FC learners at first; but when cooperative learning activities were applied in the second stage, there was a significant difference seen between the experiment and control groups. Based on this finding, it was concluded that where FC is applied properly in the classroom, it can affect the achievement of students. In another study, Lopes and Soares (2018) included 803 university students in a study that applied FC to Financial Mathematics courses. In their study, some of the students were taught with FC while others were given traditional instruction. When the developments over two different periods were examined, it was concluded that the students generally improved their achievements.

Language education research is another field reported in studies related to FC. In an experimental study by Ayçiçek and Yanpar Yelken (2018), FC was used in the English courses of secondary school students. A total of 40 students were randomly distributed between an experimental group and a control group. It was discussed as a result of the study that although there was a difference seen between the first and the last engagement scoring in the FC group, the same significant difference was not seen in the traditional teaching method control group. Similarly, Ekmekci (2017) analyzed the effects of using FC in the writing courses of preparatory school students at the university level on their writing performance. From the results of the study, it was seen that 23 students in the experimental group showed significantly higher writing performance than the other groups. Moreover, a large majority of the students were found to have shown a positive attitude with regards to FC.

There are other fields that can also be seen in the literature on research related to FC. For example, Kurt (2018) conducted a study about the Flipped Classroom Approach, having applied the method in a classroom management course for English teacher candidates. A total of 62 students participated in the semi-experimental pretest–posttest study. Quantitative and qualitative data were collected from the students, and it was found that the FC students showed better results in terms of self-efficacy and learning outcomes than those who were applied the traditional teaching method. Moreover, the perceptions of the candidates on FC were found to be positive. In other studies, FC was used as a teaching strategy on science courses. For example, Zainuddin (2018) conducted research with the participation of 56 secondary school students that were divided between two groups. One of the groups was provided with FC activities, while the other group was taught with FC and gamification activities. Based on this instruction, the students' achievement levels were compared and is was reported that the achievements of the students in the FC group with gamification were found to be significantly higher.

Table 2.1 *Studies about FC in the Literature and their Results*

Author/s (year)	Course	Research Method	Sampling Level/ Number	Variables Examined	Outcomes
Kurt (2018)	Classroom management	Experimental	Undergraduates / 62	Self-efficacy, learning outcomes & perceptions	FC activity students showed better results for self-efficacy and learning outcomes than traditionally taught students. Perceptions for FC were positive.
Shiau et al. (2018)	Epidemiology	Mixed method	Graduates / 150	Achievement	No statistically significant differences in examination scores or students' assessment of the course between traditional and flipped. Students indicated satisfaction with the FC environment in terms of time management and freedom.
Ayçiçek & Yanpar Yelken (2018)	English	Quasi-experimental	Secondary school/ 40	Classroom engagement	Significant difference found between pretest and posttest scores of the experimental group; but no significant difference between the pretest and posttest scores for the control group.
Ekmekci (2017)	Writing	Mixed method	Undergraduates /23	Writing performance	Statistically significant difference found between the experimental and control groups in terms of writing performances.
Peterson (2016)	Statistics	Quasi-experimental	Undergraduates /43	Exam performance	FC group showed higher performance than the traditional group.
Bhagat et al. (2016)	Mathematics	Quasi-experimental	High School/82	Achievement and Motivation	FC group significantly different in terms of higher levels of success (performance) and motivation than the traditional group.
Foldnes (2016)	Statistic and mathematics	Experimental	Undergraduate Students	Achievement	No initial meaningful difference found between FC and non-FC learners, but with cooperative learning activities, a significant difference was seen between the experiment and control groups.

Table 2.1 *Studies about FC in the Literature and their Results*

Author/s (year)	Course	Research Method	Sampling Level/ Number	Variables Examined	Outcomes
Rui et al. (2017)	Medical diagnostics	Experimental	Undergraduate Students/181	Achievement	Achievements of FC students were significantly higher than traditionally taught students. Majority of students who experienced FC had a positive attitude towards it.
Bas-Sarmiento et al. (2017)	Empathy training	Quasi-experimental	Undergraduate Students/48	Perception of his/her performance	Performance of students improved significantly.
Lopes & Soares (2018)	Financial mathematics	Quasi-experimental	Undergraduate Students/803	Performance and perception	FC lessons positively affected the overall student achievement.
Zainuddin (2018)	Science	Mixed method	Secondary School/56	Performance and perceived motivation	Achievements of students in FC group with gamification were significantly higher than FC only group. Gamified FC fostered better motivation and engagement.
Lewis et al. (2017)	Surgery clerkship	Mixed method	Undergraduate Students/136	Performance and attitudes	No significant difference in exam performance between the control and intervention groups (FC).

2.11 Summary of the Literature

Initial results of the literature review showed that programming is increasingly used and applied in all areas of human life. Programming courses appear to be included in the curriculum of many departments, varying from computer science (Ismail et al., 2010) to other areas of engineering and science (Siti Rosminah & Ahmad Zamzuri, 2012). It is quite clear that most future professions are based on programming skills as a background, and it is estimated that most professions will require computer skills and knowledge (Giannakos et al., 2017).

The difficulties in learning, even though the subject may be important, are self-evident in the literature's reporting of high dropout rates. Some of the reasons given are due to the nature of the course and some due to the teaching methods (Lahtinen et al., 2005; Siti Rosminah & Ahmad Zamzuri, 2012). According to the literature, some of the students stated that practical lessons might be better than lectures with direct instruction (Siti Rosminah & Ahmad Zamzuri, 2012). It is also mentioned in some curriculum that learning by doing is important for such courses (Lahtinen et al., 2005). Because it is important to learn and practice topics, it is mentioned in the literature that lessons need to involve more practical examples (Lahtinen et al., 2005).

The Flipped Classroom Approach, which emerged as a derivative of blended learning (Lage et al., 2000; Sirakaya, 2017), has become increasingly widespread as technology, especially videography, becomes more accessible. In the literature, the advantages and benefits of using FC involve the more efficient usage of classroom time, active engagement of learners (Bergmann & Sams, 2012; Estes et al., 2016), and making lessons more practice-based. This also brings the question of how to turn these advantages into solutions in teaching programming curriculum. The current study aims to contribute to the literature by looking for answers to this area of research.

CHAPTER 3

METHODOLOGY

This chapter presents the research questions, research design, treatment conditions and materials, procedures of the study, instruments, and data analysis.

3.1 Research Questions

This study focuses on the following research questions:

1. Is there a significant difference in Computer Programming Self Efficacy mean scores between undergraduate students taught with the Flipped Classroom Approach and Traditional Instructional Approach?
    a. Is there a significant difference in Basic Programming component scores of the Computer Programming Self Efficacy Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    b. Is there a significant difference in Complex Programming component scores of the Computer Programming Self Efficacy Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

2. Is there a significant difference in General Class Engagement mean scores between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    a. Is there a significant difference in Cognitive Engagement component scores of the General Class Engagement Scale between

undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

b. Is there a significant difference in Behavioral Engagement component scores of the General Class Engagement Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

c. Is there a significant difference in Emotional Engagement component scores of the General Class Engagement Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

3. Is there a significant difference in Attitudes toward Computer Programming Languages mean scores between undergraduate students taught with the Flipped classroom Approach and the Traditional Instructional Approach?
    a. Is there a significant difference in Importance component scores of the Attitudes toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    b. Is there a significant difference in Enjoyment component scores of the Attitudes toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    c. Is there a significant difference in Self-confidence component scores of the Attitudes toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?
    d. Is there a significant difference in Motivation component scores of the Attitudes toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

4. What are the students' experiences and opinions about the Flipped Classroom Approach as a teaching strategy?

3.2 Research Design

A mixed-method research design, one that included both qualitative and quantitative approaches, was adopted to address the study's research questions. Employing the strengths of both approaches, the current study's primary emphasis was on its quantitative data through the application of a sequential explanatory design (Creswell, 2012). In this respect, the collection and analysis of quantitative data was then followed by the collection and analysis of qualitative data.

In the quantitative part of the study, a quasi-experimental research design was used in order to understand the effects of FC as a teaching strategy on students' computer programming self-efficacy, general class engagement, and attitudes towards computer programming languages. Repetitive measures design known as within-subject design in which the same group of subjects were exposed in more than one treatment. The elected treatments were Traditional Classroom Approach (TCA) and Flipped Classroom Approach (FCA). On the other hand, in the qualitative part of the study, semi-structured interviews were conducted with the students who had previously taken part in the quantitative part of the study. Table 3.1 presents the overall research design of the study.

Table 3.1 *Research Design of the Study*

Groups	Pretest	Treatment 1	Posttest 1	Treatment 2	Posttest 2
	CPSES		CPSES		CPSES
Student Group	GCES	Traditional Classroom	GCES	Flipped Classroom	GCES
	ATCPLS		ATCPLS		ATCPLS

CPSES : Computer Programming Self Efficacy Scale

GCES : General Class Engagement Scale

ATCPLS : Attitudes towards Computer Programming Languages Scale

3.3 Materials

The research was conducted within an Introductory Programming course, which was scheduled to be conducted for four hours per week. For five weeks, the traditional classroom approach was applied. In the following five weeks, Flipped Classroom Approach as a teaching strategy was used on the same group of learners. The students were provided with the subject matter of the course through the use of videos. In both learning environments, Gagne’s nine events of instruction were chosen as a guide. In brief summary, for this instructional method, the following nine steps are suggested to be followed.

1. Gaining attention: Materials related to the subject are brought to the learning environment.
2. Informing learners about the objectives: Learners are informed about the objective of the course.
3. Stimulating the recall of prior knowledge: Previous knowledge and skills are recalled.
4. Providing stimuli: Instruction is carried out with the materials brought to the learning environment.
5. Providing guidance for learning: Learners are provided with guidance in order to affect meaningful learning.
6. Eliciting performance: Learners are expected to show what they have learned by encouraging them to practice.
7. Providing feedback: Correct and incorrect behaviors of learners are detected. Correct behaviors are reinforced while incorrect behaviors are amended accordingly.
8. Assessing Performance: Learners’ performances are evaluated to determine if their learning has been fully accomplished.
9. Maintaining retention and enhancing transfer: Learned skills are retrieved and associated with new situations.

This method was used as a guide in designing the instruction. In both periods using different methods, Gagne's nine events of instruction model was employed, because the aim was to accurately capture the effect of the Flipped Classroom Approach. The design of the learning environment, be it flipped classroom or traditional classroom, is explored more elaborately in the following subsections.

3.3.1 Learning Management System

The initial instructional design action was to determine the learning management system. Learner familiarity with the system and the available facilities were the criteria used to determine the learning management system. Edmodo was selected due to similarity with the most frequently used social media environment. Another advantage offered by Edmodo is its high compatibility with video-sharing environments. The learners were able to share their presentations and class-notes using this system. The fact that it enabled learners to contact each other or the instructor was another factor that made Edmodo a suitable and credible option. See Figure 3.1 for an image of what the learning management system looks.

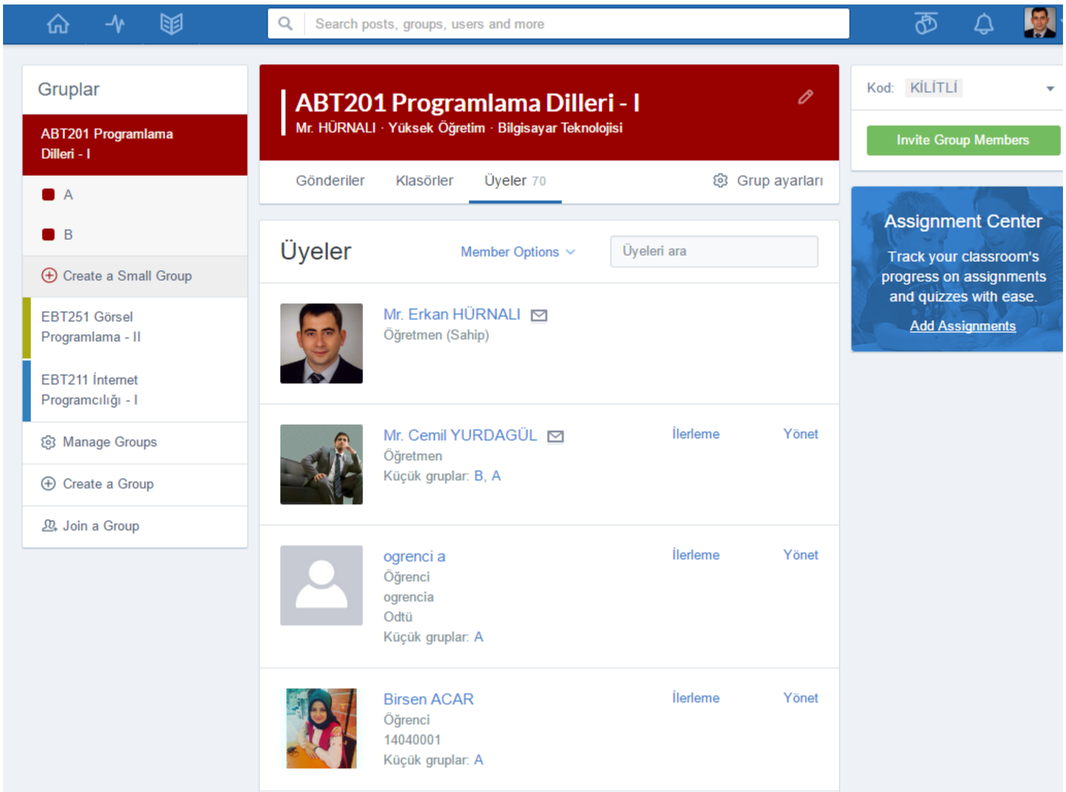


Figure 3.1 *User Interface of Learning Management System*

Edmodo also proved useful as it can award users with achievement badges in accordance with their performance in lessons. Figure 3.2 shows a learner's profile and any badges ("Rozetler") that they have been awarded.

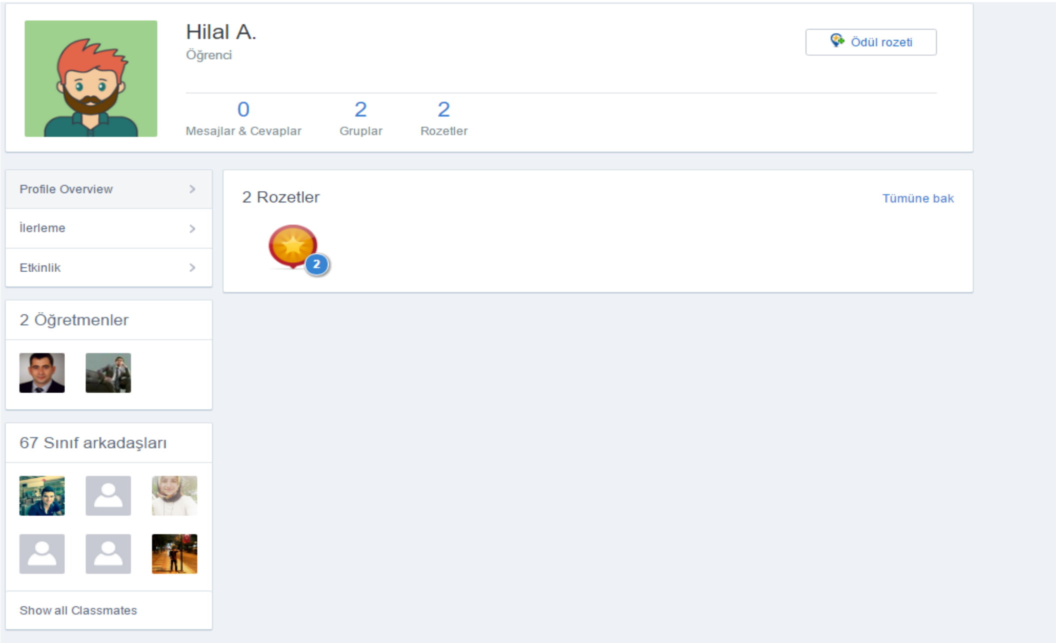


Figure 3.2 Badge System of Learning Management System

After selecting the learning management system, the next step was to design the course. The instruction was designed in accordance with the Traditional Classroom Approach and the Flipped Classroom Approach.

3.3.2 Design of the Learning Environment

3.3.2.1 Design of the Flipped Classroom Approach

In the second phase of the implementation, the Flipped Classroom Approach was chosen as the method of instruction. In using this method, Gagne's nine events of instruction were employed.

Pre-Class Video-Watching Activity

The pre-class phase of flipped learning has immense significance. The subject lecturing of the course was conducted through videos shared on the learning management system prior to the start of the class. The design of the video contents and the environment are elaborated upon as follows.

Design of the Videos

The pre-class phase of the FC represents the significant part of this kind of learning approach. For this very reason, the selection of the learning management system and devices was prioritized, and were chosen after thorough research about which would best enhance the efficiency of the environment. In light of this research, the environment developed through the collaboration of Educanon and Edmodo was considered as the best option for this course. The technological features of Edmodo enables the environment to be enhanced with videos, and features including making it possible to present questions at specific times of the videos. The device also enabled learners to answer the questions. Thus, questions were asked throughout the shared videos and learner responses were received. The learners' response could be seen by the course instructor, and the rates of learners' responses to questions reported. The environment does not allow advancing of the videos, and so the questions and flow of the video are viewed initially in strict sequence. After the first full viewing of the video, the learner is allowed to skip certain parts, but learners are required to see the video in full first time around to ensure that the instruction is follows a chronological order. The system is very user-friendly, with actions such as adding questions to videos, or editing them being easily accomplished, as can be seen in Figure 3.3.

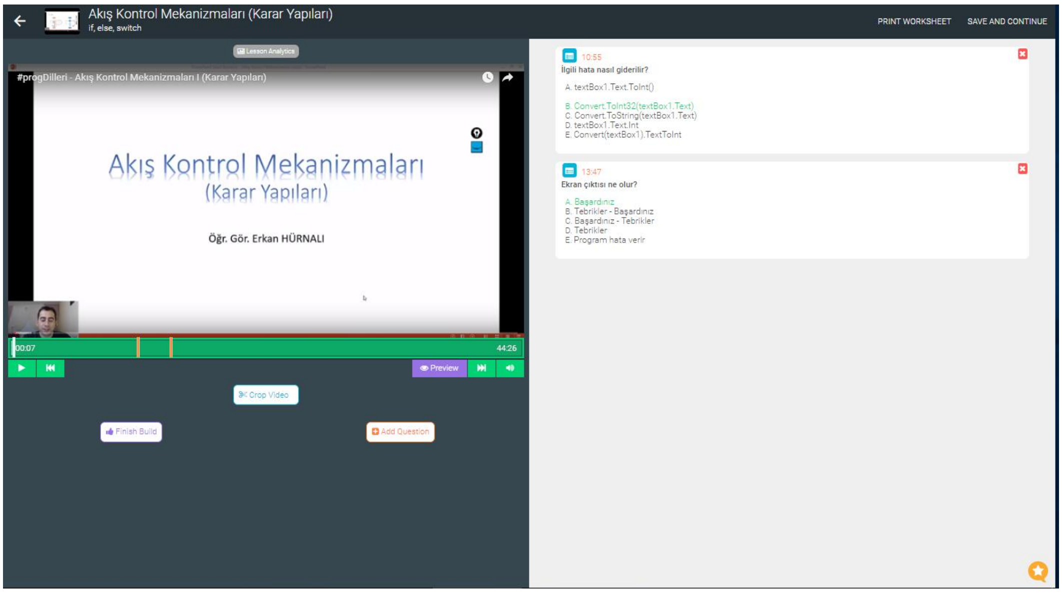


Figure 3.3 Representation of Interactive Question Preparation in Course Videos

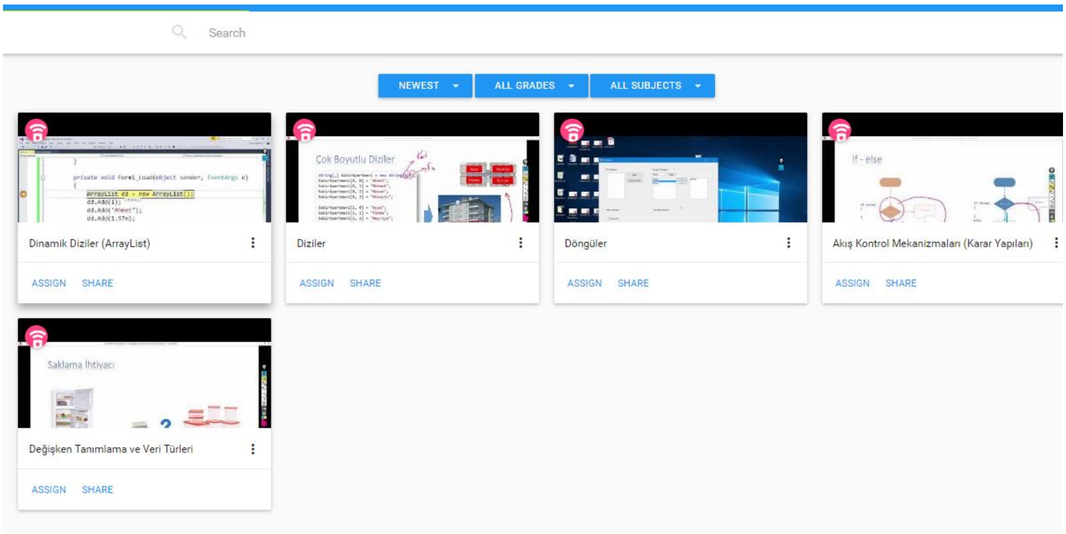
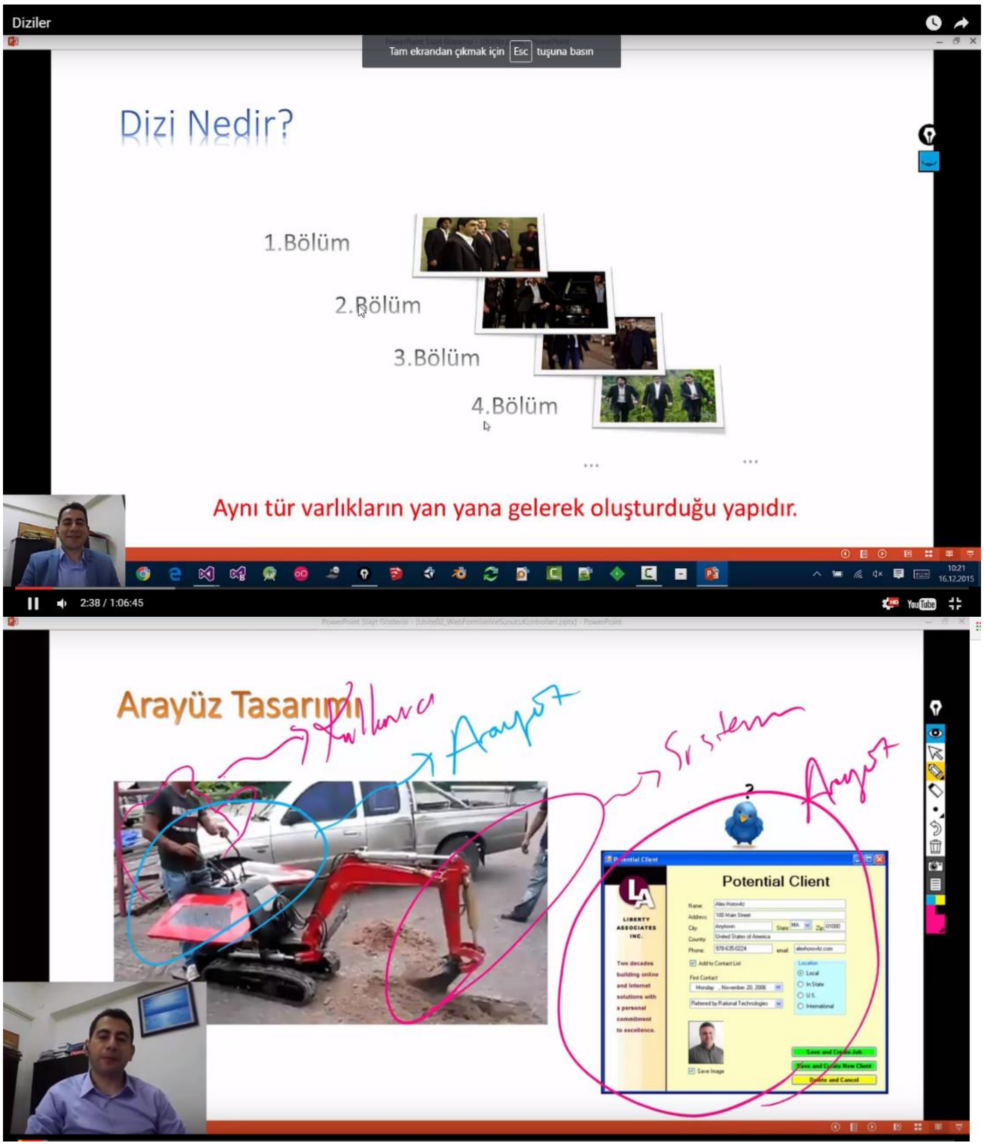


Figure 3.4 Representation of course videos in learning management system

Videos are viewed in modules (see Figure 3.4), and the reports of any chosen video can be reached through these modules. In this way, the instructor can use the learners' response as a means of evaluating the learners' performance.

Design of the Video Contents

In the process of shooting the videos, Gagne's nine events of instruction were used as a prioritization guide. The first step of this method aims at gaining the learners' attention. In order to achieve this goal, interesting images and examples are provided. For instance, when the subject of "Array" was being taught, interesting examples drawn from real life were given to attract learners' attention. Besides such examples, learners were given an insight into the importance of the subject and the reasons why. Figure 3.5 shows some of the ways to gain learners' attention.



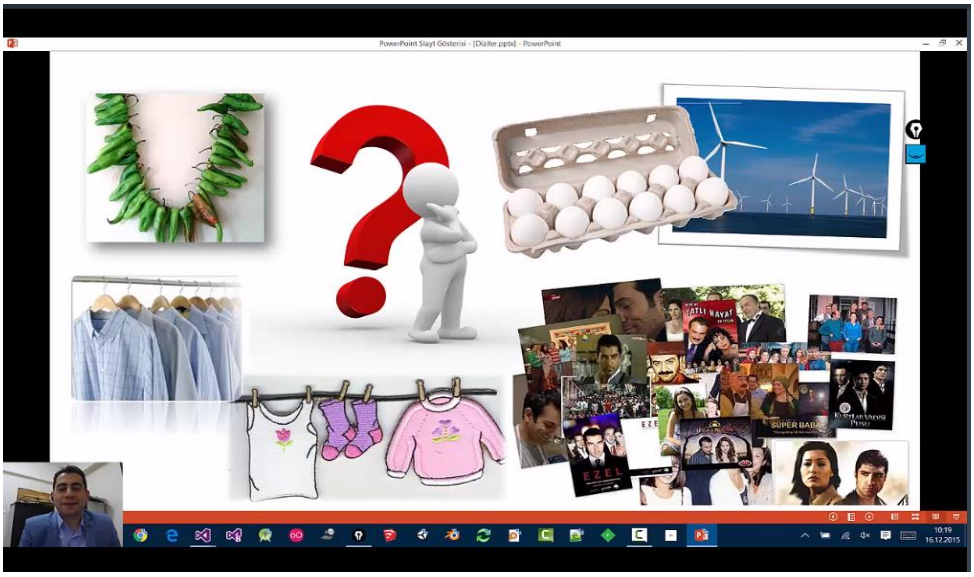


Figure 3.5 Use of Nine Events of Instruction in Course Videos

Another step of Gagne’s method requires “informing the learner about the objectives.” For this step, learners are provided with information about the subject to be learned and how to see what they would learn as a whole. In order to do this, a content page/slide was added to the video under the title “Subjects to be learned,” so as to give the learner a picture of what they were about to learn.

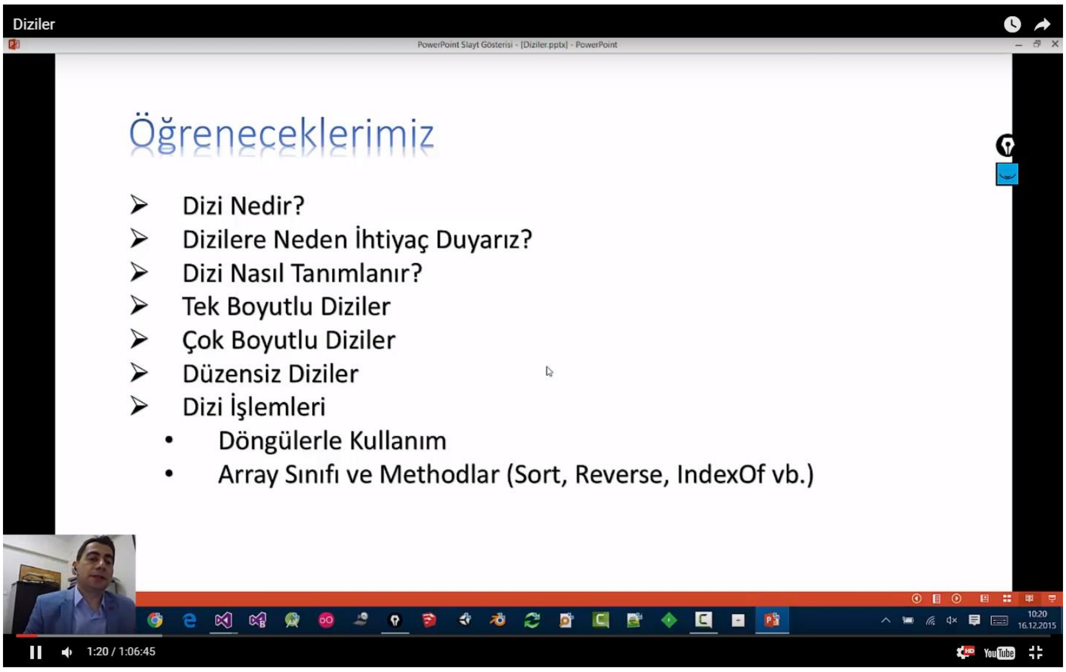


Figure 3.6 Display of Learning Objectives in Course Videos

"Stimulating the recall of prior knowledge" is another step making up the nine events of the instruction model. To enable students to recall knowledge gained earlier, another page/slide was added with the title of "What we learned" to the videos. This slide was significant in making the learners remember their prior knowledge and associate what they learned with what they will learn. Figure 3.7 shows that particular part of the video.

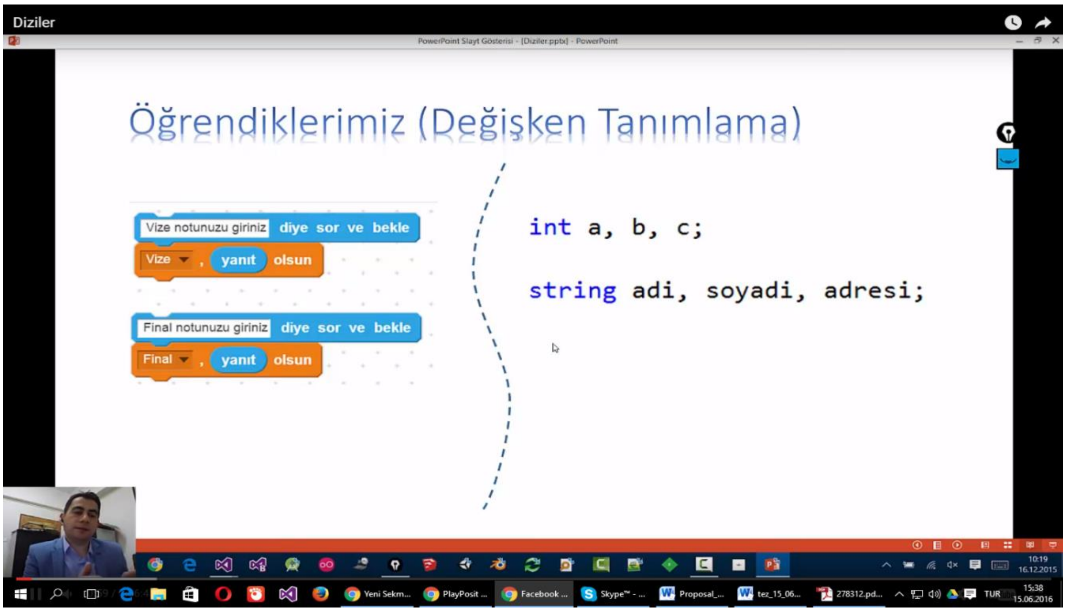


Figure 3.7 Sample; How Prior Knowledge is Recalled in Course Videos

In another step in Gagne's model, "Providing Stimuli" is when learners are provided with the learning environment. The aim was not just passively going over the presentations through the videos, but writing and drawing on the presentations when needed, so as to capture the learners' interest.

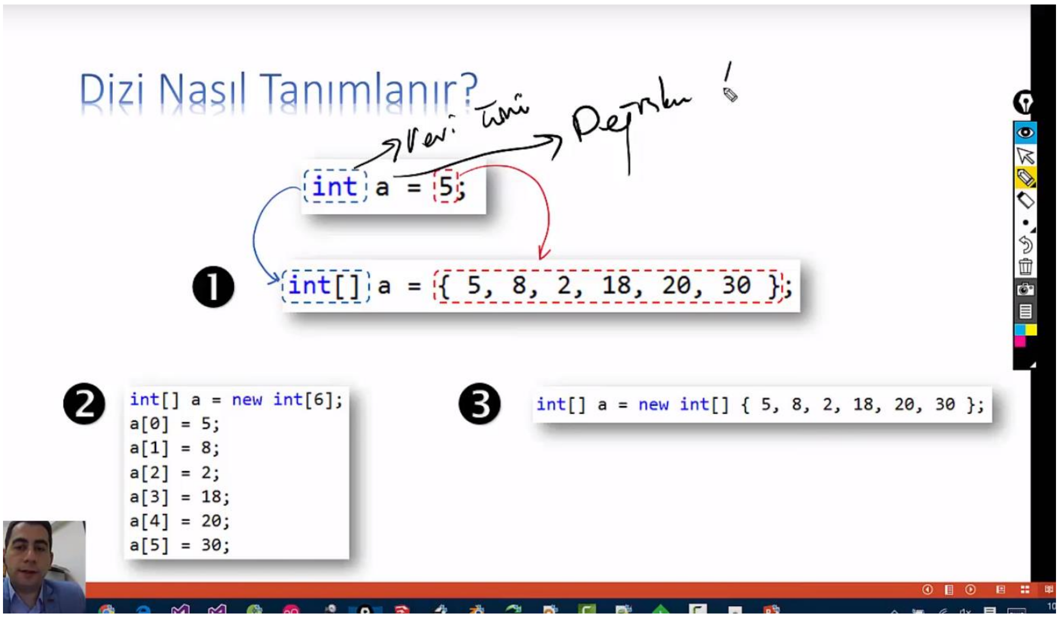


Figure 3.8 Sample; Providing Stimuli Through Course Video Writing/ Drawing

In this way, the instruction was achieved through figures and explanations when necessary. Besides this, learners were shown how codes were written by example. The subject was not only instructed, but a demonstration of its practice was also shown to the learners.

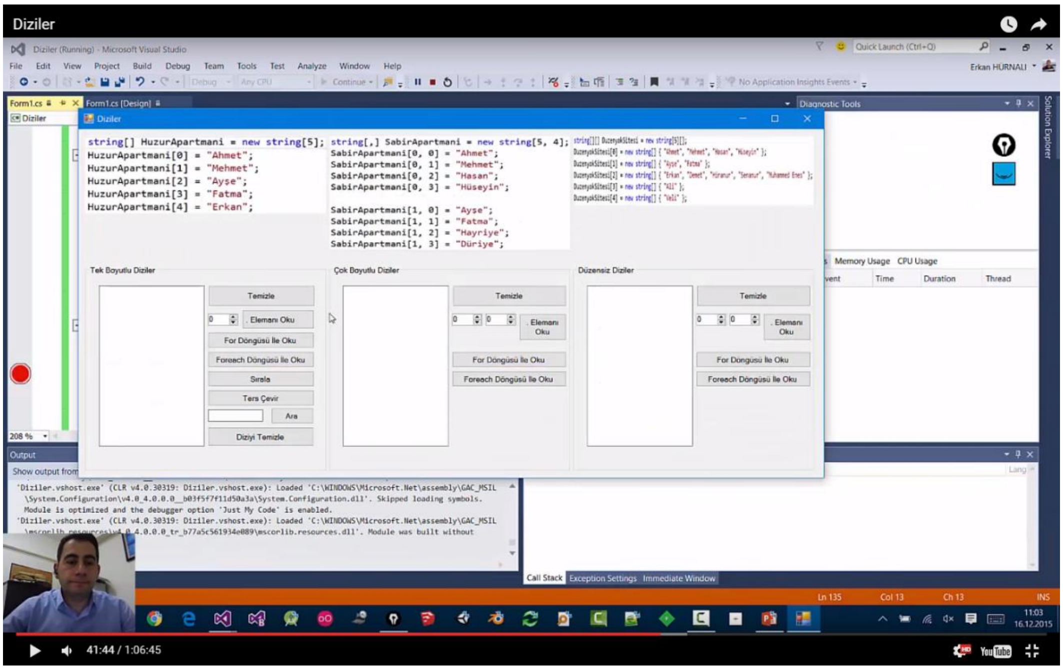


Figure 3.9 Sample; Codes Displayed in Course Videos

To increase interaction, motivate learners to view videos, and to attract their attention to some important points, questions were asked at a certain frequency. The intention being to make sure that learners understand the importance of the subject and to be more aware of their presence in the learning process. Questions were also added in order to make the videos more interesting. As can be seen from the images in Figure 3.10, the questions added were sometimes open-ended, and sometimes multiple-choice in order that learners could see whether or not they gave the correct answer.

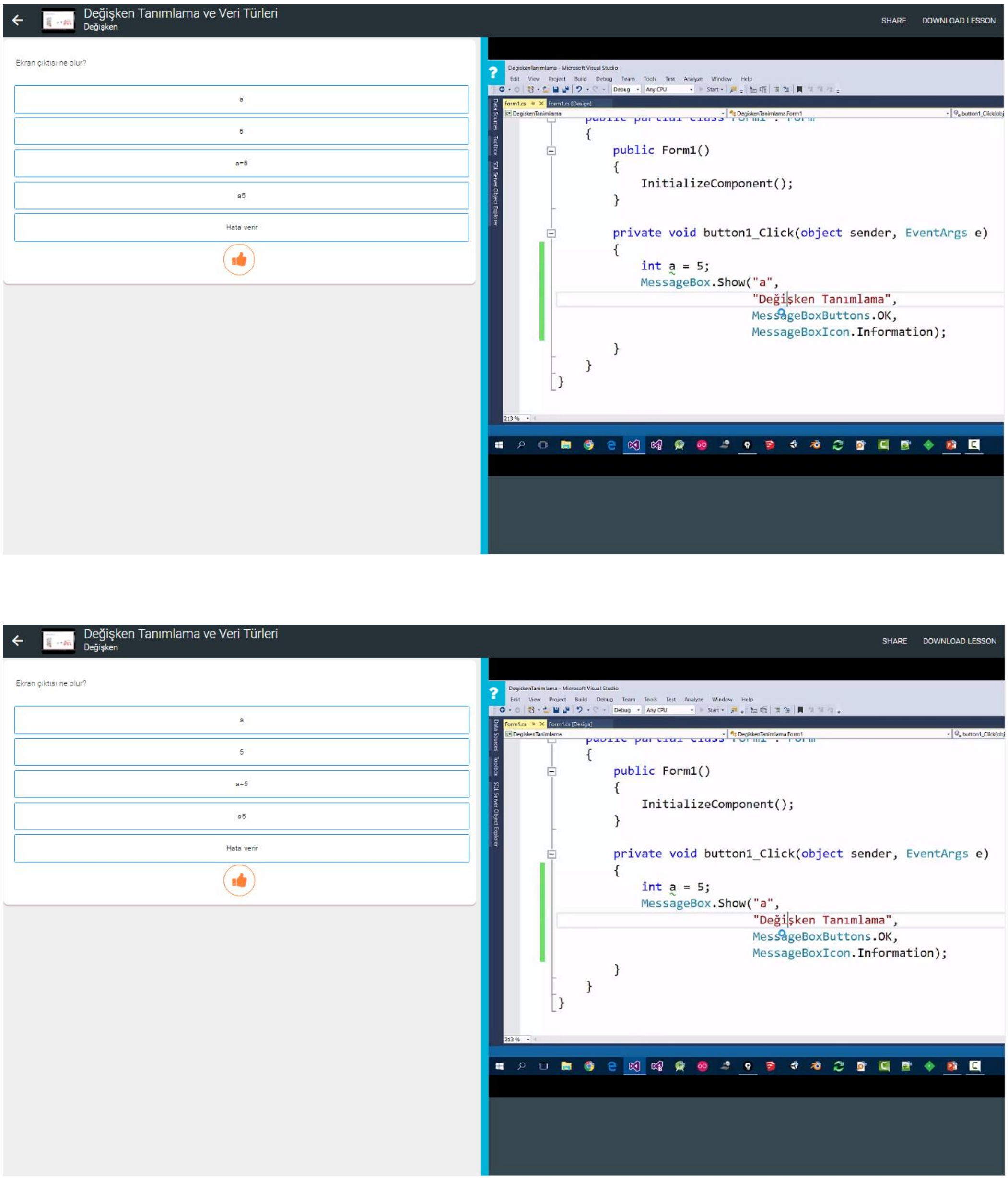


Figure 3.10 Display of Interactive Questions in Course Videos

Questions along with text overlay and instructions in the videos were aimed at making the subjects easier by guiding the learners, and thereby accomplishing the "providing guidance" step in order for meaningful learning to be achieved.

Another step, "Eliciting Performance," was achieved by learners given practice assignments after they had watched the videos. Those who successfully completed the assignments were awarded badges within the Edmodo environment. For this step, the learners were required to present what they had learned as a product. The assignments were also intended to ensure the students had an awareness of their own strengths and limitations (if any), and encouraged them to ask questions about any difficulties they encountered in doing their assignments.

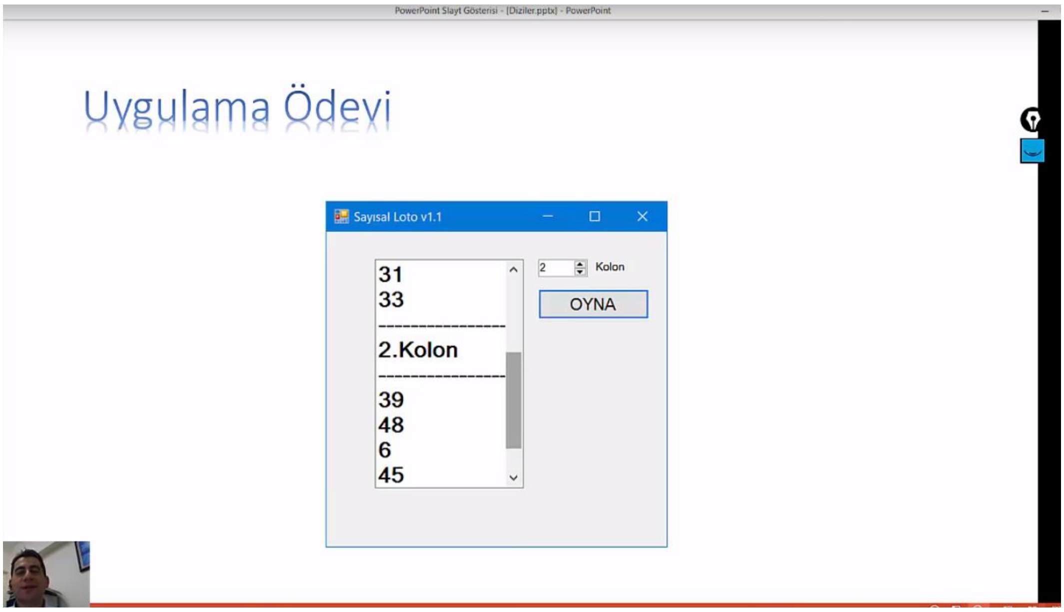


Figure 3.11 Display of How Performance was Elicited in Course Videos

Other steps of Gagne's model, "Providing Feedback" and "Assessment of Performance," were mainly achieved by looking at learners' answers to the questions added to the videos and from assessing their practice assignments within the classroom environment.

The rate of video-views and learners' answers to questions were observed through the learning management system, and feedback was given in accordance with these observations. The rate of learners' video viewings and their correct answers were reported with visuals such as diagrams.

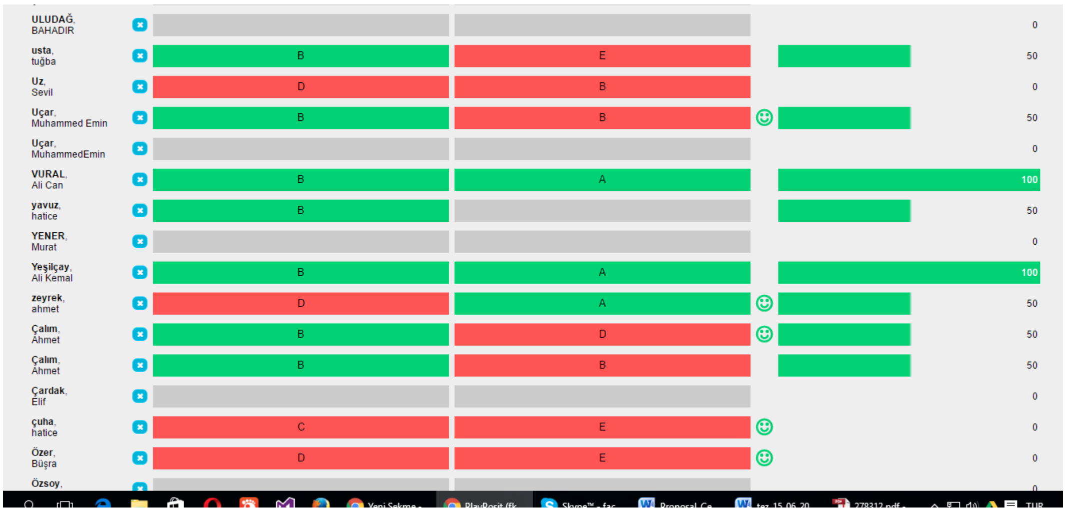


Figure 3.12 Instructor’s Panel Display of Responses to Questions

In-Class Environment

The structure of the in-class environment changed for those who attended class having watched the videos. Classes generally started with a brief summary of the content of the videos. In this part of the class, learners could ask questions about anything they failed to understand after watching the videos. The rest of the class hours were allocated to providing feedback for practice assignments and resolving matters which needed more explanation. In this way, it became possible to focus on the practice of what was learned during class-hours.

3.3.2.2 Design of the Traditional Classroom Approach

During the initial five weeks of the course, which is the time when the instruction was conducted based on the Traditional Classroom Approach, the subjects were directly lectured to the students through presentations. Gagne’s nine events of instruction model were used as the guiding method.

The nine steps of Gagne’s nine events of instruction were all applied within the classroom environment. Steps such as gaining attention, stating objectives, recalling prior knowledge, and providing stimuli were applied to presentations prepared collaboratively by the researcher and the instructor.

The design and structure of the presentations used in the videos for the FC method were preserved in the traditional approach. The discussion of subjects were started in an interesting way, with learners informed about the subjects they would be learning. Associations between what was and what would be learned were made and then the lecturing phase began. In short, the only difference between the two time-periods is that during the five weeks in which the FC was being applied, seven steps of the nine events of instruction were achieved through the application of video presentations.

3.4 Treatment Conditions

The current study analyzed the effects of the two teaching strategies on a group of students. Firstly, an introductory programming course was taught using the Traditional Classroom Approach for a period of five weeks, and then the same course was taught using the Flipped Classroom Approach for the following five weeks. Using this dual approach, the researcher was able to investigate the effects of both approaches on the students' self-efficacy, engagement and attitudes in a computer programming course. Table 3.2 provides detailed information regarding the differences in the implementation of these two teaching strategies.

Table 3.2 *Comparison of Teaching Approaches*

Teaching Strategy	Explanation
Traditional Classroom	Students do homework set at the previous week's lesson prior to attending the next class, and then the teacher checks the homework in the class. The teacher then introduces the next new topic, and a question and answer session is held. In the remaining time, code writing practice is undertaken based on that day's topic. Homework for the following week is then assigned, and the students are to complete their homework.
Flipped Classroom	The teacher shares a video with the students about the next week's topic before coming to the classroom. Students answer the interactive questions within the video. Then, the classroom session starts off with a short summary of the video, and students are questioned about any parts they did not understand. Code writing practice is then undertaken related to that week's topic. Finally, students are reminded to watch the video about the next week's topic prior to attending class.

The main differences between the two approaches in terms of practice are that in one the teaching takes place in the classroom, while in the other the teaching happens outside of the classroom. Table 3.3 outlines the differences in time usage between the two approaches to the course.

Table 3.3 *Course Timeline*

	Traditional Classroom		Flipped Classroom	
	Activity	Time	Activity	Time
Pre-class	Previous week's homework	45 min	Watching the next week's video	45 min
In class	Checking homework	20 min	Summary of the topic in the video	20 min
	Teaching of that week's topic	70 min	Question and answer session regarding any problems in the video	25 min
	Question and answer session	15 min	In-class code writing practice	110 min
	In-class code writing practice	55 min	Reminder to watch next week's video	5 min

3.5 Procedure of the Study

3.5.1 Flipped Classroom Pilot Study

First of all, because the instructor was unfamiliar with the practice of flipped learning and in order to avoid difficulties in the main study, a pilot study was conducted. The pilot study was implemented at the Vocational School of Ankara University during an Internet Programming I course.

Videos were shared with 80 students prior to the classroom-based lesson taking place. During the study, learners were informed about what was required of them through the Edmodo learning management system. The announced information is shown in Figure 3.13. The instructor, who would later implement the flipped learning method for the purposes of the current study's main research, had the opportunity to experience flipped learning prior to implementing the main study. This approach was taken in order to practice flipped learning so as to make the main study more effective. Learners were observed to be quite content with this practice, and the main study's environment was later designed in accordance with suggestions of the pilot study's participants, which proved to be beneficial as it enabled the instructor to experience the instructional model in action.

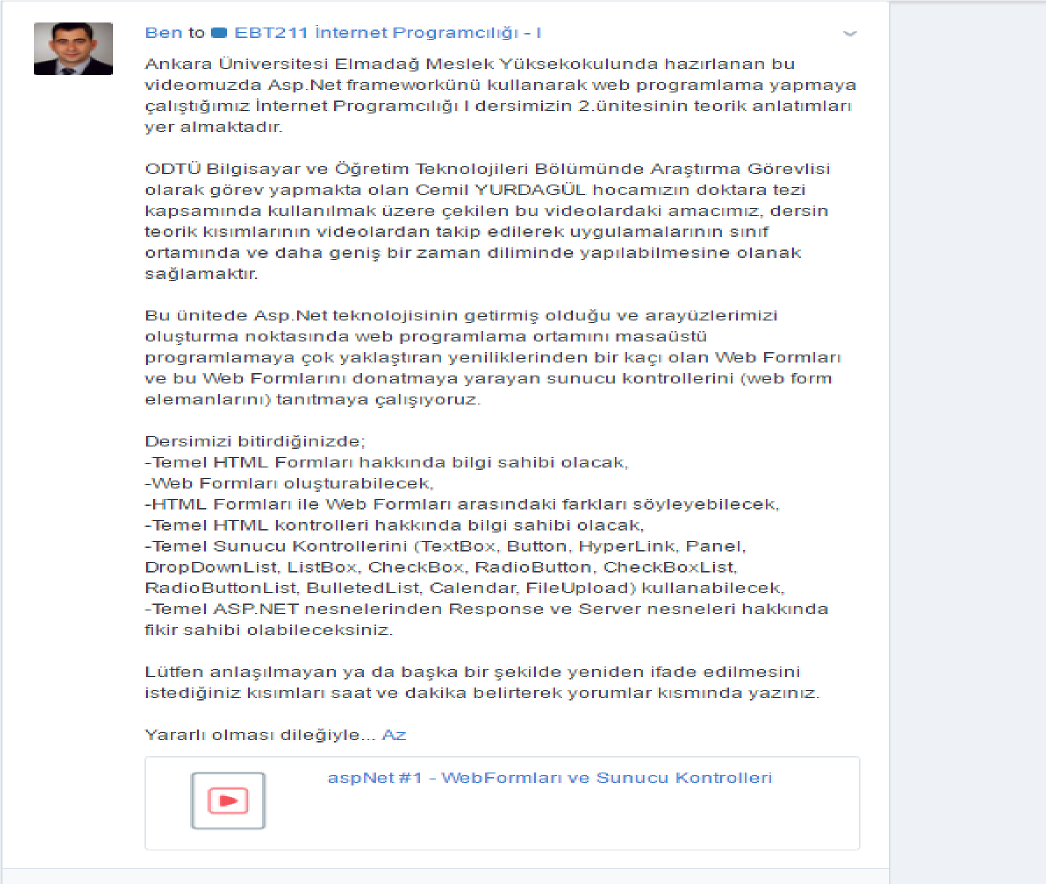


Figure 3.13 Display of Announcement Statement for Pilot Study

3.5.2 The Process of the Main Study

Participants of the main study consisted of students from the Computer Education and Instructional Technology (CEIT) department of Ankara University. The research was implemented during ten weeks of the Introductory Programming I course during the fall semester.

In order to collect the research data, the necessary permissions were obtained in advance from the university's ethics committee. Also, it was ensured that the participants voluntarily participated in the study. The ten-week period of the study was divided into two distinct phases. During the first phase, the students were applied the Traditional Classroom Approach, while during the second phase the same group of students were instructed through the Flipped Classroom Approach.

The first phase started at the beginning of the semester. Prior to instructing the learners, they were asked to complete three data collection instruments and one demographic information form. The instruments presented to the students were the Computer Programming Self-efficacy Scale, the General Class Engagement Scale, and the Attitudes towards Computer Programming Languages Scale. Soon after the pretest application of the instruments, the students were then given five weeks of instruction using the Traditional Classroom Approach. After the five week period of tuition, the students completed the same three instruments as a posttest.

Next, the teaching strategy was changed to that of the FC for the second phase. Students in the second phase were also applied five weeks of instruction, but this time using the Flipped Classroom Approach. During this second five week period, the instruction about the course subject's topics were presented to the students through videos that they watched prior to attending class.

Upon completion of the second five week phase, the learners were presented with the same three instruments as a posttest along with an additional new survey that investigated the learners' opinions about Flipped Classroom Approach.

After quantitative analysis of the data, 15 students were selected from those with high, medium, and low mean scores according to the results of the posttest implementation of the scales, and then interviews were conducted with them. This selection method for the 15 interviewees aimed at capturing diversified student opinions. Figure 3.14 presents the process of the main study.

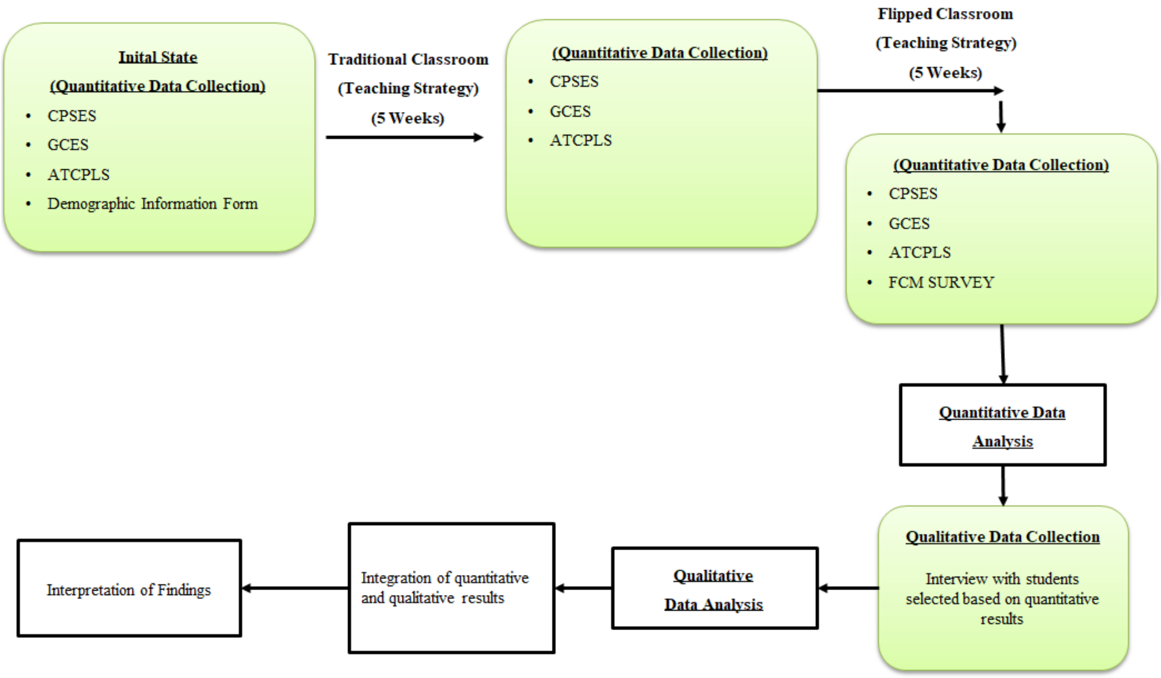


Figure 3.14 Process of the Main Study

3.6 Sample of the study

The participants of the study were second year undergraduate students studying in the Computer Education and Instructional Technology Department. Convenience sampling method was used to recruit students who were taking an Introductory Programming course. Although a total of 45 students joined the course at the beginning of the semester, 35 students attended all of the course sessions, and therefore, only these 35 students were included as the sample for the current study. The students were mostly graduates of regular high schools or vocational high schools. Table 3.4 provides demographic information of the participant students based on gender, school type, programming language experience, and age.

Table 3.4 *Demographic Characteristics of Participants (N = 35)*

	n	%
Gender		
Female	23	66
Male	12	34
School Type		
Vocational High School	22	66
Regular High School	13	34
Programming Language Experience		
No experience	8	23
Less than 1 year	9	26
1-2 years	10	29
3-5 years	7	20
6-8 years	1	3
9 years or more	-	
Age		
29	1	03
22	5	14
21	7	20
20	13	37
19	9	26
Daily time spent using the Internet		
Less than 1 hour	2	6
1-2 hours	12	34
3-5 hours	12	34
6-8 hours	4	11
9 hours or more	5	14

Note: Percentages totals may not be 100 for all characteristics due to rounding

As Table 3.4 shows, 66% of the students were female whereas 34% were male. Moreover, 66% of the participants had graduated from vocational high schools. They had varying levels of prior experience in programming; for example, while nine students had no prior experience in programming, ten students had 1-2 years' experience. Related to the students' ages, they were mostly around the same age.

As to the selection of the participants for the qualitative part of the study, purposive sampling method was employed. Among the 35 students who participated in the quantitative part of the study, 15 students were purposively chosen to participate in semi-structured interviews for the qualitative part of the study. In order to achieve maximum variety among the interview participants, students were selected based on high, medium, and low mean scores in self-efficacy, engagement, and attitude scales.

3.7 Instruments

In the quantitative part of the study, the Computer Programming Self-efficacy Scale, the General Class Engagement Scale, the Attitudes towards Computer Programming Languages Scale, and the Flipped Classroom Survey were used. Moreover, a student interview form was developed for the collection of the qualitative data. Table 3.5 summarizes the data collection instruments.

Table 3.5 *Characteristics of Instruments Used in the Study*

Instrument Name	Computer Programming Self-efficacy Scale	General Class Engagement Scale	Attitudes Towards Computer Programming Languages Scale	Flipped Classroom Survey
Number of Items	9	15	40	19
Subscales	▪ Self-efficacy in Basic Programming ▪ Self-efficacy in Complex Programming	▪ Cognitive Engagement ▪ Behavioral Engagement ▪ Emotional Engagement	▪ Importance ▪ Enjoyment ▪ Self-confidence ▪ Motivation	

3.7.1 Computer Programming Self-efficacy Scale (CPSES)

The Computer Programming Self-efficacy Scale was used in order to measure students' self-efficacy. The CPSES was adopted from Altun and Mazman (2012) who created the scale for the measurement of student computer programming self-efficacy levels. The scale consists of nine items on a seven-point, Likert-type scale. It has two factors: Self-efficacy in Basic Programming, and Self-efficacy in Complex Programming. Altun and Mazman (2012) reported the Cronbach's alpha coefficient of the scale as being .92. Similarly, the current study found the Cronbach's alpha coefficient to be .94, showing the scale to be reliable.

3.7.2 General Class Engagement Scale (GCES)

In this study, the General Class Engagement Scale was used to evaluate student engagement in the class. Eryılmaz (2014) developed the GCES to measure the engagement of undergraduate students in the classroom. The scale consists of 15 items measured on five-point, Likert-type scale. The GCES includes three sub-factors: Cognitive Engagement, Emotional Engagement, and Behavioral Engagement. For the reliability of the scale, Eryılmaz (2014) reported the Cronbach's alpha coefficient as being .92, and the current study found the Cronbach's alpha coefficient to be .91, showing the scale to be reliable.

3.7.3 Attitudes Towards Computer Programming Languages Scale (ATCPLS)

The Attitudes Towards Computer Programming Languages Scale was applied to evaluate students' attitudes towards computer programming language. The scale was adopted from Durak (2013). The scale consists of 40 items measured on a five-point, Likert-type scale. It has four factors: Importance, Enjoyment, Self-confidence, and Motivation. The Cronbach's alpha coefficient was reported by Durak (2013) as being .93. For the current study, the Cronbach's alpha coefficient was found to be .96, showing the scale to be reliable.

3.7.4 Flipped Classroom Survey (FCS)

The Flipped Classroom Survey was used to understand students' opinions and experiences regarding the Flipped Classroom Approach. The researcher of the current study developed the FCS in collaboration with his thesis advisor and a fellow PhD student. There are 19 questions in the survey, measured on a five-point, Likert-type scale.

While forming some of the survey questions, the focus was on the most commonly mentioned advantages of the FC reported in the literature. The reason for adding questions about the advantages of the FC was to understand if students consider the advantages of FC for a programming course in the same way as it was considered in the literature.

There are also several survey questions that aim to reveal students' thoughts and opinions related to the FC videos, and their intention of attending other FC-based courses in the future. In addition, several questions were added to compare flipped and traditional classroom teaching from the students' perspective. The validity of the survey questions was established through feedback from subject matters.

3.7.5 Student Interview Form

The semi-structured interview method was used to elicit the students' opinions towards the Flipped Classroom Approach. While preparing the interview questionnaire, the opinion of the researcher's thesis advisor and other colleagues who completed their doctoral studies in the same department were consulted and the questions were subsequently clarified.

The interview questionnaire was piloted with two students. Analysis of the pilot interviews led to changes being applied based on the pilot students' feedback, and the addition of one question to the interview form. The added question was included in order to reveal students' opinions and experiences related to their interaction with other students. During the pilot interviews, the two students mentioned that while watching the FC videos, they communicated, collaborated and shared ideas with each other. Additionally, the number of questions that received similar answers were reduced.

The first five interview questions are general and were designed to understand the students' insight into the FC teaching approach. The next four questions aimed to obtain the students' general views about the videos used in the FC and their thoughts on the sequencing and interactive questions used in the design of the videos. Finally, the last four questions aimed to understand which features of the Flipped Classroom Approach contributed to the students' self-efficacy and attitudes towards programming and to changes in their in-class engagement.

3.8 Data Analysis

After the data collection, SPSS software (Statistical Package for Social Science) version 23 was used to analyze the quantitative data. The level of significance was set as .05 by the researcher. The dependent variables were students' results from the Computer Programming Language Self-efficacy Scale, the General Class Engagement Scale, and the Attitudes Towards to Computer Programming Languages Scale. The independent variable of the study was the different classroom approach of the instruction (FCA or TCA). The technique for answering each of the research questions is presented in the following parts.

Analysis of Research Question 1

In order to address the first research question about Computer Programming Self-efficacy, one-way repeated measures ANOVA was conducted. The aim of this analysis was to examine the effects of FC as an independent measure on computer programming self-efficacy, and as a dependent measure on a computer programming course.

<u>Hypothesis 1:</u> There is no statistical difference as to the students' computer self-efficacy mean scores between undergraduate students taught with flipped classroom and traditional instructional approaches.

Hypothesis 2: There is a statistical difference as to the students' computer self-efficacy mean scores between undergraduate students taught with flipped classroom and traditional instructional approaches.

Analysis of Research Question 2

In order to address the second research question about General Class Engagement, one-way repeated measures ANOVA was conducted. The aim of this analysis was to examine the effects of FC as an independent measure on general class engagement, and as a dependent measure on a computer programming course.

Hypothesis 3: There is no statistical difference as to the students' General Class Engagement mean scores between undergraduate students taught with flipped classroom and traditional instructional approaches.

Hypothesis 4: There is a statistical difference as to the students' General Class Engagement mean scores between undergraduate students taught with flipped classroom and traditional instructional approaches.

Analysis of Research Question 3

In order to address the third research question about Attitudes Towards Computer Programming Languages, one-way repeated measures ANOVA was conducted. The aim of this analysis was to examine the effects of FC as an independent measure on attitudes towards computer programming languages, and as a dependent measure on a computer programming course.

Hypothesis 5: There is no statistical difference in Attitudes Towards Computer Programming Languages mean scores between undergraduate students taught with flipped classroom and traditional instructional approaches.

Hypothesis 6: There is a statistical difference in the Attitudes Towards Computer Programming Languages mean scores between undergraduate students taught with flipped classroom and traditional instructional approaches.

Analysis of Research Question 4

In order to address the fourth research question, qualitative data analysis was conducted. First, the selected subset of students were subjected to semi-structured interviews that were conducted and audio-recorded. After that, the interview recordings were transcribed in full.

Then, the interview transcript data were analyzed with MAXQDA analytical software and meaningful codes generated. The code pool was created by combining codes with similar expressions. Upon examining the code pool, common features of the codes were categorized and themes created. For the purposes of reliability, the intercoder agreement process was followed (see Section 3.10.1). The qualitative data analysis process is presented visually in Figure 3.15.

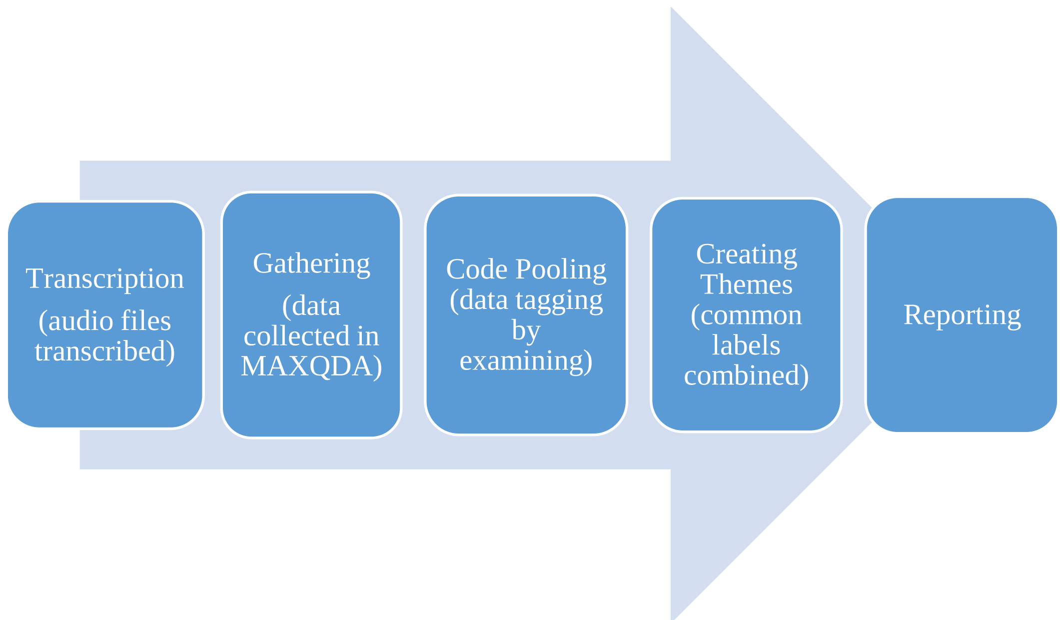


Figure 3.15 Process of Qualitative Data Analysis

3.9 Reliability and Validity

Validity can be defined as meaningfulness, correctness, and usefulness of the result of the study (Fraenkel & Wallen, 2006). On the other hand, reliability refers to the consistency of scores or answers from one implementation of an instrument to another. Both of these aspects are important in the correctness of presenting the results of research.

3.9.1 Reliability

In the quantitative part of the study, there were four instruments applied. Although reliability and validity testing results were published by the original developers of the first three scales (Computer Programming Self-efficacy Scale, General Class Engagement Scale, and Attitudes Towards Computer Programming Languages Scale), for the current study the reliability was recalculated and the reliability coefficient value for each scale reported (see Section 3.6). On the other hand, as only descriptive statistical methods were used for the Flipped Classroom Survey, there was no requirement to perform reliability testing, although opinion was sought from the researcher's advisor and colleagues with regards the issue of the survey's validity.

3.9.2 Validity

Internal Validity

Creswell (2012) stated that internal validity "relates to the validity of inferences drawn about the cause and effect relationship between the independent and dependent variables" (p. 303). Fraenkel and Wallen (2006) stated that some threads can affect internal validity.

In order to prevent internal validity threats, the following techniques were employed. At the beginning of the current study, 45 students completed a pretest of the three scales, but 10 of them did not attend all of the course teaching sessions and were therefore excluded from the study. Moreover, the researcher standardized the conditions under all steps of the study. The 10-week study duration was implemented in order to eliminate maturation effect. Also, the current study's subject design ensured that the researcher could control subject characteristics.

External Validity

External validity is more related to a generalization of the result of a study to other settings, times, and participants. The current study followed a within-subject design and the sample size was insufficient to generalize in other settings.

3.10 Trustworthiness for Qualitative Part

3.10.1 Internal Validity (Credibility)

Peer Debriefing

According to Creswell and Miller (2000), "A peer review or debriefing is the review of the data and research process by someone who is familiar with the research or the phenomenon being explored" (p. 129). It is a strategy for enhancing the credibility of a study. The thesis committee and advisor for the current study supported and checked the research in order to enhance the study's credibility by way of peer debriefs.

Intercoder Agreement

The data analysis process employed the intercoder agreement procedure in which an additional coder was used to clearly and deeply analyze the content of the data (Creswell, 2012). An experienced colleague with knowledgeable and experience of qualitative data analysis provided guidance in this process. The colleague was informed about the objectives and the phases of the study prior to the coding. Next, all the interview transcripts were coded simultaneously by both the researcher and the colleague (intercoder).

After the coding process was complete, both coders met in order to discuss their analysis findings, specifically any coding similarities and differences. Similar codes with different categories referring to the same concepts were combined into one single code. In addition, differences between codes that referred to different concepts were discussed between the two coders and agreement reached as to whether or not to include or exclude each specific code. Finally, all of the codes were controlled and reviewed with the purpose of making sure that agreement was reached between both coders. By using MAXQDA's "intercoder agreement" function, it was found that the intercoder reliability score was .85.

Prolonged Engagement

Prolonged engagement allows the researcher to build a close relationship with study's participants by attending the research setting over a prolonged period of time (Creswell & Miller, 2000). In the current study, the researcher worked as an assistant throughout the course alongside the course instructor; having attended all of the teaching sessions during the semester. It may be proposed that prolonged engagement was achieved in the current study due to the researcher's involvement level and time spent throughout the course.

3.10.2 External Validity (Transferability)

Thick, rich description

According to Creswell and Miller (2000), "Another procedure for establishing credibility in a study is to describe the setting, the participants, and the themes of a qualitative study in rich detail" (p. 128)

The current study's sample and settings are detailed in Chapter 3 – Methodology. In addition, the qualitative results are reported exactly as transcribed from the students' comments recorded in their respective interviews.

3.11 Summary

In the current study, the sequential explanatory mixed-method was used in order to determine the effects of using FC as a teaching strategy on students' self-efficacy, engagement, and attitudes towards computer programming languages, and to elicit students' opinions about the application of FCA as the method of instruction in the classroom. The research sample involved a total of 35 students studying in their second year at the CEIT department of Ankara University. In this current chapter, the application process of the research has been explained in detail, and the research questions, research design, treatment conditions and materials, procedures of the study, instruments, and data analysis discussed. In the following chapter, the results of the study are presented based on the research questions.

CHAPTER 4

RESULTS

4.1 Quantitative Findings

This section of the thesis reports on the quantitative findings related to the effects of the flipped classroom on different aspects of students-relevant factors. Both descriptive and inferential statistics were exploited in order to distill important and essential information or statistics from the raw data. While descriptive statistics were used to describe and summarize the data in a meaningful and understandable way, inferential statistics were applied in order to go beyond describing the data by proving findings drawn from a sample that can be generalized to the population.

The types of descriptive statistics employed in the current study were mean, standard deviation, skewness, kurtosis, and percentage. On the other hand, repeated measures of One-Way Analysis of Variance (ANOVA) were applied for the comparison of the students' scores over time. A measure of the participants' scores for the subscales of several tests was repeated for three conditions: before application of the Traditional Classroom Approach, after the Traditional Classroom Approach, and after application of the Flipped Classroom Approach. As a result, the quantitative findings are presented and reported as follows along with the related statistics.

Descriptive Statistics for Perceptions of Students on Flipped Classroom Approach

Table 4.1 *Mean, Standard Deviation, and Numbers of Responses for Variables of Flipped Classroom Survey (N = 35)*

Variable	*M*	*SD*	*Strongly Disagree*	*Disagree*	*Neutral*	*Agree*	*Strongly Agree*
During the course I frequently communicated with my classmates	3.66	1.03	2	3	5	**20**	5
During the course, I frequently communicated with my instructor	3.71	0.79	1	1	8	**22**	3
Flipped classroom teaching strategy and educational videos helped improve my learning	4.06	0.76	1	0	3	**23**	8
Watching videos prepared in advance helped me make progress in the programming language course	4.06	1.00	1	2	4	**15**	13
I requested help from someone to watch the videos already given for the lesson	2.97	1.34	6	9	4	**12**	4
In order to better understand the course, I watched the previously prepared videos more than once	3.47	1.17	4	2	8	1	**16**
I was pleased to be able to view the previously prepared videos after the course	4.17	0.92	1	1	3	**16**	14
Flipped classroom teaching strategy allowed me to participate more in classrooms than with the traditional method	4.03	0.98	1	1	7	**13**	**13**
I would suggest courses with the flipped classroom teaching strategy to my friends	4.06	0.91	1	0	7	**15**	12
Flipped classroom teaching strategy enabled me to communicate more with my friends during the lesson	3.86	1.06	1	3	7	**13**	11
I enjoyed watching videos supplied in advance of the lessons	3.89	1.13	2	2	6	**13**	12
I regularly watched the videos supplied in advance of the lessons	3.97	0.86	1	0	7	**18**	9
I would like to take part in other courses designed with the flipped classroom teaching strategy	3.86	0.97	2	1	4	21	7

Table 4.1 *Mean, Standard Deviation, and Numbers of Responses for Variables of Flipped Classroom Survey (N = 35)*

Variable	*M*	*SD*	*Strongly Disagree*	*Disagree*	*Neutral*	*Agree*	*Strongly Agree*
We spent more time practicing with the course designed with the flipped classroom teaching strategy	4.23	0.84	2	0	3	**15**	**15**
Flipped classroom teaching strategy provides learning opportunities suited to my own learning pace	3.97	0.95	1	1	7	**15**	11
Disseminating flipped classroom teaching strategy is useful for society	3.94	0.94	1	1	7	**16**	10
Flipped classroom improves the sense of responsibility	3.94	1.03	1	1	10	10	**13**
After the course I watched the videos again that were supplied in advance of each lesson	4.14	0.85	1	0	4	**18**	12
I think that the flipped classroom teaching strategy is boring	2.69	1.49	**12**	5	4	10	4

Note: Boldface indicates highest values

A series of questions were asked to the students in order to reflect their perceptions towards the implementation of the Flipped Classroom Approach. Responses of the students to these questions are shown in Table 4.1. As can be clearly seen, the majority of the students were either neutral or in strong agreement in their perceptions on the Flipped Classroom Approach.

It might be useful to further explore some of the responses shown in Table 4.1. For example, the statements that were perceived highly (agree or strongly agree) by a large number of students were related to the student's high level of communication with their classmates (*n* = 20) or the course instructor (*n* = 22), and the benefit of pre-class assigned or watched educational videos to their learning (*n* = 23) and improvement in programming language (*n* = 15). In addition, there were some statements in which the majority of students strongly agreed with or reflected high perception (agree). Those were about the watching of videos more than once in advance of the course so as to better understand the content (n = 16), and more time spent practicing in FC (n = 15) than TC (n = 13). Furthermore, a significant number of students thought that learning opportunities provided by the FC teaching strategy were in alignment with their own pace learning (n = 15).

4.1.1 Research Question #1: Is there a significant difference in Computer Programming Self Efficacy mean scores between undergraduate students taught with the Flipped Classroom Approach and Traditional Instructional Approach?

One-way repeated measures ANOVA were used to examine the effects of the Flipped Classroom Approach on students' self-efficacy in programming language. There was one independent and two dependent variables. While the independent variable was three conditions, also called levels, dependent variables were two dimensions of the Computer Programming Self-efficacy scale: Self-efficacy in Basic Programming; and, Self-efficacy in Complex Programming. The same groups of students were exposed to three different conditions over different times: Initial State, Traditional Classroom Approach, and Flipped Classroom Approach. Two aspects of students' self-efficacy in programming language were measured after each intervention. In the repeated measures ANOVA analyses, the differences in the mean scores of students' self-efficacy in programming language under three conditions were examined.

Mauchly's test of sphericity tests was used to inspect the assumption of sphericity. The assumption of sphericity had not been violated for Self-efficacy in the Basic Programming and Complex Programming subscales. Since there was no violation of sphericity, no adjustment or correction strategy was applied. Moreover, analysis of Skewness and Kurtosis values indicated normal distribution of variables (see Appendix E).

Table 4.2 *Means and Standard Deviation for Computer Programming Self-efficacy Scale*

Subscales	Initial state (Level 1)		Traditional Classroom Approach (Level 2)		Flipped Classroom Approach (Level 3)	
	M	*SD*	*M*	*SD*	*M*	*SD*
Self-efficacy in Basic Programming	4.62	1.84	5.30	1.74	5.82	1.20
Self-efficacy in Complex Programming	3.30	1.56	3.85	1.54	4.86	1.37

Table 4.3 *Repeated Measures Analysis of Variance for Subscales of Computer Programming Self-efficacy Scale*

Source	*df*	*SS*	*MS*	*F*(2, 68)	*p*	η^2
Self-efficacy in basic programming	2	25.37	12.69	11.49*	.00	.253
	68	75.07	1.10			
Self-efficacy in complex programming	2	44.15	22.07	21.78*	.00	.390
	68	68.93	1.01			

Note. * Mean difference significance at $p < .05$

Repeated measures ANOVA were conducted in order to compare the mean scores of students' self-efficacy in basic and complex programming language over two different conditions. As clearly indicated in Table 4.3, there was a significant difference in the Basic Programming component scores of the Computer Programming Self Efficacy Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Classroom Approach (F (2, 68) = 11.49, $p < .05$) (see Table 4.3). Analysis of post hoc tests using the Bonferroni correction showed that the mean scores of Self-efficacy in Basic Programming after Level 2 (M = 5.30; SD = 1.74) and Level 3 (M = 5.82; SD = 1.20) were significantly higher than the mean score in Level 1 (M = 4.62; SD = 1.84) (see Table 4.2). These findings suggest that students' Self-efficacy in Basic Programming increased after experiencing either the Traditional Classroom Approach or the Flipped Classroom Approach. However, there was insufficient evidence to conclude that the difference occurred between the Self-efficacy scores in the first condition and after the third condition were caused uniquely by the application of FC strategies.

As for Self-efficacy in Complex Programming, a significant difference was seen for the Complex Programming component scores of the Computer Programming Self Efficacy Scale between the undergraduate students taught with the Flipped Classroom Approach and the Traditional Classroom Approach (F (2, 68) = 21.78, $p < .05$) (see Table 4.3). Analysis of post hoc tests using the Bonferroni correction revealed that the mean score of Self-efficacy in Complex Programming after Level 3 (M = 4.86; SD = 1.37) was significantly higher than the mean score in Level 1 (M = 3.30; SD = 1.56) and Level 2 (M = 3.85; SD = 1.54) (see Table 4.2). Based on this result, it could be concluded that FC strategies created a significant increase in students' Self-efficacy in Complex Programming when compared to the two other previous conditions.

4.1.2 Research Question #2: Is there a significant difference in General Class Engagement mean scores between undergraduate students taught with the Flipped Classroom Approach and the Traditional Instructional Approach?

One-way repeated measures ANOVA were conducted in order to examine the effects of FC strategies on students' engagement in an introductory programming course. There was one independent and three dependent variables. While the independent variable was three conditions also called levels, dependent variables were three dimensions of course engagement scale: behavioral, emotional and cognitive engagement in an introductory programming course. The same groups of students were exposed to three different conditions over different times: initial state, Traditional Classroom Approach and Flipped Classroom Approach. Three aspects of students' engagement in an introductory programming course were measured after each intervention. In the repeated measures ANOVA analyses, the differences in the mean scores of the students' engagement in an introductory programming course under these three conditions were sought.

Mauchly's test of sphericity tests was used to inspect the assumption of sphericity. The result of the test showed that assumption of sphericity had not been violated for each dimension of the Engagement scale. Moreover, analysis of Skewness and Kurtosis values indicated normal distribution of variables (see Appendix E).

Table 4.4 *Means and Standard Deviation for General Class Engagement Scale*

	Initial state (Level 1)		Traditional Classroom Approach (Level 2)		Flipped Classroom Approach (Level 3)	
Subscales	*M*	*SD*	*M*	*SD*	*M*	*SD*
Behavioral engagement	3.62	0.77	3.75	0.78	4.06	0.63
Emotional engagement	3.44	0.71	3.42	0.72	3.77	0.73
Cognitive engagement	3.80	0.61	3.84	0.72	3.94	0.58

Table 4.5 *Repeated Measures Analysis of Variance for Subscales of General Class Engagement Scale*

Source	*df*	*SS*	*MS*	*F*	*p*	η^2
Behavioral engagement	2	3.58	1.79	6.46*	0.00	0.160
	68	18.81	0.28			
Emotional engagement	2	2.66	1.33	5.88*	0.00	0.147
	68	15.39	0.23			
Cognitive engagement	2	0.38	0.19	0.68	0.51	0.020
	68	18.95	0.28			

Note. * Mean difference significance at $p < .05$

Repeated measures analysis of variance was conducted in order to compare the mean scores of students' behavioral, emotional, and cognitive engagement in an introductory programming course over three different conditions. Mauchly's sphericity test showed that the variances of the differences between all combinations of the within-subject conditions were equal (see Appendix E). Therefore, the analysis proceeded without any need for adjustment. The result, as clearly indicated in Table 4.5, shows a significant difference in the Behavioral Engagement component scores of the General Class Engagement Scale between the undergraduate students taught with the Flipped Classroom Approach and the Traditional Classroom Approach (F (2, 68) = 6.46, $p < .05$) (see Table 4.5).

Also, there was a significant difference in the Emotional Engagement component scores of the General Class Engagement Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Classroom Approach (F (2, 68) = 5.88, $p < .05$) (see Table 4.5). Post hoc tests were conducted in order to detect any differences between the three conditions.

Regarding the Behavioral Engagement component, analysis of post hoc tests using the Bonferroni correction showed that the mean score of Behavioral Engagement after the third condition (M = 4.06; SD = .63) was significantly higher than the mean score in the first condition (M = 3.62; SD = .77) and after the second condition (M = 3.75; SD = .78) (see Table 4.4). Based on this finding, it could be concluded that the Flipped Classroom Approach was considerably more influential than the Traditional Classroom Approach in terms of its impact on augmenting students' engagement within an introductory programming course.

Related to Emotional Engagement, analysis of post hoc tests using the Bonferroni correction showed that the mean score of Emotional Engagement after the third condition (M = 3.77; SD = .73) was significantly higher than the mean score in the first condition (M = 3.44; SD = .71) and after the second condition (M = 3.42; SD = .72) (see Table 4.4). This finding suggests that the Flipped Classroom Approach was better in augmenting students' emotional engagement than conditions involving either the Traditional Classroom Approach or initial state.

4.1.3 Research Question #3: Is there a significant difference in Attitudes toward Computer Programming Languages mean scores between undergraduate students taught with the Flipped classroom Approach and the Traditional Instructional Approach?

One way repeated measures ANOVA was conducted in order to examine the effects of the Flipped Classroom Approach on students' Attitudes Toward Computer Programming Languages. There was one independent and four dependent variables. While the independent variable was three conditions also called levels, dependent variables were four dimensions of the Attitudes Towards Computer Programming Languages Scale: importance, enjoyment, motivation, and self-confidence. The same groups of students were exposed to three different conditions over different times: Initial State, Traditional Classroom Approach, and Flipped Classroom Approach. Four aspects of students' attitudes toward programming languages were measured after each intervention. In the repeated ANOVA analyses, the differences in the mean scores of the students' attitudes on programming language under three conditions were examined.

Table 4.6 *Means and Standard Deviation for Attitudes Toward Computer Programming Languages Scale*

	Initial state (Level 1)		Traditional Classroom Approach (Level 2)		Flipped Classroom Approach (Level 3)	
Subscales	*M*	*SD*	*M*	*SD*	*M*	*SD*
Importance	4.03	0.78	4.07	0.68	3.98	0.85
Enjoyment	3.50	0.77	3.79	0.56	3.90	0.63
Self-confidence	3.53	0.81	3.74	0.75	3.91	0.69
Motivation	3.57	0.76	3.68	0.74	3.75	0.73

Mauchly's test of sphericity tests the assumption of sphericity. A significant result means that the assumption of sphericity has been violated. In the case of violation of the assumption of sphericity, two adjustments, known as epsilon, are suggested: Greenhouse-Geisser and Huynh-Feldt correction. Greenhouse-Geisser correction was used instead of Huynh-Feldt correction. The assumption of sphericity had been violated for the Importance and Enjoyment subscales (see Appendix E).

Table 4.7 *Repeated Measures Analysis of Variance for Four Dimensions of Attitudes Toward Computer Programming Languages Scale*

Source	*df*	*SS*	*MS*	*F*	*p*	η^2
Importance	1.66	.15	.09	.203	.78	.006
	56.47	24.52	.43			
Enjoyment	1.70	3.09	1.81	7.25*	.003	.176
	57.93	14.48	.25			
Self-confidence	2	2.55	1.27	4.37*	.016	.114
	68	19.82	.29			
Motivation	2	.57	.29	.77	.47	.022
	68	25.26	.37			

Note. * Mean difference significance at $p < .05$

As can be seen in Table 4.7, for both the Importance and Motivation subscales, no significant difference was detected between the mean scores under three different conditions. On the other hand, there was a significant difference in the Enjoyment component scores of the Attitudes Towards Computer Programming Languages Scale between the undergraduate students taught with the Flipped Classroom Approach and the Traditional Classroom Approach (F (1.70, 57.93) = 7.25, $p < .05$). Also there was a significant difference in the Self-confidence component scores of the Attitudes Toward Computer Programming Languages Scale between undergraduate students taught with the Flipped Classroom Approach and the Traditional Classroom Approach (F (2, 68) = 4.37, $p < .05$). In summary, changes in the mean scores over three conditions were found to be significant for both the Enjoyment and Self-confidence subscales.

Related to the Enjoyment subscale of the Attitudes Toward Computer Programming Languages Scale, post hoc tests using the Bonferroni correction revealed that enjoyment-related attitudes after the second condition (*M* = 3.79; *SD* = .56) and the third condition (*M* = 3.90; *SD* = .63) were significantly higher than the mean score in the first condition (*M* = 3.50; *SD* = .77) (see Table 4.7). It could be concluded that both the Flipped Classroom Approach and the Traditional Classroom Approach created positive impacts on the Enjoyment subscale of the Attitudes Toward Programming Languages Scale.

Related to the Self-confidence subscale of the Attitudes Towards Programming Languages Scale, post hoc tests using the Bonferroni correction revealed that self-confidence-related attitudes after the third condition (*M* = 3.91; *SD* = .69) were significantly higher than the mean score in the first condition (*M* = 3.53; *SD* = .81) (see Table 4.7). As a result, it could be concluded that the Flipped Classroom Approach elicited a significant increase in the Self-confidence subscale of the Attitudes Towards Programming Languages Scale.

4.2 Summary of The Quantitative Findings

A tabular summary of the quantitative results is presented in Tables 4.8 to 4.10. These tables only provide information related to the significant findings. To ensure better and clearer understanding, results which were not found to be significant were purposely excluded from these three tables. The word label denotes the condition. The three conditions (Initial State, Traditional Classroom Approach, and Flipped Classroom Approach) in the study are represented as Level 1, Level 2 and Level 3 respectively. A repeated measure of ANOVA was conducted in order to examine the effect of these three conditions on students' self-efficacy, engagement, and attitudes on a programming language course. The "larger than" sign (>) and the "smaller than" sign (<) is shown between levels so as to indicate which mean score of one level was significant larger or smaller than the other level. For example, in the first row of Table 4.8, the statement Level 2 > Level 1 shows that the mean score of the Traditional Classroom Approach was significantly higher than that of Initial State in terms of Self-efficacy in Basic Programming.

Table 4.8 *Summary of Repeated ANOVA Results Showing Significance of Related Groups for Self-efficacy in Computer Programming Scale*

Subscale	Initial State (Level 1)	Traditional Classroom Approach (Level 2)	Flipped Classroom Approach (Level 3)
Self-efficacy in Basic Programming	Level 1 < Level 2 Level 1 < Level 3	Level 2 > Level 1	Level 3 > Level 1
Self-efficacy in Complex Programming	Level 1 < Level 3	Level 2 < Level 3	Level 3 > Level 1 Level 3 > Level 2

When Table 4.8 is evaluated in terms of the Basic Programming subscale of Self-efficacy, it is seen that the mean score in Level 2, which was after the students were taught using the Traditional Classroom Approach, is significantly higher than the one in Level 1, which was measured at the Initial State. Table 4.8 also shows that the mean score in the measurement of Level 3, which was applied after the students were taught with the Flipped Classroom Approach, is significantly higher than the one in Level 1, which shows the Initial State. Upon examining Table 4.8 in terms of Self-efficacy in the Complex Programming subscale, it can be concluded that the mean score measured in Level 3 is significantly higher than both the Level 2 and Level 1 scores, which respectively shows the measurements after applying the Traditional Classroom Approach and the Initial State.

Table 4.9 *Summary of Repeated ANOVA Results Showing Significance of Related Groups for General Class Engagement Scale*

Subscale	Initial State (Level 1)	Traditional Classroom Approach (Level 2)	Flipped Classroom Approach (Level 3)
Behavioral Engagement	Level 1 < Level 3	Level 2 < Level 3	Level 3 > Level 1 Level 3 > Level 2
Emotional Engagement	Level 1 < Level 3	Level 2 < Level 3	Level 3 > Level 1 Level 3 > Level 2

As can be seen in Table 4.9, the behavioral engagement mean score in Level 3, which was applied after the Flipped Classroom Approach, is significantly higher than the scores in both Level 1 and Level 2. Examining the means scores of the Emotional Engagement subscale, it is clear that the mean score in Level 3 is significantly higher than both Level 2 and Level 1.

Table 4.10 *Summary of Repeated ANOVA Results Showing Significance of Related Groups for Attitudes Toward Computer Programming Languages Scale*

Subscale	Initial State (Level 1)	Traditional Classroom Approach (Level 2)	Flipped Classroom Approach (Level 3)
Enjoyment	Level 1 < Level 2 Level 1 < Level 3	Level 2 > Level 1	Level 3 > Level 1
Self-confidence	Level 1 < Level 3	-	Level 3 > Level 1

According to the figures in Table 4.10, the mean score measurement at Level 2 is significantly higher than Level 1 in terms of the Enjoyment subscale. Additionally, the mean score of the Enjoyment subscale in Level 3 is significantly higher than in Level 1. Considering from the perspective of Self-confidence, the mean score in Level 3, which was measured after applying the Flipped Classroom Approach, is significantly higher than in Level 1.

4.3 Qualitative Findings

This section presents the analysis of the findings from the semi-structured interviews held with some of the students. Student participants were questioned about their opinions regarding the programming classes taught using the Flipped Classroom Approach, and their opinions were noted and then analyzed using MAXQDA analytical software in order to reveal the themes and subthemes. The themes are “Disadvantages of Traditional Classroom Approach,” “Advantages of Flipped Classroom Approach,” “Prospective Opinions Related to Flipped Classroom,” and “Suggestions for Better Flipped Classroom” (see Figure 4.1).

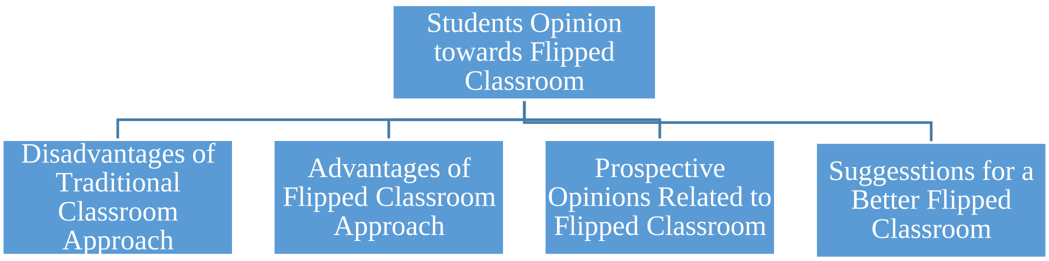


Figure 4.1 Coding Results of Qualitative Data Analysis

4.3.1 Research Question #4: What are the students' experiences and opinions about the Flipped Classroom Approach as a teaching strategy?

Four themes emerged based on the answers to this research question. These themes are "Disadvantages of Traditional Classroom Approach", "Advantages of Flipped Classroom Approach", "Prospective Opinions Related to Flipped Classroom", and "Suggestions for a Better Flipped Classroom". Each theme and its corresponding subthemes are explained in the following subsections.

4.3.1.1 Disadvantages of Traditional Classroom Approach

In the interviews with students, it was found that they had some negative thoughts towards the Traditional Classroom Approach. These opinions are analyzed and examples given from the student's comments as follow.

Crowded Classroom

The students generally talked about classrooms being overcrowded and sometimes too loud when asked for their thoughts about traditional classroom environments. Sometimes they even said that it was a problem to be able to see the whiteboard.

> *I sit at the back of the class because it is usually crowded. Therefore, I cannot follow the lessons properly or cannot concentrate at all.* [Student 10]

Teacher-centered Learning

Another common opinion of the students was that they find the traditional learning environments as teacher-centered. In other words, the lessons are taught at a certain level without regard for students at a lower level.

> *At the beginning, the teacher showed us how to do certain things; however, if we fell behind at times, the teacher was not interested in helping all of the students at the same time.* [Student 6]

It was also revealed during the interviews that some students had difficulty in following the lesson when they first encountered new course subjects.

> *The first time I encounter learning new things I ask myself what should I do. Then I continually received help from my friends on how to do the steps as I could not follow the teacher. I needed help from my classmates in order to follow the course.* [Student 8]

Lack of Lesson Time

The analysis showed that classroom time can be inadequate for some students, with opinions expressed that there is not enough time for learning to take place during the course.

> *You need longer lessons in programming courses. I don't think it's a lesson that can be taught in two or three hours like a math class. I remember having six hours of programming courses in high school.* [Student 14]

Boring Classroom Time

The interviews with the students showed that students usually come to the classes unprepared and find the lessons boring. It was also revealed that this led to a lack of participation in the classes.

> *Previously, it was very enjoyable as I prepared by watching the necessary videos, but if I do not do this, it becomes boring for me. So, I think it is very important.* [Student 9]

4.3.1.2 Advantages of Flipped Classroom Approach

Classroom Readiness

Results of the qualitative data analysis revealed a theme that the researcher titled "Advantages of Flipped Classroom Approach." The most significant subtheme for this related to students' opinions about readiness in the classroom. They talked about the positive influence of the videos they watched before attending the classes, and believed that it increased their readiness for the class topic.

> *When you watch videos before the lesson, you have prior information and comment on it when asked by the teacher. You can also follow the lessons in a better way. So I think this positively affected me.* [Student 11]

> *In a sense, it was good to prepare before coming to class. When you have extra points, this also passes the responsibility to us.* [Student 9]

As a result of students' coming to classes already prepared, they approved of the strategy as they had the opportunity to ask about matters they did not understand and to take in their own notes.

> *I came to classes having more readiness and the chance to take notes before the class when watching the videos. When I had problems, I also had the opportunity to search the Internet or look through the books.* [Student 6]

Watching Videos Repeatedly

The most significant point that students shared about lessons taught using the FC method is the possibility of watching the videos repeatedly. Students saw this aspect as significantly advantageous and therefore a subtheme of "Watching Videos Repeatedly" was created.

> *After the lesson, we cannot re-watch the teacher, but I downloaded the videos to my computer and now I have the chance to watch them on-the-go. As I live in a small village with no Internet connection, it was very convenient for me to download and watch them again. I now have an archive of programming lessons and I listed them properly in this archive including all the year's videos. It is now like a resource book.* [Student 8]

Another student, who had already taken other programming lessons and had prior knowledge stated being satisfied with the opportunity to pause the lesson and then re-watch it.

> *Even though we learned different things such as array listing, Visual Basic and C#, I thought that they had the same logic of coding. I tried to adapt it but found it was different. So, watching videos, pausing and playing them at any time I want made it easier to learn. I think it has been very efficient for me to have these videos to hand.* [Student 5]

Active Classroom Environment

Another theme analyzed from the students' opinions is that the learning environment should be more active, and as such a subtheme titled "Active Classroom Environment" was created. Under the FC method, some students stated that they were more active in the classes.

> *Without the flipped classroom method, I think the classes are too crowded for the teachers to deal with all of us. But this method helps us greatly. All of us have different learning styles; with some of us having graduated from vocational high schools and some from other types of high school. Those who have background in programming are*

> *better than the others. With the flipped classroom, I felt like I learned that class beforehand and I associated things with what I had watched previously. Thus, it is more lasting and could be more active in the lessons as a result.* [Student 5]

Another student stated that they could participate in discussions with the teacher and answer questions more easily when attending classes with prior knowledge.

> *I think we can participate more in the discussions and naturally answer the questions asked by the teacher by means of the videos we watched before the lesson. It is very different coming to lessons with no prior knowledge and attending classes already prepared.* [Student 14]

They also stated that they could more easily answer their teachers' questions during the lessons when they come having already watched the videos.

> *I can comfortably answer the questions that the teachers ask and the videos I watched before the lessons help me a lot. If I did not watch them, I would have been unable to answer the questions.* [Student 12]

The students stated that their preparations before class have increased thanks to these videos and that now they can follow the lessons.

> *When you have prior information, you can comment on teachers' opinions as you have already watched videos about it. You can also follow the classes better. I think it has had a positive effect on me.* [Student 11]

They also showed their contentment at being able to attend class with prior knowledge. Even when they do not completely understand everything in the videos, they stated that it was better than not knowing anything about the topic at all, which they suggested helped them to be more active during the lesson.

I want to give examples from the lessons. If I were someone who did not watch the videos, I would not have any information about what the teacher was talking about. But now that I watched them, I know about certain topics. Although I don't know the code in full, I can recall some of it from the videos, and this makes lessons more active and participatory. [Student 8]

Increasing Responsibility of Students' Own Learning

As a result of the qualitative analysis, it was revealed that students' pre-class tasks regarding their learning had impacted on their responsibility and awareness.

> *I think that it actually enhances the students' sense of responsibility because you have a class tomorrow and you have to watch the video. If you do not do this, there is enforcement, but also you cannot engage in the classroom tasks without having watched them. I think this is closely related to the responsibility of students, and I changed my way of studying after this method. I started to learn from videos instead of books and I looked for the videos of that class on YouTube. I frequently study with these videos as if they were my books and notebooks.* [Student 6]

Advantages for Low Prior Knowledge Students

As shown in the demographic data, there were differences between the levels of prior knowledge of students who graduated from vocational high schools and those that did not. It was discussed that vocational high school graduates have higher levels of prior knowledge, while the others are biased against programming to some extent. However, the students' opinions revealed that watching the videos for the FC had positively contributed to lessening the gap between these two student groups.

> *It is very different learning in the classroom. For instance, without flipped learning it is harder to conduct all activities in classroom lessons because they are very crowded. But the FC method helps us greatly. All of us have different learning styles; some of us graduated*

from vocational high schools and some from other high school types. Those who have prior background in the subject are better off than the others. With the flipped classroom, I felt like I had learned that class beforehand and I associated things with what I had watched previously. Thus, it is more lasting and we can be more active in the lessons as a result. [Student 5]

It was found that especially those students who had never taken programming lessons before felt that the FC method of training represented a great chance for them.

I am not a graduate of a vocational high school, I graduated from an Anatolian high school and therefore there are new things in computer-related lessons that I had never learned before. But I am very curious to learn new things and I often undertake my own research. In this way, I can comfortably develop myself by revising on the day before the lesson. Flipped classrooms, and the technique of watching videos before class, is of significant importance and helps you develop yourself. Thus, I can build upon this. [Student 6]

Similarly, those students who were not graduates of vocational high schools stated that it helped them tackle prejudice against the lessons.

Initially, I thought that I would be the least successful in the classroom because I am not a graduate of a vocational high school. I thought my classmates would exclude me, but that did not happen. I am happier now as I know more about my own level, my classmates' level and my efforts to keep up with them. [Student 8]

Also, students with similar thoughts commented that lessons under the FC model helped them to overcome their fear about the lessons.

I felt very nervous about computers and programming at first because I was not knowledgeable about programming. Later on, with the help *of the videos and the teachers, I was shocked that I achieved 100 out of 100. It was great for me.* [Student 7]

Another student who had never taken programming lessons stated that they could actively participate in the lessons.

> *I am graduate of a regular high school. I never learned about programming. So, I was more prepared for the classes thanks to the videos that I watched in the week before the lessons. These videos contributed to my knowledge a lot and helped me participate more in the lessons.* [Student 13]

Collaborative Learning

Another subtheme that arose from the students' opinions was that of "Collaborative Learning." It was seen that the students had been working on the videos before the lessons, and helping each other about any parts they did not understand. The analysis also revealed that they tried to learn from one another by discussing the topics in the videos.

It was also found that when students came together, they discussed the minutes of the videos.

> *When I am with my friends, we always talk about the videos we watched. I was asking about my mistakes that I made in the seventh minute of the video and tried to learn how to do it correctly. We discussed these questions very deeply. Some said 'no, it is not true', and I opposed them and explained why.* [Student 5]

It was observed that they used communication tools such as WhatsApp in order to connect with each other. It is obvious from the following comment that they were able to questions of each other through this communication tool with regard to the minutes of the videos.

We usually chatted on WhatsApp about what was going on in the videos and talked to each other. My friends would say there is something important in a specific minute of the video and we would then check it. It was very useful to talk about the videos. [Student 6]

Increasing Self-confidence

As a result of the qualitative analysis, it was seen that the FC positively affected students' self-confidence. When the students watch the videos, they felt safe because they would then be able to respond with less difficulty, even if questions were asked of them. They also expressed the view that if they were unable to attend classes in this way, that they might be in a state of some distress in the lessons.

> *I feel safer because I know more, so at least I can answer questions. Otherwise, I am just stuck."* [Student 12]

> *I come to classes in a good mood because our teacher is cheerful and talks to us in a good way and has a nice dialogue with us. If the teacher did not have a good relationship with us, I would exclude myself from the lessons and programming as well. I would feel unsuccessful about the lessons, but these videos have contributed a lot. I did not have the necessary resources and could not easily follow the instructions in the lessons. I also could not connect what I had learned in the past with what we were learning now. So, whilst it looked impossible, now it is much better. I can easily learn the coding and programming by watching the videos, and I also like my teacher. The teacher assigns homework, checks it afterwards, and he is more helpful. It is so nice.* [Student 8]

It can be seen from the following comments that students observed that even if they had prior knowledge about programming, their friends who are less knowledgeable attend classes with greater self-confidence having watched the videos beforehand and come to class already prepared.

It is so different, actually. As I have prior knowledge, I can myself do something in the lessons, but one of my friends had no knowledge about programming. Coming to classes having watched the videos increases our motivation and self-confidence. It changes our perspective towards lessons, such as when my classmate says 'I know programming now and I can do what I can'. They were sometimes prejudiced against programming because they had no knowledge of it, but this has been replaced by self-confidence. [Student 5]

Increasing Motivation by Using Video Questions

In interviews about the videos used in the lessons, it was found that the questions asked in the videos especially had a positive impact on student motivation. Therefore, the subtheme of "Increasing Motivation by Using Video Questions" was created. In addition, it was observed that students were fond of watching these videos. There were also opinions regarding the most useful parts of the videos being the question parts.

I was asked about one of the parts that I missed, and I understood that I had to be stricter about watching the videos. In that sense, the video questions were indeed helpful and that was the part that I regard as being the most important throughout the videos. I could open other videos on YouTube, but these specific videos were more attuned to the topic and we had to watch them. These videos were more advantageous and useful than other ones found on the Internet. [Student 14]

The students considered that the questions within the videos motivated them.

It is nice to be graded by these assignments because it controls whether or not we watched the videos, and this makes us more motivated and confident. I was not always sure about the correct answers, but at least I did my best. [Student 12]

Students reflected their ideas in the interviews that the questions in the videos made the videos more interesting rather than ordinary activities and that they helped them.

> *It is like a routine activity, the teacher would teach and then we would repeat. I sometimes used to think about taking screenshots of the steps, but thanks to the videos I do not need this anymore. The videos are more helpful for us. You go to the lessons having already prepared beforehand, and this is more impressive in the classes.* [Student 5]

Making Exam Preparation Easier

Another point that was underlined in the students' opinions regarding the FC is their feeling more comfortable while preparing for exams. As the pre-class videos are still accessible before the exams, it is easier for students to access these videos as a learning source in order to repeat what they have learned. A subtheme named "Making Exam Preparation Easier" was created based on this feedback.

> *If it was not for the videos, I would have had nothing to refer back to with nothing in writing. I only had a book, but it was not informative enough. I watched the videos continually and then took notes.* [Student 3]

Students mentioned studying before their exams and using the videos for revision of topics.

> *It was very useful for exams in all aspects as I had the opportunity to use them for revision.* [Student 11]
>
> *It was very good to revise the topics before the exams.* [Student 7]

Efficient Time in Classroom Sessions

In the analysis of the students' opinions it was stated that the FC is effective in more efficiently managing classroom time. These opinions were categorized under the subtheme of “Efficient Time in Classroom Sessions.” Thanks to the FC teaching model, students claimed that their lessons included more practice.

> *If we had a traditional classroom environment, we would not be able to experience practice in the lessons, and the teacher would not be able to allocate time for everyone in the classroom. Now, with this model, the teacher comes and asks whether or not there is a problematic area for us, and everyone can give their feedback one by one. This means that all of us have a level of readiness for the lessons.* [Student 6]

Students shared their views that if they did not use the lesson hours effectively, they would not find the opportunity to do this much practice.

> *Maybe we could leave the practice session to the following weeks, but then we would not cover all of the topics. But in this way, we can watch the videos at home and practice in the classroom.* [Student 1]

Students stated that they learned the challenging parts of the lesson subject in the class, therefore the amount of time loss decreased.

> *Problematic parts can be resolved through the practice sessions. It is more time-efficient. It is closely related to the timing. Instead of losing time in the classroom, we come to the class prepared and we directly continue with the practice activities. Thus, we can learn better. It saves a lot of time.* [Student 15]

The students indicated that the pre-lesson activities were advantageous as it gives them more time to practice.

> *We can allow more time to practice. That is, videos are very useful in using our time efficiently.* [Student 14]

Self-paced Learning

A subtheme named "Self-paced Learning." was created based on the students' opinions about FC. It was revealed that all of the students paused and played the videos, and that sometimes they even restarted from the beginning based on the topic to be learned. In this way, they were observed to self-pace their own learning process.

> *I was trying to repeat what the teacher was doing and watching the video simultaneously. When I made a mistake, I used to go back a bit on the video and find the correct answer. I can check what I missed minute by minute.* [Student 6]

Students stated that they could watch the videos by going forwards or backwards according to their own pace of learning.

> *Based on our own learning pace, we could forward or back on the video.* [Student 7]

Additionally, it was seen in the interviews that some students paused the videos and took notes whilst watching.

> *I watched the videos on different parts. I took notes and kept watching for five minutes so that I could have some sources to hand. I also paused the parts I missed through the lessons and rechecked them again.* [Student 12]

Cheap Lesson Source for Learning

It was observed that the students were quite satisfied that the videos used in the FC are reasonable and cheap as a source of learning.

Although a course book is recommended by the teachers, it is also suggested that they can be expensive to purchase.

> *The teacher recommended us a course book as a source in the very first lesson, but either I could not find it and access it, or it was*

expensive. Apart from that, downloading and watching the videos was much easier for us, it was that simple! Also, it was very nice that I could go back and view parts of the videos as many times as I want. [Student 8]

Comfortable Learning Environment

Another finding was that students feel comfortable about learning anywhere they want with the Flipped Classroom Approach, which led to the “Comfortable Learning Environment” subtheme.

> *The Flipped Classroom Approach both provides opportunity for practice and prevents working in unnecessarily noisy classrooms because everyone is focused on following the video. So, learning becomes more comfortable. You feel like you are at home. As they say, home is where you start your education; and with this method, this really takes place in the classroom. So, I find this method quite a good fit for our lessons.* [Student 6]

Also, in other interviews some students pointed out their contentment with listening to the lessons comfortably at home.

> *I am at my house; it is not like a classroom. I feel more comfortable at home and it makes learning easier and better. I can give breaks anytime I wish.* [Student 1]

Increasing Concentration on Lesson

Opinions were stated regarding the Flipped Classroom Approach having a positive effect on students’ concentration on their lessons.

> *As I know about the steps, I can behave accordingly. In the past, I used to get more help from my friends but now I can do on my own. And this increases my concentration.* [Student 8]

The students pointed to the positive effect of coming to lessons with prior knowledge, and being able to participate in the lessons through their improved concentration.

> *I can recall easily when the teacher asks because I have watched the video. I give the answer immediately and this increases my attention and concentration.* [Student 9]

Appropriate for Students' Learning

Students expressed opinions that watching videos before the lessons were more appropriate for themselves. They also stated that if they were assigned homework from a course book, they probably would not do it; however, watching videos were better and easier for them as a way to learn.

> *If our teacher told to us that we needed to study for a specific topic before the lesson and did not share any video with us, I would come to class with no preparation and I would not be able to answer any of the questions. But when they instruct us to watch these videos, I am able to gain prior knowledge about the topic. There are many such advantages…* [Student 1]

4.3.1.3 Prospective Opinions Related to Flipped Classroom

The qualitative analysis produced another theme, named "Prospective Opinions Related to Flipped Classroom," from the students' opinions in relation to the application of the Flipped Classroom Approach in their prospective classes. Having examined in which courses this approach could be used, it was suggested that the method would prove useful for programming courses and other courses that included an aspect of practical activity.

> *I would use the Flipped Classroom Approach if I were to teach programming. I could also use it with other courses related to computing.* [Student 1]

Students added their views that such a lesson structure would be very effective as a way to increase their success rate with the subject of Physics, with which they experienced difficulty in learning.

> *I wish that the Flipped Classroom Approach was applied to other courses too, such as Physics. I believe we would be more successful if the Flipped Classroom Approach was used in those courses as well.* [Student 5]
>
> *I am not good at Physics. But with this method, I could have a better chance by watching the important parts before or after the class.* [Student 11]

They also remarked that they could use this method when they become teachers in the future.

> *As we are training to become computer teachers, we will teach software such as Word and Excel to primary school students. I would make a video recording myself on how to use this software and the logic behind them and we would then practice this in the classroom.* [Student 15]

They also shared their views about the effectiveness of this teaching method for lessons which require practice.

> *I think using FC in courses that require computing and/or practice would be useful. For example, it might not be so beneficial in a subject like Educational Sciences, but for courses involving practice, it is time-saving and increases the students' readiness. So, it is more advantageous for such courses.* [Student 6]

4.3.1.4 Suggestions for a Better Flipped Classroom

Students also provided their opinions regarding the theme of "Suggestions for a Better Flipped Classroom." In response to the interview questions regarding the design of the FC environment, the students who shared their views about the pre-class videos stated that the duration of the videos could be shorter.

> *The duration is too long. This can create time problems; although personally I haven't got any problems with the videos apart from that. I mean, I was content with them. Also, the teacher's lectures were also very good.* [Student 15]

As can be seen, some students suggested that the video durations could be shortened. Similarly, another student also mentioned this.

> *The videos could have been shorter; which X teacher was very careful about; but it did not work out.* [Student 13]

Also another student suggested that the videos could be provided in parts, instead of complete topic-length videos.

> *The videos could be shorter; I don't mean essential. I don't know how to put it, but they could provide two videos per week of 30 minutes length. When you watch all 50 minutes at once, although you have the opportunity to pause and take a break, it is not very effective. It would be more efficient if they were 30 minutes long.* [Student 1]

It can also be inferred from the students' views that they were content with the questions in the videos, which are used to create a connection between them and the videos.

> *They were beneficial because, you know, the videos were too stable and the questions made us focus more on the videos.* [Student 3]

Other views about the questions on other videos showed that videos must be watched more carefully.

> *It is already a very good practice, which contributed to the videos being watched, because otherwise I could miss some parts of the video or may not listen to it very carefully. But with the videos I was aware that a question would be asked later, and so I was able to better focus on the videos.* [Student 14]

Also, some students mentioned that the number of questions should be increased. They not only pointed out their appreciation for the questions, but also suggested that the number of questions be increased.

> *Questions could have been more in number in order that we could better understand the lesson.* [Student 3]
>
> *Well, I would increase the number of questions, and I would (do something) according to the questions.* [Student 1]
>
> *Well, my suggestion is: videos are already very good; maybe at the end there should be something like a document to be graded, or questions to answer. We could answer those questions and thus prepare for the lessons. This is what I think.* [Student 9]

The views just stated regarded increasing the number of questions. In addition, students showed their contentment about being unable to continue watching the videos before answering the questions.

> *I liked this last year for instance. We could not see the rest of the video before answering the questions. That is what I liked most. Even when we skipped some parts without watching, it requires you to watch the whole video which is very cool. You have to watch it and answer the questions if you want to see the rest of the video.* [Student 8]

It was inferred from the interviews that the students were generally content with the nine events of instruction model used to design the videos. Their contentment with the introduction to the videos in connection with their prior knowledge of the context was also observed from the interviews.

> *At least you get some idea about the topic. Let me put it this way: it tells about it in relation with real life. For example, basically you need to keep everything. Everything needs to be preserved and fridges do it. So do the computers; they also store things. So they are close to each other as it is related to real life, which was reinforced through training, as well.* [Student 6]

To summarize; the students suggested the following points in relation to the Flipped Classroom Approach;

- Their contentment with the questions in the videos,
- The number of questions in the videos should be increased,
- Duration of the videos should be shortened or videos divided into parts,
- Contentment with videos designed based on the nine events of instruction.

4.4 Summary of Qualitative Findings

The results of the semi-structured interviews held with the students were analyzed thematically using the MAXQDA analytical software. As a result, themes and subthemes were created for the study. The following scheme shows the structure of the themes analyzed based on the students' opinions with regards to the flipped classroom approach.

Students Opinion towards Flipped Classroom Approach

1) Disadvantages of Traditional Classroom Approach
    a) Crowded Classroom
    b) Teacher-centered Learning
    c) Lack of Lesson Time
    d) Boring Classroom Time

2) Advantages of Flipped Classroom Approach
    a) Classroom Readiness
    b) Watching Videos Repeatedly
    c) Active Classroom Environment
    d) Increasing Responsibility of Students' Own Learning
    e) Advantages for Low Prior Knowledge Students
    f) Collaborative Learning
    g) Increasing Self-confidence
    h) Increasing Motivation by Using Video Questions
    i) Making Exam Preparation Easier
    j) Efficient Time in Classroom Sessions
    k) Self-paced Learning
    l) Cheap Lesson Source for Learning
    m) Comfortable Learning Environment
    n) Increasing Concentration on Lesson
    o) Appropriate for Students' Learning

3) Prospective Opinions Related to Flipped Classroom

4) Suggestions for a Better Flipped Classroom

CHAPTER 5

DISCUSSION AND CONCLUSION

This chapter presents discussion of research findings in the study, conclusion, suggestions for future research, and implications. This study aimed to examine the Flipped Classroom Approach (FCA) and the Traditional Classroom Approach (TCA) by comparing the mean scores of undergraduate students' computer programming self-efficacy, general class engagement, and attitudes toward computer programming in a computer programming course taught across both teaching environments. In addition, it also aimed to reveal the opinions of students who had experienced the Flipped Classroom Approach.

5.1 Computer Programming Self-Efficacy

The first research question investigated whether or not the courses delivered by way of the Flipped Classroom Approach influenced the students' computer programming self-efficacy. Computer programming self-efficacy was divided into two sub-dimensions: self-efficacy in basic programming tasks and self-efficacy in complex programming tasks. Analysis results showed that in both sub-dimensions, three measurements (no intervention, teaching with traditional classroom, and teaching with flipped classroom) significantly increased. In order to better understand the meaning of this difference, the three measurements were evaluated separately, and whether or not the FCA contributed to this change was examined.

Evaluation of self-efficacy in basic programming tasks, which was the first sub-dimension of the CPSES, showed that the measurements made after the lessons had been taught with both TCA and FCA were significantly higher than that of the initial measurement. According to this result, it can be interpreted that both the

TCA and FCA approaches contributed positively to the students' level of self-efficacy in programming. Although there was an increase seen between the measurement after teaching with TCA and then with FCA, this increase was not found to be statistically significant. Based on this finding, it cannot be interpreted that the contribution of the FCA approach to self-efficacy in basic programming was found to be better than that of the TCA approach. However, the continuation of the increase may allow us to comment that the FCA is as equally effective as the TCA. These results are corroborated by previous experimental studies in which the self-efficacy levels of an experimental FC group was found to be significantly higher than the control group (AlJaser, 2017). In another experimental study conducted with engineering students, Amresh et al. (2013) suggested that a course taught with the Flipped Classroom Approach positively influenced the computing self-efficacy levels of undergraduate students. There are also other previous studies in which the Flipped Classroom Approach was used on programming courses. In these studies, the FC was seen to positively contribute to the students' self-efficacy levels in programming. For example, Souza and Rodrigues (2015) conducted an empirical study in the context of an Introduction to Programming course, with students in the FC group found to have higher self-efficacy and performance levels compared to those of the traditionally taught group. Accordingly, researchers have suggested that the FC has the potential to increase programming self-efficacy and performance in introductory programming courses. In parallel, Özyurt and Özyurt (2018) investigated the effects of using the Flipped Classroom Approach in an introductory programming course on software engineering students' self-efficacy and success, and found that the FC had a positive effect on both attributes.

Evaluation of the results associated with self-efficacy in complex programming tasks, the second sub-dimension of the CPSES, showed a statistically significant increase seen among all three measurements of the students' self-efficacy levels. Based on this result, it can be concluded that the FCA contributes positively to students' computer programming self-efficacy levels in complex programming tasks. Accordingly, it can be suggested that courses delivered with FCA may be useful for increasing computer programming self-efficacy levels. In addition, the significant difference in self-efficacy in complex programming revealed that the FC may be influential in gaining high-level skills in programming. These results accord with earlier studies which showed that FCA was effective in raising students' high-order thinking levels (Alsowat, 2016). In the same way, one of the most important advantages of the FC is that it encourages students to use high-level thinking skills during classroom activities (Bergmann & Sams, 2012; Strayer, 2012). When the studies in the literature are evaluated together, the advantages of the FC to higher order thinking skills of students can be interpreted in a positive way to students' self-efficacy in complex programming levels.

Based on the aforementioned results, this study suggested that FC can be used as a teaching approach in improving students' self-efficacy. A number of studies found a positive association between students' self-efficacy and their achievement. For example, in the study conducted by Yusuf (2011), the effect of self-efficacy levels on students' academic achievements was investigated. As a result of path analysis in that study, it was concluded that self-efficacy level had a direct effect on the academic achievement of the students. Another study with parallel results was conducted by Pajares and Miller (1994), in which path analysis found a significant relationship between performance and self-efficacy levels. In addition, self-efficacy was examined with regards to its effects on success in the fields of programming education. In such studies, it was found that self-efficacy was an important construct in the teaching of programming. For example, Nilsen and Larsen (2011) underlined that low levels of self-efficacy and motivation were found to be underlying reasons for lack of understanding and algorithmic structure in terms of programming courses. Also, Lishinski et al. (2016) focused on self-efficacy and its effects in their study; revealing that self-efficacy was one of the most significant elements in terms of performance in the Computer Science field. Moreover, Altun and Mazman (2012) and Baser (2013) expressed that low levels of self-efficacy at the beginning of a programming course might be attributable to the students failing the course. As a result, in this context, it can be suggested that students' self-efficacy is a significant indicator of student performance and success in programming courses. When the result of the current study, with the FC having positively affected self-efficacy, is considered with studies that showed a positive correlation between students' self-efficacy levels and academic achievement, the current study's finding takes on increased importance.

Quantitative data were complemented with qualitative data obtained from semi-structured student interviews in order to determine the reasons for which characteristics of the FCA affect students' increased self-efficacy levels. The results showed that videos watched by the students before each lesson helped them to more easily prepare for their class. In addition, students' coming prepared for their course contributed positively to their interaction with their teachers, as well as strengthening their interaction with their peers. Students stated that these benefits provided by the FC could have a positive impact on their self-efficacy. In addition, students with low levels of prior knowledge, i.e., those who had not previously taken a programming course, were able to watch the instructional videos before attending class, helping them to attain similar levels of prior knowledge as those with actual higher prior knowledge levels. The students underlined this to be a very important advantage for them. These results are similar to those obtained from other qualitative studies found in the literature. For example, according to a study conducted by Bhagat et al. (2016), the Flipped Classroom Approach significantly contributed to the performance of low achievers. Based on these findings, it can be interpreted that this benefit of FC teaching for students with low background knowledge of programming may be beneficial to reducing high dropout rates, which is considered one of the most important problems in programming education. The opportunity to re-watch videos even after class was also indicated by the interviewed students as one of the reasons that contributed to their self-efficacy. Additionally, after analysis of the students' interviews, some of the themes that emerged as advantages of the FC could be considered as helpful findings in explaining the impact of the FC on students' self-efficacy. Some of these themes that emerged in the qualitative analysis were "Classroom Readiness," "Increasing Responsibility of Students'

Own Learning," "Increasing Self Confidence," and "Active Classroom Environment." In these themes, the advantages of being prepared, feeling safer, and knowing what to do in the class can be interpreted as having had a positive effect on self-efficacy levels. In the interviews, it was found that students' experienced taking responsibility for their own learning through watching videos provided to them, which in a sense meant that they were controlling their own learning. This responsibility may have helped increase their confidence in the programming course. If the course was not taught through the FCA, the students could have attended their course unprepared and the practice activities could have been more difficult for them as a result, and thereby could have negatively impacted on their self-efficacy levels in programming. In summary, it could be interpreted that the advantages provided by the FCA may have increased the confidence of students in their programming lessons and may have positively impacted on their self-efficacy levels.

5.2 General Class Engagement

The second research question explored whether or not the course delivered using the Flipped Classroom Approach impacted on the students' general class engagement. General class engagement is divided into Cognitive Engagement, Behavioral Engagement, and Emotional Engagement. The study results indicated that except for Cognitive Engagement, the FCA positively influenced both Behavioral Engagement and Emotional Engagement.

Despite the observable increase of cognitive engagement after each intervention, neither increase was found to be statistically significant. Accordingly, the FCA did not positively contribute to the cognitive dimension of general class engagement. However, it was revealed that the FCA positively affected students' self-efficacy in complex programming. Additionally, it is stated that the FCA provides an environment in which students have a high attendance rate and students can participate effectively in deep learning activities (Hung, 2014). In addition, cognitive engagement is widely known to be related to higher order thinking skills, critical thinking, and deep learning strategies. Yet, the FCA did not lead to a significant impact on cognitive engagement, hence the result was partly surprising.

The results of this study indicate that the FCA had a significant positive impact on emotional engagement and behavioral engagement; meaning that students were both emotionally and behaviorally engaged and involved in their classroom work and activities. These results accord with previous research studies which showed that after the FCA was applied, student engagement increased (Nouri, 2016), active learning was enhanced (Eichler & Peeples, 2016), students became more active and had more chances to perform practice work (Bergmann & Sams, 2012), teachers had the advantage of being able to listen to and interact with students' views (Stone, 2012), and students were more likely to participate in problem-solving and in-class activities (Moore & Chung, 2015). There have also been other studies which have focused on the positive effects of FC on students' in-class participation. In his doctoral dissertation, Clark (2013) revealed that the level of attendance in a Mathematics FC course was higher. These results are consistent with that of McLaughlin et al. (2013), who applied the FCA to a pharmaceutics course and through qualitative analysis found that students were more active in the course. These results are similar to those of the current study's qualitative results, suggesting that students have higher levels of classroom readiness and that they can better follow in-class activities. Apart from other course areas, there are also studies to be found in the literature which suggests an increase in student engagement when the FC is applied as a teaching strategy in programming courses (Puarungroj, 2015). To summarize, the current study suggests that the FCA can provide students with an active learning environment and become a potential teaching strategy for increasing students' general class engagement.

It is widely accepted that having high level engagement in a course is very important for students in many aspects, as a significant increase in engagement can result in a pronounced improvement in students' course success and motivation (Fredricks et al., 2004). A recent study showed that student engagement is positively related to academic success and critical thinking (Carini et al., 2006; Pintrich & De Groot, 1990). The result of the current study, therefore, is meaningful and has many important implications as it empirically reveals that FCA positively impacted on students' engagement in a programming course. Based on these results, the current study suggests that students' academic achievement is likely to be enhanced when FCA is applied as a teaching strategy in programming courses.

Through the current study's qualitative data analysis and the survey conducted, the reasons behind the increase in general class engagement in lessons taught with FC were examined. As a result of the FC surveys applied to the participant students, the majority who were taught with the FCA in their programming course expressed that they were better able to communicate with their instructors and peers on a more frequent basis. The increased communication with their instructors and peers may have contributed positively to the students' engagement levels. In addition, students stated that they experienced increased levels of in-class participation in the FCA course compared to when the course was applied through the TCA. These results can be seen as a finding that supports the increase in student engagement levels.

When the reasons for increased course attendance were investigated through semi-structured student interviews, it was stated that the videos watched before attending classroom lessons contributed to their attendance. The students stated that this increased their courage to answer questions asked by the teacher in the classroom. Besides, they also mentioned that it was beneficial for them to allocate class time to more practice. From these views it can be concluded that students see the FCA as advantageous by watching course videos prior to attending class, and that this has a knock-on effect on the activities undertaken in the course. This activity may also be considered a factor that contributes positively to student engagement levels.

5.3 Attitudes toward Computer Programming Languages

The third research question in the study focused on exploring the effects of lessons taught with the Flipped Classroom Approach on students' attitudes toward computer programming languages. There are four sub-dimensions of the attitude scale which are Importance, Enjoyment, Self-confidence, and Motivation. The study results indicated that among these four sub-dimensions, the FCA had a significant positive impact merely on students' attitudes about Self-confidence and Enjoyment on a programming course.

With regards to the Self-confidence sub-dimension, students' attitudes toward computer programming languages was significantly improved with the FCA in comparison to no intervention. From this, it can be deduced that the FCA does have a positive effect on students' attitudes toward computer programming languages in the Self-confidence sub-dimension. These results suggest that it might be better to teach course with the FCA in order for students to have increased self-confidence in the course. Previous studies reported partly similar findings, showing that the FCA had a positive impact on students' attitudes toward an undergraduate mathematics course (Guerrero, Beal, Lamb, Sonderegger, & Baumgartel, 2015). The same study also reported that FC provided teachers with a classroom environment that allowed for the application of student-centered and problem-solving techniques by using class time more efficiently.

In terms of Enjoyment, students' attitudes were found to be significantly increased after application of the FCA in favor of the FC in comparison with the measurement conducted at the beginning of the programming course. This finding is a clear indication of how efficient the FCA is in making courses more enjoyable and entertaining. Furthermore, the TCA was found to be significantly better than no intervention. These results suggest that either the TCA or the FCA positively impacts on students' attitudes toward enjoyment in a programming class. In accordance with the current study's results, previous research demonstrated that students' attitudes toward mathematics when applied with the FCA showed a statistically significant increase in both the Enjoyment and Self-confidence sub-dimensions (Esperanza et al., 2016). The fact that the current study and another from the literature contributed to the same two attitude sub-dimensions may be interpreted that the FCA can have a positive contribution to students' attitudes regarding the Self-confidence and Enjoyment sub-dimensions. In other studies, students' attitudes toward courses were found to improve significantly and successfully after the FCA was applied (Young, 2015; Zack, Fuselier, Graham-Squire, Lamb, & O'Hara, 2015). Despite corroborative findings, conflicting results were also reported. For example, in a study conducted by Özyurt and Özyurt (2018), FC did not have a statistically significant effect on students' attitudes toward programming.

The current study suggests that the FCA might have significant potential in playing a crucial role in the development of students' course performance. Corroborating evidence has been provided in previous studies. For example, Baser (2013) found a significant and positive correlation between students' attitudes toward a course and achievements in programming. According to this correlation, the fact that a significant increase was seen in students' attitudes toward programming languages in a programming course where FC was employed as a teaching strategy, can be interpreted that FC may be beneficial in terms of students' achievements in programming courses. It is stated in the literature that students can experience problems and difficulties in programming courses and that such courses have a known high dropout rate. Korkmaz and Altun (2014) argued that students may have negative attitudes toward programming and that they may a reason behind such problems. The positive contribution of the FC to students' attitudes toward programming languages can be considered as significant evidence that the FC can be beneficial in terms of programming courses.

During the current study's student interviews, it was investigated as to which aspects of the FC contributed to a positive change in students' attitudes toward programming. The students expressed that the classes were more enjoyable thanks to the videos they watched prior to attending class, and that they felt safer because they knew more about each class before it started. They also underlined that coming to class after watching the videos decreased their demands to request help from others and contributed positively to their self-efficacy in lessons. These advantageous aspects of the FC, as indicated by the students of the current study, can be considered as reasons for the high levels of positive attitudes exhibited towards programming courses.

5.4 Students' experiences and opinions about Flipped Classroom Approach as a Teaching Strategy

The interviews held with the students revealed that they were satisfied with the Flipped Classroom Approach. The reason for their satisfaction may have been due to aspects of the FC that are seen as advantageous, and these can be grouped into in-class, out-of-class, and general advantages.

Based on the in-class advantages, students stated their satisfaction that they attended classes with prior knowledge; arriving pre-prepared thanks to the extracurricular activities and therefore in-class lesson hours were more active. McLaughlin et al. (2013) put forward a parallel result after their study proved that when FC was applied, students' preparation was reflected positively in their participation in classroom lessons. Students in the current study also mentioned their opportunity to undertake practice activities in the classroom lesson and that they were able to answer the teacher's questions more easily. In addition, they claimed that they could figure out unclear issues which they could learn by themselves outside of the class; leaving the lesson hours to cover more activities. This could be seen as a reason to support their opinion that the lessons were more effective. Also, they mentioned their contentment with the opportunity to repeat and ask about any unclear subject areas during the lessons, which could also be considered as another finding that the lessons were seen as more active and participatory. The studies show that learning programming requires much more work and that practicing is important (Brito & De Sá-Soares, 2014). Therefore, it is important that programming lessons be assessed better and include more practice time. The Flipped Classroom Approach can be considered as beneficial in terms of utilizing time more efficiently during lessons so that more time can be allocated to practice activities.

When the opinion of the students were analyzed with regards to their out-of-classroom studies, it was seen that the students' opinions focused on their training being carried out in a more comfortable atmosphere. Another study in this direction was conducted by Nouri (2016), in which the results showed that the Flipped Classroom Approach environment provided students with a more flexible learning environment. Students also stated that they could ask questions to each other and help one another while studying with the FCA. This may in turn lead students to interact with each other inside and/or outside of the classroom, and thereby take on increased responsibility for their own self-learning process. Furthermore, it was revealed that the interactive environment provided with the questions in the videos made what were ordinary instructional videos much more attractive. Therefore, it can be inferred that these kinds of interactive videos should be introduced or further developed within the Flipped Classroom Approach.

The students' interviews revealed that the Flipped Classroom Approach had general advantages for students. They reflected on their contentment with the opportunity to learn by themselves. The results of other studies found in the literature showed that the Flipped Classroom Approach supported students who wanted to learn by themselves (Davies, Dean, & Ball, 2013; Gilboy, Heinerichs, & Pazzaglia, 2015; Morgan, 2014). The reason for this may be due to the ability to re-watch videos. The possibility to pause the videos and take notes may be representative of promoting self-paced learning. Also, the students mentioned studying by taking notes when watching the videos, and as a result of the current study, this may have raised awareness for their self-learning. Therefore, the FCA may be seen as having a positive effect on the students' skills to control their own learning. It was also found that during exam preparations, students' ability to reaccess the videos and the fact that the videos went hand in hand with the lesson plan helped them to prepare for their exams. It was also found that students considered this approach as an affordable source of learning. Besides, it was concluded that the model helped those students with lesser prior knowledge about programming to come to class with less fear thanks to their preparation for lessons in advance. Students were also observed to have overcome beliefs that they would automatically fail such a course. Esperanza et al. (2016) proved in his study with the Flipped Classroom Approach that students were more self-confident and enjoyed their lessons more, which parallels the results of the current study. Especially, it was observed that students were provided with the opportunity to catch up with their friends who were more knowledgeable than them. These views can be considered as a finding that the Flipped Classroom Approach is beneficial for students with lower levels of pre-existing knowledge. Nouri (2016) also supported these results in the findings of his work. In his study conducted with prospective teachers, students reflected their views that they could use the Flipped Classroom Approach in their own lessons when they qualified as teachers. Also, the interviewed students' views from the current study demonstrated that the Flipped Classroom Approach could also be useful for their Physics lessons in which they faced some difficulties. Their views that they could use this approach in the future could be considered as evidence for their contentment with the FCA. With regards to the question as to what courses the FCA would be suited to, the students generally stated that it suited lessons which require an element of practice.

While the students' contentment with the Flipped Classroom Approach demonstrated some advantages to this approach, their views on the disadvantages of traditional teaching were also revealed. The students stated that in the traditional teaching environment, that the classrooms were overcrowded. Thus, it was found that the lessons' effectiveness was lessened due to noise levels in the classroom. It was also concluded that the more active role of the teacher in the traditional classroom caused a lack of student participation. The students mentioned that traditional lessons were covered mostly by lecturing, which supports the concept that the traditional learning environment is teacher-centered. Other thoughts mentioned by the students about the traditional learning environment were that where lessons are mostly taught as lecture-based, that left little or no time for practice activities. This finding showed that the teaching hours are not long enough for some of the lessons. As to the hardships of teaching programming, Brito and De Sá-Soares (2014) mentioned that it was not student-centered and that students were unable to receive instant feedback. The advantages of the Flipped Classroom Approach show that it can be considered as an effective teaching strategy for programming lessons as it can make the process more productive.

5.5 Conclusion

Programming is emerging as a skill that is important to learn for most professions and science fields today. Research shows that there are difficulties experienced in the learning and teaching of programming. In order to overcome these difficulties, the FCA was used as a teaching strategy in order to analyze whether or not it made the teaching of programming any more effective. Within the scope of the current study, CEIT students who attended an Introductory Programming course were taught with TCA for the first five weeks of their course and then subsequently with FCA in the following five weeks. The aim of the study was to investigate the effects of the FCA on students' self-efficacy in programming, their general class engagement, and their attitudes toward computer programming languages.

In terms of self-efficacy for programming, it was observed that the FC provided a significant increase in the dimension of students' Self-efficacy in Complex Programming. This result can be interpreted as a finding that FC contributes positively to students' self-efficacy towards programming. Considering the effects of FC on the general class engagement of the students, it was seen that FC provided a significant increase in the dimensions of Emotional Engagement and Behavioral Engagement of the students. It can be evaluated that the lessons taught with FC increased students' engagement. Given the students' attitudes toward programming course, it was observed that teaching with FC significantly increased the levels of the students' attitudes in terms of the Self-confidence sub-dimension. This result shows that FC contributes positively to attitudes toward programming languages of students.

In the interviews with students after having taught with the FCA, their opinions about the teaching of programming with FC were examined. From these interviews, it was revealed that students found it to be advantageous when the topic was given by video in advance of the classroom lesson. Thanks to these videos, students are able to attend class pre-prepared. They expressed their opinion that the lessons passed more actively, that they experienced increased levels of concentration in the lesson, and that the course time was used more efficiently. Students stated that teaching with the FCA, which includes out-of-class learning, provides them with a freer and more comfortable learning environment. In addition, an important advantageous aspect of the FC is that it can provide more equaled opportunities for students, especially in helping them to address the traditional disadvantage of not having previously taken a programming course and therefore it being their first ever programming course, which can be a daunting experience. In short, it can be concluded that the FC may offer advantages to students with low level or no prior knowledge.

Regarding the students' opinions about their future teaching career, they suggested that they would use the FC as a teaching strategy once they became qualified teachers, and would recommend others to use it also in their other courses. This result showed that the students were satisfied with the use of the FCA for their programming course, especially when it came to the interactive questions in the videos and their positive contributions to their learning experiences. The increase of such questions in the videos might also help them to become more motivated while watching them. These questions were shown in order to provide students with the opportunity to understand the most important topics of the subject, and also to check whether or not the videos were in fact being watched. Students also proposed that the videos should not be too long or that they should be divided up into parts. They added that it may be better if the number of questions in the videos was increased and that there could be different interactive activities included with the videos. These suggestions can be taken into consideration for the design of future courses in which the FC will be employed as a teaching strategy.

Overall, it was seen that the students had positive opinions towards the programming course delivered using the FCA as a teaching strategy. The students also stated that the use of the FC for other courses may also be good for them in the future. These results also showed that this approach may be appropriate for the learning styles of many of today's students. Moreover, it can be seen as an important finding that the FC is a teaching approach that has the potential to increase the efficiency of programming education on the basis of self-efficacy in computer programming and attitudes toward computer programming languages, which are among the key factors affecting students' success in programming education. In addition, considering the importance of students' active participation in the learning of programming, the positive contribution of the FCA to students' general class engagement can be considered as another finding that the FC may be a teaching strategy that can be efficiently employed in programming education.

5.6 Suggestions for Future Research

- FCA was observed to have positive results for an undergraduate-level programming course. It could also be applied to other courses that require practice at the high school or primary school level.
- The study was conducted with a sample group. The study could be repeated with an increased range of sample groups.
- The length of the current study was 10 weeks. Its long-term effects could be analyzed by extending the length of the practice.
- In future studies, the success rate of the programming students who receive training with the Flipped Classroom Approach could be analyzed.

- An experimental, multi-group study could be conducted in order to assess the effect of the Flipped Classroom Approach on students' achievement in lessons.
- The study indicated that students were generally content with the questions in the videos. Using this interactive video design, other interactive video media for use in the Flipped Classroom Approach could be researched.
- Research could be made about the number of questions and their frequency. Also, an additional study could be conducted in order to increase the interactivity levels of the videos.
- The current study was applied on a programming course. It could also be applied to other courses that require practice activities to be conducted in order to test its wider effectiveness.
- A design-based research could be conducted that focuses on the design of FC's out-of-class content such as videos.
- The current study focused on students as the sample. Further research could also be conducted which focused on the instructors.

5.7 Implications

- Teachers who desire to cover lesson hours more effectively and save time for practice could utilize the Flipped Classroom Approach in their lessons.
- FCA could be utilized in lessons such as programming, mathematics, and physics in which an element of practice is important.
- Utilizing the FCA can be beneficial in classes which include students with different levels of prior knowledge.
- The videos used in the FCA should be designed in such a way as to draw the students' attention. Thus, they could be designed to include a level of interactivity; including questions which enable students to interact.
- The length or partiality of the videos could be important to helping the students listen more attentively. While designing pre-lesson videos, this aspect could be taken into consideration.

- In the extracurricular environment, a video viewing environment could be provided in order to enable the students to interact with each other. Thus, the opportunity for students to gain help from each other while preparing for lessons through videos in the extracurricular environment can be increased.
- In order for students to do what they need to do within an extracurricular environment, necessary precautions should be taken in order to avoid students attending class without having first watched the videos.
- Teachers with hesitation about the Flipped Classroom Approach could be provided with training about how to prepare interactive pre-lesson videos in order to overcome this hesitation.
- It could be beneficial if teachers made their lessons more reachable by making videos to help students prepare for their exams, and to watch them as a means of review their lessons.
- Trying out new teaching methods for students who are surrounded by information technologies in life could contribute positively to their learning and could be a more effective method.
- In this time, when open course materials and lecturing with videos are becoming more widespread, the Flipped Classroom Approach could be easily integrated into course syllabi.

REFERENCES

Abeysekera, L., & Dawson, P. (2015). Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research. *Higher Education Research & Development*, *34*(1), 1-14. https://doi.org/10.1080/07294360.2014.934336

Akpinar, Y., & Altun, A. (2014). Bilgi Toplumu Okullarında Programlama Eğitimi Gereksinimi. *Elementary Education Online*, *13*(1), 1-4.

Al-Ghamdi, M., & Al-Bargi, A. (2017). Exploring the Application of Flipped Classrooms on EFL Saudi Students' Speaking Skill. *International Journal of Linguistics*, *9*(4), 28. https://doi.org/10.5296/ijl.v9i4.11729

Alhazbi, S. (2017). Using flipped classroom approach to teach computer programming. In *Proceedings of 2016 IEEE International Conference on Teaching, Assessment and Learning for Engineering* (pp. 441-444). IEEE. https://doi.org/10.1109/TALE.2016.7851837.

AlJaser, A. M. (2017). Effectiveness of Using Flipped Classroom Strategy in Academic Achievement and Self-Efficacy among Education Students of Princess Nourah bint Abdulrahman University. *English Language Teaching*, *10*(4), 67-77. https://doi.org/10.5539/elt.v10n4p67.

Alsowat, H. (2016). An EFL flipped classroom teaching model: Effects on English language higher-order thinking skills, student engagement and satisfaction. *Journal of Education and Practice*, *7*(9), 108-121.

Altun, A., & Mazman, S. G. (2012). Programlamaya ilişkin öz yeterlilik algısı ölçeğinin Türkçe formumun geçerlilik ve güvenirlik çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, *3*(2), 297-308.

Amresh, A., Carberry, A. R., & Femiani, J. (2013). Evaluating the effectiveness of flipped classrooms for teaching CS1. In *Proceedings - Frontiers in Education Conference* (pp. 733-735). IEEE. https://doi.org/10.1109/FIE.2013.6684923.

Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *TOJET: The Turkish Online Journal of Educational Technology*, *8*(1), 26-32.

Ayçiçek, B., & Yanpar Yelken, T. (2018). The Effect of Flipped Classroom Model on Students' Classroom Engagement in Teaching English. *International Journal of Instruction*, *11*(2), 385-398. https://doi.org/10.12973/iji.2018.11226a.

Bandura, A. (1986). *Social foundation of thought and action: A social-cognitive view*. Englewood Cliffs, NJ: Prentice-Hall.

Barnes, D. (2002). Teaching Introductory Java through LEGO MINDSTORMS Models. *ACM SIGCSE Bulletin*, *34*(1), 147-151. https://doi.org/10.1145/563340.563397.

Baser, M. (2013). Attitude, gender and achievement in computer programming. *Middle-East Journal of Scientific Research*, *14*(2), 248-255. https://doi.org/10.5829/idosi.mejsr.2013.14.2.2007.

Bas-Sarmiento, P., Fernández-Gutiérrez, M., Baena-Baños, M., & Romero-Sánchez, J. M. (2017). Efficacy of empathy training in nursing students: A quasi-experimental study. *Nurse Education Today*, *59*, 59-65. https://doi.org/10.1016/j.nedt.2017.08.012.

Bayman, P., & Mayer, R. E. (1988). Using Conceptual Models to Teach BASIC Computer Programming. *Journal of Educational Psychology*, *80*(3), 291-298. https://doi.org/10.1037/0022-0663.80.3.291.

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, *39*(2), 32-36. https://doi.org/10.1145/1272848.1272879.

Bergmann, J., & Sams, A. (2012). *Flip Your Classroom : Reach Every Student in Every Class Every Day*. Eugene, Or: International Society for Technology in Education. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=e000xww&AN=475951&site=eds-live&authtype=ip,uid.

Bhagat, K. K., Chang, C.-N., & Chang, C.-Y. (2016). The Impact of the Flipped Classroom on Mathematics Concept Learning in High School. *Journal of Educational Technology & Society*, *19*(3), 134-142. https://doi.org/10.2307/jeductechsoci.19.3.134.

Boyle, T., Bradley, C., Chalk, P., Jones, R., & Pickard, P. (2003). Using Blended Learning to Improve Student Success Rates in Learning to Program. *Journal of Educational Media*, *28*(2–3), 165-178. https://doi.org/10.1080/1358165032000153160.

Brito, M. A., & De Sá-Soares, F. (2014). Assessment frequency in introductory computer programming disciplines. *Computers in Human Behavior*, *30*, 623-628. https://doi.org/10.1016/j.chb.2013.07.044.

BSA The Software Alliance. (2017). *The Growing $1 Trillion Economic Impact of Software*. Retrieved from https://software.org/wp-content/uploads/2017_Software_Economic_Impact_Report.pdf.

Bull, G., Ferster, B., & Kjellstrom, W. (2012). Connected classroom-inventing the flipped classroom. *Learning and Leading with Technology*, *40*(1), 10.

Butz, C. J., Hua, S., & Maguire, R. B. (2004). A Web-based intelligent tutoring system for computer programming. *Proceedings - IEEE/WIC/ACM*

International Conference on Web Intelligence, WI 2004, 159–165. https://doi.org/10.1109/WI.2004.10104

Cabi, E. (2004, July). *Web Destekli Pascal Öğretimine Yönelik Örnek Bır Çalışma*. Paper presented at the XIII. Ulusal Eğitim Bilimleri Kurultayı, Malatya, Turkey.

Carini, R. M., Kuh, G. D., & Klein, S. P. (2006). Student engagement and student learning: Testing the linkages. *Research in Higher Education*, *47*(1), 1-32. https://doi.org/10.1007/s11162-005-8150-9.

Clark, K. R. (2013). *Examining The Effects Of The Flipped Model Of Instruction On Student Engagement And Performance In The Secondary Mathematics Classroom : An Action Research Study*. Doctoral Dissertation, Capella University, Minneapolis, Minnesota. Retrieved from http://caite.fed.cuhk.edu.hk/projects/wp2016/wp-content/uploads/2016/06/Examining-the-effects-of-the-flipped-model-of-instruction-on-student-engagement-and-performance.pdf.

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

Creswell, J. W., & Miller, D. L. (2000). Determining validity in qualitative inquiry. *Theory into practice*, *39*(3), 124-130.

Çetin, İ., & Top, E. (2014). ACE Cycle in Programming Education by Using Visualization. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *5*(3), 274-303. https://doi.org/10.16949/turcomat.72987.

Çölkesen, R. (2017). *Veri Yapıları ve Algoritmalar*. Papatya Yayıncılık.

Davies, R. S., Dean, D. L., & Ball, N. (2013). Flipping the classroom and instructional technology integration in a college-level information systems spreadsheet course. *Educational Technology Research and Development*, *61*(4), 563-580. https://doi.org/10.1007/s11423-013-9305-6.

de Jesus Gomes, A., Mendes, A. J., & Marcelino, M. J. (2015). Computer Science Education Research: An Overview and Some Proposals. In R. Queirós (Ed.), *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming* (pp. 1-29). IGI Global.

Demirer, V., & Sak, N. (2016). Programming Education and New Approaches Around The World and in Turkey / Dünyada ve Türkiye'de Programlama Eğitimi ve Yeni Yaklaşımlar. *Eğitimde Kuram ve Uygulama*, *12*(3), 521-546. https://doi.org/10.17244/eku.05743.

Durak, G. (2013). *Programlama Dillerinin Cevrimici Ogretimi: Ogrenenlerin Tutumlarinin, Memnuniyetlerinin ve Akademik Basarilarinin Incelenmesi* (Unpublished Doctoral dissertation). Anadolu Universitesi Sosyal Bilimler Enstitusu.

Eichler, J. F., & Peeples, J. (2016). Flipped classroom modules for large enrollment general chemistry courses: a low barrier approach to increase active learning and improve student grades. *Chemistry Education Research and Practice*, *17*(1), 197-208. https://doi.org/10.1039/C5RP00159E.

Ekmekci, E. (2017). The flipped writing classroom in Turkish EFL context: A comparative study on a new model. *Turkish Online Journal Of Distance Education (TOJDE)*, *18*(2), 151-167. https://doi.org/10.17718/tojde.306566.

Enfield, J. (2013). Looking at the Impact of the Flipped Classroom Model of Instruction on Undergraduate Multimedia Students at CSUN. *TechTrends*, *57*(6), 14-27. https://doi.org/10.1007/s11528-013-0698-1.

Ersoy, H., Madran, R. O., & Gülbahar, Y. (2011). Programlama Dilleri Öğretimine Bir Model Önerisi : Robot Programlama. In *Akademik Bilişim'11 - XIII. Akademik Bilişim Konferansı Bildirileri*, (pp. 731-736). https://ab.org.tr/ab11/kitap/ersoy_madran_AB11.pdf.

Eryılmaz, A. (2014). Üniversite öğrencileri için derse katılım ölçeklerinin geliştirilmesi. *Uşak Üniversitesi Sosyal Bilimler Dergisi*, 18, 203-214.

Esperanza, P., Fabian, K., & Toto, C. (2016). Flipped Classroom Model: Effects on Performance, Attitudes and Perceptions in High School Algebra. In K. Verbert, M. Sharples, & T. Klobučar (Eds.), *Adaptive and Adaptable Learning, Proceedings of the 11th European Conference on Technology Enhanced Learning, EC-TEL* (Vol. 9891, pp. 85-97). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-45153-4_7.

Estes, M., Ingram, R., & Liu, J. C. (2016). A Review of Flipped Classroom Research, Practice, and Technologies. *International HETL Review*, *4*, Article 7. https://www.hetl.org/a-review-of-flipped-classroom-research-practice-and-technologies/.

Fetaji, M., Fetaji, B., Sukic, C., Gylcan, A., & Ebibi, M. (2016). Case Study Analyses of the Impact of Flipped Learning in Teaching Programming Robots. *TEM Journal*, *5*(4), 560-565. https://doi.org/10.18421/TEM54-21.

Foldnes, N. (2016). The flipped classroom and cooperative learning: Evidence from a randomised experiment. *Active Learning in Higher Education*, *17*(1), 39-49. https://doi.org/10.1177/1469787415616726.

Fraenkel, J. R., & Wallen, N. E. How to design and evaluate research in education 2006. *Mc Grawall Hill*.

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School Engagement: Potential of the Concept, State of the Evidence. *Review of Educational Research*, *74*(1), 59-109. https://doi.org/10.3102/00346543074001059.

Giannakos, M. N., Pappas, I. O., Jaccheri, L., & Sampson, D. G. (2017). Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies*, *22*(5), 2365-2382. https://doi.org/10.1007/s10639-016-9538-1.

Gilboy, M. B., Heinerichs, S., & Pazzaglia, G. (2015). Enhancing Student Engagement Using the Flipped Classroom. *Journal of Nutrition Education and Behavior*, *47*(1), 109-114. https://doi.org/10.1016/j.jneb.2014.08.008.

Guerrero, S., Beal, M., Lamb, C., Sonderegger, D., & Baumgartel, D. (2015). Flipping Undergraduate Finite Mathematics: Findings and Implications. *PRIMUS*, *25*(9-10), 814-832. https://doi.org/10.1080/10511970.2015.1046003.

Gunes, U. D. (2015, October 17). Milli gücün anahtarı yerli yazılım. *Yeni Şafak*. Retrieved from https://www.yenisafak.com/ekonomi/milli-gucun-anahtari-yerli-yazilim-2325143.

Gunuc, S., & Kuzu, A. (2015). Student engagement scale: development, reliability and validity. *Assessment & Evaluation in Higher Education*, *40*(4), 587-610. https://doi.org/10.1080/02602938.2014.938019.

Haladyna, T., Shaughnessy, J., & Shaughnessy, J. M. (1983). A Causal Analysis of Attitude toward Mathematics. *Journal for Research in Mathematics Education*, *14*(1), 19-29. https://doi.org/10.2307/748794.

Helminen, J., & Malmi, L. (2010). Jype - a program visualization and programming exercise tool for Python. In *SOFTVIS 2010, Proceedings of the 5th International Symposium on Software Zisualization* (pp. 153-162). ACM. https://doi.org/10.1145/1879211.1879234.

Hew, K. F. (2018). Unpacking the Strategies of Ten Highly Rated MOOCs: Implications for Engaging Students in Large Online Courses. *Teachers College Record*, *120*(1), n1.

Hanus, D., Marulo, F., & Bauer, P. (2017). *Aviation and the 4th Industrial Revolution: The Prominent Role of Networks of Excellence PERSEUS View project Development of an aeronautical crashworthy seat View project*. https://doi.org/10.19249/ams.v96i2.314

Hung, H.-T. (2014). Flipping the classroom for English language learners to foster active learning. *Computer Assisted Language Learning*, *28*(1), 81-96. https://doi.org/10.1080/09588221.2014.967701.

Ibrahim, M., & Callaway, R. (2014). Students ’ Learning Outcomes and Self-efficacy Perception in a Flipped Classroom. In T. Bastiaens & G. Marks (Eds.), Proceedings of *E-Learn World Conference on eLearning in Corporate, Government, Healthcare, and Higher Education 2014* (pp. 899-908). Chesapeake, VA: Association for the Advancement of Computing in Education (AACE).

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: A need assessment analyses. *Turkish Online Journal of Educational Technology, 9*(2), 125-131.

Jamaludin, R., & Osman, S. Z. M. S. (2014). The Use of a Flipped Classroom to Enhance Engagement and Promote Active Learning. *Journal of Education and Practice, 5*(2), 124-131. Retrieved from http://iiste.org/Journals/index.php/JEP/article/view/10648.

Kaila, E., Laakso, M.-J., Rajala, T., & Kurvinen, E. (2018). A model for gamifying programming education: University-level programming course quantified. In K. Skala, M. Koricic, T. G. Grbac, M. Cicin-Sain, V. Sruk, S. Ribaric…M. Janjic (Eds.), *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO),* (pp. 689-694). IEEE. https://doi.org/10.23919/MIPRO.2018.8400129.

Kennedy, E., Beaudrie, B., Ernst, D. C., & Laurent, R. S. (2015). Inverted pedagogy in second semester calculus. *Primus, 25*(9), 892-906. https://doi.org/10.1080/10511970.2015.1031301.

Kinnunen, P., & Malmi, L. (2008). CS minors in a CS1 course. In *Proceeding of the Fourth International Workshop on Computing Education Research - ICER ’08* (pp. 79-90). New York: ACM. https://doi.org/10.1145/1404520.1404529.

Korkmaz, Ö. (2016). The Effects of Scratch-Based Game Activities on Students’ Attitudes, Self-Efficacy and Academic Achievement. *International Journal of Modern Education and Computer Science, 8*(1), 16-23. https://doi.org/10.5815/ijmecs.2016.01.03.

Korkmaz, Ö., & Altun, H. (2014). Adapting Computer Programming Self-Efficacy Scale and Engineering Students’ Self-Efficacy Perceptions. *Participatory Educational Research, 1*(1), 20-31. https://doi.org/10.17275/per.14.02.1.1.

Korkmaz, Ö., & Demir, B. (2012). MEB hizmetiçi eğitimlerinin öğretmenlerin bilgi ve iletişim teknolojilerine ilişkin tutumlarına ve bilgisayar öz-yeterliklerine etkisi. *Eğitim Teknolojisi Kuram ve Uygulama, 2*(1), 1-18.

Kurt, G. (2018). International Forum of Educational Technology & Society Implementing the Flipped Classroom in Teacher Education : Evidence from Turkey. *Educational Technology & Society, 20*(1), 211-221.

Lage, M. J., Platt, G. J., & Treglia, M. (2000). Nverting the Classroom: a Gateway To Creating an Inclusive Learning Environment. *The Journal of Economic Education*, *31*(1), 30-43.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, *37*(3), 14-18. https://doi.org/10.1145/1151954.1067453.

Lee, M.-K. (2018). Flipped classroom as an alternative future class model?: implications of South Korea's social experiment. *Educational Technology Research and Development*, (20160018). https://doi.org/10.1007/s11423-018-9587-9

Lewis, C. E., Chen, D. C., & Relan, A. (2017). Implementation of a flipped classroom approach to promote active learning in the third-year surgery clerkship. *American Journal of Surgery*, *215*(2), 298-303. https://doi.org/10.1016/j.amjsurg.2017.08.050.

Liao, Y.-K. C., & Bright, G. W. (1991). Effects of Computer Programming on Cognitive Outcomes: A Meta-Analysis. *Journal of Educational Computing Research*, *7*(3), 251-268. https://doi.org/10.2190/E53G-HH8K-AJRR-K69M.

Lindberg, R. S. N., & Laine, T. H. (2018). Formative evaluation of an adaptive game for engaging learners of programming concepts in K-12. *International Journal of Serious Games, 5*(2), 3-26. https://doi.org/10.17083/ijsg.v5i2.220.

Lishinski, A., Yadav, A., Good, J., & Enbody, R. (2016). Introductory Programming : Gender Differences and Interactive Effects of Students' Motivation, Goals and Self-Efficacy on Performance. In *Proceedings of the 12th International Computing Education Research Conference* (pp. 211-220). New York: ACM. https://doi.org/10.1145/2960310.2960329.

Lopes, A. P., & Soares, F. (2018). Perception and performance in a flipped Financial Mathematics classroom. *International Journal of Management Education*, *16*(1), 105-113. https://doi.org/10.1016/j.ijme.2018.01.001.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014). Educational games for teaching computer programming. In *Research on e-Learning and ICT in Education* (pp. 87-98). Springer, New York, NY.

McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. *ACM SIGCSE Bulletin*, *34*(1), 38-42. https://doi.org/10.1145/563517.563353.

McLaughlin, J. E., Griffin, L. T. M., Esserman, D. A., Davidson, C. A., Glatt, D. M., Roth, M. T.,…Mumper, R. J. (2013). Pharmacy student engagement, performance, and perception in a flipped satellite classroom. *American Journal of Pharmaceutical Education*, *77*(9), Article 196. https://doi.org/10.5688/ajpe779196.

McLeod, D. B. (1992). Research on affect in mathematics education: A reconceptualization. In D. A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning: A project of the National Council of Teachers of Mathematics* (pp. 575-596). New York, NY: Macmillan.

Miller, B., & Ranum, D. (n.d.). What Is Programming. *Problem Solving with Algorithms and Data Structures using Python* (Section 1.4). Interactive Python. Retrieved from http://interactivepython.org/courselib/static/pythonds/Introduction/WhatIsProgramming.html.

Milli Eğitim Bakanliği. (2011). *MEB 21. Yüzyıl Öğrenci Profili*. Ankara, Milli Eğitim Basımevi.

Moore, C., & Chung, C.-J. (2015). Students' Attitudes, Perceptions, and Engagement within a Flipped classroom model as Related to Learning Mathematics. *Journal of Studies in Education*, *5*(3), 286-308. https://doi.org/10.5296/jse.v5i3.8131.

Morgan, H. (2014). Focus on Technology: Flip Your Classroom to Increase Academic Achievement. *Childhood Education*, *90*(3), 239-241. https://doi.org/10.1080/00094056.2014.912076.

Nilsen, H., & Larsen, A. (2011, November). Using the personalized system of instruction in an introductory programming course. In T. Fallmyr (Ed.), *Norsk konferanse for organisasjoners bruk av informasjonsteknologi NOKOBIT 2011 [Proceedings of the 18th NOKOBIT Conference, University of Tromsø]* (pp. 27-38). Trondheim, Norway: Tapir Akademisk Forlag. http://dro.deakin.edu.au/eserv/DU:30062167/balandin-initialexperience-evid-2011.pdf.

Nouri, J. (2016). The flipped classroom: for active, effective and increased learning – especially for low achievers. *International Journal of Educational Technology in Higher Education*, *13*(1), Article 33. https://doi.org/10.1186/s41239-016-0032-z.

O'Flaherty, J., & Phillips, C. (2015). The use of flipped classrooms in higher education: A scoping review. *Internet and Higher Education*, *25*, 85-95. https://doi.org/10.1016/j.iheduc.2015.02.002.

Özmen, B., & Altun, A. (2014). Undergraduate Students' Experiences in Programming: Difficulties and Obstacles [Üniversite Öğrencilerinin Programlama Deneyimleri: Güçlükler ve Engeller]. *Turkish Online Journal of Qualitative Inquiry*, *5*(3), 9-27. https://doi.org/10.17569/tojqi.20328.

Özyurt, H., & Özyurt, Ö. (2018). Analyzing the effects of adapted flipped classroom approach on computer programming success, attitude toward programming, and programming self-efficacy. *Computer Applications in Engineering Education* (Advanced Online Issue, May 1, 2018). https://doi.org/10.1002/cae.21973.

Pajares, F., & Miller, M. D. (1994). Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal of Educational Psychology*, *86*(2), 193-203. https://doi.org/10.1037/0022-0663.86.2.193.

Pawelczak, D. (2017). Comparison of traditional lecture and flipped classroom for teaching programming. In J. D. i Soria, M.-C. Vincent-Vela, E. de la Poza, & D. Blazquez (Eds.), *Proceedings of the 3rd International Conference on Higher Education Advances (HEAD 17)* (pp. 391-398). València, Spain: Editorial Universitat Politècnica de València. Retrieved from http://www.headconf.org/wp-content/uploads/pdfs/5226.pdf.

Peterson, D. J. (2016). The Flipped Classroom Improves Student Achievement and Course Satisfaction in a Statistics Course: A Quasi-Experimental Study. *Teaching of Psychology*, *43*(1), 10-15. https://doi.org/10.1177/0098628315620063.

Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology*, *82*(1), 33-40.

Puarungroj, W. (2015). Inverting a Computer Programming Class with the Flipped Classroom. In *Proceeding of the Twelfth International Conference on eLearning for Knowledge-Based Society* (pp. 40.1-40.7). Retrieved from http://www.elearningap.com/eLAP2015/Proceedings/02_40_Inverting.pdf.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education, 13*(2), 137-172. https://doi.org/10.1076/csed.13.2.137.14200.

Rui, Z., Lian-Rui, X., Rong-Zheng, Y., Jing, Z., Xue-Hong, W., & Chuan, Z. (2017). Friend or Foe? Flipped Classroom for Undergraduate Electrocardiogram Learning: A Randomized Controlled Study. *BMC Medical Education*, *17*(1), 1-9. https://doi.org/10.1186/s12909-017-0881-8.

Sayın, Z., & Seferoğlu, S. S. (2016, February). *Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi*. Paper presented at Akademik Bilişim 2016, Aydın, Turkey.

Scott, M. J., Counsell, S., Lauria, S., Swift, S., Tucker, A., Shepperd, M., & Ghinea, G. (2015). Enhancing Practice and Achievement in Introductory Programming with a Robot Olympics. *IEEE Transactions on Education*, *58*(4), 249-254. https://doi.org/10.1109/TE.2014.2382567.

Sener, S., & Elevli, B. (2017). E ndüstri 4.0'da Yeni İş Kolları ve Yüksek Öğrenim. *Journal Of Engineer Brains, 1*(2), 25-37.

Shiau, S., Kahn, L. G., Platt, J., Li, C., Guzman, J. T., Kornhauser, Z. G.,…Martins, S. S. (2018). Evaluation of a flipped classroom approach to learning introductory epidemiology. *BMC Medical Education*, *18*(1), 1-8. https://doi.org/10.1186/s12909-018-1150-1.

Siegle, D. (2014). Technology. *Gifted Child Today*, *37*(1), 51-55. https://doi.org/10.1177/1076217513497579.

Sirakaya, D. A. (2017). Oyunlaştırılmış Tersyüz Sınıf Modeline Yönelik Öğrenci Görüşleri. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi [Ondokuz Mayis University Journal of Faculty of Education]*, *36*(1), 114-132. https://doi.org/10.7822/omuefd.327393.

Siti Rosminah, M. D., & Ahmad Zamzuri, M. A. (2012, September). *Difficulties in learning programming: Views of students.* Paper presented at the 1st International Conference on Current Issues in Education (ICCIE 2012). https://doi.org/10.13140/2.1.1055.7441

Skinner, E. A., & Belmont, M. J. (1993). Motivation in the classroom: Reciprocal effects of teacher behavior and student engagement across the school year. *Journal of Educational Psychology*, *85*(4), 571-581.

Skinner, E. A., & Pitzer, J. R. (2012). Developmental dynamics of student engagement, coping, and everyday resilience. In S. L. Christenson, A. L. Reschly, & C. Wylie (Eds.), *Handbook of Research on Student Engagement* (pp. 21-44). Springer.

Souza, M. J. D., & Rodrigues, P. (2015). Investigating the Effectiveness of the Flipped Classroom in an Introductory Programming Course. *The New Educational Review*, *40*(2), 129-139. https://doi.org/10.15804/tner.2015.40.2.11.

Staker, H., & Horn, M. B. (2012). *Classifying K – 12 blended learning.* Innosight Institute. Retrieved from https://files.eric.ed.gov/fulltext/ED535180.pdf.

Stone, B. B. (2012, May). *Flip Your Classroom to Increase Active Learning and Student Engagement.* Paper presented at the 28th Annual Conference on Distance Teaching & Learning. Madison, Wisconsin. Retrieved from https://moodle.ruhr-uni-bochum.de/m/pluginfile.php/278895/mod_resource/content/3/ICM_Effects%20of%20Active%20learning%20strategies%20and%20student%20feedback.pdf.

Strayer, J. F. (2012). How learning in an inverted classroom influences cooperation, innovation and task orientation. *Learning Environments Research*, *15*(2), 171-193. https://doi.org/10.1007/s10984-012-9108-4.

Student-centred learning. (n.d.). In *Wikipedia*. Retrieved July 29, 2018, from https://en.wikipedia.org/wiki/Student-centred_learning.

Su, Y.-S., Ding, T.-J., & Lai, C.-F. (2017). Analysis of Students Engagement and Learning Performance in a Social Community Supported Computer Programming Course. *Eurasia Journal of Mathematics, Science and Technology Education*, *13*(9), 6189-6201. https://doi.org/10.12973/eurasia.2017.01058a.

Sundre, D., Barry, C., Gynnild, V., & Ostgard, E. T. (2012). Motivation for Achievement and Attitudes toward Mathematics Instruction in a Required Calculus Course at the Norwegian University of Science and Technology. *Numeracy, 5*(1), Article 4. https://doi.org/10.5038/1936-4660.5.1.4.

Talbot, G. L. (1997). *Helping Teachers and Students Talk about Learning Strategies for Teacher-Made Tests and Assignments.* ERIC Number: ED411203. Retrieved from https://files.eric.ed.gov/fulltext/ED411203.pdf.

Trowler, V. (2010). *Student engagement literature review*. Higher Education Academy. Retrieved from https://www.heacademy.ac.uk/system/files/StudentEngagementLiteratureReview_1.pdf.

Tu, J.-J., & Johnson, J. R. (1990). Can computer programming improve problem-solving ability? *ACM SIGCSE Bulletin, 22*(2), 30-33. https://doi.org/10.1145/126445.126451.

Tucker, B. (2012). The Flipped Classroom. *Education Next, 12*(1), 82-83. https://doi.org/10.1017/CBO9781107415324.004.

Tugun, V. (2018). Impacts and Opinions on the Technology Self- Sufficiency of the Students who are Coding Education in the Flipped Classroom Adapted to the ARCS Motivation Model. *Journal of Association for Information Communication Technology, Education and Science, 7*(2), 366-371. https://doi.org/10.18421/TEM72-18.

Tune, J. D., Sturek, M., Basile, D. P., Jensen, J. L., Kummer, T. A., & Godoy, P. D. D. M. (2015). Improvements from a Flipped Classroom May Simply Be the Fruits of Active Learning How We Teach Flipped classroom model improves graduate student performance in cardiovascular, respiratory, and renal physiology. *CBE Life Sciences Education*, *14*(Spring), 1-12.

Turan, Z. (2016, May). *Ters Yüz Sınıf Yönteminin Bilgisayar Programlama Ve Algoritma Dersinde Başarı Ve Katılıma Etkisi*. Paper presented at the 10th International Computer & Instructional Technologies Symposium, Rize, Turkey.

Ünal, C., & Bay, Ö. F. (2009). Java Programlama Dili'nin Bilgisayar Destekli Öğretimi. *Bilişim Teknolojileri Dergisi, 2*(1). Retrieved from http://dergipark.gov.tr/download/article-file/75253.

Van Vliet, E. A., Winnips, J. C., & Brouwer, N. (2015). Flipped-class pedagogy enhances student metacognition and collaborative-learning strategies in higher education but effect does not persist. *CBE Life Sciences Education, 14*(3), Article 26. https://doi.org/10.1187/cbe.14-09-0141.

Wang, X.-M., Hwang, G.-J., Liang, Z.-Y., & Wang, H.-Y. (2017). Enhancing Students' Computer Programming Performances, Critical Thinking Awareness and Attitudes towards Programming: An Online PeerAssessment Attempt. *Educational Technology & Society*, *20*(4), 58-68.

WhatsApp Türkiye. (2014, February 21). WhatsApp Türkiye'nin 20 devini geçti. *CNN TURK*. Retrieved from https://www.cnnturk.com/fotogaleri/ekonomi/sirketler/whatsapp-turkiyenin-20-devini-gecti.

Winslow, L. E. (1996). Programming pedagogy—a psychological overview. *ACM Sigcse Bulletin*, *28*(3), 17-22.

Yasad. (2009). *Yazılım: Ekonominin Yeni Kalkınma Gücü*. *Yazılım: Ekonominin Yeni Kalkınma Gücü*. Retrieved from http://www.yasad.org.tr/Content/UserFiles/yasad_rapor.pdf.

Young, A. (2015). Flipping the Calculus Classroom: A Cost-Effective Approach. *PRIMUS*, *25*(8), 713-723. https://doi.org/10.1080/10511970.2015.1031298.

Yükseltürk, E., & Altıok, S. (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, *4*(1), 50-65. https://doi.org/10.17539/AEJ.96735.

Yukselturk, E., & Altiok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, *48*(3), 789-801. https://doi.org/10.1111/bjet.12453.

Yusuf, M. (2011). The impact of self-efficacy, achievement motivation, and self-regulated learning strategies on students' academic achievement. *Procedia - Social and Behavioral Sciences*, *15*, 2623-2626. https://doi.org/10.1016/j.sbspro.2011.04.158.

Zack, L., Fuselier, J., Graham-Squire, A., Lamb, R., & O'Hara, K. (2015). Flipping freshman mathematics. *Primus*, *25*(9), 803-813. https://doi.org/10.1080/10511970.2015.1031302.

Zainuddin, Z. (2018). Students' learning performance and perceived motivation in gamified flipped-class instruction. *Computers & Education*, *126*, 75-88. https://doi.org/10.1016/j.compedu.2018.07.003.

APPENDICES

APPENDIX A

Demographic Information Form

Sevgili Öğrenciler;
Sizlere ileride daha iyi eğitim verebilmek, hem de eğitimi daha öğretici ve etkin bir hale getirebilmek için bir çalışma yapmaktayım. Bu çalışmanın başarılı olabilmesi için sizlerin yardımını ihtiyacım var. Sizlerden anket doldurmanızı rica edeceğim. Bu çalışmanın başarılı olabilmesi, aşağıdaki soruları eksiksiz ve doğru bir şekilde doldurmanıza bağlıdır. Toplanan veriler sadece araştırma amaçlı kullanılacak ve başka hiçbir kişi ya da kuruluşa gösterilmeyecektir. Lütfen her soruyu ilgili grubun başında bulunan açıklamalara uygun olarak yanıtlayınız. Sizlere şimdiden çok teşekkür ederim.

Cemil YURDAGÜL
ODTÜ - BÖTE
cemilyurdagul@gmail.com

I. BÖLÜM – GENEL BİLGİLER

Lütfen, aşağıda yer alan genel içerikli her bir soruyu ilgili boşluğu doldurarak veya size en uygun olan seçeneği "X" ile işaretleyerek cevaplayınız.

1. **Adınız Soyadınız** **:** ....
2. **Cinsiyetiniz** **:** ☐ Erkek ☐ Kadın
3. **Epostanız** **:** ....
4. **Doğum Tarihiniz** **:** ....
5. **Okul Numaranız** **:** ....
6. **Bölümünüz** **:** ....
7. **Lise Türünüz** **:** ☐ Genel Lise ☐ Meslek Lisesi
8. **Programlama ile ilgili deneyiminiz kaç yıldır:**

☐ Yok ☐ 1 yıldan az☐ 1-2 yıl ☐ 3-5 yıl ☐ 6-8 yıl ☐ 9 yıl ve üzeri

9. **Kendinize ait bir bilgisayarınız var mı?(Birden fazla işaretleyebilirsiniz)**

☐ Yok ☐ Masaüstü ☐ Dizüstü ☐ Tablet ☐ Akıllı Telefon

10. **Daha önce derslerinizde kullandığınız web araçları(Moodle, Facebook vb.):** ....

    ....

11. **Günlük kaç saatinizi internette geçiriyorsunuz?**

☐ 1 saatten az ☐ 1-2 saat ☐ 3-5 saat ☐ 6-7 saat ☐ 8 saat ve üzeri

APPENDIX B

Scales of The Study

1– Programlama Dillerine Yönelik Tutum Ölçeği

Lütfen, aşağıda yer alan her bir ifade için sizin düşüncenize karşılık gelen en uygun seçeneği "X" ile işaretleyiniz.

	Kesinlikle Katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılıyorum
1. Programlama çok yararlı ve gerekli bir konudur.					
2. Programlama becerilerimi geliştirmek isterim.					
3. Programlama zihinsel gelişime yardımcı olur ve insanın düşünmesini sağlar.					
4. Programlama günlük hayatta önemlidir.					
5. Programlama insanların çalışması için en önemli konulardan biridir.					
6. Hangi düzeyde olursa olsun programlama dersleri çok yararlıdır.					
7. Okul dışında birçok yolla programlamayı kullanabileceğimi düşünüyorum.					
8. İleri düzey programlama çalışmak yararlıdır.					
9. Programlama çalışmak diğer alanlarda problem çözmede bana yardımcı olacaktır.					
10. Programlamada sağlam bir alt yapı iş hayatımda yardımcı olabilir.					
11. Programlama problemlerini çözerken memnuniyet duyarım.					
12. Okulda programlama çalışmaktan genellikle hoşlanırım.					
13. Programlamada yeni problemleri çözmeyi severim.					
14. Program yazmayı başka bir dersin ödevini yapmaya tercih ederim.					

15. Programlamayı gerçekten severim.					
16. Diğer derslere göre programlama dilleri dersinde olmaktan daha mutluyumdur.					
17. Programlama çok ilginç bir konudur.					
18. Programlamada zor bir soruya çözüm ararken kendi fikirlerimi ifade etme konusunda rahatımdır.					
19. Programlama derslerinde sorulara rahatlıkla cevap verebilirim.					
20. Programlama sıkıcıdır.					
21. Programlama en korktuğum derslerden biridir.					
22. Ne zaman programlama kelimesini duysam kendimi kötü hissederim.					
23. Programlama çalışırken aklım dağılır ve rahatça düşünemem.					
24. Programlama çalışmak beni sinirlendirir.					
25. Programlama rahatsız hissetmemi sağlar.					
26. Programlama derslerinde kendimi sürekli gerilim altında hissederim.					
27. Bir problem hakkında program yazma düşüncesi beni sinirlendirir.					
28. Programlama derslerinde sürekli kafam karışıktır.					
29. Program yazarken kendimden emin değilimdir.					
30. Programlama beni çok da korkutmaz.					
31. Programlama konusunda kendime öz-güvenim tamdır.					
32. Çok zor olmayan programlama sorularını çözebilirim.					
33. Aldığım programlama derslerinde en iyisini yapacağımı umuyorum.					
34. Programlamayı rahatlıkla öğrenirim.					
35. Program yazmada iyi olduğuma inanıyorum.					
36. Daha ileri programlamayı da öğrenebileceğim konusunda kendime güvenirim.					
37. Eğitim süresince alabildiğim kadar programlama dersi almayı planlıyorum.					
38. Programlamanın zorluğu beni hırslandırır.					
39. Programlama derslerinde gerekli olandan daha fazla programla almayı umarım.					
40. Programlama öğretmekten uzak dururum.					

2– Programlamaya İlişkin Öz yeterlilik Ölçeği

Aşağıdaki programlamaya ilişkin verilen görevleri yaparken kendinize olan güveninizi 1 ile 7 arasında derecelendirerek belirtiniz. (“1 = Hiç Güvenmiyorum”, “2 = Genellikle Güvenmiyorum”, “3 = Biraz Güveniyorum”, “4 = %50/50” “5 = Oldukça Güveniyorum”, “6 = Genellikle Güveniyorum”, “7 = Tamamen Güveniyorum”)

	Hiç güvenmiyorum	Genellikle güvenmiyorum	Biraz güveniyorum	50/50	Oldukça güveniyorum	Genellikle güveniyorum	Tamamen güveniyorum
	1	2	3	4	5	6	7
1. "Merhaba Dünya" mesajının görüntülenebileceği bir program yazabilirim.							
2. Üç sayının ortalamasını hesaplayan bir program yazabilirim.							
3. Verilen herhangi bir sayı dizisinin ortalamasını hesaplayan bir program yazabilirim.							
4. İstenilenler açıkça tanımlandığında bir problemin çözümüne yönelik oldukça karmaşık ve uzun bir program yazabilirim.							
5. Yazacağım bir programı modüler bir biçimde organize edip tasarlayabilirim							
6. Yazdığım uzun ve karmaşık bir programdaki tüm hataları ayıklayabilir ve çalışabilir hale getirebilirim.							
7. Uzun, karmaşık ve birden fazla dosya gerektiren bir programı kavrayabilirim.							
8. Bir programın daha okunabilir ve açık olması için uzun ve karmaşık kısımları yeniden yazabilirim.							
9. Çevrede bir sürü dikkat dağıtıcı olsa bile programa odaklanma yollarını bulabilirim.							

3 – Derse Katılım Ölçeği

Aşağıda genel olarak derse katılım konusunda kendinizi değerlendirebileceğiniz ifadeler yer almaktadır. Sizden istenen, her bir ifadeyi dikkatli bir şekilde okuyup sizin için ne kadar uygun olduğunu değerlendirmenizdir.

		Hiç Uygun Değil	Uygun Değil	Biraz Uygun	Uygun	Tamamen Uygun
1	Sınıfta ders anlatılırken mutluyumdur.	1	2	3	4	5
2	Derslere ilgim yüksektir.	1	2	3	4	5
3	Kendimi derslerle bütünleşmiş görürüm.	1	2	3	4	5
4	Derslerde merak duygum yüksektir.	1	2	3	4	5
5	Dersler benim için eğlenceli geçer	1	2	3	4	5
6	Derslerde söz alırım.	1	2	3	4	5
7	Derslerde hocanın sorduğu sorulara yanıt veririm.	1	2	3	4	5
8	Derslerde gerçekleşen etkinliklere katılırım.	1	2	3	4	5
9	Derslere hazırlıklı gelirim.	1	2	3	4	5
10	Derslerde yeri geldiğinde hocaya sorular sorarım.	1	2	3	4	5
11	Derslerde, başka bilgilerle öğrendiklerimi bütünleştiririm.	1	2	3	4	5
12	Derslerde, zihnimde anlatılanları canlandırırım.	1	2	3	4	5
13	Derslerde, anlatılanlarla ilgili içimden örnekleri düşünürüm.	1	2	3	4	5
14	Derslerde, zihnimde konunun gündelik yaşamla ilişkisini kurarım.	1	2	3	4	5
15	Derslerde anlatılan bilgileri alıp beynimde işlemeye çalışırım.	1	2	3	4	5

APPENDIX C

Research Survey

5-BÖLÜM – Ters Yüz Sınıf Yaklaşımına Karşı Öğrenci Algısı Anketi

Lütfen, aşağıda yer alan her bir ifade için sizin düşüncenize karşılık gelen en uygun seçeneği "X" ile işaretleyiniz.

	Kesinlikle Katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılıyorum
1. Ders boyunca sık sık sınıftaki arkadaşlarımla iletişim halindeydim.					
2. Ders boyunca öğretmenim ile sık sık iletişim halindeydim.					
3. Ters Yüz Sınıf yaklaşımı ve öğretici videolar dersi daha iyi öğrenmem için yardımcı oldu.					
4. Ders için önceden verilen videoları izlemem programlama dersini ilerletmemde yardımcı oldu.					
5. Ders için önceden verilen videoları izlemek için birinden yardım almam gerekti.					
6. Ders için önceden verilen videoları dersi daha iyi anlamak için birden fazla kez izledim.					
7. Ders için önceden verilen videolara ders sonrasında ulaşmak beni mutlu etti.					
8. Ters Yüz Sınıf Yaklaşımı geleneksel yönteme göre derslere daha fazla katılmamı sağladı.					
9. Ters Yüz Sınıf Yaklaşımı ile olan dersleri arkadaşlarıma öneririm.					
10. Ters Yüz Sınıf Yaklaşımı arkadaşlarımla ders sırasında daha fazla iletişim içinde olmamı sağladı.					
11. Ders öncesi verilen videoları izlemekten zevk aldım.					
12. Ders öncesi verilen videoları düzenli olarak izledim.					
13. Ters Yüz Sınıf Yaklaşımı ile tasarlanmış başka derslerede katılmak isterim.					

	Kesinlikle Katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılıyorum
14. Ters Yüz Sınıf Yaklaşımı ile tasarlanmış ders içinde uygulama yapmaya daha fazla vakit ayırdık.					
15. Ters Yüz Sınıf Yaklaşımı, kendi hızıma uygun öğrenme fırsatı sağlıyor.					
16. Ters Yüz Sınıf Yaklaşımı, yaygınlaştırılması toplum için yararlıdır					
17. Ters Yüz Sınıf Yaklaşımı, sorumluluk duygusunu geliştirmektedir					
18. Ders için önceden verilen videolar, ders sonrasında da tekrar amaçlı izlerim.					
19. Ters Yüz Sınıf Yaklaşımın sıkıcı olduğunu düşünüyorum.					

APPENDIX D

Interview Questions

1. Ters Yüz Sınıf yaklaşımının size göre avantajları nelerdir?
2. Ters Yüz Sınıf yaklaşımının size göre dezavantajları nelerdir?
3. Sizce Ters Yüz Sınıf yaklaşımının hangi derslerde yararlı olabilir? Neden?
4. Ters Yüz Sınıf yaklaşımının uygulandığı dersin daha iyi olması için sizce ne gibi değişiklikler/eklemeler yapılabilir?
5. İlerde öğretmen olunca bu TYS yaklaşımını derslerinde kullanmayı düşünüyor musun? Neden?
6. Videoların yapısı/tasarımı hakkında düşünceleriniz nelerdir?
7. Videolarda çıkan sorular hakkında düşünceleriniz nelerdir?
8. Videoları ders öncesinde izlerken nasıl bir çalışma yöntemi izliyordunuz?
    a. Arkadaşlarınız ile videolardaki konular hakkında konuşup/tartışıyormuydunuz?
9. Sınav öncesi videolara ulaşabilmenin sınav hazırlığına etkisi oldu mu?
10. Ters Yüz Sınıf uygulanarak yapılan programlama dersinin, program yazabilmeye karşı kendine güven açısından etkileri nasıl oldu?
    a. TYS’ın hangi unsurları/özellikleri bu etkiyi sağladı?
11. Ters Yüz Sınıf uygulanarak yapılan programlama dersinin, derste aktifliğin/katılım açısından etkileri nasıl oldu?
    a. TYS’ın hangi unsurları/özellikleri bu etkiyi sağladı?
12. Ters Yüz Sınıf uygulanarak yapılan programlama dersinin, programlama dersine karşı düşüncelerine/bakışlarına etkileri nasıl oldu?
    a. TYS’ın hangi özellikleri bu etkiyi sağladı?
13. TYS’ın programlama dersinde, öğrenimine sağladığı başka faydalar olduysa, bu faydalar neler açıklayabilir misin?
14. Başka her hangi eklemek istediğiniz bir şey var mı?

APPENDIX E

Assumptions Check Table

Means, Standard Deviation, Skewness and Kurtosis for Self-efficacy in Programming Language Scale

IV	Condition	DV	M	SD	Skewness	Kurtosis	N
Level 1	No intervention	Self-efficacy in basic programming	4.62	1.84	-0.55	-0.76	35
		Self-efficacy in complex programming	3.30	1.56	0.03	-1.05	35
Level 2	Traditional classroom approach	Self-efficacy in basic programming	5.30	1.74	-1.01	0.11	35
		Self-efficacy in complex programming	3.85	1.54	-0.52	-0.68	35
Level 3	Flipped classroom approach	Self-efficacy in basic programming	5.82	1.20	-0.74	-0.55	35
		Self-efficacy in complex programming	4.86	1.37	-0.78	1.14	35

Table 4.5 Mauchly's W Test for Variables of Self-efficacy in Programming Language *Scale*

Measure	Mauchly's W	χ^2	*df*	*p*	Greenhouse Geisser (ε)	HuynhFeldt (ε)	Lowerbound
Self-efficacy in basic programming	0.93	2.46	2	0.29	0.93	0.98	0.5
Self-efficacy in complex programming	0.96	1.33	2	0.52	0.96	1.00	0.50

Means, Standard Deviation, Skewness and Kurtosis for Engagement in Programming Language Scale

IV	Condition	DV	M	SD	Skewness	Kurtosis	N
Level 1	Initial state	Behavioral engagement	3.62	0.77	-0.13	-0.20	35
		Emotional engagement	3.44	0.71	-0.96	3.12	35
		Cognitive engagement	3.80	0.61	-0.49	1.11	35
Level 2	Traditional classroom approach	Behavioral engagement	3.75	0.78	-0.72	0.25	35
		Emotional engagement	3.42	0.72	-0.07	-0.10	35
		Cognitive engagement	3.84	0.72	-0.82	1.71	35
Level 3	Flipped classroom approach	Behavioral engagement	4.06	0.63	-0.80	1.08	35
		Emotional engagement	3.77	0.73	-1.45	3.55	35
		Cognitive engagement	3.94	0.58	-0.15	-0.47	35

Mauchly's W Test for Variables of Engagement in Programming Language Scale

Measure	Mauchly's W	χ^2	*df*	*p*	Greenhouse Geisser (ε)	HuynhFeldt (ε)	Lowerbound
Behavioral engagement	.86	4.95	2	.08	.88	.92	.50
Emotional engagement	.98	.65	2	.72	.98	1.00	.50
Cognitive engagement	.84	5.75	2	.06	.86	.90	.50

Means, Standard Deviation, Skewness and Kurtosis for Attitude Towards Programming Languages Scale

IV	Condition	DV	M	SD	Skewness	Kurtosis	N
Level 1	Initial state	Importance	4.03	0.78	-1.35	1.45	35
		Enjoyment	3.50	0.77	-0.46	-0.26	35
		Self-confidence	3.53	0.81	-0.43	-0.41	35
		Motivation	3.57	0.76	-0.98	1.44	35
Level 2	Traditional classroom approach	Importance	4.07	0.68	-1.78	3.60	35
		Enjoyment	3.79	0.56	-0.07	0.32	35
		Self-confidence	3.74	0.75	-0.26	-0.81	35
		Motivation	3.68	0.74	-0.54	0.38	35
Level 3	Flipped classroom	Importance	3.98	0.85	-1.21	0.75	35
		Enjoyment	3.90	0.63	-0.11	-0.67	35

Means, Standard Deviation, Skewness and Kurtosis for Attitude Towards Programming Languages Scale

IV	Condition	DV	M	SD	Skewness	Kurtosis	N
	approach	Self-confidence	3.91	0.69	0.02	-1.29	35
		Motivation	3.75	0.73	-0.04	-0.31	35

Mauchly's W Test for Variables of Attitude Towards Programming Languages Scale

Measure	Mauchly's W	χ^2	*df*	*p*	Greenhouse Geisser (ε)	HuynhFeldt (ε)	Lowerbound
Importance	.80	7.534	2	.023	.83	.87	.50
Enjoyment	.83	6.305	2	.043	.85	.89	.50
Self-confidence	.97	1.021	2	.60	.97	1.00	.50
Motivation	.89	3.853	2	.146	.90	.95	.50

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name	Yurdagül, Cemil
Nationality:	Turkish (TC)
Date and Place of Birth:	November 18, 1984, Bolu
Marital Status:	Married
Phone:	+90 312 595 52 51
email:	cemilyurdagul@gmail.com

EDUCATION

Degree	**Institution**	**Year of Graduation**
BS	METU, CEIT	2009
High School	Bolu Anatolian Vocational High School(Computer Software)	2002

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2017 - present	Ankara University	Instructor
2010 - 2017	METU	Research Assistant
2009 - 2010	SEBİT	Software Developer

FOREIGN LANGUAGES

Advanced English

PUBLICATIONS AND PRESENTATIONS

Yurdagül, C. (2018, May). *Examination of Digital Data Security Awareness Levels: Vocational School of Justice Students*. Paper presented at the International Computer & Instructional Technologies Symposium.

Yurdagül, C., Arslantas, T., Ak, N., Gül, A., & Yıldırım, Z. (2017, October). *Improving English Vocabulary Knowledge of Visually Impaired People: A Design-Based Research Approach*. Paper presented at the International Instructional Technologies and Teacher Training Symposium.

Yurdagül, C., & Alper, A. (2016, May). *Opinions of Teachers of Information Technologies on Coding Course.* Paper presented at the Uluslar Arası Bilgisayar ve Öğretim Teknolojileri Eğitimi Sempozyumu.

Yurdagül, C., & Öz, S. (2016, May). *Attitudes towards Mobile Learning in English Teaching*. Paper presented at the Uluslar Arası Bilgisayar ve Öğretim Teknolojileri Sempozyumu.

Alper, A., & Yurdagül, C. (2015). Trends in Articles Related to ICT Supported Language Learning in BJET Content Analysis. *Journal of Instructional Technologies & Teacher Education*, *4*(2), 1-15.