



T.C.
YEDİTEPE UNIVERSITY
MASTER'S PROGRAM IN CURRICULUM AND INSTRUCTION

**THE EFFECTS OF FACE-TO-FACE AND BLENDED LEARNING
ENVIRONMENTS IN CODING TRAINING ON STUDENTS'
COMPUTATIONAL THINKING SKILLS**

MUSTAFA UĞURCAN ŞENGİZER

İSTANBUL, 2021

THE EFFECTS OF FACE-TO-FACE AND BLENDED LEARNING
ENVIRONMENTS IN CODING TRAINING ON STUDENTS' COMPUTATIONAL
THINKING SKILLS

MASTER'S THESIS

PROGRAM IN CURRICULUM AND INSTRUCTION

MUSTAFA UĞURCAN ŞENGEZER

YEDİTEPE UNIVERSITY

FEBRUARY, 2021



YEDİTEPE UNIVERSITY

GRADUATE SCHOOL OF EDUCATIONAL SCIENCES DIRECTORATE

THESIS SUBMISSION and APPROVAL FORM

TITLE: The Effects of Face-To-Face and Blended Learning Environments in Coding Training on Students' Computational Thinking Skills

APPROVAL:

Doç. Dr. Gonca Kızılkaya Cumaoglu

(Advisor)

(Signature)

Doç. Dr. Yelkin Diker Coşkun

(Member)

(Signature)

Dr. Öğr. Üye. Ferhat Kadir Pala

(Member)

(Signature)

SUBMITTED BY : Mustafa Uğurcan Şengezer

DATE OF THESIS DEFENSE :

DATE OF THESIS APPROVAL :

ACKNOWLEDGEMENT

Bu tezin ortaya ıkmasında teekkr etmek istediėim kiřiler var.

ncelikle lisans hayatıma bařladıėımdan beri ėrettikleriyle, bakıř aısıyla, yardımlarıyla bana yardım eden ve yanımda olan tez danıřmanım Do. Dr. Gonca Kızılkaya Cumaoglu'na teekkr ederim.

Yksek lisans derslerinden biri olan istatistik dersiyle iyi ki tanı mıřım, iyi ki bu dersi o hocadan almıřım dediėim ders hocam olan Oėuzhan Doėan'a teekkr ederim.

Tez uygulamamda yer alan enerjisi hi bitmeyen ėrencilerime teekkr ederim.

Ailem. Her daim yanımda olan, motive eden, beni destekleyen ve desteklerini hi esirgemeyen ablalarıma, anneme, babama teekkr ederim.

Lisans hayatımdan itibaren en byk destekcim, her tkezlediėimde kalkmamı saėlayan, kořulsuz en byk kolaylařtırıcım eřim Duygu'ya ve bu srete bizimle olan oėlumuza, Can Demir'e sonsuz teekkrler.

TABLE OF CONTENTS

THESIS SUBMISSION AND APPROVAL FORM.....	1
ACKNOWLEDGEMENT	2
TABLE OF CONTENTS.....	3
LIST OF TABLES.....	6
LIST OF FIGURES.....	7
LIST OF ABBREVIATIONS	8
ABSTRACT	9
ÖZ	11
CHAPTER I	13
INTRODUCTION.....	13
1.1 PROBLEM STATEMENT	13
1.2 RESEARCH AIM	18
1.3 IMPORTANCE OF THE RESEARCH	18
1.4 THE MAIN RESEARCH QUESTION.....	20
1.5 SUBPROBLEMS OF THE RESEARCH.....	20
1.6 RESEARCH LIMITATIONS	21
1.7 ASSUMPTIONS.....	23
1.8 DEFINITIONS	23
CHAPTER II LITERATURE REVIEW.....	25
2.1 EDUCATION SYSTEM IN TURKEY	25
2.1.1 History, Development and Structures of ITS Course.....	26
2.1.1.1 ITS in 1997	27
2.1.1.2 ITS in 2005	27
2.1.1.3 ITS in 2007	27
2.1.1.5 ITS in 2012 and 2013.....	29
2.1.1.6 ITS in 2017	29

2.1.1.7 ITS in 2019	30
2.1.2 Main Goals of ITS Course Curriculum	30
2.1.3 Learning Objectives of ITS Course by Grade Level	32
2.1.4 ITS Course Curriculum	33
2.1.4.1 The Goals of Curriculum for the ITS Course in Recent Years	35
2.2 CODING & PROGRAMMING ON COMPUTER	36
2.2.1 Programming Training	38
2.2.2 Programming Training in the World	41
2.3. PROGRAMMING LANGUAGES	43
2.3.1 Arduino Programming.....	43
2.3.2 Drone Programming.....	44
2.4 EDUCATION	45
2.4.1. FTFL Education.....	46
2.4.2. BL Education	47
2.5 COMPUTATIONAL THINKING (CT) SKILLS	50
2.5.1 Definition and Theory of CT	50
2.5.2 Importance of CT.....	53
2.5.3 Features of CT.....	54
2.6 PROGRAMMING TRAINING AND COMPUTATIONAL THINKING.....	55
2.6.1 Place in Curriculum of CT.....	59
2.6.1.1 CT in Preschool.....	61
2.6.1.2 CT in Primary School.....	63
2.6.1.3 CT in Secondary School.....	64
2.6.1.4 CT in High School.....	65
2.6.1.5 CT in University	66
CHAPTER III	67
METHOD	67
3.1 THE ROLE OF THE RESEARCHER AND IMPLEMENTATION PROCESS.....	67

3.2 RESEARCH MODEL.....	68
3.3 DEMOGRAPHIC AND SAMPLING	69
3.4 THE DESIGN OF THE PROCESS FOR CODING TRAINING.....	75
3.5 RESEARCH VARIABLES.....	76
3.6 DATA COLLECTION TOOLS	77
3.6.1 CT Skills Scale.....	78
3.6.2 Personal Information Form.....	80
3.7 RESEARCH APPLICATION STEPS	80
3.8 DATA COLLECTION AND ANALYSIS	88
3.8.1 CT Skills Scale.....	90
CHAPTER IV.....	93
FINDINGS	93
CHAPTER V.....	109
CONCLUSIONS, DISCUSSION, AND SUGGESTIONS	109
5.1 SUGGESTIONS FOR THE FURTHER RESEARCH.....	121
CHAPTER VI.....	124
REFERENCES.....	124
CURRICULUM VITAE (CV)	147

LIST OF TABLES

<i>Table 1: Education System in Turkey by Grade and Age</i>	<i>26</i>
<i>Table 2: History of ITS Teaching in Turkey.....</i>	<i>26</i>
<i>Table 3: Programming Languages Used by Countries</i>	<i>42</i>
<i>Table 4: Changes in The Education System From The Industrial Age To The Information Age.....</i>	<i>45</i>
<i>Table 5: Students' Class, Gender and Frequency Data</i>	<i>71</i>
<i>Table 6: Students' Groups, Gender, Variables and Frequency Data.....</i>	<i>73</i>
<i>Table 7. Used Data Collection Tools</i>	<i>78</i>
<i>Table 8. Internal consistency coefficients of CT Skills Scale.....</i>	<i>79</i>
<i>Table 9: Arduino Applications Process in FTFL Environment</i>	<i>82</i>
<i>Table 10: Research Applications Parts.....</i>	<i>83</i>
<i>Table 11. Data Collection Tools and Implementations.....</i>	<i>91</i>
<i>Table 12: General CT Descriptive Statistics of the Students</i>	<i>94</i>
<i>Table 13: General Sub Factors of CT Descriptive Statistics.....</i>	<i>95</i>
<i>Table 14: General CT Descriptive Statistics of the Students' Gender</i>	<i>96</i>
<i>Table 15: T-Test and Total Results of CT Scores of Coding Training in FTFL Environment on Sub Factors</i>	<i>97</i>
<i>Table 16: T-Test Results of CT Scores of Coding Training Given in FTFL Environment by Gender</i>	<i>100</i>
<i>Table 17: T-Test Results of CT Scores of Coding Training in FTFL Environment by Gender on Sub Factors</i>	<i>101</i>
<i>Table 18: T-Test and Total Results of CT Scores of Coding Training in BL Environment on Sub- Factors</i>	<i>102</i>
<i>Table 19: T-Test Results of CT Scores of Coding Training in BL Environment by Gender</i>	<i>105</i>
<i>Table 20: T-Test Results of CT Scores of Coding Training Given in BL Environment By Gender on Sub-Factors</i>	<i>106</i>
<i>Table 21: T-Test Results of the Mean Differences of Coding Training in FTFL Environment and BL Environment.....</i>	<i>107</i>

LIST OF FIGURES

<i>Figure 1. Research Design</i>	<i>68</i>
<i>Figure 2. Thesis of Research Design.....</i>	<i>69</i>
<i>Figure 3: The Sample of the Research and Steps.....</i>	<i>74</i>
<i>Figure 4: Application Process and Stages of the Research</i>	<i>76</i>



LIST OF ABBREVIATIONS

- BID: Bilgi İşlemsel Düşünme
- BİT: Bilgi ve İletişim Teknolojileri
- BL: Blended Learning
- CoHE: The Council of Higher Education
- CSTA: Computer Science Teachers Association
- CT: Computational Thinking
- CWI: Centrum Wiskunde & Informatica
- FTFL: Face-To-Face-Learning
- ICT: Information and Communication Technologies
- IDE: Integrated Development Environment
- ISTE: International Society for Technology in Education
- IT: Information Technology
- ITS: Information Technologies and Software
- LED: Light Emitting Diode
- RET: Research Experience for Teachers
- TDK: The Turkish Language Association

ABSTRACT

THE EFFECTS OF FACE-TO-FACE AND BLENDED LEARNING ENVIRONMENTS IN CODING TRAINING ON STUDENTS' COMPUTATIONAL THINKING SKILLS

Mustafa Uğurcan Şengezer

Supervisor: Assoc. Prof. Gonca Kızılkaya Cumaoglu

This study explores the effects of coding training on computational thinking (CT) skills levels in face-to-face learning (FTFL) and blended learning (BL) environments. The research sample consists of 60 seventh grade students (30 girls, 30 boys) observed over the ITS course of the 2019-2020 academic year. These students attend a private secondary school in Istanbul and have prior programming experience, having taken Information Technologies and Software (ITS) courses over the preceding two years.

This quantitative research was undertaken following a weak experimental research design with one group pre-test, post-test experimental design, using a personal information form and secondary school CT Skills Levels Scale. These were analyzed using the SPSS 25 program. The students undertook two coding training projects, the first lasting sixteen weeks and using Arduino educational robotics coding applications in a FTFL environment, and the second (also lasting sixteen weeks) a drone coding application in a BL environment. Three tests were undertaken across the course: one

before the FTFL training; one between the FTFL and BL training; and one at the conclusion of the coding training.

The results of this study for the FTFL coding training reveal insignificant development in CT skills; no significant difference among the girls and boys sample groups; and no significant variation in CT skill subfactors either. Whilst there was again little change in the overall CT skills after training in the BL environment, the results do suggest some development in collaboration (a CT subfactors) amongst girls students; and in the BL environment, both girls and boys students also attested to improvements in collaboration and creativity skills.

Keywords; computational thinking, face-to-face learning environment, blended learning environment, drone, Arduino, coding training

ÖZ

KODLAMA ÖĞRETİMİNDE YÜZ YÜZE VE KARMA ÖĞRENME ORTAMLARININ ÖĞRENCİLERİN BİLGİ İŞLEMSEL DÜŞÜNME BECERİLERİ ÜZERİNE ETKİSİ

Mustafa Uğurcan Şengezer

Danışman: Doç. Dr. Gonca Kızılkaya Cumaoglu

Bu çalışma, kodlama eğitiminde yüz yüze ve karma öğrenme ortamlarının öğrencilerin Bilgi İşlemsel Düşünme (BİD) becerileri üzerindeki etkilerini incelemek amacıyla yapılmıştır. Araştırmanın örneklemi, 2019-2020 eğitim öğretim yılı boyunca gözlemlenen 60 yedinci sınıf öğrencisinden (30 kız, 30 erkek) oluşmaktadır. Bu öğrenciler, İstanbul'da özel bir ortaokula eğitim gören ve önceki iki yıl boyunca Bilgi Teknolojileri ve Yazılım dersleri almış, önceden programlama deneyimine sahip öğrencilerden oluşmaktadır.

Bu nicel araştırma, Kişisel Bilgi Formu ve ortaokul seviyesi için BİD Beceri Düzeyleri Ölçeği kullanılarak zayıf deneysel desenlerden tek gruplu ön test - son test deneysel deseninde yapılmıştır. Toplanan verilerin analiz edilebilmesi için SPSS 25 programı kullanılmıştır. Öğrencilerin kodlama öğretiminde, ilk olarak on altı hafta süren ve yüz yüze öğrenme ortamında gerçekleşen Arduino eğitsel robotik uygulamaları yer almaktadır. Ardından sonraki on altı hafta boyunca karma öğrenme ortamında drone kodlama uygulamaları olmak üzere iki kodlama öğretimi uygulaması

yer almaktadır. Kodlama öğretimi boyunca üç test ile verilerin toplanması gerçekleştirilmiştir. İlki, yüz yüze eğitimden önce; ikincisi, yüz yüze ve karma eğitimin arasında ve son olan eğitimin sonunda testler yapılmıştır.

Elde edilen bulgulara göre; BİD beceri düzeyleri testi puanlarının analizi, yüz yüze öğrenme ortamında gerçekleştirilen kodlama öğretiminin BİD becerileri üzerinde anlamlı bir farklılık bulunmadığı; kız ve erkek örneklem grupları arasında anlamlı bir fark olmadığı görülmüştür. Bununla birlikte yüz yüze öğrenme ortamında kodlama öğretiminin BİD becerileri alt faktörlerinde açısından farklılaşmadığı bulgusu elde edilmiştir. Ancak, öğrencilerin cinsiyetlerine göre bakıldığında BİD becerileri alt faktörlerinden biri olan işbirliklilik üzerine anlamlı bir farklılık olduğu belirlenmiştir. Ayrıca, karma öğrenme ortamında kodlama öğretiminin BİD becerileri alt faktörlerinden olan Yaratıcılık ve İşbirliklilik üzerlerinde anlamlı bir farklılık bulunmuştur.

Anahtar Kelimeler; bilgi işlemsel düşünme becerisi, yüz yüze öğrenme ortamı, karma öğrenme ortamı, drone, Arduino, kodlama öğretimi

CHAPTER I

INTRODUCTION

In this chapter, background, significance and limitations of the study, main and sub problems questions, assumptions and definitions of the terms presents.

1.1 Problem Statement

People of each generation have their own dominant values, cultural perspectives, and behaviors. These groups define their time as much as their time defines them. The generations today are named Generation X, Y, and Z based on their involvement in technology (Eğitimpedia, 2018). Generation X, born between 1965 and 1980, has difficulties in using technology and is resistant to change. On the other hand, Generation Y, who was born between 1980 and 1999, builds a connection to Generation X and Generation Z. Characteristics of this generation depend on geographic region and economic conditions. For these people, technology plays an important role in their lives (Altuntuğ, 2012). Born in 2000 and afterward, the generation named Z is distinguished from others in the sense that they are always in a rapid and continuous change. People of this generation who are addicted to technology, in a constant hurry, demanding everything gets done quickly and immediately form a consumer society (Yüzbaşıoğlu, 2012). By this situation, he explained as always will continue to be one step ahead of the previous generation themselves (Bekmezci, 2017).

The current curricula can be insufficient in meeting the needs of the youth named generation Z (Karabak & Güneş, 2013). To attract the attention of generation Z in the

classroom, to support traditional teaching methods and enhance the learning environment, these environments and curricula need to be rearranged after technological advancements (Somyürek, 2014). Thus, many countries need to aim to produce skilled and creative students who have with broad visions and entrepreneurial spirits as well as complication determine skills and technological literacy as they plan their training strategies (Yenilmez & Balbağ, 2016). It is concluded that within this framework technological advancements are widely used in every aspect of daily life. Furthermore, it is possible to say it is crucial that the digital natives born into this digital society are making use of these technological advancements, however this can solely be possible through the integration of technology into education.

With the idea that computers will support educational processes, computers have been used in schools and faculties of many countries since the 1950s (California State University, 2021). Countries that are aware of the importance of technology have included courses on the using computers in their curricula, with the idea that it will contribute to students' learning processes. However, countries that are in competition and lagging behind in terms of technology have realized that it is not enough to raise individuals who can simply use technology, in order to produce new technologies. With this awareness, departments on computer and communication technologies have been established in universities and advanced design and programming training has begun to be given in this field. In order to raise individuals who can produce science and technology over time, it was thought that these trainings should be given from undergraduate level to K-12 and even preschool. Thus, many countries that want to be the pioneer of scientific and technological developments have prepared and

implemented information technologies education programs in accordance with their education processes (Mercan, Filiz, Göçer & Özsoy, 2009).

In order to raise the generations that produce technology, information technology education should be carried out effectively (Yılmaz, 2007). In cases where information technology education cannot be provided effectively enough, it is estimated that this education has no positive effect on the technological progress of the country. In this case, information Technologies is thought to become a problem to the curriculum in Turkey. In 2010, the name of the course was started to be applied as Information Technologies and Software course in curriculums (Barut & Kuzu, 2017). Finally, a new instruction schedule for 4th, 5th and 7th grade levels was prepared in 2019.

The workshop “21st Century Student Profile Workshop in Turkey”, coordinated by the Ministry of National Education in 2010, focused on educating individuals students who can adapt to the developments in the world, reach and use the information and body of knowledge, and acquire skills such as researching and questioning, critical thinking, dealing with problem, and decision making on Ministry of National Education Education Research and Development Project in Turkey (EARGED, 2011). These skills of “critical thinking”, "problem solving" and “decision making” were also included in the National Educational Technology Standards for Students published by ISTE in 2007 (Orhan, Kurt, Ozan, Vural & Türkan, 2014). As the skill of problem-solving gains such importance, it becomes a necessity to focus on programming training to create individuals who have a good command of technological tools and use software for dealing with their daily problems (Akpınar & Altun, 2014). The Ministry of National Education has now included programming

education in its 5th grade curriculum and made it a requirement for the “ITS” training for the 2017-2018 school year; in the curriculum for a long period but left to the teacher’s preference whether to apply or not. Since between 2018 and 2019 educational school annum, it was continuously incorporate into the sixth grade educational program as well (MEB, 2018).

According to Tüzün (2007), programming is a processing which embodies various skills. Analytical thinking skills, logical thinking, understanding of algorithms, and even problem-solving can be acquired on students in this process (Ersoy, Madran, & Gülbahar, 2011). A problem that is determined by a programmer, develops an algorithm for its solution, converts the algorithm into code blocks, and debugs possible errors in this sequence. Thus, programming can be perceived as a complicated process that is challenging for beginners, requiring them to have a certain skill set (Gomes & Mendes, 2007). Nedzad & Yasmeen (2001) stated even though Computer Science is relatively new to the curriculum, it is just as changing and developing a field as Science, Humanities and Mathematics, leading students to perceive it as complicated when the students first encounter it. Students particularly struggle with the training when they encounter programming’s abstract concepts for the first time (Ersoy, Madran, & Gülbahar, 2011). Embodying these concepts in concrete practice and examples increases the motivation of students in this subject.

Since coding training can be difficult and boring if taught without due care, individuals at an early age can develop prejudices against coding as a result of poor training (Saygıner, 2017). Today, the best coding training is given in entertaining learning environments with the thank of more basic coding tools which have been developed with advancements in computer design like block-based (Genç & Karakuş,

2011). Although the majority of operations and concepts in coding training can remain abstract for students (Ersoy et al., 2011), making coding entertaining and easier within the supporting of block and text – based programming tools can get overcome the difficulties of programming training. With the development of technology, robotics sets also began to be used in coding training, which provide an ideal tool for giving students concrete applications for their learning, by programming educational robotics kits (Ersoy et al., 2011). In this context, coding training with different educational methods in different learning environments (such as with Arduino educational robotics coding applications in a FTFL environment, and drone coding applications in a BL environment), has a positive impact on the student's point of view in the direction of coding training as well as enhancing their computational thinking (CT) skills including creativeness, problem solving, association, critical and algorithmic thinking.

It has been due to advances in computer science that the economic and social life have been deeply impacted. Nowadays, almost all people regardless of their age or gender are possible to have certain fundamental CT skills parallel to innovations in technology (Wing, 2014). Unlike traditional information technology (IT) or information and communication technologies (ICT) skills, usually understood as practical or vocational, and responsive to technological demands, CT means having the awareness, skill, and attitudes that are required to use computers for finding the solutions to daily problems, and are proactive skills with far wider applications than just programming (Özden, 2015). ISTE (2015) stated that the aim of using CT in education is not only to enable students to be more successful in computer science but also to equip them with the skills they can automatically use in other courses. Besides,

the CT skills are listed where problem solving, algorithmic thinking, collaboration, creativity and and also critical thinking skills.

In the literature, many studies that argue coding training promotes to the advancement of students' various cognitive functions (Kaucic & Asic, 2011; Erol, 2015; Demirkol, 2016; Kert & Uğraş, 2009; Saygıner, 2017; Yiğit, 2016). This current study adds to this field of knowledge by analyzing the impacts of coding training in FLFL and BL environments on CT skill levels. Consequently, this study can help to the literature and be a guide for a range of professionals in education, such as science and technology teachers, teachers from other branches working on Science, Technology, Engineering and Math (STEM) course, education directors, teachers from other branches working in the field of school level drone training, educators designing curricula, and for those managing programming training in different learning environments and applications.

1.2 Research Aim

This research goals to view the varying impacts of coding training in FTFL and BL environments on the sub-factors of CT skills including creativity, algorithmic thinking, collaboration, critical thinking, and problem-solving as well as on gender.

1.3 Importance of The Research

Coding training has become crucial in today's world as the importance of computer coding increases. Training only started to be given in primary and secondary schools after universities and high schools and, now, is even included in preschool curricula. Saritepeci & Durak (2017) lay emphasize on students should be educated to be

individuals who have the competencies to apply the unexplored technologies to answer the problems of the unknown future.

This approach can be applied to the young age groups at the moment; however, it demands more studies being done for the most effective coding training to be possible. Many studies done on the young age groups and they continue being done. They seek answers to various questions related to coding training for young age groups, such as how coding training should be implemented, which methods should be used, which tools should be preferred, which skills students develop, and how effectively these skills enhance their wider learning and employment potential.

As the literature reveals, there has been a particularly large amount of research recently done on the impact of the use new technologies and robotics within learning environments (Chaudhary, Agrawal, Sureka & Sureka, 2016; Chen, Shen, Bart Cohen, Jiang, Huang & Eltoukhy, 2017; Ching, Yang, Wang, Baek, Swanson & Chittoori, 2019; Leonard, Buss, Gamboa, Mitchell, Fashola, Hubert & Almughyirah, 2016; Noble, 2013; Şimşek, 2018; Yolcu, 2018; Yuen, Boecking, Stone, Tiger, Gomez, Guillen & Arreguin, 2014). However, this important field has generally been developed entirely separately from the conceptualization of CT, which is the other key referent of this study. The idea of CT has emerged from the work of Pfeiffer (1962), who expressed that because of the prevalence of computers in our lives, their existence around us on its own can easily shift our way of thinking. After decades, Papert (1996) developed this thesis by establishing the concept of CT skills. Wing (2006) highlighted the importance of CT and popularized this concept. Even, CT was later defined by Cuny, Snyder & Wing (2010) as a thinking process where problems are formulated by a data processor (computer, machine, human) effectively. In her

article “Computational thinking is a fundamental skill for everyone, not just for computer scientists”.

This research draws from and merges these two fields by addressing the question of how effective Arduino educational robotics coding applications in a FTFL environment and drone coding applications in a BL environment are on CT skill levels. By looking at the results of this research, the study presents research-based evidence on which ways to follow for coding training.

Mannila et al. (2014) state that teachers have no consensus over the teaching of CT and there is a necessity for studies examining how CT can be included in training programs. Based on this present study, it will be possible to focus programming training on CT skills and make it more efficient and more effective. The research demonstrates which qualities and methods in training might be most useful to the teachers of information technologies in coding training as well as to new studies.

1.4 The Main Research Question

What are the impacts of FTFL and BL environments and gender in coding training on students’ CT skill levels?

1.5 Subproblems of The Research

- What are the secondary school students’ CT skill levels?
- Do the secondary school students’ CT skill levels differ when the coding training takes place in a FTFL environment?
- Do the secondary school students’ CT skill levels differ based on gender when the coding training takes place in a FTFL environment?

- Do the secondary school students' CT skill levels differ when the coding training takes place in a BL environment?
- Do the secondary school students' CT skill levels differ based on gender when the coding training takes place in a BL environment?
- Are there secondary school students's CT skill any significant differences between the mean values of CT skill levels in a FTFL environment and in a BL environment?

1.6 Research Limitations

This study, conducted in the elective course of ITS during the two semesters (Fall and Spring) of the 2019 – 2020 academic year, was undertaken within the frame of activities related to the coding training applications. The research was undertaken within the following parameters, which in several instances were governed by variables beyond the researcher's control. These criteria and restriction should be born in mind when applying the findings of the research more widely.

- The scope of this study is restricted to 60 7th-grade students from a private secondary school that is affiliated with the National Education Directorate of Istanbul.
- The application process of this study is chronologically framed by 2 school semesters, 36 weeks, and 72 course hours that were dedicated to Arduino educational robotics coding applications in a FTFL environment within fall semester and drone coding applications in a BL environment within spring semester, developed by the instructor of the elective course on ITS.
- The study is limited to “Arduino Educational Robotics Coding Applications” and “Drone Coding Applications”, which are 7th-grade elective courses of ITS.
- The application interfaces where the students can code are restricted to the

applications of Arduino Integrated Development Environment (IDE) and PyCharm.

- The training within the research was offered to 5 different classes undertaking the ITS course.
- The study is limited to Arduino educational robotics coding applications that can be programmed by Arduino IDE, and drone Coding Applications that can be programmed by PyCharm in Python language.
- The Arduino IDE platform has the C++ programming language. In addition, the PyCharm platform has Python programming language. In order to run the codes, some library codes need to be added to the Arduino IDE platform. However, with PyCharm, it is not necessary to add library codes for running code.
- During the research, individual studies were done on the impacts of coding training in FTFL and BL environments on CT skill levels. The students offered their solutions related to coding and application activities and developed the code algorithms individually.
- The individuals of the research were prevented from interacting with each other as much as possible to make sure that their solutions and outputs were uninfluenced by other participants. To explain, group working was not allowed and the computer lab has separate sitting and study areas for each student.
- The number of participants from each class was limited to 12 students during the application of the coding training for an easier way of information gathering and evaluation.
- “Arduino Educational Robotics Coding Applications” and “Drone Coding Applications” in the elective course of ITS, on the students’ CT skills subjects are limited by the research.
- All of the participants’ demographic information is self-declared.

- At the end of the CT skill levels of coding training in the BL environment that emerged as a result of the research include Arduino educational robotic and drone coding training.

1.7 Assumptions

- In the research, the students are assumed to answer the evaluation tests and demographic information form sincerely and honestly.
- The CT skills scale that was used in the study is presumed to be responded to sincerely and honestly by students.
- The researcher is presumed to be unbiased during the research process.
- The difficulties of activities that are designed to evaluate the students who are given the coding training in FTFL and BL environments are assumed to be equivalent.
- The students who cannot participate in the above-mentioned scale and be interviewed are considered as neutral in the learning environment.

1.8 Definitions

- **Algorithm;** is the sequential formulation of rules to determine a complication or complete a task (Tynker, 2019).
- **Arduino;** is designed to help students who are inexperienced in electronic or microcontroller programs to create learning prototypes that connect the physical and digital worlds.
- **Computational Thinking (CT);** is a state of mentality which covers several sub-skills like including collaboration, creativity, problem-solving, algorithmic thinking (ISTE, 2015).

- **Drone;** is a remote-controlled, pilotless aircraft. It is also named as Unmanned Aerial Vehicle (UAV)
- **Coding;** is the computer processing of an algorithm written in any programming language for a certain purpose.
- **PyCharm;** is an the integrated development environment. It is used specially for Python language that is coding and programming.
- **Python;** is an advanced-level programming language that is object-oriented, interpretive, modular, and interactive.
- **Robotics;** is an interdisciplinary area in science that concentrate the design, production, and use of mechanical robots.
- **STEM;** is the abbreviation for S means Science, T infers Technology, E is called to Engineering and last part of M is come from Mathematics.

CHAPTER II

LITERATURE REVIEW

This part involves the thesis articles most topical to the purpose of present study has five topics were comprehended in this chapter. Firstly, coding and programming on computer are explained. Secondly, the programming language is mentioned. Thirdly, the education that face to face and BL environment is described. The fourth part is concerned about computational thinking skills. Finally, the last part is argued that coding training and computational thinking that are place in curriculum in PreSchool, Primary School, Secondary School, High School and University is explained in the chapter.

2.1 Education System in Turkey

The framework of the Turkish Education System consists of the general objectives, the system steps to achieve these goals and the basic principles regarding the presentation of primary education in terms of quantity and quality. The education level in primary school is four years. Students who finish primary school move on to secondary school and the duration of education is four years. In addition, students who move on to high school receive four more years of education. The education levels here are compulsory. All schools in the country have to apply the subjects and contents in the national curriculum, although the curriculum can be arranged and implemented in accordance with the region where the school is located (MEB, 2014; MEB, 2015a; MEB, 2015b). The training program in Turkey includes: Turkish, Mathematics, Life Sciences, Science, Social Studies, History of Revolution and Kemalism, Foreign Language, Religious Culture and Moral Knowledge, Visual Arts,

Music, Gym, Physical Education and Sports, Technology and Design, Traffic Safety, Information Technologies and Software, Guidance and Career Planning, Human Rights, and Citizenship and Democracy courses. Turkey's educational structure, with each stage lasting four years, is shown in Table 1 below (MEB, 2014; MEB 2015a; MEB, 2015b).

Table 1: Education System in Turkey by Grade and Age

Grade	Age	Year
Primary School	6-10	4
Secondary School	10-14	4
High School	14-18	4

2.1.1 History, Development and Structures of ITS Course

The history of ITS teaching in Turkey, as recorded by the Information Technology Educators Association, appears in Table 2.

Table 2: History of ITS Teaching in Turkey

Year	Full Name of Course	Status of Being Chosen	Method of Selection	Evaluation	Lesson Hours per Year of Study							
					1	2	3	4	5	6	7	8
1997	Computer	Elective	School Management	Grade	-	-	-	1	1	1	1	1
2005		Elective	Teacher	No Grade	1	1	1	1	1	1	1	1
2007	Information Technology	Elective	Teacher	No Grade	1	1	1	2	2	1	1	1
2010		Elective	Teacher	No Grade	-	-	-	-	-	1	1	1
2012	ITS	Elective	Parent	No Grade	-	-	-	-	2	2	2	2

2013	ITS	Elective(7,8) Compulsory (5,6)	Parent & E-School	Grade	-	-	-	-	2	2	2	2
2017	ITS	Elective(1,2,3,4,7,8) Compulsory (5,6)	Parent & E-School	Grade	2	2	2	2	2	2	2	2
2019	ITS	Compulsory	Pilot Application	Grade	-	-	-	2	3	2	2	-

2.1.1.1 ITS in 1997

In 1997, for the first time in the weekly course schedule, the "Computer" course was included in the elective course list of the primary school. From 4th to 8th grades, 1 hour was chosen for teaching the course. In 1998, although the course itself did not change, the total number of elective courses offered to students decreased from 19 to 7, as "Computer, Drama, Speaking and Writing, Second Foreign Language, Tourism, Agriculture, Local Crafts". The curriculum of the ITS course was first published in this year. The method of selection was determined by the school administration and the evaluation of the course was made with grades.

2.1.1.2 ITS in 2005

Since 2002, Computer Education and Instructional Technology departments started to give graduates, but with this decision, the hour of the courses was reduced to 1, and grading was abolished in elective courses. The selection method was determined by the teachers' council and there was no evaluation of the course.

2.1.1.3 ITS in 2007

In this year the number of elective courses in total decreased significantly. Students were not given the chance to choose, and a schedule was published where only 2 hours of elective courses were preferred. It was stated by the Information

Technologies Association that the Foreign Language (English) course, which has questions in central exams, should be preferred over the ITS course, which is costly and requires a special class. The "Internet" unit was included for the first time in this year's program.

The cost of IT classes and the inconvenience of maintenance and repair led some IT teachers, especially those who did not have an IT class in their schools, to attend one-week courses to obtain assistant chess coaching certificates and to carry out elective Chess Lessons instead of IT. Some schools took matters into their own hands and taught the 2-hour elective Foreign Language Course for 1 hour and the IT course in the other 1 hour.

It was welcomed by parents to increase the class hours of the 4th and 5th grades to 2, and this ensured that the requirements of the course could be fulfilled in many schools. The course gained a new curriculum and workbook. Finally, the selection method was determined by the teachers' council and there was no evaluation of the course.

2.1.1.4 ITS in 2010

In 2010 IT education in our country entered its darkest period. Lessons could only be selected for one hour per week in grades 6, 7 and 8, these lessons was never published in curriculum. For the Free Activity Lesson of 1st, 2nd, 3rd, 4th and 5th grades, Classroom Teachers were expected to teach this lesson depending on their wishes; this lesson was, supposedly, no longer needed for ITS because the students come knowing the computer. Schools rapidly began to redistribute the technology in IT suites and turn them into other classrooms. IT teachers have also had to undertake

jobs in schools with unclear job descriptions under the name of trainers. Finally, the selection method was determined by the teachers' council and there was no evaluation of the course.

2.1.1.5 ITS in 2012 and 2013

In these years, IT courses were included in the weekly schedule as from elective to compulsory for the first time, and the course provision was increased to two hours per week, to ensure the integrity of education and training. Nevertheless, with changes in 2013, up to 18 hours of foreign language education were provided in schools with appropriate means and conditions in 5th grade, alongside Turkish, Mathematics, Science, Social Studies and Religious Culture and Moral Knowledge. With only 4 hours of elective courses in total being available in the 5th grade, if a foreign language was taught. As a result, IT would not be taught in the 5th grade even if it is compulsory.

2.1.1.6 ITS in 2017

The draft curriculums developed were presented to public opinion and suggestions on <http://mufredat.meb.gov.tr/Programlar.aspx> in January 2017 and updated through the feedback received (Gölbahar & Kalelioğlu, 2018). Curriculums where this consultation process have been completed were implemented as of the 2017-2018 academic year, and the ones whose updating studies are continuing, starting from the 2018-2019 academic year (MEB, 2018). One of the programs where consultation was completed in 2017, and so was put into practice as of the 2017-2018 academic year was the ITS course, which is now among the compulsory courses at secondary school level.

In the curriculum of the ITS course, a unit-based approach has been taken and five basic units have been determined to be taught in 5th and 6th grades of secondary school. These units are respectively offered to students as Information Technologies, Ethics and Security, Communication, Research and Collaboration, Product Creation, and Problem Solving and Programming. In the curriculum, the scope of each unit, its achievements, explanations about the achievements, limitations, warnings and the time allocated for these determined gains were presented clearly (MEB, 2018).

2.1.1.7 ITS in 2019

In 2019, for the first time in the weekly course schedule, the ITS course was included in the compulsory course list between 4th and 7th grades inclusive. In 4th, 6th and 7th grades, 2 hours of teaching per week was chosen for the curriculum. Also, it was decided that the ITS course for 5th grades should be taken over 3 hours weekly. The curriculum of the ITS course for the 2019 yera has not yet been published. It is a pilot application for now, undertaken only in selected schools. These schools are selected from each unit, from seven different regions and from village to city. In school selecting, teachers, students and school situations are taken into consideration by Ministry of National Education. All of these situations are experimental conditions of the program. The curriculum of ITS still continues its pilot implementation.

2.1.2 Main Goals of ITS Course Curriculum

All students at the secondary school level have the option to be taught courses in Information Technology and Software in Turkey. It is aimed to be able to use

information and communication technologies in an ethical, effective and productive manner. The Ministry of National Education suggests the following aims for students:

1. Regarding Information Literacy learning: "Demonstrate the basic knowledge and skills required to use information and communication technologies correctly and safely. Be aware of the individual and social contributions of information and communication technologies in terms of cultural and social aspects and develop a positive attitude. Take personal responsibility for lifelong learning and independent learning. Finally, be sensitive to information ethics, privacy and security issues. "
2. Regarding Communicating by using Information Technologies, Sharing Information and Expressing Oneself in the learning area: "Communicate effectively using information tools, realizing ideas and projects. Organize knowledge and ideas in a way that different target audiences can understand and share them through the media. Communicating with different groups to develop a social and cultural understanding of virtual environments. Moreover, use and manage social media effectively. "
3. Regarding Research, Structuring Knowledge and Collaborative Working: "Access knowledge, analyze information and grasp the power and importance of generating knowledge from knowledge. Use different tools and approaches in the process of structuring knowledge. Produce joint products and projects using various virtual environments, media and software types."
4. Regarding Problem Solving, Programming and Original Product Development: "Develop a strategy to solve a problem and realize a project, and use different perspectives and approaches while generating solutions.

Recognize authoring and programming languages, and use at least one authoring / programming language effectively. Create models and simulations to examine systems and topics. "

Thus, although the use of technology is emphasised at length, only a limited number of aims are expressly concerned with *creating* that technology.

2.1.3 Learning Objectives of ITS Course by Grade Level

In Turkey, the ITS curriculum learning areas are divided by grade, as follows:

- Comprehending Information Technologies,
- Accessing and Evaluating Information,
- Knowledge Management,
- Sharing Knowledge,
- Creating Knowledge,
- Sharing Knowledge as Information Literacy and Self Expression.

Each level of the ITS curriculum brings different skill areas related to the same subject. The teacher prepares the acquisitions related to the subject area by choosing a level according to the class level. MEB provides exemplary acquisitions regarding what should be taught in accordance with the levels (MEB, 2012). For this reason, the student with low skill level may stay at the level of Comprehension of Information Technologies and maybe not reach the level of Sharing Information at all.

2.1.4 ITS Course Curriculum

The objectives published by the Ministry of National Education (MEB, 2018) in the ITS course teaching program are as follows,

- **The Objectives of the Curriculum for 1st to 4th Grades in 2017**
 - To gain awareness of the correct and effective use of information technologies,
 - Use technology in an ethical and safe way,
 - Developing an understanding of the use of technology for communication and research purposes,
 - Developing products using Information Technologies
 - Gain problem solving and computational thinking skills
 - Understanding of algorithm design.
 - To be able to use different logic structures for Problem Solving
 - To undertake programming by designing their own games.
- **The Objectives of the Curriculum for 5th and 6th Grades in 2017**
 - Being individuals who understand technological concepts, systems and processes well as a digital citizen,
 - To use information technologies effectively and appropriately,
 - Access, research and use internet-based services,
 - To create a general understanding and technical knowledge of computer science,
 - Acquire and develop problem solving and computational thinking skills,
 - Follow and evaluate the reasoning process,
 - As a part of the learning process, it is learned collaborative work skills, benefit from social environments and share what have learned,
 - Searching for learning opportunities in the internet environment,
 - To be able to express them verbally and visually by developing an understanding of algorithm design,

- To be able to choose and apply the appropriate programming approach to solve problems,
- Build technical knowledge in programming,
- To be able to use at least one of the programming languages,
- To carry out studies on product design and management,
- Develop innovative and original projects to solve the problems encountered in daily life (especially problems faced by elderly and disabled individuals, etc.),
- To gain awareness about lifelong learning

- **The Objectives of the Curriculum for 7th and 8th Grades in 2017**

- To be individuals who understand technological concepts, systems and processes well as a digital citizen,
- To use information technologies effectively and appropriately,
- Access, research and use internet-based services,
- To create a general understanding and technical knowledge of computer science,
- Acquire and develop problem solving and computational thinking skills,
- To be able to follow and evaluate the reasoning process,
- As a part of the learning process, they should acquire cooperative working skills, benefit from social environments and share what they have learned,
- Searching for learning opportunities in the internet environment,
- Develop an understanding of algorithm design and express it verbally and visually,
- Apply by choosing the appropriate programming approach to solve the problems,
- Establishing technical knowledge about programming,
- Use at least one of the programming languages,
- To carry out studies on product design and management,

- Develop innovative and original projects to solve the problems encountered in daily life (problems faced by elderly and disabled individuals, etc.),
- To raise awareness about lifelong learning.

2.1.4.1 The Goals of Curriculum for the ITS Course in Recent Years

Building on these aims, the newly developed objectives of the pilot curriculum for 4th, 5th and 7th Grades in 2019, as published by the Ministry of National Education (MEB, 2019) are as follows,

- Understand that the information technologies that they will use more intensely in every stage of their daily lives are actually problem solving tools,
- To recognize the software, hardware and design features of any information technology they see in their environment,
- Recognize the components of smart devices that make up the input-process-output process,
- By accessing the existing knowledge with information technologies, helping people and creating value for life.
- experience that will enable them to realize that they can develop new knowledge and products to add,
- To realize that all kinds of information technology products they use in their daily lives are developed by using information from different disciplines, especially basic sciences,
- Understand that by using software, hardware and design components of information technologies, they can develop the technology they see in their environment and what they imagine by trial and error,
- Develop the algorithm of a smart digital product to solve a problem that is given or determined by themselves,
- Select the necessary software, hardware and design technology to turn an algorithm they have developed into a product,

- Recognize the importance of universal literacy skills such as creative thinking, critical thinking and collaborative work in the process of developing a smart digital product to solve a given or self-determined problem,
- Evaluate the smart digital systems they see and use in their environment thanks to the production skills they have gained in the field of information technologies, and determine their shortcomings and the aspects that can be developed,
- To realize that technological products did not emerge suddenly throughout history, but emerged over time by the combination of knowledge and experiences produced by people from different disciplines,
- Understand that technology has an important transformative effect throughout history that can affect every stage of life,
- Understand the common entrepreneurial characteristics of science and technology people who have added value to humanity and the world throughout history,
- Bullying, fraud, etc. that may be encountered in the internet environment. to understand the methods of protecting themselves from negativity.

2.2 Coding & Programming on Computer

Coding is the first step of the computer software program. Coding literally means all of the operations that are required to be done according to certain conditions and order. Coding is defined as the whole or part of a sequence of commands written to perform a process on the devices formed by a computer or electronic circuit and mechanism (Aytekin, Sönmez, Yücel & Kulaözü 2018).

A computer, laptop, tablet or smartphone is required as hardware to encode. The product of coding can be a web page, software, animation, digital story, computer game, robot code, smartphone application. Coding starts with a purpose and it is necessary to plan the stages that will achieve this goal. The planning is actually phasing what will be done verbally. It is seen that coding, whose importance is

increasing day by day as the alphabet of the new age, stands out as the basis of the areas that direct the information and communication technologies of our age, such as producing software, creating applications and making websites (What is Coding? Why is it Important ?, 2018).

Traditional coding languages can be difficult to learn due to their structural complexity. It is important where the software to be made will start where it will end, which parts should be applied over and over again. Even if a simple game is coded in the computer environment, a plan should be made by thinking in detail before proceeding to the programming stage. Recently, open source platforms such as Scratch, Google Blockly and Code.org, which make it relatively easy to code, are user-friendly and have many visual features, have been developed for children and individuals who are just starting to learn coding (Aytakin, Sönmez, Yücel & Kulaözü, 2018).

The commands are given to computers are called "programs". The process of bringing together the commands that tell how to behave in computers and direct the computer is called "programming" (CodingBK, 2017). Programming means generating a solution to a problem and then creating this solution as a communication pattern, term and idea that the computer can understand (Akpınar & Altun, 2014). The programming process starts with the identification of the problem. While creating the program, the problem should be analyzed first, the solution of the problem in daily life should be found and this solution should be transformed into commands that the computer can understand (Çölkesen, 2002). Computer programming requires the simultaneous use of different skills in problem solving and production processes (Ersoy et al., 2011). Programs can be defined as scripts created by programmers. The

computer is taught through a series of instructions which results in the commands, actions and actions in the program. In order for the program to perform these actions, the instructions must be written in a computer language (Cutter & Kocabaş, 2007).

With the first computer, there was a need for a search that could make the computer work and communicate between the computer user. This need was first tried to be met with machine language. In the process, programming languages that are more usable and easier than machine language have been developed. The syntax the programmer uses to explain what he wants the computer to do is called the programming language (Demirkol, 2016). The first programming language, ENIAC Coding System, appeared in 1943. The languages such as ALGOL, LISP, FORTRAN, BASIC were developed in the 1950s, and programming languages like C and Pascal were developed in the 1970s. Especially after the 1980s, with the emergence of personal computers and the developments in computer hardware, C #, C ++, Delphi and Java languages, which are high-level programming languages, were used for the creation of computer programs (Eryılmaz, 2003).

2.2.1 Programming Training

With the spread of computers, their use in problem solving, and the emergence of personal computers, the desire to write individual programs has increased. The increasing desire to write individual programs has led educational institutions to start focusing on trainings. (Kalelioğlu, Gülbahar, Akçay & Doğan, 2014). Programming language in the world and Turkey has place taken as a training course in different educational institutions and levels. One of the main purposes of programming training is to instruct the learner to develop their problem-solving skills, to look at problems

from different angles, and to employ information technology tools by providing programming training (Akpınar & Altun, 2014). The principal aim of programming lessons is to provide students with high-level thinking skills rather than the product that will emerge at the end of each lesson (Sinap, 2017). In programming applications, problem analysis, expressing the problem regularly and systematically as an algorithm, transforming the resulting algorithm into a flow chart visually, and shaping the functions with predefined symbols improve the thinking skills of the individual (Kert & Uğraş, 2009). Akpınar and Altun (2014) state that if computer programming and design tools are taught to students, the following skills will improve;

- Students' problem solving and analytical thinking skills,
- Computational thinking skills,
- Spatial thinking skills,
- Willingness to make big product-oriented projects,
- Collaborative working and learning skills,
- Development of learning habits and culture by doing and teaching to the computer.

It is important to involve programming in the curriculum of primary training institutions to pick students who have a special interest or talent or find the opportunity to improve themselves (Baz, 2018; Demirer & Sak, 2016; Yecan, Özçınar, & Tanyeri, 2017). The software to be used in programming education should be convenient for the advancement of the students. Moreover, they can make progress according to their own pace and knowledge level (Baz, 2018). There are softwares designed for students of all age groups for programming education. Choosing the right tool in programming education is crucial to attract students' attention, increase

their motivation, and ensure that they achieve their teaching goals (Sayan, 2016). The software varies in terms of age groups, language and assistance support status, and fees. Baz (2018) examined 40 softwares at his research to choosing the right programming environment. In the outcome of his research, it is seen that other softwares have less features and functions than Scratch, Code.org and App Inventor softwares. The age group these softwares appeal to; Being large, free of charge, supporting all operating systems, having language support and having a large number of sample projects increased their popularity.

Karadeniz (2017) stated that programming that uses technology not only to play games, but also to design their own games, create their own animated films instead of watching ready-made animated films, create their own social media tools rather than using social media tools, develop and patent information products in the digital world. He also stated that we need to teach "Programming" to train innovative individuals. Demirer and Sak (2016) stated that training a productive generation will increase the development levels of countries by meeting the personnel needs in software field thanks to coding training. Countries that want to be at the top in the reach of technology have begun to update their curriculum and to ensure programming education from an young age. Providing programming education in early childhood gives the ability to get solve problems encountered their daily routine, in fields such as Mathematics, Turkish, and to establish the relationship between events and to solve the problem in a shorter time (CodingBK, 2017). A group of programmers, including Mark Zuckerberg, Sergey Brin and Bill Gates, among the world's largest programmers, founded Code.org in order to spread programming learning in United States of America (USA). Various institutions and organizations organize events and courses

for anyone who wants to learn programming at an early age. Platforms such as Code Dojo, Code.org, Code Academy and Khan Academy were created in cooperation with volunteer educators to teach code writing in the field of programming (Demirer & Sak, 2016).

2.2.2 Programming Training in the World

In a study conducted by the European School Network in 2015, ministry of education of twenty-one countries were interviewed to examine how they included coding in the country's curriculum. The including programming language and countries are showed as in Table 3 (Hubwieser et al., 2015). Accordingly, a part of the national, regional and local curriculum is taken place on 16 countries (Balanskat & Engelhardt, 2015). These countries include Austria, Belgium, Bulgaria, Israel, the Republic of Ireland, Slovakia and the United Kingdom.

In the study conducted by Sayginer and Tüzün (2017), it was found that these countries discussed programming education under different topics such as computer programming, programming education, code training, coding, and algorithm. In addition, it has been found that the importance that countries give to programming education has increased after 2010 and there is a tendency to give education at an early age.

According to Gardiner (2014), Estonia teaches first grade students how to create their own computer games, England teaches computer programming to children from the age of five with the updated national curriculum program, and stated that the USA has convinced 28 million people to code with the "Hour of Code" activity. According to

Dinç (2019), USA and Europe, who understand the importance of coding, are working fast to teach coding to children.

In the ranking of the best universities coding training in the world by HackerRank, the first three places are Russia, China and Vietnam, while the USA is ranked fourth (Minsky, 2016). The study of coding training to students of Russia dates back to 1985. Namely, in the Soviet Union, the subject of "Fundamentals of Informatics" was introduced to the 9th and 10th grades of all schools in 1985 in order to create an "algorithmic thinking and computer literacy" in students. Later, between 2010 and 2012, a new Federal Education Standard was announced, making the subject of informatics compulsory in all secondary schools, as well as giving high schools the opportunity to choose to include it in their basic or advanced level.

Table 3: Programming Languages Used by Countries

Programming Languages	Country
Java	France, Finland, USA, New Zealand, Israel, Sweden, Germany, Korea
C++	USA, Isreal, Sweden, India, Korea
Python	France, New Zealand, Korea, India
AppInventor	USA, New Zealand
BASIC	Finland, Sweden
C	Isreal, Korea
C#	USA, New Zealand
HTML	France, India
JavaScript	USA, New Zealand
Pascal	Finland, Isreal
VisualBasic	USA, New Zealand
ALGOL	Sweden

COBOL	Sweden
FORTRAN	Sweden
Logic Programming	Sweden
PHP	New Zealand
VB Script	India

It has been determined that countries mostly prefer Java as the programming language and that they use it while providing programming education in K-12 schools. It is seen that while Java, App Inventor, C ++, C #, JavaScript and Visual Basic languages are preferred by the USA, it is seen that they prefer Java, C ++, BASIC, ALGOL, COBOL and FORTRAN languages in Sweden.

2.3. Programming Languages

2.3.1 Arduino Programming

Arduino is an open-source, easy-to-use, flexible, equipment and application-based physical programming platform that contains an input-output properties and an application of some language that is becoming more and more common in the teaching and training community (Rubio et al., 2013; Arduino, 2019). It is used in simple electronic learning applications and performs actions by communicating with programmable sensors or Light Emitting Diode (LED) or can be operated as a peripheral with a computer (Herger & Bodarky, 2015). They are electronic products that everyone can buy ready-made and with the components installed, or they can print their own circuits by downloading the printed circuits (Yılmaz, 2014). A wide variety of developers have chosen Arduino as a development platform for all kinds of computing systems. In addition, thanks to its open source code, it is supported by a large user group who publish their projects and solutions (Rubio et al., 2013).

Interactive objects can be developed using Arduino alone, or various products can be created by establishing connections with software running on the computer (Yılmaz, 2014). Massimo Banzi introduced Arduino to help students with no previous experience in programming to make working prototypes that connect from real world, physical world to the digital world (Arduino, 2019). It has a large number of online courses and a large user group base to ask questions. Barr & Stephenson (2011) and Grover & Pea (2013) state for this reason, it has been chosen as a suitable platform for developing teaching-learning initiative and especially cognitive thinking and problem solving processes. Arduino has achieved favorable results in the cognitive domain in order to learn an object oriented coding to students in secondary school (Jang, Lee, & Kim, 2015).

2.3.2 Drone Programming

Python programming language is used to programing to drones. Guido Van Rossum developed to Python programming language as a successor to the old ABC language. In 1980s, Python language began at Centrum Wiskunde & Informatica (CWI) in the Netherlands. Combining simplicity and ease of learning with powerful capabilities and performances, Python is one of the most well liked programming also coding languages throughout the world (Severance, 2016). Python is a programming language that is basic learning and running written code due to its simple language structure, and an easy to read feature thanks to indented code writing. Python is an ideal language for fast application development with its interpreted code structure and working between operating systems. Object oriented programming, classical structured programming or functional programming can be done in Python. Python is seen as a language that beginners can easily adapt to (Korkmaz, 2020).

It is defined as the Integrated Development Environment (IDE) that supports the necessary environment for the progress of computer programming. According to JetBrains (2019), PyCharm is an integrated development environment designed by JetBrains for software development with Python. It includes analysis of a written code and graphical debugger capabilities. The software created within the scope of the thesis was coded using Pycharm.

In their study, Grandell, Peltomäki, and Salakoski (2006) were asked to compare students with a different programming language experience with the Python language, at the end of their research they just learned that Python is much easier and fun to learn.

2.4 Education

Developing technology has affected every field significantly as well as the content of education that constitutes the structure of society. Science in the change and technology has created differences in instruction in the way of content, purpose and functioning. Especially, the rapid distribution of the Internet in all societies has been effective in changing learning and teaching environments. These changes are outlined in Table 4 (Reigeluth, 1999).

Table 4: Changes in The Education System From The Industrial Age To The Information Age

Industrial Age	Information Age
Teacher Centered	Student Centered
Learning At School	Lifelong Learning
Competitive Relationships	Cooperative Relations

Specific And Limited Program	Flexible And Open Program
Follow The Schedule.	Creating Diversity And Difference
One-Way Communication	Network
Result Oriented Education	Process Oriented Education

Developing science and technology have transform a necessary element of educational institutions, which are the basis of the social structure, as well as taking their place in every field, providing the necessary environments for the improvement and use of different methods and tools. In addition, the concept of "lifelong learning" means that education can take place in environments independent of place, time and space. Throughout the history, the most important element of the education process, "learner" has been neglected. In recent years, the learner has come to the forefront in research on education and training, what and how he learns and the factors affecting him have been among the research topics.

The rapid change in technology has created a significant difference between the student profile that was raised 10 years ago and the current student profile, and the needs, objectives and expectations of the learning group have changed. As the speed of public and the rise in the number of individuals using technology accelerated access to information, new structuring was required in the education system.

2.4.1. FTFL Education

FTFL is the type of learning where teacher and student are in the same environment and social interaction that is the mostly. As it is the oldest approach used in teacher

education, it is still the most preferred approach today due to the high social interaction.

In the literature is examined, the face-to-face education has shown many benefits. In face-to-face education, students are part of the real world and effectively communicate with their friends (Osguthorpe & Graham, 2003). Here, the quality of social interaction depends on the teacher's course management. Since it does not require skills such as using a computer, anyone can participate easily (Çilenti, 1998). In the same environment, the deficiencies are more easily understood and these deficiencies are easier to overcome in the learning environment. Teacher's come up to the subject inspires the student and encourages them to learn more (Osguthorpe & Graham, 2003).

Face-to-face education has its strengths as well as its weaknesses in the learning environment. Since face-to-face education requires a real environment for learning, students who cannot reach this environment for any reason do not have the chance to take the courses they could and were able to attend whenever they want. The notes that need to be kept during the lesson decrease the productivity in learning (Ellis, 2001). There may be difficulties in teaching difficult-to-learn subjects for learning that happen in a certain time zone and the teacher cannot compensate for this situation even if s/he asks examples of content, activities are restricted.

2.4.2. BL Education

Based on the theories of Vygotsky, Piaget, Keller, Gagné, Bloom, Merrill, Clark and Gery suggest that BL environments are of great importance for the learning group to be able to clear up the complications they encounter efficiently, and where they

consider all the conditions while solving these problems, applying what they have learned in their lives and establishing social relations with the course participants (Sasa, 2011).

Many definitions have been made for BL in the literature (Singh & Reed 2001; Finn & Bucci 2004; Clark & Mayer 2003; Horton 2000; Driscoll, 2002; Bielawski & Metcalf, 2003; Young, 2002), and many researchers have defined the definition of BL from person to person, from culture to culture (Horton, 2000).

Some researchers such as like Osguthorpe and Graham (2003) stated it is combining the take advantages of face-to-face and online approached using classroom and web based environments for being the best learning. However, Martyn (2003) and Singh (2003) define it as a mixture of various event-based learning exercises, combining face-to-face classroom, individual and e-learning environments.

According to Balci (2013), BL focuses on the perfect accomplishment of learning results, and this is done by matching the right individual learning style with the right learning technology in order to gain the right skills to the right person at the right time. __

Besides, BL can be considered as an environment where different methods and strategies are used together. BL is the use of more than one teaching method together in order to increase the quality of teaching (Rossett, 2002). According to Driscoll (2002) BL involves combining several teaching phases with or without educational technology to obtain the most appropriate teaching output. Also, BL is a combination of different methods (House, 2002).

The balance of FTFL and e-learning in BL applications also differs from lesson to lesson and from subject to subject. Therefore, while FTFL methods and strategies are used more dominantly in some courses, e-learning technologies can be used more in some courses. It also states that BL can be implemented as a succession of online and face to face activities or a combination of both environments, depending on the need. The combinations can be applied chronologically in the form of time blocks. In addition, students and / or instructors can decide on their own the levels of blending learning (Yalçın, 2020).

In this researching, according to the need for BL applications, first face-to-face activities and then online activities are applied as a combination of both learning environments. In these applications, the students come together in a FTFL environment under the supervision of an instructor, according to pre-determined coding training. For future coding applications, the same students and teachers continue to interact in the online environment.

It brings together the advantages of BL , e-learning and FTFL, minimizing the disadvantages, and making education more efficient and conveying knowledge from generation to generation in a more meaningful way. Although this is the general purpose in BL, it does not have a universal formula, and how to develop the BL lesson design varies according to many factors such as the culture, purpose, learning styles of the participants, and the conditions.

2.5 Computational Thinking (CT) Skills

2.5.1 Definition and Theory of CT

In computer science, CT has deep and long history. In between 1950s and 1960s, the concept mostly expressed the mental processes used to formulate problems such as inputs and outputs (Denning, 2009). Over time, it expanded to examine thinking at different levels of consideration, the using of math to advance algorithms, and the solutions of problems in terms of different variables (Carnegie Mellon University Center For Computational Thinking Carnegie Mellon, n.d.). The named “computational thinking” which was head referent in the study of Seymour Papert in 1996, was not defined in detail in the related research. After the concept was brought back to the agenda by Jeannette Wing in 2006, its importance has increased in recent years in parallel with its popularity and technological developments. The increasing popularity of the concept of computational thinking has led to an increase in research conducted in that field. However, According to Hemmendinger (2010), uncertainty prevails in many of the definitions made. In-depth definitions and the scarcity of examples related to these definitions cause problems in revealing exactly what CT is. As reported by Wing (2006), CT is a means of determine complication, designing systems, and comprehension personal behavior by emphasizing the basic concepts of computer science. Wing (2006) defines CT as a basic skill that people have, like reading and writing. This definition remains general and abstract at the point of teaching the concept of CT and its integration into the curriculum (Deng, Huang, & Dong, 2009). The researchers attracted this circumstance and more in-depth research has begun. One of the first definitions is the definition made by Carnegie Mellon

University (Center For Computational Thinking Carnegie Mellon,. t.y.). They define computational thinking as:

- Problem solving and system design based on the basic concepts of computer science
- Using different levels of abstraction to understand problems and solve them more effectively
- Being able to produce more efficient, logical and safe solutions, to be able to think algorithmically and to use mathematical concepts
- Defines the scale results as understanding not only for productivity, but also for economic and social reasons.

Other similar definitions have been made, and although these all contribute to the improvement of CT, they are not at a satisfactory level in terms of being a detailed definition (Weinberg, 2013). Workshops on exactly what the concept is were organized, and people's ideas about computational thinking were received. One of these workshops was the one organized by the National Academies in the USA in 2009 to discuss computational thinking. In the National Research Council (NRC) workshop, whose report was published in 2010, a common definition of computational thinking could not be made (NRC, 2010).

Nevertheless, the joint work of ISTE and CSTA emphasises that finding problem solution is CT process. The main features of this process are listed as follows:

- Using computers or different tools for solving problems
- Logically analyzing relevant data
- Organizing the data more clearly using setup and duplication
- Translation of explanation into machine language using algorithmic thinking

- Finding, analyzing and applying the most effective combination of resources and process steps
- Generalizing the solution of the problem and transferring it to different problems (CSTA & ISTE, 2011).

It is seen that with the aim of gaining computational thinking in educational environments, computational thinking must be defined educationally. At this point, it can be said that the definitions made are more practical. With the perception of computer science as programming in the 1960s and 1970s (Denning, 2009), coding training to students has gained popularity. Many computer scientists also thought that CT skills could be gained by students while instruction programming (Weinberg, 2013). Brennan and Resnick (2012) made the interpretation of CT depend on their Scratch-based studies. Their definition also stated that computational concepts, computational applications and computational perspectives consists of three basic within CT. Moreover, they mentioned seven computational concepts that are most common in Scratch projects and that fit into other programming languages or content other than programming languages. These concepts are process steps, loops, parallelism, events, conditions, operators and data. Computing at School (2014), operating in the UK, defined computational thinking with six concepts and five approaches. Their concepts are logic, algorithm, parsing, pattern, abstraction and evaluation. In addition to these five approaches: correction, creation, debugging, continuing and cooperation, are included in the explanation of CT.

In addition to these definitions, some researchers have emphasized that computational thinking is related to other thinking styles. Some of these thinkings are; procedural, algorithmic, recursive, and critical thinking. Computational thinking covers these

concepts and more (Weinberg, 2013). It is also stated that computational thinking has common aspects with analytical thinking, engineering thinking, math thinking and also scientific thinking (Wing, 2010). Some researchers thought that these definitions were not sufficient in terms of psychometric approaches and they have emphasized that computational thinking should be defined psychometrically as well (Román-González et al., 2018). They tried to reveal the relationship between computational thinking and cognitive and non-cognitive parameters with various studies. By revealing that self-efficacy and personal characteristics are related to computational thinking, they emphasize that educators should also consider these parameters while integrating computational thinking into educational environments.

2.5.2 Importance of CT

The inclusion and application of CT in almost all areas of work has changed the way today's works. Human mind which one of the most powerful problem solving skills ever owned, the ability to expand the power of human thought with computers and other digital tools has become an important part of people's lives and work. The students have to solve problems that arise when using computers and other digital tools and understand how, when, and where to help. They also require to understand how to communicate with others and can help people with the computer-aided solutions.

In various disciplines, the students already learn many elements of CT skills in various disciplines. However, all students should be offered a learning opportunity that will enable them to use all their relevant abilities together for themselves (Barr, Harrinson, & Conery, 2011). The way to do this is including computational thinking

in the educational program. However, the main purpose in training CT is not to get students to assume like a computer. Instead, it teaches them how to think like an economist, physicist, artist, and how to use CT to determine issues, calculate efficiently, and discover new questions (Hemmendinger, 2010).

2.5.3 Features of CT

According to Wing (2006), the study of computation called to “Computer science” on his study. She stated that computational thinking has the following features by saying “What can and how can it be calculated”. According to Wing, Computational Thinking;

- Thinking about people, not computers. CT is one step people determine problems; it doesn't try to get people to think of it as like a computer.
- Conceptualization is not about programming, coding, or creating code. Computer science; it is not writing code, running codes, compiling, or computer programming. Any kind of programming on a computer says more than just thinking of a computer scientist. It requires thinking at multiple levels of abstraction.
- Basic, not rote skill. A basic skill is something every person should know in modern society.
- Complementing and combining maths and engineering thinking. Computer science, by its nature, like all sciences, is based on mathematical thinking, given that its formal foundations are based on mathematics.
- Ubiquitous, not works. It is not just the software and hardware we produce that will physically be ubiquitous and touch our lives at all times. Ideas are for

approaching and solving problems, managing our constantly lives, communicating and interacting within another people, along with the computational concepts used.

- CT will become such an integral reality to human endeavor and will perish as an open philosophy for everyone and everywhere.

In addition to these explanations, Wing added the following statements to the characteristics of computational thinking in her study published in 2008.

- The type of analytical thinking is CT. It distributes the general ways of our approach to solving a problem with mathematical thinking. It also shares the general ways of designing and evaluating a large, complex system that operates within engineering thinking, within the constraints of the real world.
- The essence of CT is abstraction. In the computer, the concepts of time and space beyond the physical dimensions are abstracted. Abstractions are general because numerical abstractions are just a special case. "An algorithm is an abstraction of a step-by-step process to get input and produce some desired output." explained in the form.

2.6 Programming Training and Computational Thinking

ISTE (2015) emphasizes that high-level skills such as algorithmic thinking, creative reasoning, critical reasoning, determine complication, and collaboration learning should be acquired in order to produce solutions to real life problems in the 21st century, where information technologies are an indispensable part of life. Therefore,

computational thinking, which includes these skills, emerges as an important variable that has been examined in many branches in recent years.

Computational thinking is having the knowledge and capacity to use the computer to produce solutions to problems and to use the mental skills required to produce systematic solutions (Özden, 2015; Voogt et al., 2015; Wing, 2006). CT also includes skills such as designing computers and other technological tools for problem solving, analyzing data, developing algorithmic thinking skills for problem solving, developing the right algorithm to reach the most effective solution, and transforming the problem solution to other problem solutions (National Academies of Science, 2010). CT is a way of thinking that includes many sub-skills (algorithmic thinking, creative thinking, critical thinking, collaboration, problem solving) rather than a single skill. The computational thinking can be explained by sub-factors:

Algorithmic thinking: It is the method of reaching the most efficient result by evaluating all stages and finding creative solutions in the face of problems (Pei, Weintrop, & Wilensky, 2018; Futschek, 2006). In this way, the individual can transfer the solutions they find to different problem situations (Sedgewick & Wayne, 2011). Before solving a problem in algorithm development, problem solutions should be created in considered. Performing the steps for the rules in algorithm development, testing the planning and correcting the errors will provide the best result for problem solving (Csizmadia et al., 2015). Since the algorithm constitutes the basis of programming education, it is considered important to develop this skill (Korkmaz et al., 2017). In the study undertaken as part of this dissertation, algorithm teaching was carried out to provide a basis for students to develop their programming skills.

Creativity: Creativity means revealing a new thought or product from existing knowledge (Sternberg, 2003; Sünbül, 2002). In order to develop creativity potential, it is stated that students at every education level should have skills such as valuing different ideas, being open to criticism, and not being afraid of making mistakes (Özden, 2014). Among the suggestions for developing creative thinking skills are creative activities that develop problem-solving skills in schools, providing environments where ideas can be expressed by students in the classroom and rewarding creativity (Erden & Akman, 2004; Sternberg, 2003).

Critical thinking: It is a way of determining that is necessary to distinguish right and wrong by looking at events from different perspectives without being under the influence of others (Mason, 2009). Confirming correct information, distinguishing assumptions and facts, presenting evidence for correct information, getting rid of prejudices, and being informed one's own understanding are skills that must be acquired for critical thinking (Vural, 2005). Listening to the ideas and experiences of students in the classroom, helping to guide complex discussions, and heartening students to request questions are among the suggestions for developing critical thinking skills (Kabilan, 2000). Critical thinking significantly contributes to problem-solving skills as it gives individuals the ability to view events from different angles (Mojica, 2010).

Collaboration: It is a learning method applied in the classroom for students to develop their communication skills by doing the tasks given in groups, while realizing high-level thinking (Daniel, 2004). In order for cooperative schooling to be effective, it is important that all members in the group work actively, establish social relationships, and make decisions together (Michaelsen et al., 2014; Xin et al., 2007).

The collaboration process that takes place in this way contributes to the academic year success of the students. Also, the students can be evaluated both in groups and individually within cooperative learning (Aziz & Hossain, 2010).

Problem solving: Individuals try many ways to determine issues or problems they face in daily routine. While some of these are solved through experience, some may be realized in ways that require many thinking skills, based on previously acquired knowledge (Bayman & Mayer, 1988). The problem solving process requires creative and critical thinking skills (Alvarez & Larranaga, 2015), and other factors that might also helping in the problem solving process are: using computers and technological tools, organizing data, modeling and concretizing abstract data, producing algorithms and using these in the most effective way, and generalizing problem solving skills for other problems (Grover & Pea, 2013).

It has been observed that the definitions of sub-dimensions of CT skills are nearly related to 21st century learner skills (ISTE, 2018). In addition, it is claimed in the literature that CT skills are common skills that should be acquired by all students in terms of computational science and that every student should acquire these skills (Shute, Sun & Asbell-Clarke 2017; Voogt et al., 2015; Witherspoon & Schunn, 2019).

Oluk and Korkmaz (2016) conclude that there is a highly positive interrelation among programming skills and CT skills. Chaudhary et al. (2016), on the other hand, found that robotic applications, besides providing the knowledge and skills in course content, had a positive effect on CT skills. Atmatzidou and Demetriadis (2016) found that robotic applications contributed to the CT skills of students regardless of their gender and education level, but this contribution increased towards the end of their

studies, as it takes time to acquire these skills. In a study conducted with primary school students, robotic activities improved students' CT and reasoning skills (Chen et al., 2017). Karaahmetoğlu (2019) determined that project-based Arduino educational robot coding applications significantly contribute to the Computational thinking skill problem solving sub-factor of middle school students. Kılıç et al. (2019) examined the perceptions of vocational high school information technology students towards Computational thinking skills and found that both the total and the sub-dimensions of the variable were at medium level.

In this present study the author has examined the effect of coding training on CT skills levels of secondary school students in two different learning environments, both FTFL and BL. The study is original and important as determining the level of development of students' CT skills on coding training in different learning environments. It is hoped that the study will also be a guide for new researchers in this field due to the very few studies in this direction.

2.6.1 Place in Curriculum of CT

It has been noted that trends towards integrating computer science into the K-12 curriculum have increased significantly in the last years. This situation affects the future workforce due to many factors including economic and technological demands with computer skills (Chen et al., 2017).

It is stated that CT makes use of some concepts for computing and information science. These are essential concepts for finding solutions to problems, organizing and evaluating complex systems, and understanding human reasoning and behavior. It has been said that this way of thinking has significant effects in computer science and

in many areas. Accordingly, it was determined that the CT should be taught in primary schools and should be included in the education curriculum of every university (Buitrago et al., 2017). According to Barr and Stephenson (2011), it was stated that two efforts were required for the concepts of CT to be included in the K-12 educational program successfully. Also, education policy should be changed in order to prevent infrastructure problems and K-12 teachers should be supported with resources with a good definition and appropriate examples.

While many coding training programs state that rarely look in-depth at the styles of thinking used in CT. It has been said that there is no curriculum that forms the basis of CT in school especially in Mathematics and Science lessons. Therefore, models that support and assist CT are needed in existing classroom applications. Because there is very little opportunity for the existing school curriculum to adapt to the new content. However, many problems in the current curriculum can be solved with computational thinking (Shute et al., 2017).

Most researchers emphasize the importance of passing on CT to the next generation. In addition, teaching this idea is very necessary for them to gain skills (Grover & Pea, 2013; Hodhod, Khan, Kurt-Peker, & Ray, 2016; Kafai & Burke, 2013; Lu & Fletcher, 2009; Voogt, Fisser, Good, Mishra, & Yadav, 2015). Hemmendinger (2009) stated that it is insufficient to teach everyone to think like a computer scientist in order to make CT available for the K-12 curriculum. Instead, it should solve the emerging problems and enable all disciplines to discover new questions. According to Czerkawski (2013), it was stated that the dissemination of the CT throughout the existing education and training program would be possible with the conceptualization and implementation of the subject by the instructors.

Various professional societies and think tanks have also made attempts in the USA, England and the Netherlands to emphasize CT in the curriculum (Dede, Mishra, & Voogt, 2013). In countries such as the United Kingdom Finland and the USA, CT has been added to the existing curriculum. However, how CT is taught in these countries is changing. For example, the Education Department in the UK has determined goals for each stage and explained what the student should do or know at each stage depending on the age. In Finland, teachers are encouraged to find computational thinking through their own practice in their lessons. Although some of the teachers could choose the way they wanted to teach the subject, it was stated that they were uncertain about how they should approach the new subject (Faber, Wierdsma, Doornsbos, Van der Ver, & De Kevin, 2017).

The following studies have been conducted to include CT within curriculum in all school periods starting from pre-school period.

2.6.1.1 CT in Preschool

The programming and development of basic programming concepts has recently attracted attention especially at pre-school education levels. However, this level of education has also increased the interest of education and science unions. In studies on coding in the field of international education, it has been observed that programming of young children has a significant effect on developing their cognitive skills (Papadakis et al., 2016). Being able to write programs on the computer or coding, which means the whole of the operations that are expected to be done under certain conditions and in a certain order, is a type of literacy that has recently emerged

for the pre-school education level. Scratchjr (2017) is a programming language that is used for children between the ages of 5-7 in the preschool period to learn coding. It is a software that young children in pre-school period can code with ScratchJr. With this software, not only interacting with the computer, but also learning how to create codes written on the computer and how students can express themselves better. In this period, children who have pre-school level learn to find solutions to the problems that arise and to design projects that they can create on their own. In addition, it develops skills that form a strong foundation for academic success, which has an important place in students. It can also act as a bridge with other interdisciplinary subjects (eg mathematics and language) as meaningful, interesting and supportive. In this way, they support their numeracy and literacy development in early childhood (Scratchjr, 2017).

Papadakis et al. (2016) conducted a pilot study on ScratchJr to reveal the importance of coding training and CT to pre-school education level. This pilot study took place in the 2014-2015 academic year and was conducted in Crete, Greece. In addition, 43 pre-school students were included in this study. At the end of the study, they stated that pre-school students had active participation and that the ScratchJr application attracted their attention. Finally, it showed that the students participating in this study improved their CT and Digital skills (Abstraction, Division And Synthesis Abilities).

One study was conducted by González & Repiso (2017) with 131 kindergarten students in Spain. The purpose of this study is: It was done for the development of CT and teamwork with educational robots for pre-school students. Here, it is seen that the robots used for students are programmable (Bee-Bot). Considering the results of the

study, it was seen that teachers at the relevant level enabled them to benefit from this technology for students at the preschool level in order to improve CT and teamwork.

2.6.1.2 CT in Primary School

Sanford & Naidu (2016) emphasized the importance of CT requirement for primary school students. This requirement should be included with other branches such as reading, writing and mathematics for primary school students. Because of it, they stated that the spreadsheets included in the existing training program are the best tool to increase CT skills.

Lopez, Gonzalez & Cano (2016) worked with many primary school students in Spain to improve their CT skills. In these studies, students used the Scratch programming tool at primary level through coding projects and activities. This programming tool, which is a Visual Block Programming Language, positively improves students at primary school level.

According to Von Rekowski et al. (2015), students studying at a primary school with immigrant children in the Ruhr region in Germany were asked to perform various tasks based on CT theory. These training and programming tools were used while carrying out activities aimed at increasing the CT skills of children studying at primary school level with electronic textile (E-Textile) products and LilyPad Arduino. Within the scope of these studies, it was stated that the use of electronic textile products and the LilyPad Arduino is a very useful tool for CT. It has been stated that there are several important basic components when using these products. Among these there are factors such as structuring in CT, visual suitability, originality, making the visuals meaningful and understanding the materials used in the products.

Noh & Lee (2019) aimed to measure the level of CT and creativity of students by using Entry and a Hamster robot in their study conducted with 155 Korean primary school students. The students had the opportunity to improve their CT skills by performing their Bebras tasks. At the same time, they used a test tool developed by Torrance (1974) to measure the creativity of primary school students. As a result of the research, it has been revealed that the CT skills of students with programming and robotics courses at primary education level have increased significantly, however, the originality has developed in direct proportion to CT skills. In addition, they stated that robotic programming is of great importance for elementary school students, and it raises their CT skills to a very high level.

2.6.1.3 CT in Secondary School

Wolz, Stone, Pearson, Pulimood, & Switzer (2011) conducted a demonstration project involving secondary school 7th and 8th grade students and teachers by incorporating the CT process into the curriculum to teach students interactive journalism. As a result of the project, it has been stated that students and teachers at the secondary school level, which do not include themselves as "types of mathematics", develop a positive attitude towards computational thinking and programming.

Webb & Rosson (2013), carried out the Scratch training with four activities designed to help secondary school girls acquire CT concepts and skills using scaffold examples and minimalist workbooks. In addition, they concluded that the students received a fun and engaging learning activity, and that it was also an effective way of communicating CT concepts and skills in a short time.

Denner, Werner, Campe & Ortiz (2014) conducted a study with 320 secondary school students at K-12 level to determine the effectiveness of dual programming. In this study, the CT levels of the students on the Alice programming tool were examined regardless of the time or amount of interaction with a partner or teammate. As a result of the study, the Alice programming tool was advantageous to increase the CT levels of K12 level students.

Jenkins (2017) England, South Wales and secondary level education class by establishing dialogue with teachers, students studied the situation in the CT skill development. In the research, it was carried out by using the Snap application as a programming tool. As a result of the research, it has been said that encouraging the dialogues created in the classroom has positively improved the CT skill levels of students studying at secondary school level. However, it was stated that the dialogues established in the classroom could be a useful tool for CT skill.

2.6.1.4 CT in High School

Lee et al. (2011) examined how CT skill ways are shaped for secondary and high school youth. Cognitive and practical activities are at the forefront in this review. However, it provides a "use-change-create" framework that represents students' three stages in CT. It is suggested to research some researches for the development of CT. Some of them emphasized the need to develop qualified learning environments for students and support teachers working on the subject. It was stated that a project was conducted within the scope of Research Experience for Teachers (RET) in Chicago Laboratory Schools. In this project, some field courses at secondary and high school levels were changed among themselves. With the change happen, they showed how

CT adapted the course contents at different levels to the course contents at the relevant level (Settle et al., 2012). Caspersen & Nowack (2013) presented a new and general approach based on a conceptual framework to integrate CT in Danish high schools.

2.6.1.5 CT in University

According to Peñalvo and Mendes (2017), CT has beneficial effects on university level students even there are various opinions about how to teach technology to students more effectively. Basically, it was stated that studies in different dimensions should be used in order to reuse and interpret existing knowledge with concepts such as CT, abstraction or repetition in an active problem solving method. Therefore, considering the content of pre-university education, it was stated that the purpose of introducing CT is not only to take place as a team user, but to increase their productivity skills related to the subject. (TACCLE 3 & Erasmus +, 2016). According to Ru (2019), it has been stated that there is a qualified education and coding training plan to ensure that the students who study at university level carry CT to a better level in computer basic education. Moreover, University-level educators should adapt the existing curriculum content according to computer basics and their abilities. In addition, it was stated that the experimental education should take more place with the scientific classification of the course content. However, CT stated that the efficient application of scientific CT methods can improve.

CHAPTER III

METHOD

In this chapter, it is identified that the methodological procedures. The role of the researcher, implementation process, research model, demographic, sampling, design process, variables, data collection tools, application steps and data analysis sections will be explained.

3.1 The Role of the Researcher and Implementation Process

This chapter contains information about the researcher's role and the implementation process. Information on the description of the environment in which the research was conducted is presented based on data for the 2019-2020 academic year.

The researcher works as an information technology teacher at a free-paying secondary school in Beşiktaş, Istanbul. ITS courses are among the compulsory courses for the year 5 and 6 as twice a week, and are among the non-compulsory courses for the year 7 also twice a week. In the school, the classes have an average of 17 students and the lessons are conducted in a semi-crowded environment in the information technologies laboratory within the school. In the research, coding training is carried out individually in Arduino educational robotic coding applications in a FTFL environment and in drone coding applications in a BL environment.

The researcher is in the role of "participant teacher and observer" as the organizer of the activities, ensuring that the scales are filled by the participating students, and the conductor of the observation and interviews.

3.2 Research Model

This research is a quantitative research and it has been done according to the experimental model. Experimental model is the most valid and reliable way to observe the effects of the variable and measure cause-effect relationships when it is used appropriately (Büyüköztürk et al., 2012). The word ‘skill’ is defined as solving problems or completing tasks by applying knowledge, that is, it is the power to do a job (Güneş, 2012). Considering the CT process as a skill enables the research to be done quantitatively.

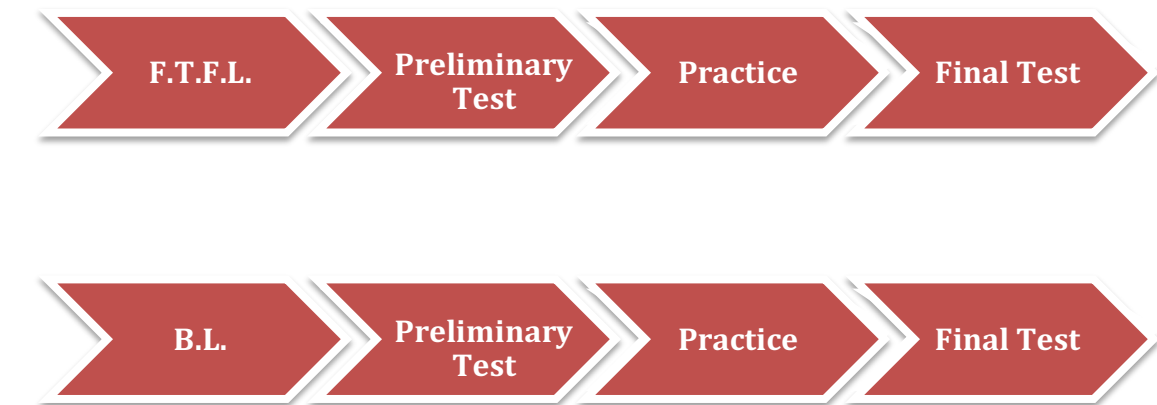
The design of the research is one group preliminary and final tests design among the weak-experimental research models. This pattern is also defined as a one group preliminary and final tests design among the weak-experimental research models. experimental within groups.

Figure 1. Research Design



In this design process, there is an intervention in the group between preliminary and final tests. At the final process, the significance of the change between the preliminary and final tests results of a single group is revealed (Büyüköztürk, Kılıç Çakmak, Akgün, Karadeniz, & Demirel, 2012). Not being able to control the factors affecting internal validity in this design is a statistical disadvantage.

Figure 2. Thesis of Research Design



The model of the research is summarised in Figure 2. Before the coding training, "Computational Thinking (CT) Skill Level" scale is used as a preliminary test for FTFL environment. After the scale, coding training course starts as practice. When the coding training courses finish, CT Skill Level scale is applied as final test. For BL, coding training courses start. At the final of the practice, the final test is used for BL.

3.3 Demographic and Sampling

The population of the research was formed by the students who were in year 7, 12-13 age group taking selective ITS course in a private secondary school in Beşiktaş, Istanbul within 2019-2020 academic year.

Before the research was done, the students took the ITS course, which is a compulsory course of the Ministry of National Education, for four terms (making up of 2 academic years) for the students in year 5 and 6 per academic year. During this period, the students have been given the learning outcomes of the Ministry of

National Education ITS course educational program, and their subject headings are given below:

- Importance of ITS in Daily Life
- Hardware and Software of Systems
- File Extension and Properties
- Digital Ethical & Citizenship
- Privacy and Security
- Networks of Web
- Security Online Researching
- Technologies and Cooperation
- Visual Processing Programmes
- Word Processing Programmes
- Presentation Programmes
- Spreadsheet Programmes
- Audio and Video Editing Programmes
- Coding, Algorithm, Concepts and Approaches
- Programming

During the students' coding training in FTFL environment with Arduino educational robotic coding applications and BL environment for drone coding, the students do not have knowledge or experience with either of the applications. The students learn both of these applications for the first time in the research process using Arduino APP and PyCharm. The students were the first to use the applications with this research. The Students were selected from five different classes by lot method. A total of 60 (N =

60) students, including 12 students from each class, were determined with the Disproportionate Stratified (Stratified) Sample Selection, with half of the students being girl and half boy. The characteristics and numbers of students in the study are shown in Table 5.

Table 5: Students' Class, Gender and Frequency Data

Groups	Class	Gender	N	%
Students	7 Grade	Girls	30	50%
		Boys	30	50%
Total			60	100%

According to Table 5, the students were seventh grade students in researching. Moreover, The total number of students participating in the researching is 60. Half of this total number of students is girls, the other half is boys. When looking at the percentages is 50% of the total number of students are girls and the other 50% are boys.

The stratified sampling is used to create samples with a higher representation by reducing sampling errors. This sampling technique is a technique that allows a much higher precision with a smaller sample size and a lower cost compared to simple random sampling (Sencer & Sencer 1978). The essence of this technique is to divide the population into some sub-layers before selected the sample, and later to select the samples from these layers (Sencer & Sencer, 1978). The population in the research has the feature of being the same everywhere. It is also divided into sub-steps and selected appropriately. However, the sample is not chosen from the entire population.

For example, if there is research about university students, first the population can be divided according to the students' classes, then the required number of students are selected among the students in each class (Earl, 2004). After the strata are determined, individuals from strata are selected again by simple random or systematic sampling selection technique. Criteria such as gender, age, socio-economic status, place can be used to stratify a research population. More than one layer can be used in stratified sampling. For example, after stratifying participating students by grade, we can also stratify them by gender and grade point average. It is important that the variables used in the study need be related with population into layers that used to be separated the criteria (Sencer & Sencer, 1978). For example, if we are going to examine the medical condition of students, their weight or height may be an appropriate criterion to stratify the sample, but it may not be a meaningful criterion because their gender is unrelated to their medical condition.

When the sample selection is examined, stratified sample selection can be seen. This sample type consists of two parts. These are proportional and disproportionate samples. When looking at the selection of the disproportionate stratified sampling, the number of samples from each section is selected first. Afterwards, its proportion in the universe is not taken into account and the sample is taken equally. It is seen that a sample determined in this way will not be well efficient. It is stated that each section in the universe will be suitable with the sample, if it is of sufficient size and quality to complement each other (Sencer & Sencer, 1978). For example, the assuming that the research population consists of 100 people working in a workplace, 20% of this population is women and 80% is men. With the disproportionate sampling technique,

it is possible to make up half of the sample from women and half from men, and if we are to choose a sample of 10 people, we can choose five from women and five from men. In order for the groups that make up the sample to indicate the sample, the sampling rate should be exactly the same as in the population. This is possible due to proportional stratified sampling, which is a sampling type (De Vaus, 1990). Considering the previous workplace example, when 10 people from this workplace of 100 people are selected by proportional stratified sampling technique, 2 women and 8 men will be selected for the sampling.

In the stratified sampling technique, units from each stratum will be selected for sampling, whether proportional or non-proportional, each stratum will be represented. In other words, there is no stratum not represented in the sample.

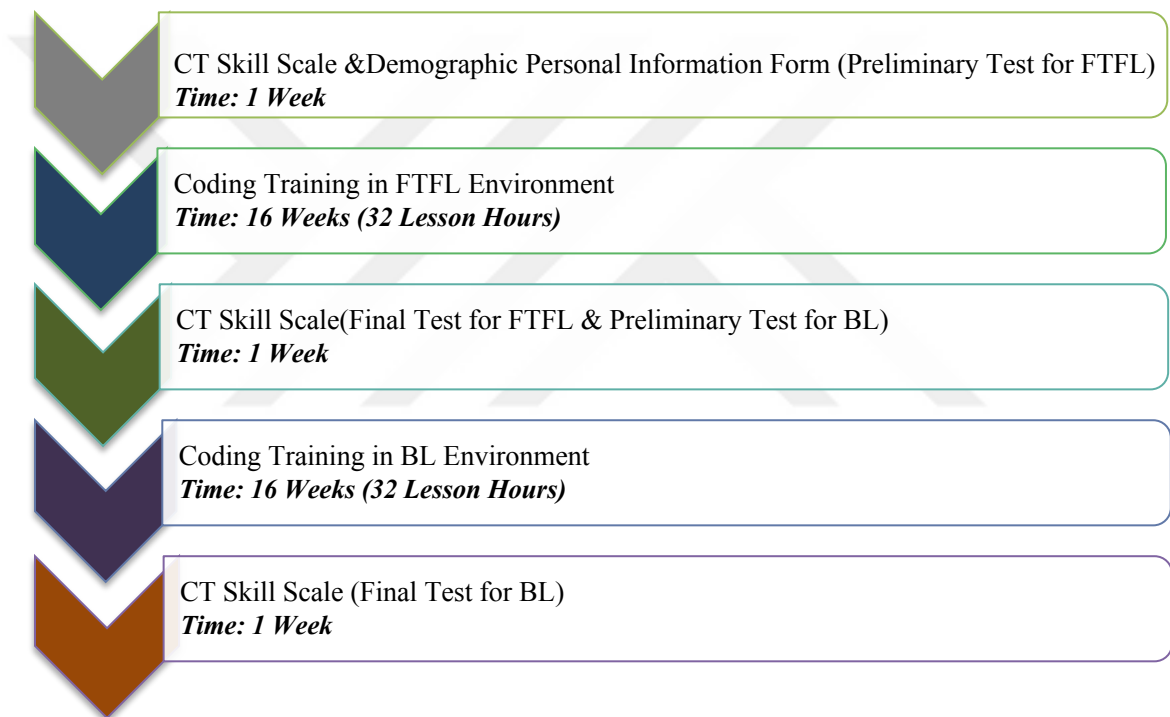
Table 6: Students' Groups, Gender, Variables and Frequency Data

Groups	Gender	N	%
FTFL Environment	Girls	30	50%
	Boys	30	50%
BL Environment	Girls	30	50%
	Boys	30	50%

The student's groups, gender, variables and frequency data are summarised in Table 6. First of all, total number of students on FTFL and BL environment is 60 students. In both learning environments, 50% of the total number is girls and the other 50% is boys. In the researching, there are 30 girls and 30 boys for each learning environment.

According to Table 6, the distribution of the students participating in the research are showed the frequency and percentage values according to the gender factor variable are given. 12 students were chosen from each class, half of them were girls and half were boys. According to this selection, for the FTFL environment of coding training, 50% is male and 50% is female students. Likewise, for the BL environment of coding training, 50% is the male and 50% is the female students.

Figure 3: The Sample of the Research and Steps



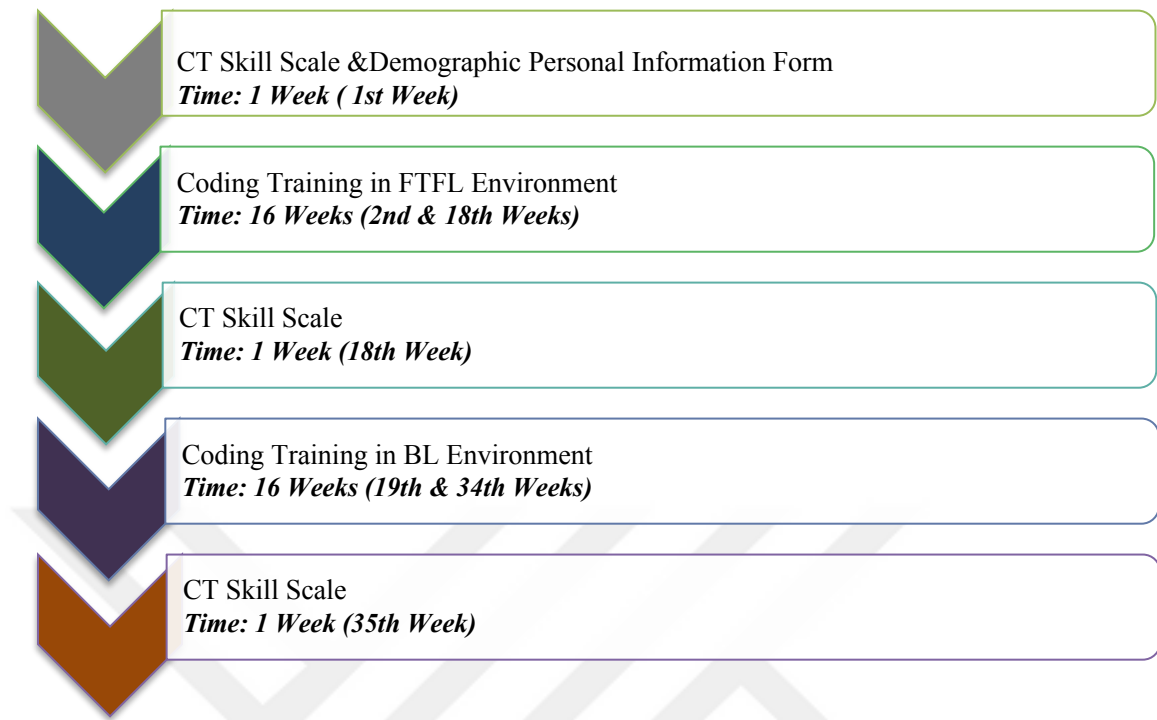
The sample of the research is summarised in Figure 3. In first steps, CT Skill Scale and demographic personal information form are applied. In coding training in FTFL environment is for thirtytwo lesson hours and then CT Skill Scale is applied. In coding training in BL environment is given for thirty two lesson hours Finally, CT Skill Scale is applied.

3.4 The Design of the Process for Coding Training

The study was appeared out in two stages ended within 35-weeks period. At the beginning of the study, CT Skills Scale (Korkmaz, Çakır, & Özden, 2015) and Personal Information Form were used as a FTFL environment for preliminary test to measure the CT skills of the students. during the next 16 weeks, the Arduino educational robotic coding application course was introduced in the FTFL environment, activities and applications were carried out for teaching the concepts related to programming, and electronic circuits were established and lessons were taught in this way. After this stage, the same scale was applied to the students again as FTFL environment for final test. At the same time, the relevant scale was included as BL environment for preliminary test. During the next 16 weeks, the students' drone coding applications course in BL environment was introduced and programming activities and applications were implemented.

In this section, theoretical information about phyton programming application and its environment and drone coding lessons with simple applications were taught face to face in the computer laboratory for the first 6 weeks. For the next 10 weeks, lectures, applications and drone coding applications were carried out in a virtual classroom environment for distance education. At the end of the process, the same scale was applied to the students as BL Environment for final test. In the 36th week, online focus group meetings were held. In Figure 4, the stages including the application process of the research are included.

Figure 4: Application Process and Stages of the Research



The application of the research process and stages is summarised in Figure 4. In first week, CT Skill Scale and demographic personal information form are applied. In the second week has coding training in FTFL environment between second and eighteenth weeks for sixteen weeks and then CT Skill Scale is applied in the eighteen week. In the next week, coding training in BL environment is given for sixteen weeks between nineteenth and thirtyfourth weeks. Finally, CT Skill scale is applied in thirtyfifth week.

3.5 Research Variables

The variables can be examined in four groups as dependent, independent, controlled and invariance variables according to their controllability.

3.5.1 Independent Variables

This independent variables of research has students' gender, learning environment supported by Arduino educational robotic coding applications in FTFL Environment, drone coding applications in BL Environment.

3.5.2 Dependent Variables

These variables of this research; Computational Thinking (CT) Skill Levels.

3.5.3 Controlled Variables

The controlled variables of this research are; the effect of CT levels of students in different learning and coding training environments.

3.5.4 Invariance Variable

The person making the application of the invariance variable of this research is also the researcher.

3.6 Data Collection Tools

Since the research is a quantitative research, the scope of the research used within quantitative data tools. The quantitative data tools used within the scope of the study are "Computational Thinking Skills Scale" and "Personal Information Form". Data collection tools and application times during the study are shown in Table 7.

Tablo 7. Used Data Collection Tools

Data Collection Forms	Implementation
Personal Information Form	FTFL Environment Preliminary Test
CT Skills Scale	FTFL Environment Preliminary Test & Final Test BL Environment FinalTest

The used data collection tools are summarised in Table 7. First of all, there are two data collection forms that are personal information form and CT skills scale. The personal information form was implemented only for FTFL environment with preliminary test. However, the CT skills scale was implemented FTFL environment for preliminary & final tests and BL environment for only final test.

3.6.1 CT Skills Scale

The "Computational Thinking Skill Level" scale developed by Korkmaz, Çakır and Özden (2015). The scale used how much the Arduino educational robotic coding applications in the FTFL Environment and drone coding applications in the BL Environment affect the CT skill levels of the students. Some researchers have designed the scale to reveal the CT skills of students studying at both secondary

school levels and University level. In addition, this scale design consists of many survey questions. The large number of sub-items was reduced in number to 22 survey questions. This scale includes five point likert scale used in education and researching. When these survey questions are examined, it is seen that the scale is divided into various categories. These options such as Never (1), Rarely (2), Sometimes (3), Usually (4), Always (5) are grouped together. For the categories created in the scale, the results of Cronbach's alpha (α) values for each option were found and confirmatory factor analyzes were made. Cronbach Alpha reliability coefficient was found as .809. The values of the Cronbach Alpha reliability coefficient, which emerged depending on the options, were observed to yield results between .640 and .867. After the confirmatory factor analysis, seven items with very low regression values were removed from the original scale. After the confirmatory factor analysis performed on the 22-item scale, the observed values were found to be acceptable. The reliability analysis results regarding the overall scale and its factors are showed in Table 8.

Table 8. Internal consistency coefficients of CT Skills Scale

Factors	Survey Questions	Cronbach Alpha
Creativity	4	0,640
Algorithmic Thinking	4	0,762
Collaboration	4	0,811
Critical Thinking	4	0,714
Problem Solving	6	0,867
Toplam	22	0,809

According to Table 8, it is seen that the scale can make reliable measurements since the internal consistency of the Computational Thinking Skill scale is sufficiently high.

3.6.2 Personal Information Form

In the demographic information form prepared by the researcher, there are questions created to obtain information about the participants' gender, age and class level. Based on the aforementioned literature, since variables such as gender, age, and class level were considered to have an effect on students' skills like computational thinking, creativity, collaboration, algorithmic thinking, problem solving, and critical thinking. Also, these variables were included in the analyzes conducted within the scope of this study.

3.7 Research Application Steps

The application part of the research was carried out with 86 seven grade students from five class selected with appropriate sampling in a private secondary school in Beşiktaş district of Istanbul during the fall and spring terms of the 2019-2020 academic year. However, a total of 60 students, including 12 students from each of the five selected classes, were determined.

During the application with data collection tools, two hours of lesson, can be also called as a double lesson (40 + 40 minutes) were used for both application. During the autumn term of the application, it was carried out for sixteen weeks within the scope of "Arduino Educational Robotic Coding Applications in FTFL Environment". Following this, it was carried out for sixteen weeks within the scope of "Drone

Coding Applications in BL Environment" by the researcher. The training process of this research, in which the effect of coding training on Arduino educational robotic coding applications in FTFL and Drone coding in the BL environment is investigated. It started on September 9, 2019 and ended on June 19, 2020.

All of the coding training in FTFL Environment took place in the classrooms where students were studying and in the computer lab. In addition, drone coding lessons with simple applications were taught face to face in the computer laboratory for the first 6 weeks. For the next 10 weeks; lectures, applications and drone coding applications were carried out in BL Environment were realized by distance learning in a virtual classroom environment without first weeks.

The code training was applied in the first week of the autumn term, when the students started their classes, and a preliminary test of the computational thinking skills scale was conducted for the FTFL environment. It was then applied again in the last lesson of the autumn term and the CT skills scale was final test given for the FTFL environment, and at the same time, the relevant scale was used as a preliminary test application for the BL Environment. In the spring term, the final test application of the computational thinking skills scale for the BL environment was applied in the last ITS course of the students. By obtaining the final test data, the stage of analysis of the data, creation of findings and reporting of the research was initiated. Detailed calendar of the implementation process is included via Table 9.

Table 9: Arduino Applications Process in FTFL Environment

Weeks	Action	Chapter & Application Processes
1st	CT Skill Scale & Personal Information Form	Collection of Preliminary Test for FTFL Environment.
2nd	Teaching Process of Arduino	Problem Solving Concepts and Approaches
3rd	Teaching Process of Arduino	
4nd	Teaching Process of Arduino	
5th	Teaching Process of Arduino	
6th	Teaching Process of Arduino	Use of Programming Tool With Arduino Arduino Educational Robotic Coding Applications in the FTFL Environment of Coding Training
7th	Teaching Process of Arduino	
8th	Teaching Process of Arduino	
9th	Teaching Process of Arduino	
10th	Teaching Process of Arduino	
11th	Teaching Process of Arduino	
12th	Teaching Process of Arduino	
13th	Teaching Process of Arduino	
14th	Teaching Process of Arduino	
15th	Teaching Process of Arduino	
16th	Teaching Process of Arduino	
17th	Teaching Process of Arduino	
18th	CT Skill Scale for FTFL Environment	Collection of Final Test for FTFL Environment
19th	Teaching Process of Drone	Problem Solving Concepts and Approaches
20th	Teaching Process of Drone	
21th	Teaching Process of Drone	
22th	Teaching Process of Drone	Using Drone Coding Applications With PyCharm
23th	Teaching Process of Drone	
24th	Teaching Process of Drone	
25th	Teaching Process of Drone	
26th	Teaching Process of Drone	Drone Coding Applications in BL Environment
27th	Teaching Process of Drone	
28th	Teaching Process of Drone	
29th	Teaching Process of Drone	
30th	Teaching Process of Drone	
31th	Teaching Process of Drone	
32th	Teaching Process of Drone	
33th	Teaching Process of Drone	
34th	Teaching Process of Drone	
35th	CT Skill Scale for BL Environment	Collection of Final Test for BL Environment

General aspects of implementation are:

- In a total of 35 weeks, Coding practices applications were carried out. In the first week, before the application, "Personal Information Form" and "Computational Thinking (CT) Skill Level" scale were used as a preliminary test for FTFL environment.
- Arduino educational robotic coding applications in FTFL Environment and drone coding applications in BL Environment are divided into weeks under four main parts. During Table 10 is shows research applications parts. The first part covers the lecture practices in the first week. The second part includes lecture practices between the 2nd and the 17th weeks. Also, in the third part, the lecture practices in the 18th week were carried out, and in the fourth part, the lecture practices of the 19th and 35th weeks were included.

Table 10: Research Applications Parts

	First Part	Second Part	Third Part	Fourth Part
Weeks	1st Week	2nd – 18th Weeks	18th Week	19th – 35th Weeks

3.7.1 First Part (1st Week)

- In the 1st week, the applications took place in the computer laboratory where the students were studying coding. In addition, a preliminary test application

of the Personal Information Form and the CT Skills Scale for the FTFL environment was conducted.

3.7.2 Second Part (2nd – 18th Weeks)

- In the 2, 3 and 4 weeks, the unit of "Problem solving concepts and approaches" of Arduino C / C ++ programming language was presented to the students with the same lesson plans. In these weeks, students:
 - Types of Data for Arduino,
 - The concept of constant and Types of Variables,
 - Concept of Constant and Variable,
 - Separating a problem within sub-problems,
 - Advancing an algorithm for the explanation of the problem or issue,
 - Checking the explanation of an algorithm is covered.
 - Using basic functions,
- In the 5th week, coding information was given about the FTFL Environment and the Arduino Application programme, which is the Arduino educational robotic coding applications programming tool. In addition, the first code samples were written and uploaded.
- In the 6th and 7th weeks, the students were informed about the programming of educational robotic applications and the rules of circuit construction with the Arduino Application programme.
- Finally, in these weeks, students have made activities that require basic usage such as "Arduino App Interface Recognition", "Recognizing Simple Circuit Elements", "Loading Codes", "Setting Up a Simple Electronic Circuit". In addition to this, students also were given:

- Purpose of Arduino educational robotic applications,
- Sensors used in applications,
- Information about programming Arduino educational robotics applications.

For the rest of the weeks, firstly, the reasearcher gave a seminar to students on their lesson project applications. During the seminar information was given on the definition of Arduino and Robotic Applications, Arduino Types, Application steps, sample projects. Listening to the requests of the students, they were seated in groups of two in the same row, but individual work was carried out. Applications of the week specified by the researcher that distributed to the students. A total of 12 application studies were conducted by the students. The students tried to complete the tasks by setting up the circuit on which they saw the screenshot and by coding the circuits they had built. Hence, the students encountered some issues in which they needed to use variable, loop and condition structures. In this way, students were supported by activities so that they could discover programming concepts. The names of the applications prepared by the students are given below.

- 2 LED Blink and 3 LED Blink Project Application
- Simple Traffic Light Project Application
- Traffic Light and Pedestrian Crossing Project Application
- "Saymon Says" Game with Led & Button Game Application
- A Photoresistor (LDR) - Night Light Project Application
- HC 04 - Digital Meter Application
- HC 04 - Light and Sound Vehicle Parking Sensor Application
- 16x2 A Liquid Crystal Display (LCD) Simple Text with Screen Application
- 16x2 LCD Digital Clock Project Application
- Weather Station Project (Humidity and Temperature) with LCD Screen
- Application

- Illuminated Digital Piano Application
- Smart Air Conditioning Application

Finally, the final test given for FTFL environment was applied to the groups as a "CT Skill Level" scale.

3.7.3 Third Part (18th Week)

- In the 18th week, the final test was used to the students that completed the Arduino educational robotic coding applications in FTFL environment. In addition, a preliminary test scale was not applied for the BL environment, instead the final test scale was used for the FTFL environment.

3.7.4 Fourth Week (19th – 35th Weeks)

- At 19, 20 and 21 weeks, the unit of “Problem solving concepts and approaches” of the Python programming language was presented to the students with the same lesson plans. During these weeks, the following subjects were taught with the students.
 - Type of Data,
 - Constant and Variable concept,
 - Using the Concept of Constant,
 - Variable,
 - Separating a problem within sub-problems,
 - In problem solving process, used basic functions,
 - Advancing an algorithm for the explanation of the problem,
 - Checking the explanation of an algorithm

In addition, an online seminar was given by the researcher on the course contents to be applied to the students on drone coding applications. In the seminar information was given on the definition of drone, such as Types of Drones, Place of the Drone in Human Life, Programmable Drones, and sample projects.

In the 22nd, 23rd and 24th weeks, each class was informed about the PyCharm programme, which is the drone coding applications programming tool, and the first code samples were written and their codes were compiled. In the first application, the students were taught to takeoff the drone and practice the landing code in Python Programming their environment, and then apply it. In this process, students have implemented applications requiring basic usage such as "Interface Recognition of Python Program", "Recognizing Basic Drone Direction Movements", "Loading Codes to Drone". In the following weeks, the list of activities students has carried out for drone coding applications using the PyCharm Program is listed below.

In the 25th week, the students were informed about drone coding applications programming with PyCharm programme and drone coding applications were carried out. The applications of the week determined by the researcher were given to the students. A total of 12 application studies were conducted with the students.

In other weeks the names of the applications carried out by the students are given below:

- Drone - Landing and Takeoffs
- Drone - Four Way Movement (Forward, Backward, Right, Left One Direction Movement)
- Two Dimensional Shapes in the Air (Square, Triangle, Rectangle Drawings)
- Drone - Rotation (Other Applications and Clockwise / Counter-Clockwise Rotations)
- Drone - Drone Track Design - 1 (Organ Transplant Project)
- Drone - Drone Track Design - 2 (Seed Supplement Project)
- Three Dimensional (3D) Shapes in the Air (Cube, Pyramid, Rectangular Prism)
- Drone - Tilt in Four Ways
- Drone - Drone Track Design - 3 (Skyscraper Cleaning Project)

- Drone - Drone Track Design - 4 (Forest Fires Extinguishing Project)
- Drone - Use Self-Drive

After the students in the group learned the commands about drone coding and the tasks were told, they tried to programme the drone and fly it. Followed by trial and errors and after some hiccups, the student encountered some serious issues by using the flight logic of the drone, including how to program 2-dimensional and 3-dimensional objects in the air and how the drone would move with this programming, using variables and loops. In this way, students were supported with activities so that they could discover programming concepts.

In the 35th week, a final test application of the CT skills scale for the BL environment was performed for the students who completed the drone coding applications in the BL Environment. Then, after collecting the experimental process and quantitative data, online focus group interviews were conducted for qualitative data.

3.8 Data Collection and Analysis

SPSS 25.0 program was used as a tool in the statistical analysis of the data obtained during the research process. The design of the research, the type of measurement, the presence of the study group, its distribution and the number of factors are the variables that should be taken as criteria for choosing the appropriate statistical test (Çokluk Bökeoğlu, 2018).

Determining the normal distribution of the data was the first of the many steps. The degree of symmetry distortion in a normal distribution is called skewness.

The distribution is called right (positive) skewed if the right-tailed is long-tailed, and left (negative) crooked if the left is long-tailed. The degree of sharpness or roundness of the normal distribution curve is called kurtosis (Yıldız, Akbulut & Bircan, 2002). While evaluating it, the skewness and kurtosis values and the presence of the study group were taken as a basis. The skewness-kurtosis value of ± 1.5 forms a basis for deciding whether it is suitable for normal distribution (Tabachnick & Fidell, 2012). In addition, according to the central limit theorem, each mass consisting of 30 or more numbers from any part of the study group of the population shows a normal distribution (Demir, Saatçioğlu, & İmrol, 2016).

The significance of the difference between the preliminary and final tests skill tasks of FTFL and BL Environments, which were found to show normal distribution, was measured with the "Dependent (Related) Sample T-Test". Collecting at least two data from the same study group means that the data are "dependent-related". This test is used whilst comparing repeated measurements from subjects or measurements from paired groups. In order to measure the t-test of the related variables, both data sets should show a normal distribution. It can be assumed that the difference in skill test scores observed with this test is significant because of the procedure applied in cases where external effects are considered constant (Büyüköztürk, 2011).

The analysis is explained below for each data collection tool separately. After the checks on the data and providing the assumptions, the process of writing findings and comments started.

3.8.1 CT Skills Scale

It was observed that the items formed within the scope of the Computer Thinking Skills Scale included in the research scale were categorized. They were formed as: Never (1), Rarely (2), Sometimes (3), Usually (4), Always (5) and became available at this scale. It was collected according to the information provided by the students for the questionnaire and classified in a suitable type for the scale. The classification type used in the relevant scale was the five-point likert type. The scores obtained by calculating the information provided by the students were not standard. The reason for this situation was that it consists of many different subcategories and that the number of items is not the same. The items that make up this scale: Creativity consists four questions, Algorithmic Thinking consists four questions, Collaboration consists four questions, Critical Thinking consists four questions, also Problem Solving consists six questions and total number of items has twenty-two Survey Questions. In addition, these substances do not have a standard feature. In order for the results to be of a standard quality, the results must be structured so that the minimum raw score is 20 and the maximum is 100 points. Because this developed scale aims to reach a standardized CT score regardless of the characteristics of the group in which the scale is applied. The formula below can be used to convert raw scores into standard scores:

$$x_{standard\ score} = \frac{x_{raw\ score}}{Number\ of\ survey\ question\ in\ the\ scale} \times 20$$

The levels corresponding to the scores obtained from the factors can be summarised as follows:

- 20-51: Low Level,
- 52-67: Medium Level,
- 68-100: High Level.

Frequency, percentage, arithmetic mean, standard deviation, independent sample t-test and related samples t-test were used on the data calculated in this way to identify the learning levels of students by computer. The difference analysis was based on $p > .05$ significance level.

The distribution should show normality in order to use parametric tests in the analysis of research data. If the distribution does not show normality, non-parametric tests are used instead of parametric tests (Büyüköztürk, 2007). In line with this information, Kolmogorov-Smirnov normality test was used to determine whether the data obtained in the study showed normality or not. The normality test results ($p > .05$) were based on the significance level.

In general, the data collection tools used in the study and the tests applied to these data as a result of entering the data into the computer environment are given in Table 11.

Table 11. Data Collection Tools and Implementations

Data Collection	Implementation	Group / Level
CT Skills Scale	Tests of Normality	7th Grade Students
	Paired-Samples T-Test	
	Independent Samples T-	
	Test	

Test of Normality, Paired Samples T-Test and Independent Samples T-Test were applied in the analysis of CT Skill Scale, and its levels consist of 7th grade secondary school students.



CHAPTER IV

FINDINGS

In this chapter, the findings acquired by statistical analysis of the data collected regarding the problems examined during the research are included. Findings and results obtained from the analysis are included in the section below.

The effects of coding training applied in the 7th grade level of a secondary school on FTFL and BL environments on CT skills were examined. On the other hand, there was a need for analysis to determine whether there is a significant difference in CT skills in terms of different variables. Whether the effect of coding training on CT skills in FTFL and BL environments, which is the main purpose of the study, is significant or not was revealed by the research. The general descriptive statistics of the study group's CT skills are given in Table 12, Table 13 and Table 14.

IBM SPSS Statistics 25.0 statistics software was complied for data analysis. The level of significance addressed in the analysis was accepted as ($p \leq 0.05$). In cases where the significance level is p value ($p > 0.05$), it is concluded that there is no important difference in the study examined. However, in cases where the p value ($p \leq 0.05$), it is signed that there is a statistically significant difference in the study examined (Can, 2019).

In the data analysis where the data obtained as a result of the research showed normal distribution. It is also taken place within t-test that are Paired-Samples T-Test and

Independent-Samples T-Test within the scope of parametric test were used for the effect and difference significance of the scales.

The findings from which there was no important difference between the scores of the scales used in the analysis of the research questions were also included.

4.1 Results Regarding the Secondary School Students' CT Skill Levels

In the thesis, CT Scale was implemented as a preliminary and final tests to examine the CT skill development of the students joining in the study before and after coding training in a FTFL environment. Whether these scales, which are dependent variables, show normal distribution or not is explained in the following data. Findings of the data are showed in Table 12.

Table 12: General CT Descriptive Statistics of the Students

Scale	Statistics	
Preliminary Test for FTFL Environment	N	60
	Mean	66,00
	Median	64.201
	Variance	48.471
	Standart Deviation	6.962
	Min	48.18
	Max	89.09
	Range	40.91
	Skewness	.271
	Kurtosis	1.363

According to Table 12, the average of the preliminary test scores for the FTF environment is defined as N = 60, Mean = 66.00, Range = 40.91, Median = 64.201, Standard Deviation = 6.962, Skewness = .271, Kurtosis = 1.306. The skewness and kurtosis values of the scores of the CT show a normal distribution as the range of ± 1.5 and the number of the study group is over 30 (Tabachnick & Fidell, 2012; Demir, Saatçioğlu, & İmrol, 2016).

Table 13: General Sub Factors of CT Descriptive Statistics

Sub Factors	FTF Learning Environment	N	Mean	S.D.	Min	Max
Creativity	Preliminary Test	60	14.43	1.89	7.27	17.27
Algorithmic Thinking	Preliminary Test	60	13.51	2.88	5.45	18.18
Collaboration	Preliminary Test	60	14.51	3.50	3.64	18.18
Critical Thinking	Preliminary Test	60	13.10	2.73	7.27	17.27
Problem Solving	Preliminary Test	60	10.42	3.58	5.45	25.45

According to Table 13, the average of preliminary test results on CT skill levels for the FTF environment of coding training was statistically analyzed and descriptive statistics were obtained.

As a result of coding training in FTF environments, one of the skill levels of CT;

- **Creativity:** mean is 14.43, the standard deviation is 1.89, the smallest value is 7.27, the highest value is 17.27,
- **Algorithmic Thinking:** mean is 13.51, the standard deviation is 2.88, the smallest value is 5.45, the largest value is 18.18,

- **Collaboration:** the mean is 14.51, the standard deviation is 3.50, the smallest value is 3.64, the maximum value is 18.18,
- **Critical Thinking:** the mean is 13.10, the standard deviation is 2.73, the smallest value is 7.27, the maximum value is 17.27,
- **Problem Solving:** mean is 10.42, standard deviation is 3.58, the smallest value is 5.45 and the largest value is 25.45.

Table 14: General CT Descriptive Statistics of the Students' Gender

Scale	Gender	N	Mean	S.D.	Std. Error	Min	Max
Preliminary Test for FTFL Environment	Girls	30	65.57	7.44	6.54	48.18	78.18
	Boys	30	66.42	6.54	1.19	54.55	89.09

According to Table 14, there are descriptive statistics for the scores of CT skill levels of the students according to the gender of the coding training given in the FTFL environment. Considering the total average for the FTFL environment, the mean score of the girls out of 100 points is 65.57, the standard deviation score is 7.44, the standard error is 6.54, the lowest 48.18 and the highest 78.18 points. Besides, the score of the boys out of 100 points is 66.42, standard deviation score is 6.54, standard error is 1.19, the lowest 54.55 and the highest 89.09 points. It is seen that the coding training takes place in a FTFL environment, CT skills means are very close to each other (Girls = 65.57, Boys= 66,42).

4.2 Do The Secondary School Students' CT Skill Levels Differ When the Coding Training Takes Place In A FTFL Environment?

Within the scope of the study, the preliminary and final tests of the activities of coding training in the FTFL environment were applied. Furthermore, it was viewed to investigate whether significant difference has among the CT skill levels of the secondary school students in the group that was conducted after the application process. For this purpose, the CT skill scores of the students who took coding training in a FTFL Environment were compared with Independent Sample T-Test as Sub-Factors and Total values, and the results indicated in Table 15.

In addition, there are preliminary and final tests analysis for the FTFL environment of the average scores of sub factors, which are five separate factors in the CT Skill Scale. The Paired Samples T-Test was applied as the study method to determine the difference between them and data regarding the Paired Samples T-Test are included in Table 15.

Table 15: T-Test and Total Results of CT Scores of Coding Training in FTFL Environment on Sub Factors

Sub Factors	FTF Learning Environment	N	Mean	S.D.	t	Df	p
Creativity	Preliminary Test	60	14.43	1.89	.772	59	.443
	Final Test	60	14.15	2.33			
Algorithmic Thinking	Preliminary Test	60	13.51	2.88	.288	59	.775
	Final Test	60	13.34	2.95			
Collaboration	Preliminary Test	60	14.51	3.50	1.146	59	.256
	Final Test	60	13.78	4.02			

Critical	Preliminary Test	60	13.10	2.73	.025	59	.980
Thinking	Final Test	60	13.09	3.30			
Problem	Preliminary Test	60	10.42	3.58	-1.701	59	.094
Solving	Final Test	60	11.78	5.12			
Total CT	Preliminary Test	60	66.00	6.69	-.128	59	.899
	Final Test	60	66.16	8.28			

Creativity: For the FTFL environment, the Preliminary test creativity mean score is 14.43, S.D. = 1.89 and the final test creativity mean score is 14.15, S.D. = 2.33 which are shown in Table 15. Also, p value of the creativity factor appears .443. The results show that there is no statistically important difference between the Creativity preliminary and final tests in the way of the creativity levels of the CT skill ($p = .443$, $p > .05$).

Algorithmic Thinking: For the FTFL environment, the preliminary test algorithmic thinking mean score is 13.51, S.D. = 2.82 and the final test algorithmic thinking mean score is 13.34, S.D. = 2.95 are shown in Table 15. Also, p value of the algorithmic thinking factor appears .775. The results show that there is no statistically important difference between the algorithmic thinking preliminary and final tests in terms of the algorithmic thinking levels of the CT skill ($p = .775$, $p > .05$).

Collaboration: For the FTFL environment, the preliminary test collaboration mean score is 14.51, S.D. = 3.50 and the final test collaboration mean score is 13.78, S.D. = 4.02 are shown in Table 15. Also, p value of the collaboration factor appears .256. The results show that there is no statistically important difference between the

collaboration preliminary and final tests in the way of the collaboration levels of the CT skill ($p = .256, p > .05$).

Critical Thinking: For the FTFL environment, the preliminary test critical thinking mean score is 13.10, S.D. = 2.73 and the final test critical thinking mean score is 13.09, S.D. = 3.30 are shown in Table 15. Also, p value of the critical thinking factor appears .980. The results display that there is no statistically important difference between the critical thinking preliminary and final tests in terms of the critical thinking levels of the CT skill ($p = .980, p > .05$).

Problem Solving: For the FTFL environment, the preliminary test problem solving mean score is 10.42, S.D. = 3.58 and the final test problem solving mean score is 11.78, S.D. = 5.12 are shown in Table 15. Also, p value of the problem solving factor appears .094. When the looking at the research results that there is no statistically powerful difference between the problem solving preliminary and final tests in terms of the problem solving levels of the CT skill ($p = .094, p > .05$).

TOTAL CT: For the FTFL environment, the total CT skill for preliminary test mean is 66.00 and S.D. is 6.69. Also, final test mean is 66.16, S.D. is 8.28. in the way of Table 15. When the looking at this research results, it shows that there is no statistically important difference in terms of the total sub-dimension of CT skill ($p = .899, p > .05$).

4.3. Do The Secondary School Students' CT Skill Levels Differ Based On Gender When the Coding Training Takes Place In A FTF Learning Environment?

The question above seeks to find an answer to the coding training given in the FTF Learning Environment of the research about the CT skill levels of secondary school students as regards to their gender. The method includes CT Scale having being applied as preliminary and final tests on Girls and Boys. In this context, Independent Samples T-Test was conducted to determine the CT Skill levels of the students according to gender. Results related to the Independent Samples T-Test for the FTF Learning Environment are dedicated in Table 16.

Table 16: T-Test Results of CT Scores of Coding Training Given in FTF Learning Environment by Gender

FTF Learning Environment					Levene's Test		Equality of Means T-Test		
	Gender	N	Mean	S.D.	F	p	t	Df	p
Preliminary Test	Girls	30	65.57	7.44	.949	.334	-.469	57.07	.641
	Boys	30	66.42	6.54					
Final Test	Girls	30	64.30	7.27	1.133	.292	-1.774	55.74	.810
	Boys	30	68.03	8.91					

According to Table 16, when the results of the CT skill scores of the FTF learning environment by gender in coding training was statistically analyzed on final test, it was observed that there was no important difference ($p = 0.810 > 0.05$). It was

observed that the mean score of CT female students is 64.30 and it was statistically lower than the mean score of boy students is 68.03. It is observed that there is an increase in the scores of boys compared to girl students' means after FTFL environment in the coding training as reported by students' gender.

Table 17: T-Test Results of CT Scores of Coding Training in FTFL Environment by Gender on Sub Factors

Sub Factors	Gender	N	Mean	S.D.	t	df	P
Creativity	Girls	30	13.84	2.42	-1.00	58	.503
	Boys	30	14.45	2.22			
Algorithmic Thinking	Girls	30	12.67	3.03	-1.82	58	.761
	Boys	30	14.03	2.75			
Collaboration	Girls	30	13.66	4.20	-.232	58	.437
	Boys	30	13.90	3.90			
Critical Thinking	Girls	30	12.90	3.63	-.425	58	.289
	Boys	30	13.27	2.98			
Problem Solving	Girls	30	11.21	4.83	-.869	58	.625
	Boys	30	12.36	5.42			

As reported by Table 17, there is no important difference in the all sub-factors of CT skill levels of coding training in FTFL environment ($p > .05$).

4. 4. Do The Secondary School Students' CT Skill Levels Differ When the Coding Training Takes Place In A BL Environment?

In this connection of the study, the preliminary and final tests of the activities of coding training in the BL environment learning were carried out. The main purpose of this study was to examine to find out whether there was an important difference between the CT skill levels of secondary school students in this group after the training process. For this, the CT Skill scores of the students who received coding training in the BL Environment were compared with Independent Sample T-Test as Sub-Factors and Total values, and the outcomes obtained are showed in Table 18.

In addition, the Paired Samples T-Test was applied as an analysis method to determine the difference between the statistic scores of creativity, algorithmic thinking, collaboration, critical thinking and problem solving, which are five separate factors to the CT Levels Scale, with preliminary and final tests in BL environments. The data regarding the Paired Samples T-Test are included in Table 18.

Table 18: T-Test and Total Results of CT Scores of Coding Training in BL Environment on Sub- Factors

Sub Factors	BL Environment	N	Mean	S.D.	t	df	p
Creativity	Preliminary Test	60	14.15	2.33	-2.273	59	.027**
	Final Test	60	14.90	2.02			
Algorithmic Thinking	Preliminary Test	60	13.34	2.95	-.883	59	.381
	Final Test	60	13.72	2.83			

Collaboration	Preliminary Test	60	13.78	4.02	-2.040	59	.046**
	Final Test	60	14.78	3.49			
Critical Thinking	Preliminary Test	60	13.09	3.30	-.959	59	.341
	Final Test	60	13.51	3.00			
Problem Solving	Preliminary Test	60	11.78	5.12	1.130	59	.263
	Final Test	60	10.86	3.63			
Total CT	Preliminary Test	60	66.16	8.28	-1.709	59	.930
	Final Test		67.80	6.16			

According to Table 18, the important difference was observed just within the sub-factors of Creativity and Cooperation between the computational thinking skill levels of coding training in the BL environment ($P_{\text{Creativity}} = .027$, $P_{\text{Collaboration}} = .046$, $p > .05$). In addition, the obtained results were explained one by one as reported by the sub-factors in the CT skills scale.

Creativity: For the BL environment, the preliminary test creativity mean score is 14.15, S.D. = 2.33 and the final test creativity mean score is 14.90, S.D. = 2.02 are shown in Table 18. Also, p value of the creativity factor appears .027. The results show that there is a statistically important difference between the Creativity preliminary and final tests in terms of the creativity levels of the CT skill ($p = .027$, $p < .05$).

Algorithmic Thinking: For the BL environment, the preliminary test algorithmic thinking mean score is 13.34, S.D. = 2.95 and the final test algorithmic thinking mean score is 13.72, S.D. = 2.83 are shown in Table 18. Also, p value of the algorithmic thinking factor appears .381. The conclusion of research shows that there

is no statistically important difference between the algorithmic thinking preliminary and final tests in terms of the algorithmic thinking levels of the CT skill ($p = .775$, $p > .05$).

Collaboration: For the BL environment, the preliminary test collaboration mean score is 13.78, S.D. = 4.02 and the final test collaboration mean score is 14.78, S.D. = 3.49 are shown in Table 18. Also, p value of the collaboration factor appears .046. The results show that there is a statistically important difference between the collaboration preliminary and final tests in terms of the collaboration levels of the CT skill ($p = .027$, $p < .05$).

Critical Thinking: For the BL environment, the preliminary test critical thinking mean score is 13.09, S.D. = 3.30 and the final test critical thinking mean score is 13.51, S.D. = 3.00 are shown in Table 18. Also, p value of the critical thinking factor appears .341. The conclusion shows that there is no statistically important difference between the critical thinking preliminary and final tests in terms of the critical thinking levels of the CT skill ($p = .341$, $p > .05$).

Problem Solving: For the BL environment, the preliminary test problem solving mean score is 11.78, S.D. = 5.12 and the final test problem solving mean score is 10.86, S.D. = 3.63 are shown in Table 18. Also, p value of the problem solving factor appears .263. The results show that there is no statistically important difference among the problem solving preliminary and final tests in terms of the problem solving levels of the CT skill ($p = .263$, $p > .05$).

TOTAL CT: For the BL environment, the total CT for preliminary test mean is 66.16 and S.D. is 8.28. Also, final test mean is 67.80, S.D. is 6.16 in the way of Table 18. As state in these results, it shows that powerful important difference is not seen in terms of the total sub-dimension of CT skill ($p = .930$, $p > .05$).

4. 5. Do The Secondary School Students' CT Skill Levels Differ Based On Gender When the Coding Training Takes Place In A BL Environment?

In order to learn an explanation to the question of coding training given in the BL Environment of the study on the CT levels of the students' gender, the CT Skill Scale was applied as a preliminary and final tests. In this context, Independent Samples T-Test was conducted to determine the CT Skill levels of the students as reported by gender. Findings regarding the T-Test that is Independent Samples for the BL Environment is included in Table 19.

Table 19: T-Test Results of CT Scores of Coding Training in BL Environment by Gender

BL Environment					Levene's Test		Equality of Means T-Test		
	Gender	N	Mean	S.D.	F	p	t	df	p
Preliminary Test	Girls	30	64.30	7.27	1.1133	.292	-1.774	55.74	.810
	Boys	30	68.03	8.91					
Final Test	Girls	30	67.36	5.29	.740	.393	-.549	54.07	.585
	Boys	30	68.24	6.89					

According to Table 19, when the outcomes of the CT skill scores of the BL environment as stated in the gender were statistically analyzed, it was based that there

was no important difference ($p = 0.585 > 0.05$). Female students' mean score is 67.36 and male students' mean score is 68.24). Statistically they are at good levels on CT. It is observed that the scores of girls increase more than boys after the BL environment in coding training as compared with students' gender.

Table 20: T-Test Results of CT Scores of Coding Training Given in BL Environment By Gender on Sub-Factors

Sub Factors	Gender	N	Mean	S.D.	t	df	p
Creativity	Girls	30	15.21	1.98	1.16	58	.854
	Boys	30	14.60	2.05			
Algorithmic Thinking	Girls	30	13.66	2.84	-.163	58	.994
	Boys	30	13.78	2.87			
Collaboration	Girls	30	14.12	4.00	-1.492	58	.033**
	Boys	30	15.45	2.80			
Critical Thinking	Girls	30	14.24	2.69	1.919	58	.347
	Boys	30	12.78	3.15			
Problem Solving	Girls	30	10.12	3.22	-1.603	58	.451
	Boys	30	11.60	3.91			

As regards to Table 20, in the creativity, algorithmic thinking, critical thinking and problem solving of CT skill levels of coding training in BL environment have no important difference on sub factors. ($p > .05$). However, there is important difference in the collaboration on sub-factors of CT skill levels of coding training in BL environment ($p = 0.33$, $p < .05$).

4.6. Are There Secondary School Students' CT Skill Any Significant Differences Between The Mean Values Of CT Skill Levels In A FTFL Environment And In A BL Environment?

Table 21: T-Test Results of the Mean Differences of Coding Training in FTFL Environment and BL Environment

Learning Environment						
	N	Mean	S.D.	t	df	p
FTFL	60	66.16	8.28	-1.709	59	.093
BL	60	67.80	6.16			

As reported by Table 21 the mean difference among CT skills of the students who received coding training in FTFL environment and BL environments is not significant ($p = .093$, $p > .05$).

In the study, descriptive opinions were expressed about the lesson activities and CT skills of secondary school students in FTFL and BL environments of the designed coding training. In line with the findings obtained from the analysis of the student interviews applied within the scope of the study, it is concluded that the coding class activities are designed with text-based programming teaching at the end of the study, despite different learning environments, affect them positively in both learning environments and the students have good lofs of fun. In line with the investigation notes taken by the researcher during the activities and the students' evaluation of the course activities as fun, it can be consired that the contents of the activities are up-to-

date and the students' interest is more especially in drone coding applications. Gupta (2004) states that while choosing a textule programming language, the target audience and the programming tool to be used should be focused on learning. The students stated that they had a pleasant time in FTFL and BL environments without any difficulty, and their results in this direction were confirmed. Course applications prepared accordingly were given according to the level of the target audience. However, in the coding training process, the teacher should be careful and should be alert to the analysis of the target audience.

At the end of the examines, the students stated that they have seen improvement in problem solving, algorithmic thinking and collaboration skills, which are subdimensions of CT skill in FTFL and BL environments. After the thorough analysis, it is showed that the students have made improvement in their creativity and collaboration skills solely in a BL environment. It was stated by Bundy (2007) that problem solving and algorithmic thinking skills are necessary for conceptual understanding in all areas of CT skill. Apart from this, Czerkowski and Lyman (2015) support the idea that problem solving skills will improve with CT skill. Thinking with coding is one of the necessary factors that strengthens dealing with problem skills and enables the students to gain CT skills (Barr, Harrison, & Conery, 2011; Wing, 2006).

CHAPTER V

CONCLUSIONS, DISCUSSION, AND SUGGESTIONS

In this thesis, the effects of FTFL and BL environments on girl and boy students' CT skills during coding training within the framework of the 7th-grade course "ITS" were compared. The results of this comparison provide research-based evidence on the best ways to follow for coding training as it is practiced in different learning environments. We have summarized our results and their place in current thinking under the headings of our initial research questions, and suggest various ways in which this research might be extended.

CT is one of the fundamental skills of the 21st century (Wing, 2006). It is included in the curricula of many countries. Coding training plays an important factor in the acquisition within these CT skills (The Partnership for 21st Century Skills, 2009). Bundy (2007) asserts that computational thinking is essential to conceptual comprehension of any subjects related to problem-solving and algorithmic thinking skills. Several researchers have expressed that having the sub-skills of computational thinking such as creativity, algorithmic thinking, collaboration, critical thinking, and problem-solving both improves children's general educational capacity, and helps them acquire digital-age learning skills in particular (Barr et al., 2011; Brown, 2015; ISTE, 2019; Lye & Koh, 2014).

The activity of writing codes in a language is named coding or programming. The output of this activity is, on the other hand, a program or application. Programming can be defined as a complicated determine complication process that is composed of writing, testing, and debugging, and requires an advanced level of cognitive skills (Kert & Uğraş, 2009; Fesakis and Serafeim, 2009). Coding can help one to gain various skills such as the concretion of abstract concepts, creativity, teamwork, and problem-solving. These skills can make coding an effective tool for acquiring computational thinking skills (ET 2020, 2016). Arduino IDE and PyCharm are software tools that have been widely used in programming training. Robotics, which is becoming more popular parallel to coding, is preferred in also requires programming training as well. This current research has sought an answer to the question of whether a FTFL or BL environment is more effective on coding training.

One of the standards set for students by ISTE in 2016 is being a computational thinker. While programming is compulsory in the primary and secondary education programs of many countries as a method adopted for each student to gain computational thinking skills, it is included by some other countries as a part of their broader educational activities (Grover & Pea, 2013; Mannila et al., 2014). In addition, some researchers emphasize that coding training is a crucial tool for the improvement of CT skills (Bocconi et al., 2016).

Introduction of students in primary education to programming affects their cognitive enhancement positively (Papert, 1980; Clements & Gullo, 1984; Lewis, 2010; Akpınar & Altun, 2014; Atmatzidou & Demetriadis, 2016) and increases their perception level understanding of programming even from this early age (Lewis, 2010; Kalelioğlu & Gülbahar, 2014).

Primary and secondary education programs of our country include computational thinking skills. This is referred to in article 5 of the main focuses of the education program as: “To aim that the students require and enhance their problem-solving and computational thinking skills” (MEB, 2017). Moreover, it is specified as a crucial element for the application of the education program that “Every student in the field of information technologies must be enabled to benefit from technology and acquire computational thinking skills” (MEB, 2017). It is clear that the Turkish education ministry, at least, acknowledges CT as a skill of the 21st century and regards it as one of the core skills.

Visual programming software offers students from young age groups ease of use for the acquisition of CT skills and provides even the inexperienced ones with the opportunity for design and production whilst minimising fatal coding errors (Grover & Pea, 2013). It is one of the tools used in coding training (Armoni et al., 2015; Brennan & Resnick, 2012).

For programming electronic devices of various uses, programming languages composed of specific words and symbols are utilized (Van-Roy & Haridi, 2004). Programs and software tailored for specific purposes emerge by writing commands (TDK, 2017). There are many programming languages for various technologies, which are run by issuing target-specific commands based on the writing rules and idioms peculiar to each language (Van-Roy & Haridi, 2004).

The software that is defined as assistants to program training or simply text-based software tools have become prevalent as they provide easy use, by adding and executing code lines on the programmer’s behalf. With certain software, it is also

possible to produce ‘live’ elements in the coding training, by programming electronic circuit components (such as drones) as part of that training.

Arduino and PyCharm applications, which are text-based software tools, were used in this research. Arduino, which is a free programming application, was developed first in 2005 by five colleagues in Italy as a college project. As a start-up enterprise, the initial manufacturing run of Arduino microcontroller cards was just 200; In the world, Arduino has since become one of the largest microcontroller companies. PyCharm, on the other hand, is an IDE used specially for Python language in computer coding within programming. It was developed by a Czech company named JetBrains. With integrated code analysis, a graphical debugger, and a version control system, it enables Django and Python web development. Furthermore, unmanned aerial vehicles including DJI drones named Tello EDU can be programmed using PyCharm. Arduino and PyCharm have cross-platform compatibility with Windows, MacOS, and Linux are free. Moreover, it provides to programmer open-source software that means source code accessible whenever.

This research aimed to compare and study the effects of FTFL and BL environments on 7th-grade students’ CT skills, based on quantitative methods. A predefined scale was used to measure students CT skills both before training began, and after the completion of their programming projects.

The students who participated in the research were selected by lot from four different classes. During the lot casting, it was made sure the number of male and female students was equal. Fifteen students were selected from each class, which made the total number 60.

Spanned over two school semesters, the research applied the coding training in a different learning environment for each semester. While the coding training took place in a FTFL environment in the first semester, it was practiced in a BL environment in the second semester.

First Arduino coding, then drone coding applications were taught to the same student groups to get the explanation to the question of whether the students being given coding training in a FTFL or BL environment and a greater effect on their CT skills.

The findings of this work were summarized while the results obtained in the light of each subproblem were examined in different sections and addressed as the results of the subproblems. Under the section of the results regarding the subproblems, the similarities and differences between this research and the other studies in the national and international literature were discussed. Lastly, some suggestions regarding the research were listed.

1. Results Regarding the Secondary School Students' CT Skill Levels And Discussion

The CT Scale was applied to the students before the coding training in the FTFL environment took place, as a preliminary test to monitor their development during the research. The scores of the preliminary test given before the training in the FTFL environment yielded the following values: the mean $\bar{x}=66.00$, the range $R=40.91$, the median $M=64.201$, the standard deviation $\sigma=6.962$, the skewness $=.271$, the kurtosis $=-1.306$. The skewness and kurtosis values of the scores resulting from the CT skills test has a normal distribution as it is between ± 1.5 and the total number of the sample group members is above 30 (Tabachnick & Fidell, 2012; Demir, Saatçioğlu, & İmrol, 2016).

The process of coding training taken place in the FTFL environment on the sub-skills of CT resulted in the minimum mean score for problem-solving, and the maximum mean score for collaboration. Based on the total mean scores on the descriptive statistics regarding CT skill levels by the students' **gender**, the following values were calculated out of 100: For the female students, the mean = 65.57, the standard deviation = 7.44, the standard error = 6.54, the minimum = 48.18, and the maximum = 78.18; for the male students, the mean = 66.42, the standard deviation = 6.54, the standard error = 1.19, the minimum = 54.55, and the maximum = 89.09. The coding training in the FTFL environment produced similarly distributed CT levels ($\bar{X}_{\text{female student}} = 65.57$, $\bar{X}_{\text{male student}} = 66.42$).

2. Results regarding the Students' CT Skill Levels in FTFL Environment and Discussion

The research examined whether there were any meaningful differences in the secondary school students' CT skill levels based on gender when the coding training was given in a FTFL environment.

a) Does FTFL Environment Make a Difference in the Students' CT Skill Levels?

The findings viewed that the mean score for the students' CT skill levels was 66.00 before the coding training in the FTFL environment took place while it was 66.16 after the training. Based on the mean scores both in the preliminary and final tests, no important differences were identified in the all subcategories of CT.

- b) Do the Students' CT Skill Levels in a FTFL Environment Differ based on Gender?

Based on the students' scores both in the preliminary and final tests taken before and after the coding training in the FTFL environment, no statistically significant differences were identified between the students based on gender. Also, no important differences were viewed in the all subcategories of CT based on gender ($p > .05$).

3. Results regarding the Students' CT Skill Levels in BL Environment and Discussion

The research examined whether there were no important differences in the secondary school students' CT skill levels based on gender when the coding training was given in a BL environment.

- a) Does a BL Environment Make a Difference in the Students' CT Skill Levels?

The research examined whether there were significant differences in the secondary school students' CT skill levels when the coding training was given in a BL environment. The findings showed that the mean scores for the students' CT skill levels was 66.16 before the coding training in the BL environment took place, while it was 67.80 after the training. Based on the mean scores produced by both the preliminary and final tests given for the students who took the coding training in the BL environment, no important differences were recognize in the CT subcategories of algorithmic thinking, critical thinking, and problem-solving, but significant differences were detected in creativity and collaboration ($P_{\text{creativity}} = .027$, $P_{\text{collaboration}} = .046$, $p < .05$).

b) Do the Students' CT Skill Levels in a BL Environment Differ based on Gender?

Based on the scores both in the preliminary and final tests given the students who took the coding training in the BL environment, no significant differences were identified between their CT skill levels based on gender. No statistically significant differences were evident in the subcategories of creativity, algorithmic thinking, critical thinking, or problem-solving, but there was a significant difference in collaboration based on gender ($P_{\text{collaboration}} = .033, p < .05$).

4. Results Regarding the Difference in the Mean Scores for the Students' CT Skill Levels in FTFL and BL Environments

The research showed no significant differences in the mean scores for the students' CT skill levels between the coding training in the FTFL and BL environments.

In brief, based on the survey undertaken after the coding training in the FTFL environment, no statistically significant differences were identified in the mean scores of the CT subcategories. The finds retrieved after the training in the BL environment showed no important differences in the mean scores of preliminary and final tests for are algorithmic thinking, critical thinking, or problem-solving, but there were significant differences in the mean scores of preliminary and final tests for the subcategories of creativity and collaboration. This corresponds with the findings of Ceylan's study (2019) which revealed that there is a statistically significant relationship between programming and creativity. Nevertheless, although several works have showed that programming training has a positive impact on creative thinking, some studies identified no significant relationships between these

parameters. Ergin (2019), for instance, concluded that coding training has no statistically significant effect on computational thinking, including its subcategory of creativity.

For the BL environment, the students' mean score for creativity was $\bar{X}=13.15$ in the preliminary test whereas it was $\bar{X}=14.90$ in the final test. Furthermore, the students' mean score for collaboration was $\bar{X}=13.78$ in the preliminary test whereas it was $\bar{X}=14.78$ in the final test. This suggests that a BL environment helps the coding training enhance the skills of creativity and collaboration. The existing literature yields both qualitative and quantitative studies that examine the impact of collaborative thinking on coding training. In his study on the effects of individual and collaborative training on the programming software Alice, Kayabaşı (2016) concluded that collaborative training was more successful. Haşlaman (2005) has studied self-regulatory learning strategies in coding training and shown that peer collaboration plays a significant role in this period. Many other studies have revealed that students can improve their creative problem-solving skills if they are provided with proper learning environments and necessary training (Özkök, 2004, 2005; Kandemir, 2006; İslim, 2009; Pannells, 2010; Olgun, 2012; Barak, 2013; Demirci, 2014; Toraman, 2017).

When the coding training in the FTFL environment was analyzed, the study resulted in no significant differences between the sub-parameters of CT skills based on gender. However, when the coding training in the BL environment was analyzed, the study indicated significant differences in the female students for collaboration factor.

According to Gonzales et al. (2017), the male students reached higher scores in CT skills than the female students did. Several studies, however, produced different results than those of this work. Sarıtepeci (2017) stated in his research that although women had a higher CT skill level than men, this difference was insignificant. The studies that were undertaken by Oluk and Korkmaz (2016) and Turan (2019) concluded that gender had no significant effects on CT skills.

On the other side, researched students' CT skills depend on the gender variable and their study indicated that the mean scores for the subcategories, creativity, algorithmic thinking, collaboration, and critical thinking were in the male students' favor whereas problem-solving was in the female students' (Kirit, Dönmez, & Çataltaş, 2018).

When in different coding training, the researching was statistically significant CT skills such as creativity, algorithmic thinking, critical thinking, which are sub-factors of CT skills, and on the total score of scale of the students. However, they also found in same researching, there is no statistical significance in the other sub-factors of CT, collaboration and problem solving sub-factors (Pala & Türker, 2019).

The findings of this study showed that there are insignificant differences between the effects of Arduino educational robotics applications in a FTFL environment on CT skills and those of drone coding applications in a BL environment. Based on the given data, we can conclude that different types of coding training or different methods can be adopted to enhance the students' CT skills. Within this framework, training activities built upon Arduino educational robotics applications in a FTFL environment and drone coding applications in a BL environment cannot be preferred for improving the students' CT skills.

The study revealed that the coding training in both types of learning environments have no significant impact on CT skills. Whilst no statistically important differences were identified in the algorithmic thinking, critical thinking, and dealing with problem subcategories of CT either, a statistically significant difference was detected in the variables of creativity and collaboration. This shows that coding training in a BL environment has an impact on CT skills in terms of creativity and collaboration.

Several works in the field that were conducted on the students from different education levels suggested that Virtual Educational Robotics are proper learning environments for enhancing the students' CT skills (Berland & Wilensky, 2015; Shoop et al., 2016; Witherspoon et al., 2017). While Witherspoon et al. (2017) stated that the learners' CT skills became better as they engaged with more activities related to robotics training in a virtual environment, Berland and Wilensky (2015) argued that both the physical and the virtual robotics learning tools contributed to the students' CT skills.

The international institutions such as CSTA, ISTE, and SLO as well as Ministry of National Education include in their curricula text-based programming training, which is given after block-based programming training. The learning outcomes and contents of these curricula aim to raise the students' CT skills. Therefore, they support the idea that text-based programming languages improve CT skills (Kandemir, 2018). Moreover, Kandemir (2018) advocates that object-based and functional text-based programming languages are the right choice for enhancing the students' CT skills during the abstract stage of their cognitive development. It is observed that the students who participate in programming activities develop in terms of their CT skills (Buss & Gamboa, 2017; Glover, 2011; Kazakoff et al., 2013).

As Delal (2019) also expressed in his work, it is important that Council of Higher Education (CoHE) National Thesis Center started should do more studies on CT skills in the future since it increases the number of programming activities in the Turkish education program, their variety as well as applicability. According to Czerkawski (2013), the CT skills which embody a new set of problem-solving strategies to tackle daily complicated issues should not be limited to the IT education program but should rather be extended to the other areas of the academy.

Finally, in order to solve the problems in the field of Information Technology, it is necessary to examine how the ITS course applied in Turkey has emerged. Changes and regulations have been made many times in information technology education programs at primary, secondary and high school levels in Turkey, especially in the last 10 years. The constant development and change of science and technology in the age in which we live has forced countries to constantly renew their educational programs in order to keep up with this change (Kaptan & Kuşakçı, 2002).

It is a well-known fact that computer technology is also developing much faster than other fields. Given this situation, continuous evaluation of the elements of the ITS course program should be considered as a necessity rather than a requirement. Even if a complete program is prepared in terms of development principles, it is always necessary to obtain the opinions of relevant teachers in order to determine the effectiveness of the curriculum in practice, and to continue to improve it over time, especially as things change so quickly.

The ITS course is compulsory at all levels of teaching in Turkey, and supports the advancement of the programming skills of individuals on behalf of a nation which

wants to capture technological advances and produce new technologies by equipping its citizens with fundamentally sound knowledge (Öztürk & Yılmaz, 2011). The curriculum also requires advanced activities of students at all levels in order to educate who generations who will follow, and who will undertake their own developments. As suggested in the introduction to this dissertation, now more than ever there is an ongoing need to update and develop the curriculum according to the requirements of the era.

5.1 Suggestions for the Further Research

In this chapter, suggestions depend on the findings of this thesis are shown to guide future studies.

- Arduino, drone, coding, and robotics are popular concepts of recent years. Robotics classes are emerging in every level of education from playschools to primary, secondary, and high schools, and they are getting the attention of people from every age group. Despite its popularity among the students, the use of robotics as the course material, however, is an understudied subject, and this should be rectified.
- Arduino and drone technologies can be utilized in the courses of many fields of teaching such as ICT, Mathematics, and Physical Sciences, as they can be adapted according to the course subject. The effects of Arduino and drone on education in different subjects has yet to be fully studied.
- A quantitative research modal is based on this study. The adoption of qualitative methods would also be a valid and rewarding subject of enquiry.

- In this research, 2 weekly course hours were allocated for Arduino and drone applications. The number of these hours can be increased for more effective training.
- Sixty students participated in this research. The sample group can be expanded and students from a wider range of backgrounds might also be analysed.
- Some difficulties with the use and circuit connections of the modules like LCD Screen were faced during the study. More practical, pluggable modules could be utilized for Arduino.
- The 21st-century skills are extremely crucial for students as these skills offer them life-long success in their educations and careers. Therefore, while deciding which learning tools to choose for future works, the training aiming to enhance various skills such as all sub factors of CT, should be considered important.
- The curriculums of the ITS course prepared by the Ministry of National Education in Curriculum Monitoring and Evaluation System will lay the groundwork for achieving these goals. In line with the results obtained in the study, the following recommendations are presented:
 - ITS courses should be made compulsory at all levels.
 - Programming and software information to ITS course should be included in all education levels.
 - ITS courses should be reorganized in a way to focus on programming and software skills.
 - National exams should be held for the evaluation of the ITS course and grades should be given in all grades.

- Teachers who will teach ITS courses and teacher candidates in higher education institutions should be equipped with programming and software skills.



CHAPTER VI

REFERENCES

- Akpınar, Y ., & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *İlköğretim Online*, 13(1), 1-4.
- Altuntuğ, N. (2012). Kuşaktan kuşağa tüketim olgusu ve geleceğin tüketici profili. *Organizasyon ve Yönetim Bilimleri Dergisi*, 4(1) .
- Alvarez, A., & Larranaga, M. (2015). Experiences incorporating Lego Mindstorms robots in the basic programming syllabus: Lessons learned. *Journal of Intelligent & Robotic Systems*, 81 (1), 117-129.
- Arduino. (2019). Arduino About Us. Retrieved from <https://www.arduino.cc/en/Main/AboutUs>
- Atmatzidou, S. & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670.
- Aytekin, A., Sönmez, Ç.F., Yücel, Y. & Kulaözü, İ. (2018). Geleceğe yön veren kodlama bilimi ve kodlama öğrenmede kullanılabilecek bazı yöntemler. *Eurasian Journal of Researches in Social and Economics*, 5(5), 24-41.
- Aziz, Z. & Hossain, M. A. (2010). A comparison of cooperative learning and conventional teaching on students' achievement in secondary mathematics. *Procedia Social and Behavioral Sciences*, 9, 53-62.
- Balanskat, A., & Engelhardt, K. (2015). Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe. In *European Schoolnet*: Belgium

- Balcı, A. (2013). *Sosyal bilimlerde araştırma: Yöntem, teknik ve ilkeler*. Ankara: Pegem Akademi
- Barak, M. (2013). Impacts Of Learning Inventive Problem-Solving Principles: Students' Transition From Systematic Searching To Heuristic Problem Solving. *Instructional Science*, 41(4), 657- 679.
- Barut, S. & Kuzu, A. (2017). The Comparison of Turkey and UK's Information Technologies Curriculum in the Context of Objectives, Acquisition, Activities, Measurement and Evaluation. *Trakya Üniversitesi Eğitim Fakültesi Dergisi*, 7(2), 721-745.
- Barr, D., Harrison, J., & Conery, L. (2011). Computational Thinking: A Digital Age Skill for Everyone. *ISTE Learning and Leading with Technology*, 5191, 20–23.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K12. *ACM Inroads*, 2(1), 48. <https://doi.org/10.1145/1929887.1929905>
- Bayman, P. & Mayer, R. E. (1988). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. *Communications of the ACM*, 26(9), 677-679.
- Baz, F. Ç. (2018). Çocuklar İçin Kodlama Yazılımları Üzerine Karşılaştırmalı Bir İnceleme. *Curr Res Educ*, 4(1), 36-47.
- Bekmezci, M. (2017). Nesillerin Karşılaştırılması ve İş Dünyasının Y Kuşağına Kurumsal Bir Bakış. *Toros University İİSB Sosyal Bilimler Dergisi*, 4(7), 101-111. Retrieved from <https://dergipark.org.tr/tr/pub/iisbf/issue/33561/368916>
- Berland, M. & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology*, 24(5), 628-647.
- Bielawski, L. & Metcalf, D. (2003). *Blended E-Learning*. Amherst: HRD Press.

- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kamyli, P., & Punie, Y. (2016). Exploring the Field of Computational Thinking as a 21st Century Skill. *EDULEARN16 Proceedings, 1*. (Haziran), 4725–4733. <https://doi.org/10.21125/edulearn.2016.2136>
- Brennan, K. & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada.
- Buitrago, F., Casallas, R., H. Hoyos, M., Reyes, A., Restrepo, S. & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research, 87*, 834-860.
- Bundy, A. (2007). Computational Thinking is Pervasive. *Journal of Scientific and Practical Computing, 1*(2), 67–69.
- Buss, A., & Gamboa, R. (2017). Teacher transformations in developing computational thinking: Gaming and robotics use in after-school settings. *In Emerging Research, Practice, and Policy on Computational Thinking* (pp. 189-203). Springer, Cham.
- B y k zt rk, Ő. (2007), *Sosyal Bilimler i in Veri Analizi El Kitabı*, Ankara: Pegem A Yayıncılık
- B y k zt rk, Ő. (2011). *Veri Analizi El Kitabı* (15 b.). Ankara: Pegem Akademi.
- B y k zt rk, Ő., Kılı   akmak, E., Akg n,  . E., Karadeniz, Ő., & Demirel, F. (2012). *Bilimsel Arařtırma Y ntemleri*. Ankara: Pegem Akademi.
- Can, A. (2019). *SPSS ile Bilimsel Arařtırma S recinde Nicel Veri Analizi*. Ankara: PEGEM Akademi.
- Caspersen, M., & Nowack, P. (2013). Computational thinking and practice - A generic approach to computing in Danish high schools. Paper presented at the Fifteenth

Australasian Computing Education Conference (ACE2013), Adelaide, Australia.

Center for Computational Thinking Carnegie Mellon (n.d.). What is Computational Thinking?. Retrieved from <https://www.cs.cmu.edu/~CompThink/>

Ceylan, V. K. (2020). *Senaryo temelli scratch öğretim programının öğrencilerin bilgisayarca düşünme becerilerine, problem çözme ve programlama ünitesi erişilerine etkisi*. (Unpublished doctoral dissertation). Aydın Adnan Menderes University, Eğitim Bilimleri Enstitüsü, Aydın, Turkey.

Chaudhary, V., Agrawal, V., Sureka, P., & Sureka, A. (2016). *An experience report on teaching programming and computational thinking to elementary level children using lego robotics education kit*. In 2016 IEEE Eighth International Conference on Technology for Education (T4E) (pp. 38-41). IEEE, Mumbai, India.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers and Education*, 109, 162-175.

Ching, Y. H., Yang, D., Wang, S., Baek, Y., Swanson, S., & Chittoori, B. (2019). Elementary School Student Development of STEM Attitudes and Perceived Learning in a STEM Integrated Robotics Curriculum. *TechTrends*, 1-12.

Clark, R.T., & Mayer, R.E., (2003). *E-Learning and The Science of Instruction*, San Fransisco: Preiffer.

CodingBK (2017). Geleceğin dili kodlama. Retrieved from <https://www.bahcesehir.k12.tr/tr/images/pdf/codingbksayi1.pdf>

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., N., T., Selby, C., & Woollard, J. (2015). Computational thinking-A guide for teachers. Retrieved from <https://www.computingatschool.org.uk/computationalthinking>

- CSTA, & ISTE. (2011). Computational Thinking in K–12 Education Leadership Toolkit. Computer Science Teacher Association. Retrieved from <http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershipToolkit-SP-vF.pdf>
- Cuny, J., Snyder, L. & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Retrieved from <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Curriculum and Instruction: A 21st Century Skills Implementation Guide (2009). The Partnership for 21st Century Skill. Retrieved from <http://goo.gl/0na2jN>
- Czerkowski, B. (2013). Instructional design for computational thinking. In R. McBride & M. Searson (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2013* (ss. 10-17). Chesapeake, VA: AACE.
- Czerkowski, B. C., & Lyman, E. W. (2015). Exploring Issues About Computational Thinking in Higher Education. *TechTrends*, 59(2), 57–65. <https://doi.org/10.1007/s11528-015-0840-3>
- Çokluk Bökeoğlu, Ö. (2018). Ankara University Açık Ders Malzemeleri - Davranış Bilimlerinde İstatistik. Retrived from <https://acikders.ankara.edu.tr/>
- Çölkesen, R. (2002). *Bilgisayar programlama ve yazılım mühendisliğinde veri yapıları ve algoritmalar*. İstanbul. Papatya Yayıncılık.
- Daniel, A. R. (2004). *How a group of elementary deaf students interact with LEGO-LOGO activities*. (Doctoral dissertation). University of Tennessee, Knoxville, USA.
- De Vaus, D. (1990). *Surveys in social research*. (5 edition). Australia: Routledge.

- Dede, C., Mishra, P., & Voogt, J. (2013). Working Group 6: Advancing computational thinking in 21st century learning. Paper presented at EDUsummIT 2013, Faculty of Behavioural, Management and Social Sciences Educational Science.
- Delal, H. (2019). *Developing middle school students' computational thinking skills using unplugged computing activities* (Master's thesis). Boğaziçi University, İstanbul, Turkey.
- Demir, E., Saatçioğlu, Ö., & İmrol, F. (2016). Uluslararası Dergilerde Yayımlanan Eğitim Araştırmalarının Normallik Varsayımları Açısından İncelenmesi. *Current Research in Education*, 2(3), 130-148.
- Demirci, N. (2014). *Sistematik Yaratıcı Problem Çözme Etkinliklerinin İlköğretim 7. Sınıf Öğrencilerinin Işık Konusundaki Kuramsal, DeneySEL ve Günlük Yaşam Problemlerini Çözmelerine Etkisi*. (Unpublished doctoral dissertation). Marmara University, Eğitim Bilimleri Enstitüsü, İstanbul, Turkey.
- 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.
- Demirkol, Z. (2016). *Çocuklar için kodlama*, İstanbul: Pusula Yayıncılık.
- Deng, Z., Huang, W., & Dong, R. (2009). *Discussion of ability cultivation of computational thinking in course teaching*. In Education Technology and Computer, 2009. ICETC'09. International Conference on (pp. 197-200). IEEE, Singapore.
- Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, 46, 277-296. <http://dx.doi.org/10.1080/15391523.2014.888272>

- Denning, P. J. (2009). The Profession of It Beyond Computational Thinking. *Communications of The ACM*, 52(6), 28-30.
- Dinç, F. N. (2019). Geleceğe gidiyoruz kodlama bilmeyen kalmasın. Retrieved from <http://www.zraporu.com/egitim/gelecege-gidiyoruz-kodlama-bilmeyen-kalmasin/>
- Driscoll, M. (2002). Blended learning: Let's get beyond the hype. Learning and Training Innovations Newslane. Retrieved from http://www.ibm.com/services/pdf/blended_learning.pdf
- EARGED (2011). MEB 21. yüzyıl öğrenci profili. Retrieved from http://www.meb.gov.tr/earged/earged/21.%20yy_og_pro.pdf
- Earl, Babbie (2004). *The Practice of Social Research* (1st ed.). Thomson, Wadsworth.
- Eğitimpedia. (2018). Bir Kuşağı Anlamak: X Kuşağı Kimdir?, Retrieved from <https://www.egitimpedia.com/bir-kusagi-anlamak-x-kusagi-kimdir>
- Ellis, A. (2001). *Student-Centered Collaborative Learning via Face-To-Face And Asynchronous Online Communication: What's The Difference?*. Meeting At The Crossroads, Proceedings Of The Annual Conference Of The Australian Society For Computers In Learning İn Tertiary Education (Ascolite 2001), Melbourne, Australia, December 9–12.
- Erden, M., & Akman, Y. (2004). Gelişim ve öğrenme. (13. Baskı) Ankara: Arkadaş Yayınevi.
- Ergin, H. (2019). *Programlama dersinde proje kullanımının öğrencilerin bilgisayarca düşünme becerilerine ve programlama öz yeterlilik inancına etkisi*. (Unpublished master's thesis). Ege University, Fen Bilimleri Enstitüsü, İzmir, Turkey.

- Erol, O. (2015). *Scratch ile Programlama Öğretiminin Bilişim Teknolojileri Öğretmen Adaylarının Motivasyon ve Başarılarına Etkisi*. (Doctoral dissertation). Anadolu University, Eğitim Bilimleri Enstitüsü, Eskişehir, Turkey.
- Ersoy, H., Madran, R. O., & Gülbahar, Y. (2011). *Programlama dilleri öğretimine bir model önerisi: robot programlama*. Akademik Bilişim 2011 Konferansı, Malatya, Turkey.
- Eryılmaz, S. (2003). *Algoritma tasarlama ve programlamaya giriş*. Ankara: Detay Yayıncılık.
- Faber, H.H., Wierdsma, M.D.M., Doornsbos, R.P., Van der Ver, J.S. & De Kevin, V.(2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network*, 12, 13-24.
- Fesakis, G. & Serafeim, K. (2009). *Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education*. In proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE-2009), 2, 258-262. Newyork. NY. USA .
- Finn, A. & Bucci, M., (2004). A case study approach to blended learning, Los Angeles: Centra Software.
- Futschek, G. (2006, November). Algorithmic thinking: the key for understanding computer science. In International conference on informatics in secondary schools-evolution and perspectives (159-168). Springer, Berlin, Heidelberg.
- Gardiner, B. (2014). Adding coding to the curriculum. Retrieved from <https://www.nytimes.com/2014/03/24/world/europe/adding-coding-to-the-curriculum.html>

- Genç, Z. & Karakuş, S. (2011). *Tasarımla öğrenme: eğitsel bilgisayar oyunları tasarımında Scratch kullanımı*. 5th International Computer and Instructional Technologies Symposium. Elazığ.
- Gomes, A., & Mendes, A. J. (2007, September). *Learning to program-difficulties and solutions*. In International Conference on Engineering Education–ICEE (Vol. 2007).
- González, Y. A., & G. V. M. Repiso, A. (2017, October). *Development of computational thinking and collaborative learning in kindergarten using programmable educational robots: a teacher training experience*. Paper presented at the Fifth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'17), New York, USA.
- Grandell, L., Peltomäki, M., Back, R. J., & Salakoski, T. (2006). *Why complicate things?:introducing programming in high school using Python*. In Proceedings of the 8th Australasian Conference on Computing Education (pp. 71-80).
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Gupta, D. (2004). What is a good first programming language? *Crossroads*, 10(4), 7– 7. <https://doi.org/10.1145/1027313.1027320>
- Güneş, F. (2012). Bologna Süreci ile Yükseköğretimde ÖngörülenBeceri ve Yetkinlikler. *Yükseköğretim ve Bilim Dergisi*, 2(1), 1-9.
- Haslaman, T. (2005). *Programlama dersi ile ilgili özdüzenleyici öğrenme stratejileri ile başarı arasındaki ilişkilerin incelenmesi: bir yapısal eşitlik modeli*. (Unpublished master's thesis). Hacettepe University,Fen Bilimleri Enstitüsü, Ankara.
- Hemmendinger, D. (2010). A Plea for Modesty. *ACM Inroad*, 1(2), 4-7.

- Herger, L. M., & Bodarky, M. (2015). *Engaging students with open source technologies and Arduino*, 2015 IEEE Integrated STEM Education Conference, 27-32.
- Horton, W. (2000). Designing Web-Based Training. How to teach anyone anything anywhere anytime. William Horton Consultign, Inc. USA. Retrieved from <http://www.irrodl.org/index.php/irrodl/article/download/321/531>
- Hubwieser, P., Giannakos, M., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., Pal, Y., Jacková, J. & Jasute, E. (2015). *A global snapshot of computer science education in k-12 schools*. Paper presented at the 2015 ITiCSE Working Group Reports (ITICSE-WGR '15), Vilnius, Lithuania, <http://doi.org/10.1145/2858796.2858799>.
- House, R. (2002). Clocking In Column. *The Spokesman-Review*. 61(2), 179-211.
- Information Technologies Education Association (2013). *History of ITS Courses*. Retrieved from <https://bte.org.tr/bt-derslerinin-tarihcesi/>
- ISTE. (2015). Computational thinking leadership toolkit first edition. Retrieved from <http://www.iste.org/docs/ct-documents/ct-leadershiptoolkit.pdf?sfvrsn=4>
- ISTE. (2019). ISTE standards for educators: Computational thinking competencies. Retrieved from <http://www.iste.org/standards>
- İslim, F. Ö. (2018). *Bilgi ve İletişim Teknolojileri Dersinin Scamper (Yönlendirilmiş Beyin Fırtınası) Tekniğine Göre İşlenmesinin Öğrencilerin Yaratıcı Problem Çözme Becerilerine ve Akademik Başarılarına Etkisi*. (Unpublished master's thesis). Gazi University, Eğitim Bilimleri Enstitüsü, Ankara, Turkey.
- Jang, Y., Lee, W., & Kim, J. (2015). Assessing the usefulness of object-based programming education using Arduino, *Indian Journal of Science and Technology*, 8(S1), 89-96.
- Jenkins, C. (2017). Classroom Talk and Computational Thinking. *International Journal of Computer Science Education in Schools*, 1(4), 1-11.

JetBrains. (2019). PyCharm, Community Edition. Retrieved from <https://www.jetbrains.com/pycharm/>

Kabilan, M. K. (2000). Creative and critical thinking in language classrooms. *The Internet TESL Journal*, 6(6), 1-3.

Kafai, Y. & Burke, Q. (2013, March). *The social turn in K-12 programming: Moving from computational thinking to computational participation*. Paper presented at the 44th ACM Technical Symposium on Computer Science Education (SIGCSE 2013), Denver, CO, USA.

Kalelioğlu, F., Gülbahar, Y., Akçay, S., & Doğan, D. (2014). *Curriculum integration ideas for improving the computational thinking skills of learners through programming via scratch*. In Local proceedings of the 7th international conference on informatics in schools: Situation, evolution and perspectives (pp. 101-112).

Kandemir, A. M. (2006). *OFMA Matematik Eğitimi Öğretmen Adaylarının Yaratıcılık Eğitimi Hakkındaki Görüşleri Ve Yaratıcı Problem Çözme Becerilerinin İncelenmesi*. (Unpublished master's thesis). Balıkesir University Fen Bilimleri Enstitüsü, Balıkesir, Turkey.

Kandemir, C.M. (2018). *Metin Tabanlı Programlama*. Y. Gülbahar, H. Karal (Ed.), Kuramdan Uygulamaya Programlama Öğretimi (ss. 299-336). Ankara: PEGEM Akademi.

Kaptan, F. & Kuşakcı, F. (2002). Fen öğretiminde beyin fırtınası tekniğinin öğrenci yaratıcılığına etkisi. *V. Ulusal Fen Bilimleri ve Matematik Eğitimi Kongresi Bildiriler Sitesi (s197-202)*, ODTÜ, Ankara.

Karaahmetoğlu, K. (2019). *Proje tabanlı arduino eğitsel robot uygulamalarının öğrencilerin bilgisayarca düşünme becerileri ve temel stem beceri düzeyleri algılarına etkisi* (Unpublished Master's thesis). Amasya University, Amasya, Turkey.

- Karabak, D. & Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 163-169.
- Karadeniz, Ş. (2017). Geleceğin dili kodlama. Retrieved from <https://www.bahcesehir.k12.tr/tr/images/pdf/codingbksayi1.pdf>
- Kaucic, B., & Asic, T. (2011). *Improving introductory programming with Scratch?*. In MIPRO, 2011 Proceedings of the 34th International Convention (pp. 1095-1100).
- Kayabaşı, E. (2016). *Öğretmen adaylarının alice deneyimi: 3b ortamda programlama*. (Unpublished master's thesis). Uludağ University/Eğitim Bilimleri Enstitüsü, Bursa, Turkey.
- Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 1-11.
- Kert, S. B., & Uğraş, T. (2009). *Programlama eğitiminde sadelik ve eğlence: Scratch örneği*. In The First International Congress of Educational Research, Çanakkale, Turkey.
- Kılıç, F. N., Korkmaz, Ö., Çakır, R., & Uğur Erdoğan, F. (2019). Meslek lisesi bilişim teknolojileri öğrencilerinin kodlama yeterlilikleri, STEM ve bilgisayarca düşünme becerilerine yönelik algıları. *Gazi Eğitim Bilimleri Dergisi*, 5(Özel Sayı), 196-218.
- Kirmit, Ş., Dönmez, İ., & Çataltaş, E. H. (2018). Üstün Yetenekli Öğrencilerin Bilgisayarca Düşünme Becerilerinin İncelenmesi. *Bilim, Teknoloji, Mühendislik, Matematik ve Sanat Eğitimi Dergisi(Journal Of STEAM Education)*, 2(1), 17-26.
- Korkmaz, M. (2020). Python programlama dili ile termoelastik gerilme analizi. (Master's thesis), Gazi University, Fen Bilimleri Enstitüsü. Ankara.

- Korkmaz, Ö., Çakır, R., & Özden, M. (2017). A Validity and reliability of the computational thinking scales (CTS). *Computers in Human Behavior*, 1-43. <http://doi.org/10.1016/j.chb.2017.01.005>
- Korkmaz, Ö., Çakır, R., & Özden, M. Y. (2015). Bilgisayarca düşünme beceri düzeyleri ölçeğinin (Bdbd) ortaokul düzeyine uyarlanması. *Gazi Eğitim Bilimleri Dergisi*, 1(2), 67-86.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., M. Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2, 32-37. <http://doi.org/10.1145/1929887.1929902>
- Leonard, J., Buss, A., Gamboa, R. et al. (2016). Using Robotics and Game Design to Enhance Children's Self-Efficacy, STEM Attitudes, and Computational Thinking Skills. *J Sci Educ Technol* 25, 860–876.
- Lewis, C. M. (2010). *How programming environment shapes perception. learning and goals: logo vs. scratch*. In Proceedings of the 41st ACM technical symposium on computer science education, 41, 346-350. ACM.
- Lopez, J.M., R. Gonzalez, M., & V. Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129-141.
- Lu, J., & Fletcher, G. (2009). Thinking about Computational Thinking. *ACM Sigcse Bulletin*, 41(1). 260-264.
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. & Settle, A. (2014). *Computational thinking in k-9 education*. Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference (ss. 1-29). New York. USA: ACM.

- Martyn, M. (2003). The Online Hybrid model: Good Practice. *Educause Quaterly*. *Educause Quarterly*, 1, 18-23.
- Mason, M. (2009). *Critical thinking and learning*. Hoboken. WileyBlackwell.
- Mercan, M., Filiz, A., Göçer, İ., & Özsoy, N. (2009). Bilgisayar destekli eğitim ve bilgisayar destekli öğretimin dünyada ve Türkiye’de uygulamaları. *Akademik Bilişim’09-XI. Akademik Bilişim Konferansı Bildirileri*, Şanlıurfa: Harran University.
- Michaelsen, L. K., Davidson, N., & Major, C. H. (2014). Team-based learning practices and principles in comparison with cooperative learning and problem-based learning. *Journal on Excellence in College Teaching*, 25(4), 57-84.
- Milli Eğitim Bakanlığı (MEB). (2012). *Ortaokul Ve İmam Hatip Ortaokulu Bilişim Teknolojileri Ve Yazılım Dersi (5, 6, 7 ve 8. Sınıflar) Öğretim Programı*. Retrieved from <http://ttkb.meb.gov.tr/program2.aspx?islem=1&kno=196>
- Milli Eğitim Bakanlığı (MEB). (2014). 19. *Millî eğitim şûrası tavsiye kararları*. Retrived from <http://www.meb.gov.tr/19-mill-egitim-srasi-sona-erdi/haber/7594/tr>
- Milli Eğitim Bakanlığı (MEB). (2015a). *Millî Eğitim Bakanlığı 2015–2019 Stratejik Planı*. Retrieved from http://sgb.meb.gov.tr/meb_iys_dosyalar/2015_09/10052958_10.09.2015sp17.15imzasz.pdf
- Milli Eğitim Bakanlığı (MEB). (2015b). *2015 Yılı Temel Eğitim Yapım Programı Hazırlama Esasları*. Retrieved from http://iedb.meb.gov.tr/meb_iys_dosyalar/2015_02/17052429_2015yltemeleitimyapmprogramhazrlamaesaslar.pdf
- Milli Eğitim Bakanlığı (MEB). (2017). Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5 ve 6. Sınıflar). Retrieved

from https://can.meb.gov.tr/meb_iys_dosyalar/2017_08/24134656_BiliYim_Teknolojileri_ve_YazYIYm_Dersi_YYretim_ProgramY.pdf

Milli Eğitim Bakanlığı (MEB). (2018). Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5 ve 6. Sınıflar). Retrieved from <https://mufredat.meb.gov.tr/Dosyalar/2018124103559587-Bili%C5%9Fim%20Teknolojileri%20ve%20Yaz%C4%B1%C4%B1m%205-6.%20S%C4%B1n%C4%B1flar.pdf>

Milli Eğitim Bakanlığı (MEB). (2019). Bilişimle Üretim Öğretim Programı (4, 5 ve 7. Sınıflar). Retrieved from https://yegitek.meb.gov.tr/meb_iys_dosyalar/2019_10/21171539_BilisimleuretimPilotUygulamasidegerlendirmeRaporu2019.pdf

Minsky, C. (2016). Best universities in the world for learning to code. Retrieved from <https://www.timeshighereducation.com/student/news/best-universities-world-learning-code>

Mojica, K. D. (2010). Ordered effects of technology education units on higher-order critical thinking skills of middle school students (Order No. 3468158). Retrieved from ProQuest Central; ProQuest Dissertations & Theses Global. (885145844). <https://www.proquest.com/docview/885145844>

National Academies of Science. (2010). Report of a workshop on the scope and nature of computational thinking. Washington DC: National Academies Press.

Nedzad, M., & Yasmeen, H., (2001). Challenges in teaching Java technology. *Challenges Informing Clients A Transdiscipl Approach*, 365–371.

Noble, J. (2013). *Building a LEGO-based Robotics Platform for a 3 rd Grade Classroom*. (Doctoral dissertation). Tufts University, Boston, USA.

- Noh, J. & Lee, J.(2019). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development, 1*, 1-22.
- NRC. (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking*. Washington D.C.: The National Academies.
- Olgun, E. (2012). *A Non-Formal Learning Program For The Contribution Of Creative Problem Solving Skills: A Case Study*. (Unpublished master's thesis), Bilkent University, Ankara, Turkey.
- Oluk, A., & Korkmaz, O. (2016). Comparing students' Scratch skills with their Computational thinking skills in terms of different variables. *Online Submission, 8*(11), 1-7.
- Orhan, D., Kurt, A. A., Ozan, Ş., Vural, S. S., & Türkan, F. (2014). Ulusal eğitim teknolojisi standartlarına genel bir bakış. *Karaelmas Eğitim Bilimleri Dergisi, 2*(1).
- Orhan, D., Filiz, O., & Kurt, A. A. (2014). Bilişim teknolojileri ve yazılım dersi öğretmen adaylarının esnek programa yönelik içerik önerileri. *Eğitimde Kuram ve Uygulama, 10*(5), 1190-1205.
- Osguthorpe, R. T., & Graham, C. R. (2003). Blended Learning Systems: Definitions And Directions. *Quarterly Review Of Distance Education, 4*(3), 227–234.
- Özden, M. Y. (2015). Computational thinking. Retrieved from <http://myozden.blogspot.com.tr/2015/06/computational-thinking-bilgisayarca.html>
- Özden, Y. (2014). *Öğrenme ve öğretme*. Ankara: Pegem Akademi.

- Öztürk, H. T., & Yılmaz, B. (2011). Bilişim Teknolojileri ve Yazılım Dersi'nin seçmeli statüsünün dersin pedagojik değerine yansımalarının öğretmen bakış açısı ile değerlendirilmesi. *Ege Eğitim Dergisi*, 12(2), 63-82.
- Pala, F. K., & Mıhçı Türker, P. (2019). *The effects of different programming trainings on the computational thinking skills*. Interactive Learning Environments, 1-11. <https://doi.org/10.1080/10494820.2019.1635495>
- Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. *International Journal of Mobile Learning and Organisation*, 10(3), 187-202. <https://doi.org/10.1504/IJMLLO.2016.077867>.
- Papert, S. (1993). *The Children's Machine*. New York: Basic Books.
- Pei, C., Weintrop, D., & Wilensky, U. (2018). Cultivating computational thinking practices and mathematical habits of mind in lattice land. *Mathematical Thinking and Learning*, 20(1), 75-89.
- Peñalvo, F., & Mendes, A. (2017). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411. <https://doi.org/10.1016/j.chb.2017.12.005>.
- Pfeiffer, J. (1962). *The thinking machine*. Philadelphia, PA: Lippincott, USA.
- Rossett, A. (2002). *The ASTD E-Learning Handbook*; New York: McGraw-Hill.
- Ru, Z. (2019). *Cultivation of computational thinking in college computer basic teaching*. Journal of Physics: Conference Series, 1237, 1-5.
- Rubio, M. A., Hierro, C. M., & Pablo, A. P. D. M. (2013). *July, Using arduino to enhance computer programming courses in science and engineering*. Proceedings of EDULEARN13 conference, IATED Barcelona, Spain, 1-3.

Sanford, J. F., & Naidu, J. T. (2016). Computational Thinking Concepts for Grade School. *Contemporary Issues in Education Research*, 9(1) 23-32.

Saritepeci, M. (2017). *Ortaöğretim düzeyinde bilgi-işlemsel düşünme becerisinin çeşitli değişkenler açısından incelenmesi*. 5th International Instructional Technologies & Teacher Education Symposium Proceedings Kitabı. İzmir: Dokuz Eylül University, 218-226.

Saritepeci, M., & Durak, H. (2017). Analyzing the effect of block and robotic coding activities on computational thinking in programming education. *Educational research and practice*, 49, 490-501.

Sayan, H. (2016). Okul öncesi eğitimde teknoloji kullanımı. *21. Yüzyılda Eğitim ve Toplum*, 5(13).

Saygıner Ş. & Tüzün H. (2017). *İlköğretim düzeyinde programlama eğitimi: yurt dışı ve yurt içi perspektifinden bir bakış*. 19. Akademik Bilişim Konferansı'nda sunulmuş bildiri, Aksaray University, Aksaray.

Saygıner, Ş. (2017). *Blok tabanlı görsel ve metin tabanlı programlama öğretimlerinin erişil, mantıksal düşünme ve motivasyona etkileri*. (Master's thesis), Hacettepe University, Eğitim Bilimleri Enstitüsü, Ankara, Turkey.

ScratchJr (2017). Information. Retrieved from <https://scratchjr.org/about/info>

Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). USA, Pearson Education.

Sencer, M. & Sencer, Y. (1978). *Toplumsal araştırmalarda yöntem bilim*. Ankara: Türkiye ve Orta Doğu Amme İdaresi Enstitüsü Yayınları.

- Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., R. May, C. & Wildeman, B. (2012, July). *Infusing computational thinking into the middle-and high-school curriculum*. Paper presented at the 17th ACM annual conference on innovation and technology in computer science education, New York, ABD.
- Severance, C. (2015). Guido van Rossum: the early years of Python. *Computer*, 48(2), 7–9.
- Shoop, R., Flot, J., Friez, T., Schunn, C. & Witherspoon, E. (2016). Can computational thinking practices be taught in robotics classrooms. Retrieved from https://www.cmu.edu/roboticsacademy/PDFs/Research/CTP_RoboticsClassrooms.pdf
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158.
- Sinap, V. (2007). *Programlama Eğitiminde Probleme Dayalı Öğrenmeye Yönelik Arduino Etkinliklerinin Kullanılması: Bir Eylem Araştırması*. (Master's thesis). Süleyman Demirel University, Eğitim Bilimleri Enstitüsü, Isparta, Turkey.
- Singh, H. (2003). Building effective blended learning programs. *Educational Technology* 43, 51-54.
- Singh, H. & Reed, C. (2001). *A White Paper: Achieving success with blended learning*. American Society for Training and Development[White Paper]. Centra Software. Retrieved from <https://maken.wikiwijs.nl/userfiles/f7d0e4f0bd466199841ede3eea221261.pdf>
- Somyürek, S. (2014). Öğretim sürecinde z kuşağının dikkatini çekme: artırılmış gerçeklik. *Eğitim Teknolojisi Kuram ve Uygulama*, 4(1), 63-80.
- Sternberg, R. J. (2003). Creative thinking in the classroom. *Scandinavian Journal of Educational Research*, 47(3), 325-338.

- Sünbül, M. (2002). *Yaratıcı Düşünme Yaklaşımı*. M. Sünbül (Ed.), Eğitime Yeni Bakışlar içinde, Micro Basım. 130-180.
- Şimşek, E. (2018). *Programlama öğretiminde robotik ve scratch uygulamalarının öğrencilerin bilgi işlemsel düşünme becerileri ve akademik başarılarına etkisi*. (Master's thesis), 19 Mayıs University, Eğitim Bilimleri Enstitüsü, Samsun, Turkey.
- Tabachnick, B. G., & Fidell, L. S. (2012). *Using Multivariate Statistics* (6 b.). Boston, MA: Pearson.
- TACCLE 3 & European Union Erasmus+. (2016). TACCLE 3, O5: An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers. Retrieved from <https://gredos.usal.es/bitstream/handle/10366/131863/TACCLE3O5Literaturereview%20-%20final.pdf?sequence=1&isAllowed=y>
- Toraman, Ç. (2017). *Lise Öğrencilerinin Yaratıcı Çözü Elde Etme Düzeyleri: İstatistiksel Söylem Analizi*. (Unpublished doctoral dissertation). Eskişehir OsmanGazi University Eğitim Bilimleri Enstitüsü, Eskişehir, Turkey.
- Turan, B. (2019). *Ortaokul Öğrencilerinin Geliştirdiği Oyun ve Robot Projelerinde Probleme Dayalı Öğrenmenin Problem Çözme ve Bilgi İşlemsel Düşünme Becerilerine Etkisi*. (Unpublished master's thesis) , Van Yüzüncü Yıl University Eğitim Bilimleri Enstitüsü, Van, Turkey.
- Türk Dil Kurumu (TDK). (2017). Türk Dil Kurumu. Retrieved from <http://www.tdk.gov.tr>
- Van-Roy, P. & Haridi, S. (2004). *Concepts, techniques and models of computer programming*. London. England: MIT Press.

- Von Rekowski, T., Weibert, A., Rode, J., Booker, J., Marshall, A., Aal, K., & Mimoni, H. (2015, September). *From computational thinking to computational making*. Paper presented at the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Osaka, Japan. <https://doi.org/10.1145/2800835.2800926>
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Educ Inf Techno*, 20, 715-728.
- Vural, B. (2005). *Öğrenci Merkezli Eğitim ve Çoklu Zeka*. B. Vural, Eleştirel Düşünme Yetkinliği Kazanmak içinde, Hayat Yayıncılık, 103-151.
- Webb, H., & Rosson, M. B. (2013). *Using scaffolded examples to teach computational thinking concepts*. Paper presented at the 44th ACM technical symposium on Computer science education, Denver, Colorado, USA.
- Weinberg, A.E. (2013). *Computational Thinking: An Investigation Of The Existing Scholarship And Research*. (Doctoral dissertation), Colorado State University, School of Education, Colorado.
- Wing, J. (2006). Computational Thinking. *Communications of The ACM*, 49(3), 33-35.
- Wing, J. (2010). Computational Thinking: What and Why? Center for Computational Thinking Carnegie Mellon. Retrieved from <https://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf>
- Wing, J. M. (2014). *Computational thinking benefits society*. 40th Anniversary Blog of Social Issues in Computing, 2014.
- Witherspoon, E. & Schunn, C. D. (2019). Teachers' goals predict computational thinking gains in robotics. *Information and Learning Sciences*. <https://doi.org/10.1108/ILS-05-2018-0035>.

- Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C. & Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education (TOCE)*, 18(1), 4-24.
- Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (2011). Computational thinking and expository writing in the middle school. *ACM Transactions on Computing Education*, 11(2), 1-22. <http://doi.acm.org/10.1145/1993069.1993073>
- Xin, Z., Lin, C., Zhang, L., & Yan, R. (2007). The performance of Chinese primary school students on realistic arithmetic word problems. *Educational Psychology in Practice*, 23, 145–159.
- Yalçın, B. (2020). *Harmanlanmış Öğrenme Ortamında 7.Sınıf Öğrencilerinin Öğrenme Düzeylerinin Araştırılması*. (Master Thesis) . Manisa Celal Bayar University, Fen Bilimleri Enstitüsü, Manisa, Turkey.
- Yecan, E., Özçınar, H. & Tanyeri, T. (2017). Bilişim teknolojileri öğretmenlerinin görsel programlama öğretimi deneyimleri. *Elementary Education Online*, 16(1), 377-393.
- Yenilmez, K. & Balbağ, M. Z. (2016). Fen bilgisi ve ilköğretim matematik öğretmeni adaylarının Stem'e yönelik tutumları. *Journal of Research in Education and Teaching*, 5(4), 301- 307.
- Yıldız, N., Akbulut Ö. & Bircan, H. (2002). *İstatistiğe Giriş*. İstanbul: Aktif Yayınevi.
- Yılmaz, E. (2014). Arduino nedir, özellikleri nelerdir ve nerelerde kullanılır. Retrieved from <https://www.hataverdi.com/arduino-nedir-ozellikleri-nelerdir-ve-nelerelerde-kullanilir>
- Yılmaz, M. (2007). Sınıf öğretmeni yetiştirmede teknoloji eğitimi. *Gazi Eğitim Fakültesi Dergisi*, 27(1), 155-167.

- Yiğit, M. F. (2016). *Görsel programlama ortamı ile öğretimin öğrencilerin bilgisayar programlamayı öğrenmesine ve programlamaya karşı tutumlarına etkisinin incelenmesi*. (Master Thesis), Ondokuz Mayıs University, Eğitim Bilimleri Enstitüsü, İzmir, Turkey.
- Yolcu, V. (2018). *Programlama eğitiminde robotik kullanımının akademik başarı, bilişimsel düşünme becerisi ve öğrenme transferine etkisi*. (Master Thesis) . Süleyman Demirel University, Eğitim Bilimleri Enstitüsü, Isparta, Turkey.
- Yuen, T., Boecking, M., Stone, J., Tiger, E. P., Gomez, A., Guillen, A., & Arreguin, A. (2014). Group tasks, activities, dynamics, and interactions in collaborative robotics projects with elementary and middle school children. *Journal of STEM Education*, 15(1).
- Yüzbaşıoğlu, S. (2012). Kuşaklar X, Y, Z Diye Ayırıştı Pazarlamacıların Kafası Karıştı. Retrieved from <https://www.dunya.com/ozel-dosya/isim-iletisim/kusaklar-x-y-z-diye-ayristi-pazarlamacilarin-kafasi-karisti-haberi-171279>