

## Ortaokul Öğrencilerinin Bilgi-İşlemsel Düşünme Becerisi Öz Yeterlilik Algıları: Yeni Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programının Etkinliğinin Bir İncelemesi

### Computational Thinking Skills Self-Efficacy Perceptions in Secondary Education: A Review of The Effectiveness of The New Information Technology and Software Curriculum

Emin İBİLİ<sup>1</sup>, Mustafa Serkan GÜNBATAR<sup>2</sup>

**ÖZ:** Bu çalışmada ortaokul öğrencilerinin bilgi-işlemsel düşünme becerisine yönelik öz yeterlilik algıları incelenmiştir. Ayrıca bilgi-işlemsel düşünme becerisine yönelik öz yeterlilik ölçümleri farklı değişkenlere göre karşılaştırılmıştır. Çalışma ilişkisel tarama modelinde yürütülmüştür. Araştırma verileri 2018-2019 eğitim-öğretim yılında Van ili merkez ilçelerinde öğrenim gören dört farklı ortaokuldaki 332 öğrenciden elde edilmiştir. Veriler 2018 yılında güncellenen Bilişim Teknolojileri ve Yazılım dersi öğretim programının ilk kez uygulanmasının ardından toplanmıştır. Ölçme aracı olarak ise Gülbahar, Kert ve Kalelioğlu (2018) tarafından ortaokul öğrencileri için geliştirilen Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlilik Algısı (BİDBÖA) ölçeği kullanılmıştır. Elde edilen verilerin analizinde ilişkisiz örneklem t testi, ilişkisiz örneklem için tek faktörlü varyans analizi ve Bonferroni çoklu karşılaştırma testleri kullanılmıştır. Elde edilen bulgulara göre öğrenciler ağırlıklı olarak orta düzey öz yeterlilik algısına sahiptirler. Bunu %39'luk bir oranla yüksek öz yeterlilik algısına sahip olanlar takip etmektedir. Çok az bir kısmı ise düşük öz yeterlilik algısına sahiptir. Öğrenim görülen sınıf değişkeni açısından öğrenciler hem BİDBÖA ölçeği toplam puanına göre hem de alt faktörlere göre farklılaşmaktadırlar. Blok tabanlı programlama aracı kullanma süresi açısından toplam ölçek puanı, Algoritma tasarlama ve Temel programlama alt boyutlarına göre farklılaşmaktadırlar. Cinsiyet değişkeni açısından farklılaşma yoktur.

**Anahtar sözcükler:** Bilgi-işlemsel düşünme, öz-yeterlilik, ortaokul öğrencileri..

**ABSTRACT:** In this research self-efficacy perceptions of secondary school students' about computational thinking skills were investigated. In addition, self-efficacy perception measures for computational thinking skills were compared according to different variables. Descriptive survey model was employed. The data was collected from 232 students in four secondary schools in the central districts of Van province during the 2018-2019 academic year. Data were collected after the first implementation of the Information Technology and Software course, which was updated in 2018. The Self-Efficacy Perception Scale for Computational Thinking Skill (CTSSP) developed by Gülbahar, Kert and Kalelioğlu (2018) was used. The data were analyzed by using t-test for unrelated samples, single-factor analysis of variance for unrelated samples, and Bonferroni multiple comparison tests. The findings show that students usually have moderate self-efficacy, and 39% have high self-efficacy, while only a few have low self-efficacy. In terms of grade variable, the students differ according to the total score of the CTSSP and its sub-scales. While the total scale score in terms of block-based programming tool usage time showed significant difference according to algorithm design and basic programming, no difference was observed in terms of gender variable.

**Keywords:** Computational thinking, self-efficacy, secondary school students.

#### Bu makaleye atf vermek için:

İbili, E. ve Günbatır, M. S (2020). Ortaokul öğrencilerinin bilgi-işlemsel düşünme becerisi öz yeterlilik algıları: Yeni bilişim teknolojileri ve yazılım dersi öğretim programının etkinliğinin bir incelemesi. *Trakya Eğitim Dergisi*, 10(2), 303-316

#### Cite this article as:

İbili, E.,& Günbatır, M. S (2020). Computational thinking skills self-efficacy perceptions in secondary education: A review of the effectiveness of the new information technology and software curriculum. *Trakya Journal of Education*, 10(2), 303-316

## UZUN ÖZET

### Giriş

Bilgisayarları öğretim aracı olarak ilkökul ve ortaokul öğretim müfredatına entegre etmek hem öğrenciler ve öğretmenler hem de eğitim sistemi açısından açılması gereken bazı zorluklar getirebilir

<sup>1</sup> Dr. Öğr. Üyesi, Afyonkarahisar Sağlık Bilimleri Üniversitesi, Sağlık Bilimleri Fakültesi, Sağlık Yönetimi Bölümü, eminibili@aku.edu.tr, ORCID: 0000-0002-6186-3710

<sup>2</sup> Dr. Öğr. Üyesi, Van Yüzüncü Yıl Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü, mustafaserkanyu.edu.tr, ORCID: 0000-0003-0485-3038

(Peng et al., 2014). 2018-2019 öğretim yılında uygulanmaya başlanan Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı'nda önceki yıllara göre değişiklikler yapılmıştır (Mercimek ve İlic, 2017). Bu sebeple bazı programlama ve bilgisayar becerilerin sınıf düzeyine göre farklılaşması ortaya çıkabilir. Ayrıca 21. Yüzyıl becerileri açısından oldukça önemli olan Bilgi İşlemsel Düşünme (BİD) becerilerine yönelik öz yeterlilik algılarında da farklılaşmalar muhtemeldir. Bu sebeple bu araştırmada orta öğretim öğrencilerinin BİD becerilerine yönelik öz yeterlilik algıları araştırılmış ve farklı değişkenlerin BİD becerileri üzerindeki etkisi incelenmiştir. Öte yandan ilk defa 2018-2019 öğretim yılında uygulanmaya başlanan Bilişim Teknolojileri ve Yazılım Dersi öğretim programının BİD becerileri üzerindeki etkisinin incelenmesi ise bu araştırmanın diğer bir odağıdır. Bu kapsamda bu araştırmada aşağıdaki araştırma sorularına cevap aranmıştır;

1. Ortaöğretim öğrencilerinin BİD becerilerine yönelik öz yeterlilik algıları hangi düzeydedir?
2. Sınıf düzeyine göre BİD becerilerine yönelik öz yeterlilik algıları değişkenlik göstermekte midir?
3. Blok tabanlı programlama aracı kullanma süresine göre BİD becerilerine yönelik öz yeterlilik algıları değişkenlik göstermekte midir?
4. Cinsiyete göre BİD becerilerine yönelik öz yeterlilik algıları değişkenlik göstermekte midir?
5. Bilgisayara sahip olup olmam durumuna göre BİD becerilerine yönelik öz yeterlilik algıları değişkenlik göstermekte midir?

## **Yöntem**

### ***Araştırma Modeli***

Bu çalışmada ortaokul öğrencilerinin BİD öz yeterlilik algıları ölçülmüştür. BİD öz yeterlilik algıları öğrenim görülen sınıf, blok tabanlı programlama aracı kullanma süresi, cinsiyet ve bilgisayar sahipliği değişkenleri açısından karşılaştırılmıştır. Çalışma kapsamında elde edilen veriler olduğu gibi sunulmuştur. Buna ek olarak karşılaştırmalar yapılarak veriler analiz edilmiştir. Bu özellikleri ile çalışma ilişkisel tarama modeline uymaktadır (Karasar, 2005)

### ***Çalışma Grubu***

Çalışma kapsamında elde edilen veriler 2018-2019 eğitim-öğretim yılında Van ili merkez ilçelerinden İpekyolu ve Edremit' te bulunan dört farklı ortaokuldan elde edilmiştir. Veriler 5., 6. ve 8. sınıflarda öğrenim gören ve 171'i kız ve 161'i erkek olmak üzere toplam 332 öğrenciden elde edilmiştir.

### ***Veri Toplama Aracı***

Veri toplama aracı iki bölümden oluşmaktadır. İlk bölümde katılımcıların cinsiyet, bilgisayar sahipliği, Bilişim teknolojileri ve yazılım dersi alıp almama, öğrenim görülen sınıf ve Blok tabanlı programlama aracı kullanma süresi gibi bilgilerinin yer aldığı form bulunmaktadır. İkinci bölümde ise Gülbahar, Kert ve Kalelioğlu (2018) tarafından geliştirilen Ortaokul Öğrencileri İçin Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği (BİDBÖA) yer almaktadır. Ölçekte bulunan tüm maddeler "Evet", "Kısmen" ve "Hayır" şeklinde cevaplandırılmaktadır. Veriler kodlanırken 1=Evet; 2=Kısmen; 3=Hayır şeklinde kodlanmaktadır. BİDBÖA ölçeği beş alt boyuttan oluşmaktadır. Algoritma Tasarlama Yeterliği olarak isimlendirilen birinci alt faktör 9 maddeden; Problem Çözme Yeterliği faktörü 11 maddeden; Veri İşleme Yeterliği faktörü 7 maddeden; Temel Programlama Yeterliği faktörü 6 maddeden; Özgüven Yeterliği faktörü 6 maddeden oluşmaktadır.

### ***Verilerin Analizi***

Ölçekten elde edilen ortalama puan hesap edilirken tüm maddeler tersten kodlanmıştır. Yani veriler kodlanırken 3=Evet; 2=Kısmen; 1=Hayır şeklinde kodlanmıştır. Bunun sonucunda da yüksek puan alan öğrencilerin BİD becerisine yönelik öz yeterlik algılarının yüksek; düşük puan alanların ise düşük olması sağlanmıştır. Veriler analiz edilirken karşılaştırmalar yapılmıştır. Karşılaştırmalar yapılırken ilişkisiz örneklem t testi ve ilişkisiz örneklem için tek faktörlü varyans analizi kullanılmıştır. İlişkisiz örneklem için tek faktörlü varyans analizi sonucunda ortaya çıkan

farklılaşmaların kaynağını tespit edebilmek için eğitim bilimleri alanında sıklıkla tercih edilen Bonferroni çoklu karşılaştırma testi tercih edilmiştir.

### **Bulgular ve Yorum**

BİDBÖA ölçümü sonucunda maksimum 3; minimum 1 puan alınabilmektedir ve alınabilecek puan aralığı (3-1=2) 2' dir. Katılımcılar üç kategoriye ayrılarak incelenmiştir. Dolayısıyla 2/3=0,66' lık aralıklarla gruplara ayrılmışlardır. 1-1.66 arası düşük öz-yeterlilik; 1,67-2,33 orta düzey öz-yeterlilik; 2,34-3 arası yüksek öz-yeterlilik olarak belirlenmiştir. Bundan hareketle, öğrencilerin %58'i orta düzeyde BİD öz yeterlilik algı düzeyine sahipken %39.2'si yüksek, %2.1'i ise düşük öz yeterlilik algısına sahiptir. Öğrencilerin BİDBÖA ölçümlerinin ve alt faktör ölçümlerinin sınıf düzeyine göre karşılaştırılmasına ilişkin veriler, ölçeğin toplam puanı ve alt faktörlerinin tümü için istatistiksel farklılaşmaların olduğunu göstermektedir ( $p<0.05$ ). Farklılaşmaların kaynağının tespiti için Bonferroni Post Hoc testi yapılmıştır. Buna göre Algoritma Tasarlama alt boyutu açısından 5. sınıfların ve 6. sınıfların 8. sınıflardan yüksek puanlara sahip oldukları görülmüştür. Problem Çözme alt boyutu açısından da 5. sınıfların ve 6. sınıfların 8. sınıflardan yüksek puanlara sahip oldukları görülmüştür. Veri işleme alt boyutuna göre 6. sınıfların ve 8. sınıfların 5 sınıflara oranla daha yüksek puanlara sahip oldukları görülmüştür. Temel programlama alt boyutu açısından sadece 6. sınıfların 8. sınıflara göre yüksek puanlara sahip oldukları görülmüştür. Özgüven faktörüne göre 5. sınıfların 6. ve 8. sınıflara oranla daha düşük puanlara sahip oldukları görülmüştür. BİDBÖA ölçeğinin geneli açısından ise 6. sınıf öğrencilerinin 5. sınıflardan ve 8. sınıflardan daha yüksek öz yeterlilik puanlarına sahip oldukları görülmüştür.

Öğrencilerin BİDBÖA ölçümleri ve ölçeğin alt faktörlerine göre Blok tabanlı programlama aracı kullanma süresi açısından karşılaştırmalar yer almaktadır. Gerçekleştirilen ilişkisiz örneklem için tek faktörlü varyans analizi sonuçlarına göre ölçeğin algoritma tasarlama alt boyutu açısından farklılaşmalar bulunmaktadır ( $F_{(5,328)}=24.304$ ,  $p<0.05$ ). Bonferroni testi ile Post Hoc karşılaştırması yapıldığında hiç blok tabanlı programlama aracı kullanmayanların 1 dönem ve altı, 2 dönem, 3 dönem ve üzeri kullananlara göre daha düşük puana sahip oldukları görülmektedir. Problem çözme ( $F_{(5,328)}=0.499$ ,  $p>0.05$ ), Veri işleme ( $F_{(5,328)}=1.362$ ,  $p>0.05$ ) ve Özgüven ( $F_{(5,328)}=0.334$ ,  $p>0.05$ ) alt boyutları açısından farklılaşma yoktur. Temel programlama alt boyutu açısından farklılaşmalar bulunmaktadır ( $F_{(5,328)}=13.908$ ,  $p<0.05$ ). Temel programlama alt boyutu için yapılan Bonferroni testine göre, blok tabanlı programlama aracını hiç kullanmayanların 1 dönem ve altı, 2 dönem, 3 dönem ve üzeri kullananlara oranla daha düşüktür. BİDBÖA ölçeğinin tamamı açısından ise blok tabanlı programlama araçlarını 1 dönem ve altı, 2 dönem, 3 dönem ve üzeri kullananların özyeterlilikleri hiç kullanmayanlara oranla anlamlı düzeyde daha yüksektir ( $F_{(5,328)}=8.529$ ,  $p<0.05$ ).

BİDBÖA ölçümü ve ölçeğin alt boyutları açısından cinsiyete göre gerek ölçeğin tamamı için gerekse alt boyutları için farklılaşma yoktur ( $p>0.05$ ). Öğrencilerin bilgisayara sahip olup olmama durumuna göre ölçeğin Özgüven alt boyutu hariç diğer boyutlarına göre farklılaşma görülmemiştir ( $p>0.05$ ). Özgüven alt faktörü açısından ise bilgisayarı olan öğrencilerin ölçümleri olmayanlara kıyasla istatistiksel olarak daha yüksektir ( $p<0.05$ ).

### **Sonuç ve Tartışma**

Bu çalışmada ortaokul öğrencilerinin bilgi işlemsel düşünme becerilerine yönelik öz yeterlilik algıları araştırılmıştır. Öğrenim görülen sınıf düzeyi, blok tabanlı programlama aracı kullanma süresi, cinsiyet ve bilgisayar sahipliği, açısından BİDBÖA ölçümlerinin farklılaşp farklılaşmadığı tespit edilmiştir.

2018 yılından itibaren Talim Terbiye Kurulu tarafından 5. ve 6. sınıflar için Bilişim Teknolojileri ve Yazılım dersi öğretim programına Problem Çözme ve Programlama olarak eklenen ünitenin süresi dersin % 50'lik bir bölümünü teşkil etmektedir (MEB, 2019a). Bu oran benzer şekilde 7. ve 8. sınıflar için % 44'tür (MEB, 2019b). Bu programlara 2018-2019 eğitim öğretim yılı itibarıyla topyekün olarak geçilmiştir ve bu kapsamda bilgi-işlemsel düşünme becerilerinin öğrencilere kazandırılması da hedeflenmektedir. Araştırma verilerinin toplandığı süreçte bu öğretim programları ilk defa uygulanmıştır. Sınıf düzeyi açısından yapılan karşılaştırmalarda BİDBÖA toplam ölçek puanına göre 6 sınıf öğrencilerinin ölçümleri 5. ve 8. sınıflara göre anlamlı düzeyde yüksektir. Ölçeğin diğer alt boyutları için de farklılaşmalar vardır. Algoritma tasarlama alt boyutuna göre 8. sınıflar 5. ve 6. sınıflara

göre daha düşük; Problem çözme alt boyutu için 8. sınıflar 5. ve 6. sınıflara göre daha yüksek; Veri işleme ve Özgüven alt boyutuna göre 5. sınıflar 6. ve 8. sınıflara oranla daha düşük; Temel programlama alt boyutu açısından 6. sınıflar 8. sınıflara göre daha yüksek puanlara sahiptirler.

Bu çalışmada elde edilen veriler Bilişim Teknolojileri ve Yazılım dersi öğretim programının ilk kez uygulandığı öğrencilerden elde edilmiştir. Buna ek olarak sınıf düzeyi farklılaşsa bile, birbirine benzer sürelerde Problem Çözme ve Programlama konusunu alan bu öğrencilerin geçmiş altyapıları bilinmediği için tespit edilen bu farklılaşmaların olası nedenlerinin ne olduğu kestirilememektedir. Ancak 5. ve 6. sınıf öğrencilerinde BİD'in alt faktörlerinden algoritma tasarlama ve temel programlamaya yönelik öz yeterlilik algıları 7. ve 8. Sınıflara göre yüksek bulunmasının sebebi 5. sınıflarda daha çok temel bilgi ve kavrama düzeyinde kazanımlar yer alırken 6. Sınıflar temel bilgi ve kavrama düzeyinde kazanımların yanısıra uygulama düzeyinde kalması olabilir. 5. sınıflarda daha çok temel kavramların öğretimine yönelik bilişsel kazanımların yer alması ve uygulamaya yönelik kazanımların ise daha az yer almasından dolayı ise veri işleme becerilerinin daha düşük düzeyde kaldığı düşünülmektedir. Öte yandan 7. ve 8. sınıflarda daha çok problem çözme becerilerinin gelişmesine yönelik kazanımlar yer alması 7. ve 8. sınıfların problem çözme becerilerine yönelik öz yeterlilik algılarının daha yüksek çıkmasına neden olmuş olabilir. Ayrıca problem çözme becerilerinin öz güven üzerinde etkili olduğuna yönelik literatürde çeşitli araştırmalar mevcuttur (Hair, 2003; Yenice, 2012). Bu sebeple 7 ve 8. sınıfların özgüven algıları 5. ve 6. Sınıflara göre daha yüksek olabilir. Öğrenim görülen sınıf değişkenine ek olarak Problem Çözme ve Programlama konusu dâhilinde kullanılan Blok tabanlı programlama araçlarını kullanım süresinin incelenmesi ile daha sağlıklı yorumlar yapılabileceği düşünülmüştür ve bu değişkene göre karşılaştırmalar da yapılmıştır.

Araştırmada Blok tabanlı programlama aracı kullanım süresi açısından öğrencilerin BİDBÖA ölçeği sonuçları farklılaşmaktadır. Alt boyutlardan Algoritma tasarlama ile Temel Programlama ölçümlerine göre de farklılaşmalar tespit edilmiştir. Blok tabanlı programlama araçlarını kullanan öğrencilerin hiç kullanmayanlara göre BİDBÖA ölçümleri anlamlı düzeyde daha yüksektir. Fakat kullanım süresi arttıkça farklılaşma meydana gelmemektedir. Alan yazında genel olarak blok tabanlı programlama araçlarının kullanımının BİD becerisine olumlu katkıların olduğu ifade edilmektedir (Yünkül, Durak ve Çankaya, 2018). Blok tabanlı araçları kullanmanın BİD becerisine katkı sunmadığını ifade eden çalışmalar ise öğrencilerin duyuşsal olarak bu araçları kullanma noktasında olumlu düşündüklerini ortaya koymaktadır (Kalelioğlu ve Gülbahar, 2014; Ataman-Uslu, Mumcu ve Eğin, 2018). Bundan hareketle MEB (2019a;2019b)' in yaptığı düzenlemenin bir eksikliği giderdiği ve bu konu için bir ünitenin ayrılmasının BİDBÖA değişkeni açısından yeterli olduğu söylenebilir. Yani blok tabanlı programlama araçlarını belli bir süre kullanmak öğrenciler açısından duyuşsal olarak olumlu etki yapmaktadır.

Araştırmada ortaokul öğrencilerinin BİDBÖA ölçümlerinin ve ölçeğin alt boyutlarının cinsiyete göre farklılaşmadığı bulunmuştur. Çalışma kapsamındaki örneklem gurubunun orta düzeyde BİD becerisine sahip öğrencilerden oluşması ayrıca örneklem gurubunun 11-15 yaş aralığındaki öğrencilerden oluşması cinsiyetin BİD üzerinde etkisinin ortaya çıkmamasında etkili olduğu şeklinde düşünülmüştür.

Ayrıca araştırmada bilgisayara erişim kolaylığının BİDBÖA ölçüm puanları üzerinde etkili olmadığı, ölçeğin alt faktörleri açısından ise sadece Özgüven faktörü üzerinde etkili olduğu bulunmuştur. Benzer şekilde Sarıtepeci (2018) teknolojiye erişim kolaylığının BİD üzerinde etkili olmadığını bulmuştur. Alt faktörler açısından ise sadece işbirliği alt faktörü açısından teknolojiye erişim kolaylığı çok yüksek ve çok düşük olan öğrenciler arasında anlamlı farklılığın oluştuğunu diğer alt faktörler açısından ise her hangi bir farklılığın oluşmadığını bulmuştur. Bu sebeple bilgisayara erişim imkânının yanısıra internet erişim olanağı gibi diğer teknolojik erişimlerinde birlikte incelenmesi gerekmektedir. Bu araştırmadan farklı olarak Werner vd. (2012) bilgisayar erişim imkanı daha kolay ve akademik performansı daha iyi olan öğrencilerin deneysel araştırma süreci sonunda BİD becerilerinin de yüksek olduğunu ancak bilgisayar kullanım sıklığının ise BİD becerisi üzerinde etkisinin olmadığını bulmuşlardır. Bu bulgu BİD'in geliştirilmesine yönelik gerçekleştirilecek deneysel araştırmalarda bilgisayara sahip olma durumunun etkili olabileceğini ortaya koymaktadır.

Bu araştırma sonuçları orta öğretim öğrencilerinin BİD öz yeterlilik algıları üzerinde farklı değişkenlerin etkisini ortaya koymaktadır. Bu sebeple sonraki araştırmalarda yerleşim yeri, anne ve baba mesleki ve ekonomik durumları ya da düşünme stilleri gibi BİD öz yeterlilik algıları üzerinde etkisi muhtemel olan farklı değişkenlerin etkisi araştırılmalıdır. Bunun yanısıra 2018 yılında ilk defa

uygulanmaya başlanan Bilişim Teknolojileri ve Yazılım dersi öğretim programının BİD öz yeterlilik algıları üzerindeki etkisi bu araştırmada sınıf düzeyine göre ortaya konmuştur. Sonraki araştırmalarda ise yeni öğretim programının etkililiğinin öğretmen, öğrenci görüşlerinin yanısıra deney ve gözlem verilerini de elde ederek sınıf düzeyine göre ortaya konması önerilmektedir. Ayrıca bu araştırmada 7. Sınıf öğrencilerinden BİD özyeterlilik algılarına ilişkin veriler toplanmamıştır. Özellikle problem çözme becerisine ilişkin kazanımların 7. Sınıf öğretim programından itibaren daha fazla yer verilmeye başlanmasından dolayı sonraki araştırmalarda bu öğrencilerin de örnekleme dahil edilmesi önerilmektedir.

## INTRODUCTION

The level of belief that individuals can overcome a task is one of the essential internal dynamics for starting a behavior and performing well (Erol & Avci-Temiz, 2016). In addition, this belief, which is also defined as self-efficacy, is the source of many behaviors such as overcoming the obstacles that an individual can meet in the process, coping with various situations and maintaining this belief (Bandura, 1989a). For the first time, Bandura mentioned self-efficacy perception as one's belief in him/herself. He argued this perception is the basis of human determination, success, and motivation (Bandura, 1977). One of the most important factors leading to success is self-efficacy perception. If the individual does not believe in the desired effect of his/her achievement, then there will be little motivation (Bandura, 2010). Self-efficacy perception is influenced by individuals' own experiences, observations, social norms and psychological factors (Bandura, 1982). The perception of self-efficacy also develops with the experiences of the individual (Delcourt & Kinzie, 1993). Even if the individual has the necessary skills and knowledge in a related field, one may not have sufficient awareness and therefore has doubts. In this case, the individual may not be able to use her/his skills effectively and consistently. For this reason, the individual must first have the knowledge and skills; in other words, they should have self-efficacy belief (Bandura, 1989b). Since self-efficacy belief is directly related to self-confidence, self-efficacy belief enhances and increases the individual's effort and achievement (Moore and Anderson, 2003). When the self-efficacy of the individual is sufficient, then one is unafraid of problems and sees them as an opportunity for personal development. In addition, when an individual is not afraid of problems, one can set higher goals (Bandura, 1993). For this reason, self-efficacy perception should be supported and strengthened. There are four factors that feed self-efficacy perception (Bandura, 1977). These are personal experiences, emotional situations, indirect experiences and verbal persuasion. Personal experiences positively affect self-efficacy perception by increasing the motivation of the individual because of the reward effect on the success of the tasks. On the other hand, emotional states increase self-efficacy perception by increasing the likelihood of an individual to intervene. Indirect experiences seeing other individuals as successful in similar tasks have a positive effect on self-efficacy perception. The words that others hear about the achievement of the individual can contribute to the courage of the individual and the increase of self-efficacy.

An individual's judgment on computer skills is defined as Computer Self-Efficacy Perception (Compeau & Higgins, 1995). Computer technologies are used as an effective tool for learning, as well as for accessing and transmitting information. In order to use computer systems more effectively, individuals need to feel safe and competent while using computers. Otherwise, the individual cannot use the computer efficiently (Geer, White & Barr, 1998). Therefore, it is argued that having a high Computer Self-Efficacy perception will have positive effects on the individual's participation in computer-mediated activities, his patience with problems during his duty, and his performance (Murphy et al., 1989).

Computational Thinking (CT) has become increasingly important in the field of learning in recent years. CT emphasizes that an individual thinks like a computer scientist in solving a problem (González, M. R., 2015). CT involves problem solving, system design, and understanding human behavior in line with the basic concepts of computer science (Wing, 2006). With CT, the problem is simplified (abstraction), generalized, the algorithm is created, the problem can be modified (modularity) and decomposed (Weintrop, Holbert, Horn and Wilensky, 2016). CT is a system formed by bringing together multiple thinking skills, such as creative, critical, algorithmic, cooperative thinking, and

supporting sub-skills, such as communication skills. (ISTE-International Society for Technology in Education, 2015; Grover, S., & Pea, R., 2013).

CT simplifies and eases students' activities, such as abstraction, problem comprehension, and focus on solution (Grover & Pea, 2013). CT enables students to think algorithmically and allows them to follow a regular path of solving and expressing problems. It also provides innovation in many fields, such as engineering, health, informatics, humanities, arts (Grover & Pea, 2013). For this reason, RFI skills enable students to select and use the necessary tools and appropriate strategies in problem solving and provides significant benefits in using algorithms to solve similar problems (Yadav, Zhou, & Mayfield, 2011). Lee et al., (2011) argue that CT and self-efficacy perception can be increased through programs that increase the student's experiences or through group work. In addition, supporting the student with positive words may contribute to the development of CT self-efficacy perceptions. In-school and out-of-school activities are recommended to improve CT skills. An example of this is to create computationally rich and 'use-modify-create' educational environments in which the student starts using pre-existing programs, makes changes, and creates new programs (Lee et. Al, 2011).

Integrating effective use of computers in primary and secondary education is undoubtedly a challenge for students, teachers, and the education system (Peng et al., 2014). Compared to previous years, there have been fundamental changes in the Information Technology and Software Course Curriculum, which was implemented in the 2018-2019 academic year (Mercimek and İlic, 2017). There are differences between 5<sup>th</sup>,6<sup>th</sup> and 7<sup>th</sup>,8<sup>th</sup> grade students. The Information Technology and Software Curriculum developed for 5<sup>th</sup> and 6<sup>th</sup> grades are based on a unit-based approach and includes the following five basic units.

1. Information Technologies
2. Ethic and Security
3. Communication, Research and Cooperation
4. Problem Solving and Programing
5. Product creation (Design)

The only difference can be noted in the 7<sup>th</sup> and 8<sup>th</sup> grade curricula. They don't have a unit of Ethics and Security. The aim of the Problem Solving and Programming unit is to equip students with the following skills: develop understanding of algorithm design (search, sort, etc.); verbal and visual expression; variable assignment for problem solving; using sequential logic, decision structure, loop and function structures; choosing and applying appropriate programming approach to solve problems (MoE, 2019b). In 5<sup>th</sup> and 6<sup>th</sup> grades, the Problem Solving and Programming unit constitutes 50% of the course, while this ratio is 44% in 7<sup>th</sup> and 8<sup>th</sup> grades. In addition, when the cognitive learning outcomes of the course are compared, the outcomes related to Problem Solving Concepts and Approaches take place at the level of basic knowledge and comprehension in 5<sup>th</sup> grade. In 6<sup>th</sup> grade, it includes the basic knowledge and comprehension level, as well as the practical level outcomes (MoE, 2019a). While the curriculum of the 7<sup>th</sup> grade includes the achievements at the application and analysis level, it is seen that the 8<sup>th</sup> grade is mostly at the synthesis level (MoE, 2019b). According to cognitive outcomes related to programming perspective, 5<sup>th</sup> grade curriculum contains the design of algorithms for recognizing block-based programming interfaces, learning linear logic structures, decision structures and loops, and acquiring basic knowledge. The curriculum of 6<sup>th</sup> grade includes outcomes in the design of algorithms and basic knowledge, as well as the use and application of block-based programming tools (MoE, 2019a). In the 7<sup>th</sup> and 8<sup>th</sup> grades, there are gains for the development of problem-solving skills, and in the 8<sup>th</sup> grade, there are also outcomes at the synthesis level and product design (MoE, 2019b).

The new information technologies and software course was first implemented in the 2018-2019 academic year, and as explained above, it may result in skills to be differentiated according to the grade level. There may also be differences in self-efficacy perceptions of CT skills, which are among the 21<sup>st</sup> century skills. Therefore, in this study, self-efficacy perceptions of secondary school students about CT skills were investigated, and the impact of variables on CT skills was investigated. Examining the effect of the Information Technology and Software course's curriculum on the skills of CT is another focus of this research. In this context, this research seeks to answer the following research questions:

1. What are the perceptions of secondary school students' self-efficacy towards CT skills?
2. Do self-efficacy perceptions of CT skills vary by grade level?
3. Do self-efficacy perceptions of CT skills vary according to the duration of block-based programming tool usage?
4. Do self-efficacy perceptions of CT skills vary by gender?
5. Do self-efficacy perceptions of CT skills vary depending on whether the students have a computer or not?

## METHOD

### 2.1. Research Model

In this study, computational thinking skills self-efficacy perceptions of secondary school students were measured. CT self-efficacy perceptions were compared in terms of class, block-based programming tool usage time, gender, and availability of a computer at the student's home. Therefore, the study employed the correlational descriptive model (Karasar, 2005).

### 2.2. Participants

Data were collected in four secondary schools in Ipekyolu and Edremit, which are the central districts of Van province in the 2018-2019 academic year. The participants of the study are 332 5<sup>th</sup>, 6<sup>th</sup> and 8<sup>th</sup> grade students (171 girls and 161 boys).

### 2.3. Data Collection Tools

The data collection tools have two parts. In the first part, there is a form that includes demographic information, such as gender, whether the student owns a computer, and whether the student has ever taken information technology and software courses, class and student's block-based programming usage time. The second part includes the Self-Efficacy Perception Scale for Computational Thinking Skills (CTSSP) for Secondary School Students, developed by Gülbahar, Kert and Kalelioğlu (2018). The responses of the scales are as follows: 1 – Yes, 2 – Partially agree, 3 – No. The scale has 5 subscales. Algorithm Design Competence consists of 9 items, Problem Solving Competence consists of 11 items, Data Processing Competence has 7 items, Basic Programming Competence is composed of 6 items, and Self Confidence Competence includes 6 items. Cronbach's Alpha ( $\alpha$ ) coefficients ranged from 0.930 to 0.762. The Cronbach's Alpha ( $\alpha$ ) coefficient of the 39-item scale was 0.943. The scale explains 55.33% of the total variance. According to Confirmatory Factor analysis, fit indices are as follows:  $\chi^2/sd=2.52$ ; RMSEA=0.04; AGFI=0.90; CFI=0.95; GFI=0.91; NFI=0.91; IFI=0.95. The fit indexes are acceptable and have good fit values. Based on these results, it was decided the scale was sufficiently valid and reliable.

### 2.4. Data Analysis

While calculating the mean score of the scale, all items were coded in reverse. So all items were scored as follows: 3 – Yes, 2 – Partially agree, 1 – No. As a result, students with high scores meant they had higher self-efficacy perceptions of CT skills; low scores meant they had low scores of CT. In comparison analysis, uncorrelated samples t test and one-factor analysis of variance were used. Bonferroni Post Hoc test was used to determine the source of differentiation.

## FINDINGS

The tables below present the findings regarding whether the students' CTSSP levels and measurements differ according to the grade, block-based programming tool usage time, gender, and owning a computer. The frequency and percentages of the CT self-efficacy perception levels of the secondary school students are in Table 1.

**Table 1.** *The classification of students according to CTSSP*

CTSSP level	n	%
Low self-efficacy	7	2,1
Moderate self-efficacy	195	58,7
High self-efficacy	130	39,2
Total	332	100,0

For easier interpretation of the total scores obtained from the scale, it was assumed that the scale ranges were equal and the score range (Maximum Value-Minimum Value/number of categories =  $(3-1)/3 = 0.66$ ) was found to be 0.66 (Şahin, 2013). So the students with a mean of CTSSP between 1 and 1.66 were categorized as having low self-efficacy; the ones between 1,67 and 2,33 were at intermediate level; those between 2,34 and 3 were categorized as having high levels of self-efficacy. The lowest average is 1.42; the highest average is 3.0. As seen in Table 1, 58% of students had moderate CT self-efficacy perception, while 39.2% had high self-perception, and 2.1% had low self-efficacy.

Table 2 shows the average scores of secondary school students from the sub-components of the CTSSP by grade level.

**Table 2.** *ANOVA test result of CTSSP in terms of grade variable*

Measurement	Grade level	n	$\bar{X}$	Sd	F	p	Difference
Algorithm Design Competency	5 <sup>th</sup> grade	21	2.06	0.40	38.931	0.000*	5 & 8
	6 <sup>th</sup> grade	250	2.27	0.47			6 & 8
	8 <sup>th</sup> grade	61	1.65	0.61			
Problem Solving Competency	5 <sup>th</sup> grade	21	2.23	0.42	7.395	0.001*	5 & 8
	6 <sup>th</sup> grade	250	2.37	0.36			6 & 8
	8 <sup>th</sup> grade	61	2.53	0.27			
Data Processing Competency	5 <sup>th</sup> grade	21	1.96	0.53	6.745	0.001*	5 & 6
	6 <sup>th</sup> grade	250	2.32	0.42			5 & 8
	8 <sup>th</sup> grade	61	2.35	0.53			
Basic Programming Competency	5 <sup>th</sup> grade	21	2.02	0.50	30.081	0.000*	6 & 8
	6 <sup>th</sup> grade	250	2.28	0.45			
	8 <sup>th</sup> grade	61	1.76	0.57			
Self Confidence Competency	5 <sup>th</sup> grade	21	2.09	0.48	5.414	0.005*	5 & 6
	6 <sup>th</sup> grade	250	2.40	0.46			5 & 8
	8 <sup>th</sup> grade	61	2.46	0.44			
Total CTSSP	5 <sup>th</sup> grade	21	2.09	0.29	11.044	0.000*	5 & 6
	6 <sup>th</sup> grade	250	2.33	0.32			6 & 8
	8 <sup>th</sup> grade	61	2.16	0.30			

The data of the students in Table 2 on the comparison of CTSSP and its sub-scales according to grade level show the total score of the scale and the statistical differences of the sub-scales ( $p < 0.05$ ). Bonferroni Post Hoc test was used to determine the source of differentiation. Accordingly, the 5<sup>th</sup> and 6<sup>th</sup> grades have higher scores than the 8<sup>th</sup> grade in terms of the Algorithm Design sub-scale. As far as



the problemsolving subscale is concerned, 5<sup>th</sup> and 6<sup>th</sup> grades have higher scores than 8<sup>th</sup> grade. Grade 6 and grade 8 have higher scores than grade 5 in data-processing subscale. In terms of basic programming, 6<sup>th</sup> grade students have higher scores than grade 8 students. 5<sup>th</sup> grade students had lower scores than 6<sup>th</sup> and 8<sup>th</sup> grades in self-confidence subscale scores. In terms of CTSSP, 6<sup>th</sup> grade students had higher self-efficacy scores than 5<sup>th</sup> grade and 8<sup>th</sup> grade.

Table 3 shows the CTSSP and its subscale mean scores of the secondary school students according to the duration of using a block-based programming tool.

**Table 3.** ANOVA test result of CTSSP measurements in terms of block-based programming tool usage time variable.

Measurement	Duration of use	n	$\bar{X}$	Sd	F	p	Difference
Algorithm Competency	Never	89	1,76	0.60	24.3	0.000*	Never & 1 term and below
	1 term and below	50	2.35	0.45			
	2 terms	173	2.25	0.47			
	3 terms or more	20	2.41	0.34			
Problem Competency	Never	89	2.43	0.37	0.50	0.683	
	1 term and below	50	2.38	0.42			
	2 terms	173	2.38	0.34			
	3 terms or more	20	2.41	0.28			
Data Competency	Never	89	2.23	0.53	1.36	0.254	
	1 term and below	50	2.35	0.45			
	2 terms	173	2.31	0.42			
	3 terms or more	20	2.41	0.39			
Basic Competency	Never	89	1.89	0.57	13.9	0.000*	Never & 1 term and below
	1 term and below	50	2.37	0.46			
	2 terms	173	2.23	0.47			
	3 terms or more	20	2.31	0.29			
Self Competency	Never	89	2.37	0.44	0.33	0.801	
	1 term and below	50	2.36	0.51			
	2 terms	173	2.41	0.47			
	3 terms or more	20	2.37	0.45			
Total CTSSP	Never	89	2.14	0.31	8.53	0.000*	Never & 1 term and below
	1 term and below	50	2.36	0.35			
	2 terms	173	2.32	0.31			
	3 terms or more	20	2.39	0.26			

Table 3 shows the comparisons in terms of the duration of using the block-based programming tool. The results of the single factor analysis of variance show there is a significant difference ( $F_{(5,328)}=24.304$ ,  $p<0.05$ ). When Bonferroni test and Post Hoc comparison were used, it was seen that those who did not use any block-based programming tools had lower scores than those who used it for 1 term and below, 2 terms, 3 terms or more. There is no significant differentiation in problem-solving ( $F_{(5,328)}=0.499$ ,  $p>0.05$ ), data- processing ( $F_{(5,328)}=1.362$ ,  $p>0.05$ ), and Self-Confidence ( $F_{(5,328)}=0.334$ ,  $p>0.05$ ) subscales. There is a meaningful difference in terms of basic programming ( $F_{(5,328)}=13.908$ ,  $p<0.05$ ). According to the Bonferroni test for basic programming, those who never use the block-based programming tool are lower than those who use 1 term and below, 2 terms, 3 terms or more. In terms

of CTSSP, self-efficacy was significantly higher for those who used block-based programming tools for 1 term and below, 2 terms, 3 terms and more than those who had never used it ( $F_{(5,328)}=8.529$ ,  $p<0.05$ ).

Table 4 presents the results according to gender variable.

**Table 4.** *T-Test Results of CTSSP according to gender variable.*

Measurement	Gender	n	$\bar{X}$	Sd	df	t	p
Algorithm Design Competency	Female	171	2.17	0.54	330	0.984	0.326
	Male	161	2.11	0.57			
Problem Solving Competency	Female	171	2.42	0.35	330	1.725	0.086
	Male	161	2.35	0.36			
Data Processing Competency	Female	171	2.31	0.46	330	0.542	0.588
	Male	161	2.29	0.46			
Basic Programming Competency	Female	171	2.15	0.55	330	-0.766	0.444
	Male	161	2.19	0.50			
Self Confidence Competency	Female	171	2.40	0.47	330	0.453	0.651
	Male	161	2.38	0.46			
Total CTSSP	Female	171	2.30	0.33	330	0.999	0.318
	Male	161	2.26	0.33			

Table 4 presents the results of CTSSP and its sub-scales according to gender variable. T-test results show there was no significant difference in gender variable ( $p>0.05$ ). Table 5 presents the results according to availability of computer at student's disposal.

**Table 5.** *T-Test results according ownership of a computer.*

Measurement	Computer ownership	n	$\bar{X}$	Sd	df	t	p
Algorithm Design Competency	Yes	138	2.10	0.59	330	-1.173	0.242
	No	194	2.17	0.50			
Problem Solving Competency	Yes	138	2.42	0.36	330	1.381	0.168
	No	194	2.37	0.35			
Data Processing Competency	Yes	138	2.32	0.50	330	0.729	0.467
	No	194	2.28	0.42			
Basic Programming Competency	Yes	138	2.20	0.56	330	0.895	0.371
	No	194	2.15	0.49			
Self Confidence Competency	Yes	138	2.45	0.43	330	2.085	0.038*
	No	194	2.35	0.48			
Total CTSSP	Yes	138	2.30	0.34	330	0.728	0.467
	No	194	2.27	0.32			

The results show there was no significant difference according to self-confidence sub-scale ( $p>0.05$ ). In terms of self-confidence scale, students with a computer have higher scores than those who do not own a computer ( $t=2.085$ ,  $p<0.05$ ).

## CONCLUSION AND DISCUSSION

This research examined the self-efficiency perceptions of secondary school students of computational thinking skills based on the variables of grade, block-based programming tool usage time, gender, and computer ownership.

In 2018, the Problem Solving and Programming unit was added to the Information Technologies and Software course by the Board of Education for 5<sup>th</sup> and 6<sup>th</sup> grades. The unit constitutes 50% of the course (MoE, 2019a). This ratio is 44% for 7<sup>th</sup> and 8<sup>th</sup> grades (MoE, 2019b). As of the 2018-2019 academic year, these programs were fully implemented to equip students with computational thinking skills. In the process of data collection, the curricula were applied for the first time. In the comparisons made in terms of grade level, the measurements of the 6<sup>th</sup> grade students according to the total scale score of the CTSSP were significantly higher than the 5<sup>th</sup> and 8<sup>th</sup> grades. There are also differences for other sub-scales. 8<sup>th</sup> grade student scores were lower than 5<sup>th</sup> and 6<sup>th</sup> grades according to the sub-scale of designing algorithms; as for problem solving subscale, 8<sup>th</sup> grade students performed higher than 5<sup>th</sup> and 6<sup>th</sup> grades, and in terms of data processing and self-confidence sub-scale, 5<sup>th</sup> grade scores were lower than 6<sup>th</sup> and 8<sup>th</sup> grades. When basic programming sub-scale is considered, 6<sup>th</sup> grade students have higher scores than 8<sup>th</sup> grade students. Contrary to these findings, there are conclusions that the CT skills do not differ in terms of learning level (Atmatzidou & Demetriadis, 2016). The data was collected from the students who took the first Information Technology and Software course. Therefore, it is not possible to predict the possible causes of these differences. However, perceptions of self-efficacy towards algorithm design and basic programming were higher in 5<sup>th</sup> and 6<sup>th</sup> grade students compared to 7<sup>th</sup> and 8<sup>th</sup> grade students. The reason may be that, in the 5<sup>th</sup> grade, students master the basic knowledge and comprehension level, whereas in the 6<sup>th</sup> grade, the basic knowledge and comprehension level reaches a higher level. It is thought that data processing skills remain at a lower level in 5<sup>th</sup> grade, because the focus is teaching more cognitive but less practical skills. On the other hand, in the 7<sup>th</sup> and 8<sup>th</sup> grades, students develop problem solving skills. This may have caused higher self-efficacy perceptions of 7<sup>th</sup> and 8<sup>th</sup> grade problem solving skills. In addition, there are various studies in the literature that problem-solving skills affect self-confidence (Hair, 2003; Yenice, 2012). Therefore, self-confidence perceptions of 7<sup>th</sup> and 8<sup>th</sup> grades may be higher than 5<sup>th</sup> and 6<sup>th</sup> grades. In addition to the class variable, it was thought that better comments could be made by examining the usage time of Block-based programming tools.

The results of the CTSSP scale of the students differ in terms of block-based programming tool usage time. Differences were identified according to sub-categories of Algorithm design and Basic Programming measurements. Erdem (2018) found that, even though different teaching strategies were used, self-efficacy perceptions of CT skills increased as a result of using block-based programming tools. Oluk and Korkmaz (2016) found that Scratch, a block-based coding and application development environment, increased students' BCT skills and that students with high skills in Scratch programming had higher CT skills. Similarly, Oluk, Korkmaz and Oluk (2018) argued 5<sup>th</sup> grade students studying with the block-based programming tool developed their CT skills more than the traditional method learners. Resinovic (2015) states that using a humanoid robot and visual programming will increase students' CT skills. In addition, Kalelioğlu and Gülbahar (2014) stated that visual programming activities do not have an impact on CT skills in 5<sup>th</sup> grade. However, Kalelioğlu and Gülbahar (2014) indicated that students liked programming and were eager to improve themselves. Similarly, Ataman-Uslu, Mumcu and Eğin (2018) found there was no increase in computational thinking skills as a result of the teaching activity carried out with the use of block-based tools with 6<sup>th</sup> grade students. They concluded that students produced positive affective statements about the process. The use of block-based programming tools generally contributes positively to CT skills (Yünkül, Durak & Çankaya, 2018). Studies show that using block-based tools does not contribute to CT skills; however, they reveal that students think positively about using these tools.

Teachers state that there is no common education program before 2018, and it is one of the difficulties experienced in coding education (Ceylan & Gündoğdu, 2018). Before 2018, the teachers who taught programming in information technologies class used different methods, and their views on the success of the process were positive (Özçınar, Yecan & Tanyeri, 2016). In 2018, the Ministry of National Education added a coding related unit to the curriculum of Elective Information Technologies and Software course in the 5<sup>th</sup>, 6<sup>th</sup>, 7<sup>th</sup> and 8<sup>th</sup> grades (MoE, 2019a; MoE, 2019b). Yet, duration of unit

differentiated between 5<sup>th</sup>, 6<sup>th</sup> grades and 7<sup>th</sup>,8<sup>th</sup> grades. According to the findings obtained in this study, students using block-based programming tools had significantly higher CTSSP measurements than those who never used them. However, differentiation does not occur as the duration of use increases. Therefore, it can be said that the regulation made by MoE (2019a; 2019b) eliminates a deficiency, and it is sufficient to allocate one unit. In the literature, code teaching by using block-based programming tools was found to be entertaining and instructive by primary school students (Tağci, 2019). Also, research shows that teacher candidates' attitudes towards using block-based programming tools are positive, and they develop positive attitudes after using these tools (Arslan and Akçelik, 2019). In other words, using block-based programming tools for a certain period of time has a positive effect on students.

In this research, CTSSP of secondary school students and the sub-scales did not differ according to gender. Similarly, Oluk and Korkmaz (2016) found the CT skills of fifth grade students did not change according to gender. However, there are different studies indicating that gender plays a role on CT skills. Kirit, Dönmez and Çataltaş (2018) evaluated the CT skills of 26 gifted female and 33 gifted male secondary school students in terms of gender. Research shows that men are more successful in terms of algorithmic, creative, critical thinking and cooperative learning; women were found to be more successful in terms of problem-solving skills. In addition, in the research of Atmatzidou and Demetriadis (2016) with 164 students in the 15-18 age range shows that CT skills were dependent on gender and age. For this reason, it was thought that the sample group of the study was composed of students with moderate CT skills. Moreover, the sample group consisted of students between the ages of 11 and 15 who were considered to be effective in avoiding the effect of gender on CT.

The research showed that ease of access to computer did not play a role in CTSSP scores, but in terms of the sub-scales, it was found to be effective only on the self-confidence scale. Similarly, Sarıtepeci (2018) found that ease of access to technology does not affect CT. In terms of cooperation subscale, there was a significant difference only among the students with very high and low ease of access to technology. For other sub-scales, there was no significant difference. Therefore, in addition to the possibility of access to the computer, other technological accesses, such as internet access, should be examined. Unlike this study, Werner et al. (2012) found that students with better computer access and better academic performance had higher CT skills. However, they found that frequency of computer use had no effect on CT ability. This finding reveals computer access may be effective in developing CT skills

The results of these studies reveal the effect of different variables on secondary school students' perceptions of CT self-efficacy. For this reason, the effects of different variables on the perceptions of CT self-efficacy should be investigated in future research. The variables such as location, occupational and economic status of parents, and thinking styles can be analysed. The impact of Information Technology and Software course curriculum on CT self-efficacy perceptions was introduced according to grade level in this study. Future studies may investigate the impact of the new curriculum according to grade level by working with teachers, collecting student opinions, and incorporating observation data. In addition, data were not collected from 7<sup>th</sup> grade students regarding their self-efficacy perceptions. Since problem-solving skills are emphasized more in the 7<sup>th</sup> and 8<sup>th</sup> grade curriculum, these students should be included in the sample of the following research.

## REFERENCES

- Arslan, K., & Akçelik, M. (2019). Using Scratch in programming language: Teacher candidates' attitudes and perceptions. *National Journal of Education Academy*, 3(1), 41-61.
- Ataman-Uslu, N., Mumcu, F. & Eğin, F. (2018). The effect of visual programming activities on secondary school students' computational thinking skills. *Journal of Ege Education Technologies*, 2(1), 19-31.
- 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.
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological review*, 84, 191-215.
- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs Prentice Hall
- Bandura, A. (1989a). Human agency in social cognitive theory. *American Psychologist*, 44, 1175-1184.
- Bandura, A. (1989b). Regulation of cognitive processes through perceived self-efficacy. *Developmental Psychology*, 25, 729-735.
- Bandura, A. (1993). Perceived self-efficacy in cognitive development and functioning. *Educational Psychologist*, 28, 117-148.
- Bandura, A. (1997a) Current directions in self-efficacy research. *Advance in Motivation and Achievement*, 10, 1-49.
- Bandura, A. (1997b). *Self-efficacy: The exercise of control*. New York: W.H. Freeman and Company
- Bandura, A. (2010). *Self-Efficacy*. *The Corsini Encyclopedia of Psychology*. doi:10.1002/9780470479216.corpsy0836
- Ceylan, V. K. & Gündoğdu, K. (2018). A phenomonological study: What's happening in coding education? *Educational Technology Theory And Practice*, 8(2), 1-34.
- Delcourt, M., & Kinzie, M. (1993). Computer technologies in teacher education: The measurement of attitudes and self-efficacy. *Journal of Research and Development in Education*, 27, 31-37.
- Erdem, E. (2018). *The Investigation of Different Teaching Strategies During Teaching Programming Process in Block Based Environment in Terms of Different Factors*. (Master thesis, Başkent University Institute of Education).
- Erol, M., & Avci-Temizler, D. (2016). A catalyst that put into action "perception of self-efficacy": a study on university students. *Hacettepe University Journal Of Education*, 31(4), 711-723.
- González, M. R. (2015). Computational thinking test: Design guidelines and content validation. In *Proceedings of EDULEARN15 conference* (pp. 2436-2444).
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- Gülbahar, Y., Kert, S. B., & Kalelioğlu F. (2018). The Self-Efficacy Perception Scale for Computational Thinking Skill: Validity and Reliability Study. *Turkish Journal of Computer and Mathematics Education*, Advance online publication. doi:10.16949/turkbilmate.385097.
- Şahin, İ. (2013). Öğretmenlerin iş doyumları düzeyleri. *Journal of Yüzüncü Yıl University Faculty of Education*, 10(1), 142-167.
- Kalelioğlu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33-50.
- Karasar, N. (2005). *Bilimsel araştırma yöntemi* (14. Baskı). Ankara: Nobel Yayın Dağıtım.
- Kirmit, Ş., Dönmez, İ., & Çataltaş, H. E. (2018). The study of gifted students' computational thinking skills. *Journal of STEAM Education*, 1(2), 17-26.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads*, 2(1), 32-37.
- Ministry of Education [MoE] (2019a). *Information technologies and software course curriculum (5<sup>th</sup> and 6<sup>th</sup> Grades)*. Retrieved from: <http://mufredat.meb.gov.tr/Dosyalar/2018124103559587-Bili%C5%9Fim%20Tecnolojileri%20ve%20Yaz%C4%B1m%20-%205-6%20S%C4%B1n%C4%B1flar.pdf> (Date of Access: 09.08.2019)

- Ministry of Education [MoE] (2019b). *Information technologies and software course curriculum (7<sup>th</sup> and 8<sup>th</sup> Grades)*. Retrieved from: [http://mufredat.meb.gov.tr/Dosyalar/2018813171426130-2-2018-81Bili%C5%9Fim%20Teknolojileri%20ve%20Yaz%C4%B1%C4%B1m%20Dersi%20\(7%20ve%208.%20S%C4%B1n%C4%B1flar\).pdf](http://mufredat.meb.gov.tr/Dosyalar/2018813171426130-2-2018-81Bili%C5%9Fim%20Teknolojileri%20ve%20Yaz%C4%B1%C4%B1m%20Dersi%20(7%20ve%208.%20S%C4%B1n%C4%B1flar).pdf) (Date of Access: 09.08.2019)
- Oluk, A., & Korkmaz, Ö. (2016). Comparing Students' Scratch Skills with Their Computational Thinking Skills in Terms of Different Variables. *Online Submission*, 8(11), 1-7.
- Oluk, A., Korkmaz, Ö., & Oluk, H. (2018). Effect of scratch on 5<sup>th</sup> graders' algorithm development and computational thinking skills. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 9 (1), 54-71.
- Özçınar, H., Yecan, E & Tanyeri, T. (2016, 26-29 April). Öğretmen gözüyle görsel programlama öğretimi. 3<sup>rd</sup>. *International Conference on New Trends in Education*, İzmir-Turkey.
- Pajares, F. (1996). Self-efficacy beliefs in academic settings. *Review of Educational Research*, 66, 543-578.
- Pajares, F. (2003). Self-efficacy beliefs, motivation, and achievement in writing: A review of the literature. *Reading and Writing Quarterly: Overcoming Learning Difficulties*, 19, 139-158.
- Pajares, F., & Kranzler, J. (1995). Self-efficacy beliefs and general mental ability in mathematical problem-solving. *Contemporary Educational Psychology*, 20, 426-443.
- Pajares, F., & Miller, D. M. (1994). Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal of Educational Psychology*, 86, 193-203.
- Pajares, F. & Valiante, G. (1997). Influence of self-efficacy on elementary students' writing. *The Journal of Educational Research*, 90, 353-360.
- Resinovic, B. (2015). The use of Nao, a humanoid robot, in teaching computer programming. *In The Proceedings of International Conference on Informatics in Schools: Situation, Evolution and Perspectives—ISSEP 2015*. (p. 63). Retrieved from: <http://eprints.fri.unilj.si/3185/1/issep2015-proceedings.pdf>. (Date of Access: 30.03.2016.)
- Sartepeci, M. (2018). Analysis of computational thinking skill level in secondary education in terms of various variables. *Fifth International Instructional Technologies & Teacher Education Symposium (ITTES)*, İzmir.
- Tağci, Ç. (2019). *Research of Effects of Coding Education on Primary School Students*. (Master Thesis, Afyon Kocatepe University Institute of Science).
- Üzümcü, Ö., & Bay, E. (2018). A new 21<sup>st</sup> century skill in education: Computational thinking. *Uluslararası Türk Kültür Coğrafyasında Sosyal Bilimler Dergisi*, 3(2), 1-16.
- Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012, February). The fairy performance assessment: measuring computational thinking in middle school. *In Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 215-220). ACM.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-36.
- Yünkül, E., Durak, G. & Çankaya, S. (2018, October 26-28). Blok tabanlı yazılımların kodlama öğretiminde kullanımı. *International Necatibey Educational and Social Sciences Research Congress (NESAK 2018)*, Balıkesir-Türkiye.