

Ortaöğretim Proje Hazırlama Dersinde Android ile Robotik Programlama Etkinliklerinin Öğrencilerin Programlama Öz Yeterliliklerine ve Performanslarına Etkisi

Esra Cesur^{1*} 
Tuğba Yanpar Yelken² 

¹ Mersin Üniversitesi, Eğitim Programları ve Öğretimi Ana Bilim Dalı, Mersin, Türkiye
esrasesur@gmail.com

*Sorumlu Yazar

²Mersin Üniversitesi, Eğitim Programları ve Öğretimi Ana Bilim Dalı, Mersin, Türkiye
tyanpar@gmail.com

Geliş tarihi: 22.05.2024
Kabul tarihi: 14.11.2024
Yayın tarihi: 27.10.2024

Özet: Bu çalışmada, Ortaöğretim Proje Hazırlama dersinde Android programlama ile robotik programlama öğrenen öğrencilerin programlama öz yeterliliklerine ve performanslarına etkileri araştırılmaktadır. Araştırmada karma yöntem ve yakınsayan paralel desen kullanılmıştır. Araştırmanın yarı deneysel bir çalışma olmasından dolayı, otuz üç ortaöğretim öğrencisi kontrol grubu, otuz üç ortaöğretim öğrencisi deney grubu olarak toplamda altmış altı öğrenci araştırmanın örneklemini oluşturmaktadır. Araştırmada, deney grubu için Android programlama öğretim tasarımı ve kontrol grubu için robotik programlama öğretim tasarımı hazırlanmıştır. İlk başta programlama öz yeterlilik ölçeği geliştirilmiştir. Daha sonra performans değerlendirme formu geliştirilmiştir. Aynı anda kontrol grubu ve deney grubuna programlama öz-yeterlilik ölçeği ön-test, öğretim programları içerisindeki etkinlikler yapıldıktan sonra programlama öz-yeterlilik ölçeği son-test olarak uygulanmıştır. Çalışmada veri toplama araçları olarak öz yeterlilik ölçeği, performans değerlendirme formları, günlükler ve odak grup görüşmeleri soruları kullanılmıştır. Elde edilen bulgular sonucunda, ön test olarak uygulanan programlama öz yeterliliği ölçeğinin kontrol grubu ve deney grubu puan ortalamalarında anlamlı bir fark olduğu ortaya çıkmıştır. on haftalık programlama dersleri sonucunda öğrencilerin programlama öz yeterlilik düzeyleri, başlangıca göre yüksek çıkmıştır. Deney ve kontrol gruplarında, programlama öz yeterlilik erişim puanlarına bakıldığında anlamlı bir fark görülmüştür. Odak grup görüşme verileri analizi sonucunda, kontrol grubu ve deney grubu öğrencilerinin programlamaya yönelik düşüncelerinin hem olumlu hem olumsuz olduğu görülmektedir.

Anahtar Kelime: Programlama, Öz yeterlilik, Android, Arduino, Orta öğretim öğrencileri

GİRİŞ

Öğrenenlerin 21. yüzyıl yeteneklerini ilerletmeyi amaçlayan programlama ya da kodlama eğitimi ile, problemleri çözen, bilgiler üreten ve bilgiyi aktif olarak kullanan, eleştirel düşünen, girişimci ve eş duyum yapan bireyleri topluma hazırlamayı hedeflemektedir. Uluslararası yapılan kodlama etkinlikleri ve açık kitlesel çevrimiçi öğrenme ortamlarında hizmete sunulan programlama öğretimleri bilgi işlemsel düşünme becerisinin 21. yüzyıl becerileri arasında yerini almasından sonra daha çok önem kazanarak yaygınlaşmıştır. (Yauney, Bartholomew & Rich, 2023). Öğrenenlerin bu yetenekleri kazanıp kullanmaları için uygulanması gereken yöntemlerden birisinin, programlamanın ya da kodlamanın öğretilmesi olduğunu gösteren çalışmalar vardır (Williams & Cernochova, 2013:2; Günüş, Odabaşı & Kuzu, 2013:436).

Hızla ilerleyen yaşamda yeni yetişen genç nesillere varolan programları tüketmek yerine, gençlere yeni geliştirilen uygulamaları üretebileceklerinin gösterilmesi amaçlanmaktadır. Bilim ve teknoloji merkezli bir dünyada üretebilen bir ülke olabilmesi adına bireylerin kodlama yetkinliklerine önem verilmektedir. Böylece programlama ve kodlama yeteneği, ayrı bir araştırmanın bilim ve faaliyet alanlarında ve öğrenme-öğretme süreçlerinde temel düzeyde sahip olunması gereken bir yeterlilik haline gelmiştir (Köse & Tüfekçi, 2015).

Bilişim teknolojileri temellerinin esaslarından biri olan programlama eğitimi, yazılım uygulamalarının varoluşunu ve sürekliliğini gerçekleştiren önemli bir öğretim alanı olarak karşımıza çıkmaktadır (Kert & Uğraş, 2009). Öğrencilere erken yaşlardan başlayarak algoritma, kodlama ve programlama gibi kavramları zorunlu temel eğitim kapsamında kazandırılmaya başlanması onlara hayatlarının ilerleyen yıllarında bilginin işlenebilirliği açısından artı değer katacaktır (Bybee, 2010; Çepni, 2017; Okuyucu, 2019). Programlama veya kodlama öğretiminin çok önemsenen ilk basamağında programlama mantığı öğretimi yer almaktadır (Arabacıoğlu, 2006).

Programlama, öğrenenlerin matematiksel- kompüsyonel becerilerini (değişken ve koşul ifadeleri gibi) geliştirmekle birlikte, öğrenenler kod yazarken problem çözmeye ilişkin öğrenme stratejilerini, projeleri

tasarlamayı ve görüşler arasında bağlar kurmayı öğrenmektedirler. Yetenekler sadece uzman bilgisayarlıların değil her yaşta, uğraşan ve meslekteki bireyler için gerekli yeteneklerdir (Resnick, 2013).

Şimdiki zamanlarda programlama yeteneği, 21. yüzyıl yeteneklerinde sahip olunması beklenen önemli yetkinliklerden birisidir. Bilhassa bilinen bireylerin bu kavramı ifade etmesiyle birlikte pek çok platformda tartışıldığı, eğitim sistemlerine girmesi için projelerin başlatılıp yürütüldüğü, özel sektörün de bu girişimleri büyük bir istekle desteklediği görülmektedir. Bu durumun sonucunda, programlamayı okul öncesi öğrenenlere öğretmek için bazı programlama dilleri, alanları ve uygulama yazılımları oluşturulmuştur. Çoğu ücretsiz olan bu programlama dili araçlarıyla kod kullanmadan, sürükle bırak veya puzzle oynar gibi, uygulamalar oluşturma imkanı elde edilmiştir. Bu ortamlara Blok Programlama Ortamları adı verilmektedir. Alice, Blockly, App Inventor, Code Org ve Code Studio, Scratch, KoduLab ve Snap gibi bu tür uygulama ortamları yenilik ifadesiyle nitelendirilebilir (Numanoğlu & Keser, 2017).

Öğrenci düzeylerine göre yapılan araştırmalar incelendiğinde, programlamanın öğrencilerin performanslarına etkisini olumlu şekilde etkilediği görülmektedir. Diğer bir araştırmada ise, lise çağındaki bireylerin belirli bir öğretim programı kapsamında programlama dersini başarı ile öğrenmesinin nadir gerçekleşen bir durum olduğu saptanmış ve o çağlardaki bir birey için sınırları çizilmiş bir program dâhilinde bir eğitim süreci tamamlamanın sıkıcı gelme durumunun programlama eğitiminde de görüldüğü belirtilmiştir (Gerják, 2017). Bu sebepten dolayı, ortaokul 5. sınıf itibarıyla algoritma, programlama, veri tabanı gibi kavramların temel eğitim bünyesinde kazandırılması gereken bilgi ve yetenekler olarak kabul edilmesi gerektiğine vurgu yapılmaktadır (Akpınar & Altun, 2014). Ülkemizde, 2012 yılından sonra kodlama eğitimi ve bilişim teknolojileri eğitimi 5. sınıftan itibaren kademeli olarak eğitim müfredatına alınmaya başlanmıştır (Sayın & Seferoğlu, 2016).

Öğrencilerin ileri düzey programlama öğrenimleri ile ilgili olarak ortaöğretim ve üniversite eğitimlerinde akademik başarılarının artmasını sağladığı alanyazında görülmektedir. Konvalina, Stephens ve Wileman (1983) araştırmalarında üniversitedeki kodlama dersindeki başarının ve performansın; lisede yıl sonu not ortalaması, lise matematik yıl sonu not ortalaması ve ortaöğretimde verilen bilişim teknolojileri dersleriyle ilişkili olduğunu ileri sürmüşlerdir. Newsted (1975) araştırmasında ise, Wisconsin Üniversitesi'nde programlamaya giriş dersindeki başarıya etki eden faktörleri araştırdığını; programlamaya giriş başarısı ile yıl sonu not ortalaması (genel akademik başarı) arasında anlamlı bir ilişki olduğu görülmüştür.

Klasik öğretim metodunun programlama derslerinde kullanımı, programlama dilinin yetersiz öğrenimine, problem çözme becerileri ve ileride kullanacağı analiz ve tasarım becerilerinin az gelişmesi neden olur. Bu sebeple, programlama eğitim-öğretiminin başarısı, günümüz yaşamında ortaya çıkan sorunlara uygulanabilirlik düzeyi ile ölçülür (Karabak & Güneş, 2013). Bu sayede programlama eğitimi ile ülkelerin gelişmişlik düzeyleri artarak teknoloji alanında zirveye ulaşabilecekleri söylenebilir (Demirer ve Sak, 2016).

Rountree, Robins ve Rountree (2003)' in araştırmasında, programlama öğretiminin karşılaştıkları engellerin ve zorlukların; programlama ve kodlama öğretiminin kompleks yapısından, öğrenilenlere yönelik bilgileri öğretme, bu öğrenilenlerle ilgili yeni metodlar geliştirme ve uygulamalarda program yazabilme yeteneği olma gibi gerekliliklerden dolayı olduğunu belirtmektedirler. Bu durumun sebebi araştırıldığında öğrenciler, programlama ve kodlama öğreniminin teori ve uygulama çalışmalarının çok zor olduğunu ve bu konuda kaygı çektiklerini belirtmişlerdir (Meisalo, Suhonen, Torvinen & Sutinen, 2002).

Programlama eğitimi, öğrencilerde her kademedeki farklı çalışmalarla gerçekleştirilmektedir. Bununla birlikte programlama öğreniminin diğer derslerdeki başarıya ve performansa etkilerinin olumlu düzeyde olduğu araştırmalar sonucunda görülmektedir.

Programlama eğitiminin başlangıç ile gelişim süreçlerinin, ülkemizde ve dünya genelindeki detayları aşağıda verilmektedir.

Ülkemizde Programlama Eğitimi

Ülkemizde, okullardaki programlama ve kodlama eğitimi ve öğretiminin geliştirilmeleri ile ilgili yapılmış araştırmalar şu şekildedir; 2006 senesinde yürürlükte olan Bilişim Teknolojileri (BT) dersinde, ilköğretim programında, ileri düzey uygulamaları öğrenme alanları oluşturulmuştur. Bu öğrenme alanları içinde bulunanlar; programlama kavramı, nesne tabanlı programlama ve web tasarım konularıdır (Gülbahar &

Kalelioğlu, 2018). 2012 yılında yürürlüğe giren 69 sayılı kararı doğrultusunda BT dersinin ismi değiştirilerek “Bilişim Teknolojileri ve Yazılım” olmuştur, öğretim programına programlama ile yazılım öğretimi ile ilgili üniteler eklenmiştir ve 5. sınıf itibariyle temel programlama öğrenimini öğrenenler almaya başlamıştır. Bu şekilde, MEB’ nin yayınladığı BT ve Yazılım dersinin öğretim programı ile bilişim teknolojileri temellerini en iyi ve etkin şekilde kullanan bireyler olarak yetiştirilmesi amaçlanmıştır (Talim Terbiye Kurulu Başkanlığı, 2012). 2016 yılında yapılan yenileme sonucunda Bilgisayar Bilimi ismini alan ders, ortaöğretimde Kur 1- Kur 2 olarak uygulanmaya başlanmıştır. Ortaöğretim düzeyinde Bilgisayar Bilimi dersi öğretim programı ile seçmeli ders şeklinde verilmeye devam etmektedir. 2025 yılı itibari ile Bilgisayar Bilimi dersinin ismi seçmeli Bilişim Teknolojileri ve Yazılım dersi olarak değiştirilmiştir ve ders aynı öğretim programı kapsamındadır.

Dünyada Programlama Eğitimi

Estonya, Ekim 2012 tarihi ile 7 yaşı itibariyle küçük yaştaki çocuklar ve gençler programlama öğretimi almaya başlamışlardır (Demirer & Sak, 2016). Kanada’ da programlama eğitimi ve öğretimi, 6. sınıflarda zorunludur ve 2017 yılı itibariyle öğretim devam etmektedir (Şimşek, 2019). Güney Kore ülkesinin Bilim ve Gelecek Planlama Bakanlığı, ilkökul düzeyinden başlayıp programlama ile ilgili derslerin zorunlu olacağını, 2017 senesinde ilkökul öğrencilerinin, 2018’ de ise ortaöğretim öğrencilerinin aşamalı olarak programlama öğretimi alacakları açıklanmıştır (Özçakmak, 2014). 2013 yılı Kasım ayı itibari ile İngiltere’nin tüm okullarında Temel Bilgisayar Programlama eğitimi ve öğretimine başlanmıştır (Salter, 2013). İngiltere hükümetinin ortaöğretim öğretim programında yapılacak olan yenileme uygulamasından önce, 2014 yılını "kodlama yılı" ilan edilmiştir ve bu şekilde Avrupa Birliği (AB)’ nde "çocuklara kod öğretme" bilincinin oluşturulması açısından ilerleme yapılmıştır. Yapılan ilerleme ile birlikte, AB, 2013 senesi Kasım ayında Avrupa Kod Haftası (Europe Code Week) kutlamaları adı altında çeşitli atölye çalışmaları, uygulamalar ve etkinlikler düzenlemiştir (Öymen, 2014; Demirer & Sak, 2016). Ayrıca Norveç, Finlandiya ve Kuzey Avrupa’ nın diğer ülkelerinde bilişim Teknolojileri öğretimi erken yaşlarda başlarken, İtalya, İspanya ve Fransa’ da 6. sınıf ta başlamıştır (Ceylan, 2020).

Türkiye ve dünya genelinde 21. Yüzyıl becerileri ile çocuk ve yetişkinlerin programlama yeteneklerini geliştirmesi sağlanmaktadır. Ayıca bireylerin düzeylerine göre program oluşturmak için yazılan kodları gerçekleştiren çeşitli etkinlikler ve organizasyonlar düzenlenme ihtiyacı doğmuştur. Bu organizasyonların, etkinliklerin ve platformların birkaçı şunlardır: “Code Org”, “Bilge Kunduz Etkinliği”, “Haydi Kızlar Kodlamaya”, “Bilişim Garaj Akademisi”, “Kodlama Saati”, “Code Dejo”, “Code Club” ve “Khan Academy” dir (Elçiçek, 2019).

Dünyada da ülkemizde olduğu gibi programlama eğitimi standartları benzer bir süreçte yürütülmektedir. Bu sebeple, uluslararası bilişim platformunda gerçekleştirilen tüm etkinlikler ve aktiviteler öğrenenlerin programlama bilgisini ve öz yeterliliğini artırmaya yöneliktir.

Android (Mobil) Programlama

Günelik hayatta kullanılan mobil cihazlara yeni özellikler eklenmekte ve bu sayede güçlü ve dinamik bir öğrenme aracı oldukları ortaya çıkmaktadır. Mobil cihazlar için geliştirilen uygulama programlarında yer alan özellikler eğitimcilerin ve akademik camiasının ilgisini çekmiş, bu yenilikler ile birlikte mobil öğrenme, eğitim-öğretimde yer almaya başlamıştır (Bozkurt, 2015).

Android, Google tarafından tüm mobil cihazları için geliştirilmekte olan Linux tabanlı açık kaynak (Open Source) işletim sistemidir. Sistemin açık kaynak olmasındaki sebep diğer tüm açık kaynak projelerle hemen hemen aynı amacı benimsemesidir. Google Play oyun ve uygulama mağazası, Android işletim sistemi üzerinden çalışmaktadır. Ayrıca Android uygulamalara internet üzerinden farklı sitelerden de ulaşabilmektedir (Arslan & Gülnar, 2014).

Google Play’ in desteklediği uygulama (çalışır dosya) uzantısı “.apk’ dır. Android, aygıtlarda kullanılan uygulamalar yazan büyük ve geniş bir geliştirici grubuna sahiptir. Android işletim sistemindeki ara katman yazılımı, kütüphaneler ve API, C programlama diliyle yazılmıştır. Uygulama yazılımları(.apk) ise, Java uyumlu kütüphaneler içeren Apache harmony üzerine kurulu uygulama iskeleti üzerinden çalışır (Stallman, 2021).

Robotik Programlama

Robot kelimesi, ilk olarak Karel Čapek (1920)' de yazılan Rossum' s Universal Robots adlı eserinde yer almıştır ve daha sonra bu kelime tüm dünyada kullanılmaya başlanmıştır. 1990 yılında Maja Mataric'in robot sözcüğü tanımlamasında, çevresinden gördüğü bilgiler ile robotta var olan bilgi ile harmanlayarak çalışma prensibine uygun mantıklı hareket edebilen makine olarak belirtilmektedir (Kurşuncu, 2016). Daha sonra, robot kelimesinden türetilen robotik sözcüğü, 1941 yılında Isaac Asimov tarafından robotik kelimesi 1941' de kullanılmıştır (Kılınç, 2014).

Robotik kavramı; motorları, mekanik malzemeleri, sensörleri ve kodlama ile ilgili kavramları kapsayan geniş bir alanı oluşturmaktadır. Günümüzde robotik alanda artan yoğun ilgi ile birlikte eğitsel robotik kavramı ilk defa söylenmiştir. Robotik çalışmalarında kullanılan eğitsel robotik kitleri ilkokuldan ortaöğretime kadar tüm kademelerde yaygınlaşmaya başlamıştır (Rogers, Wendell & Foster, 2010).

Simülasyon yoluyla eğitsel robot kitleri, bilişsel uyumu gösteren gerçek aktif ve etkin öğrenme nesnelere (Ruzzenente vd, 2012). Eğitici robotik, öğrenenlerin hayatta yaşanan problemlerle karşılaşım çözümler üretmeleri ile uygun bir sistem yaratır ve oluşturdukları ve geliştirdikleri robotlar aracılığıyla öğrenenler tarafından öğrenirler (Eguchi, 2014).

Robotik kodlama donanım ve kodlamanın öğrenciler tarafından birlikte işe koşulması gerektiği yoğun bir zihinsel çaba gerektiren zor bir süreç olarak değerlendirilmektedir (Çakıroğlu & Kiliç, 2020). Yaşları küçük olan öğrenenlere “Kodlama ve Robotik” eğitimi ile, kodlamaya dayalı robotik çalışmalar yapmaları pek çok beceriyi kazandırmak için eğitim- öğretim sistemleri güncellenmektedir (Göksoy & Yılmaz, 2018). MEB (2019) öğretim programlarını incelediğinde ders adı değişmese de öğretim programında Robot Programlama ünitesinin yer almaya başladığı görülmektedir. Robotikle öğrenmeyi geliştiren öğrenme yaklaşımları, nihai üründen ziyade öğrenme sürecine odaklanılan proje tabanlı, probleme dayalı, tasarım yoluyla öğrenme, öğrenci merkezli ve yapılandırmacı öğrenme yaklaşımlarıdır (Çakıroğlu & Kiliç, 2020). Gelişmiş ülkelere bakıldığında, ülkelerin kodlama öğretimlerini anaokulu eğitimi verilen yaş grubuna kadar öğrendikleri ve eğitsel robot kitleri sayesinde küçük yaşlardan itibaren fen, teknoloji ve mühendislik eğitimi vermekte oldukları görülmektedir (Sullivan & Bers, 2016).

Robotik programlama eğitimlerin, bireylerin ilgi alanlarında olması ve bu ilgi alanının geliştirilmesi, ülkemizdeki programlama bilgi ve becerinin başarı düzeyini artıracakını ortaya çıkarmaktadır. Bu sebeple hem donanım hem yazılım ile programlamanın geliştirilmesi robotik kodlama eğitimleri ile gerçekleştirilebilir boyuttadır. Aynı şekilde mobil programlamanın geliştirilmesi ve ilgi alanı olarak görülmesi, programlamanın ve eğitimlerinin ülkemizdeki gelişimini artırabilecektir. Bu şekilde, uluslararası platformlarda yetkin ve nitelikli projeler ile yer edinilebilir.

Yapılan araştırmalar incelendiğinde, ortaöğretim öğrencileri ile yapılan programlama öğretimi araştırmalarının, diğer eğitim seviyesindeki öğrencilere göre çok az sayıda olduğu görülmüştür. Araştırma ile, ortaöğretim öğrencilerinin programlama öz yeterlilik ve akademik başarı düzeyleri hakkında bilgi edinilmesi sağlanmıştır.

Araştırmada, Ortaöğretim Proje Hazırlama Dersinde Android Programlama ile Robotik Programlamanın öğrencilerin programlama öz yeterliliklerine ve performanslarına etkisinin incelenmesi amaçlanmıştır.

YÖNTEM

Araştırmanın Deseni

Karma yöntem kullanılarak gerçekleştirilen araştırmada, nitel ve nicel veriler aynı anda toplanmıştır. Bu kapsamda yakınsayan paralel desen araştırma modelidir. Araştırma modeli yaklaşımında araştırmacı nitel ve nicel verileri aynı anda toplandıktan sonra, analizleri ayrı ayrı yapar ve elde edilen tüm verilerin sonuçları karşılaştırılarak bulgularda verilerin sonuçlarını doğrulayıp doğrulamadığını inceler (Creswell ve Plano Clark, 2015). Yarı deneysel desen kullanılan araştırma da dahil olmak üzere yarı deneysel araştırmalar, örneklemin rastgele seçilmediği durumlarda kullanılan, deney(D) ve kontrol(K) gruplarının olduğu ve deneysel

uygulamanın yapıldığı araştırmalardır (Yıldırım ve Şimşek, 2003). Bu sebeple iki sınıfın benzer özellikteki şubeleri arasında seçkisiz atama yolu ile biri deney(D) diğeri kontrol(K) grubu olarak iki sınıf belirlenmiştir. Deney grubunda Android mobil programlama öğretimi, kontrol grubuna ise Arduino robotik programlama öğretimi uygulanmıştır. Ön test ve son test olan nicel verilerinin toplanmasında kullanılan programlama özyeterliliği ölçeği sonuçları ile öğrencilerin performans değerlendirme puanları hesaplanmıştır. Odak grup görüşme ve günlükler sonucunda elde edilen veriler, nitel verilerdir. İlk başta, nicel veriler toplanmıştır. Daha sonra nitel verilerin toplanması sonrasında analizleri aynı zaman içerisinde gerçekleştirilmiştir. D ve K gruplarından elde edilen nicel ve nitel verilerin analiz sonuçları karşılaştırılmıştır. Nitel ve nicel verilerinin sonuçlarının, birbirini destekleyip desteklemediği gözlenmiştir.

Birinci basamak ortaöğretim öğrencilerinin programlama öz yeterliliklerinin belirlenmesi, ön test(Ö-T) ve son test(S-T) olarak “Programlama Öz-yeterliliği Ölçeği” ve öğrencilerin performans değerlendirme puanları ile gerçekleştirilmiştir. Ayrıca, aynı esnada öğrencilerden ders bitimi görüşlerini almak için günlükler ve ders öğretim programının sonunda odak grup görüşme ile nitel veriler toplanmıştır. Bu şekilde, ortaöğretim öğrencilerinin programlama öz yeterliliklerini ve performanslarını belirlemek için yapılan nitel ve nicel analizlerin karşılaştırılması arasında uyumun olup olmadığının daha iyi anlaşılması amaçlanmıştır.

Deney ve kontrol grubu olarak belirlenen ortaöğretim öğrenci sayısı (N=66) ve programlama öz-yeterliliği ölçeği oluşturma çalışmasında ortaöğretim öğrenci sayısı (N=362) örnekleme seçilmiştir. Bu araştırmada nitel ve nicel verilerin birleştirilmesinde 66 öğrenciden elde edilen veriler kullanılmıştır. Araştırmada, nicel veri toplamak için öğrencilerin programlama öz yeterlilikleri ile ilgili iki ölçüm ve performanslarını belirlemek için bir ölçüm toplam üç ölçüm yapılmıştır. İlk ölçüm, programlama öğrenimi ile ilgili 20 saatlik öğretim programı başlangıcında programlama öz-yeterliliği ölçeği öğrencilere ön test olarak uygulanmıştır. İkinci ölçüm olarak, programlama öğrenimi öğretim programının bitiminde programlama öz-yeterliliği ölçeği öğrencilere son test olarak uygulanmıştır. Üçüncü ölçüm ise, öğrencilerin öğrendikleri programlama dili ile ilgili performans değerlendirme puanlarıdır. Nicel verileri analiz ederken, ön test(Ö-T) ve son test(S-T) puanlarının uyumlu olup olmadığı sonucuna varılmıştır, bununla birlikte öğrencilerin performans değerlendirme puanları için de aynı istatistiki işlemler yapılmıştır. Daha sonra, tanımlayıcı istatistikler ortaya konulmuş ve diğer değişkenler açısından deney ve kontrol gruplarında anlamlı bir farkı belirlemek için istatistiksel gruplar arasında karşılaştırmaları yapılmıştır.

Araştırmada, öğrencilerin programlama öz yeterlilikleri hakkında yarı yapılandırılmış görüşme soruları oluşturulmuş ve odak grup görüşmeler hem kontrol hem deney grubunda yapılmıştır. Görüşmeler yazılı hale getirilmiş ve içerik analizi yapılmıştır. Aynı içerik analizi, günlükler için de gerçekleştirilmiştir. Bu analiz nicel analizden bağımsızdır. Böylece hem nitel hem nicel bilgilerin ilişkilendirilmesi için, birbirinden farklı ve eş zamanlı olarak nitel ve nicel veri setlerini belirleyip analiz edilmiştir. Bu doğrultuda, her iki veri türü de araştırmanın amacını ortaya koymak için eşit düzeyde önemli görülmüştür. Şekil 1.'de yakınsayan paralel desen akış şemasının araştırmadaki detayları gösterilmektedir.

Şekil 1*Araştırmanın Yakınsayan Paralel Desen Akış Şeması*

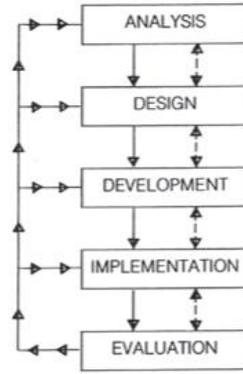
1.ADIM	<p>Nicel Kısmı Tasarlama Programlama öz yeterliliği ölçeği ve performans değerlendirme formu oluşturulmuştur. Nicel veri için izinler alınmıştır. İki sınıf belirlenmiştir ve yarı deneysel desen yaklaşımı uygulanmıştır.</p>	<p>Nitel Kısmı Tasarlama Odak grup görüşme ve günlük uygulanmıştır. Nitel veri için izinler alınmıştır ve veriler toplanmıştır.</p>
2.ADIM	<p>Nicel Verilerin Analizi Deney ve Kontrol gruplarına geliştirilmiş Programlama Öz yeterliliği Ölçeği ön test ders başlangıcında uygulanır. Ders bitiminde ölçek son test olarak uygulanır. Performans değerlendirme formu ile öğrencilerin performans puanları belirlenir. Ön test ve son test ile diğer değişkenlerin analizi yapılır. Ayrıca performans değerlendirme puanları analizi de yapılır. Tanımlayıcı istatistikleri yapılır.</p>	<p>Nitel Verilerin Analizi Tema gelişim prosedürleri kullanılarak nitel veri analizi yapılmıştır. Odak grup görüşme ve günlüklerin içerik analizi yapılmıştır.</p>
3.ADIM	<p>Kontrol ve Deney Grubu Sonuçlarını Birleştirmek için Strateji Kullanımı Kontrol ve deney grubu analiz sonuçları karşılaştırılır. Kontrol ve deney grubundaki farklılıklar diğer değişkenlere dayandırılarak belirlenir ve elde edilen boyutlar sonucunda organize edilmiş bir görüntüde farklılıklar incelenir. İçerik analizi, kontrol ve deney grubu için karşılaştırılır hale getirilir. Temalar sayılara dönüştürülür ve karşılaştırılarak ilişkilendirilir.</p>	
4.ADIM	<p>Birleştirilmiş Sonuçların Yorumlanması Kontrol ve deney grubu için çıkan sonuçlar birbirinden bağımsız özetlenir ve yorumlanır. Ö-T ve S-T sonuçları gruplar adına yorumlanır. Performans değerlendirme puanları her grup için yorumlanır. Odak grup görüşmeleri ve günlükler her grup için yorumlanır. Kontrol ve deney grubu verilerinin sonuçlarının birleşim noktaları, birbirlerinden ayrıldıkları tarafları ve birbirleriyle ilişkili olup olmadıkları açıklanır.</p>	

ADDIE Modeli

Araştırmada ADDIE tasarım modeli kullanılmıştır. Modelin adı İngilizce sözcükler olan Analysis(A) (Analiz), Design(D) (Tasarım), Development(D) (Geliştirme), Implementation(I) (Uygulama), Evaluation(E) (Değerlendirme) sözcüklerinin ilk harflerinin bir araya gelmesiyle oluşmuştur. Şekil 2.' de ADDIE Tasarım Modelinin şematik gösterimi aşağıda verilmektedir.

Şekil 2

ADDIE Modeli Şeması



Kaynak: (Grafinger, 1988)

ADDIE tasarım modeli, eğitim için kullanılan bir model olarak planlanmayı, geliştirilmeyi, uygulanmayı ve son olarak değerlendirilmeyi kapsar ve bu kapsamda öğretim alanını, öğretene, öğreneni, ölçme değerlendirmeyi ve dış faktörleri de dahil eden bir öğretim tasarımı modelidir (Gustafson & Branch, 2002).

ADDIE modeline göre, araştırmanın analiz basamağında, ortaöğretim öğrencilerinin donanım ve yazılım teknolojilerinin birleştirilmesi ile daha rahat yaşamların tüm dünyada etkili olabilmesini sağlayacak spesifik bir çalışma planlanmıştır ve bu çalışma için programlama öz yeterliliği kavramı doğrultusunda eğitim ve öğretim ortamının daha etkili ve kalıcı olacağı düşünülmüştür. Tasarım aşamasında ise, etkinliklerin planlanması ve sonrasında uygulanması gerçekleştirilmiştir. Yakınsayan paralel desen yöntemi ile önce nicel sonra nitel verilerin toplanıp analiz edilmesi sonucunda deney ve kontrol grupların sonuçlarını birleştirmek için strateji belirlenir. Öncelikle, kontrol ve deney grubu analiz sonuçları karşılaştırılır. Ortaya çıkan farklılıklar diğer değişkenlere dayandırılarak belirlenir ve elde edilen temalar sonucunda organize edilmiş bir görüntüde farklılıklar incelenir. Geliştirme aşamasında deney grubunun mobil programlama etkinlikleri ve kontrol grubunun robotik programlama etkinliklerinin sınıf ortamına uygun olarak hazırlanması çalışılmıştır. Uygun öğrenme ortamı yaratılmaya çalışılmıştır. 20 saatlik robotik ve mobil programlama öğretim programları oluşturulmuş ve bu doğrultuda her iki grup için de çalışma kağıtları oluşturulmuştur. Öğretim yöntemi olarak, anlatım ve gösteri yöntemleri kullanılması uygun görülmüştür. Değerlendirme sürecinde her iki grup içinde oluşturulması uygun görülmüş olan performans değerlendirme formları tasarlanmıştır. Uygulama aşaması, 20 saatlik öğretim programlarının, uygun sınıf ortamlarında, çalışma kağıtları ile anlatım ve gösteri yöntemleri etkinliklerin gerçekleştirilmesi ve performans değerlendirme formlarının öğrencilerin etkinliklerdeki performanslarına göre doldurulması ile sonuçlanır. Değerlendirme aşamasında, Kontrol-G ve Deney-G' larına ön test(Ö-T) ve son test(S-T) adına uygulanmakta olan ortaöğretim düzeyi programlama öz yeterliliği ölçeği analiz sonuçları ile performans değerlendirme formu sonuçlarının nicel analizleri yapılmıştır. Ders sonrası öğrencilerin yazdıkları günlüklerin ve odak grup görüşmelerin içerik analizi yapılmıştır. Günlüklerin ve görüşmelerin içerik analizleri temalar oluşturularak gerçekleştirilmiştir. Bu doğrultuda analizler sonucunda deney ve kontrol gruplarının birleşim noktaları, birbirinden ayrıldıkları tarafları ve birbirleriyle ilişkili olup olmadıkları açıklanmış olur.

Çalışma Grubu

Bu araştırma, yarı deneysel desen olarak uygulamalı bir çalışma olması sebebiyle bir ortaöğretim okulunda Proje Hazırlama dersini alan iki sınıfta uygulanmıştır. Mersin' in Yenişehir ilçesinin bir anadolu lisesinde 10. sınıfta öğrenim gören 66 öğrenciye deneysel çalışma yapılmıştır. Deney(D-G) grubunda 33 öğrenen (mobil programlama), kontrol(K-G) grubunda ise 33 öğrenen (robotik programlama) bulunmaktadır. Araştırmanın yarı deneysel çalışma grubunda bulunan öğrenenlerin cinsiyetlerine göre dağılımları Tablo 1.' de gösterilmektedir.

Tablo 1*Yarı Deneysel Uygulama Çalışma Grubunda Bulunan Öğrencilerin Cinsiyet(C) Değişkenine Göre Dağılımları*

(C)	(DG)		(KG)		(T)	
	f	%	f	%	f	%
Kız	16	48	19	58	35	53
Erkek	17	52	14	42	31	47
Toplam	33	100	33	100	66	100

Tablo 1 incelendiğinde, çalışma grubunda yer alan öğrenenlerin sayısı, 35' i kadın (%53), 31' i ise erkek (%47)' dir. Deney grubunda 33 (%50) kontrol grubunda ise 33 (%50) öğrenen yer almaktadır. Deney grubundaki öğrencilerin sayısı; 16'sı kız (%48), 17'si erkektir (%52). Kontrol grubundaki öğrencilerin sayısı ise; 19' u kız (%58), 14' ü erkektir (%42).

Araştırmada, nitel olarak odak grup görüşme soruları ile günlük formunun oluşturularak görüşlerin alındığı çalışma grubu, uygulama yapılan 66 öğrenciyi kapsamaktadır. Odak grup görüşme soruları için çalışma grubu, kontrol grubu için 5-6' şar öğrenciden oluşan 6 grup ve deney grubu için 5-6' şar öğrenciden oluşan 6 grup toplam da 12 gruptan oluşturmaktadır. Gruptaki öğrenci sayısı, uygulama yapılan çalışma grubu ile aynı olup toplamda 66 öğrenciyi içermektedir.

Tablo 2*Odak Grup Görüşme Çalışma Grubundaki Öğrencilerin Cinsiyet Değişkenine Göre Dağılımları*

G'lar	DG		KG		T	
	K	E	K	E	K	E
Grup1	5	-	3	3	8	3
Grup2	3	3	2	4	5	7
Grup3	5	-	4	2	9	2
Grup4	-	5	5	-	5	5
Grup5	-	6	3	2	3	8
Grup6	3	3	2	3	5	6
T	16	17	19	14	35	31

Tablo 2.' de görüldüğü üzere, odak grup görüşme çalışma grubundaki öğrencilerin deney grubu sırasına göre sayıları, 5 kız(K) öğrenci, 3 kız(K) 3 erkek(E) öğrenci, 5 kız(K) öğrenci, 5 erkek(E) öğrenci, 6 erkek(E) öğrenci ve 3 kız(K) 3 erkek(E) öğrenci olmak üzere toplam(T) 33 öğrencidir. Kontrol grubu sırasına göre öğrenci sayıları, 3 kız(K) 3 erkek(E) öğrenci, 2 kız(K) 4 erkek(E) öğrenci, 4 kız(K) 2 erkek(E) öğrenci, 5 kız(K) öğrenci, 3 kız(K) 2 erkek(E) öğrenci ve 2 kız(K) 3 erkek(E) öğrenci olmak üzere toplam(T) 33 öğrencidir.

Veri Toplama Araçları

Araştırmadaki yer alan veri toplama araçları; ortaöğretim öğrencileri düzeyine yönelik programlama özyeterlilik ölçeği, performans değerlendirme formu, öğrenci günlüğü ve odak grup görüşmeleridir. Aşağıda tüm veri toplama araçları ayrıntılı bir şekilde açıklanmaktadır.

Ortaöğretim Öğrencileri için Programlama Öz-yeterliliği Ölçeği

Programlama veya kodlama bilgisi az olan ve Bilgisayar Bilimi dersi seçen lise öğrencilerine yönelik, programlama özyeterliliği konusunda saptamaların tespit edilmesi öğretim sürecine etkili ve olumlu bir katkı sağlamaktadır. Programlama ile ilgili bilgisi olan küçük yaş grubu öğrencilerinin hazırbulunuşluk düzeyine bakıldığında, lise öğrencilerinin ilkokul- ortaokul ve üniversite öğrencileri arasında geçiş süreci oluşmasından dolayı, özyeterlilik çalışması yapılması gerekmektedir. Bundan dolayı, "Lise düzeyi öğrencilerine yönelik programlama öz yeterliliği ölçeği" geçerlik ve güvenirlik çalışmasının gerçekleştirilmesi gerekliliği görülmüştür."

Ölçeğin geliştirilmesi ile ilgili tüm analizler yapılmış ve sonuç olarak, ölçeğin güvenilirliği ve geçerliğini sağlandığı görülmüştür.

Performans Değerlendirme Formu

Araştırmada hem deney hem kontrol grubu öğretim programı uygulanması esnasında ders sonu değerlendirme amacıyla kullanılmak üzere hazırlanmış performans değerlendirme formları, her öğrencinin ortaya çıkardığı ürünlere ve elde edilen kazanımlara yönelik puanlar içermektedir. Deney ve kontrol grupları için ayrı ayrı kazanımlardan oluşan 29 maddelik performans değerlendirme formları, üç alan uzmanı tarafından incelenerek ölçme değerlendirme amacıyla kullanıma uygun bulunmuştur.

Deney ve kontrol grupları için 29' ar maddeden oluşan performans değerlendirme formları kazanım ifadelerini içermektedir. Her dersin sonunda ders öğretmeni tarafından doldurularak her grubun öğrencileri için 10 haftalık derslerin sonunda performans değerlendirme puanları elde edilmektedir. Bu şekilde hem deney hem kontrol grubundaki öğrencilerin performans değerlendirmeleri hesaplanmıştır.

Öğrenci Günlüğü

Araştırmada deney ve kontrol grubu için her dersin sonunda öğrencilerin görüşlerini almaya yönelik günlük formu oluşturulmuştur. Formu dolduracak öğrenciler, deney grubu öğrencileri 10-E şubesi (Android-Mobil programlama), kontrol grubu öğrencileri 10-D şubesi (Arduino- Robotik kodlama) olarak belirlenmiştir. Bu doğrultuda, her grubun öğrencileri tarafından her hafta derste öğrendiklerine yönelik fikir ve duygularını günlüğe aktarılmaları sağlanmıştır. Google Formlarda oluşturulmuş olan öğrenci günlüğü formu, günlük adı (ders konusu), öğrenci adı soyadı, numarası, sınıfı, düşünceleri ve tarihini içermektedir. MEB'nin bilgi paylaşım platformu olarak belirlenen Eğitimde Bilişim Ağı (EBA) platformunda ders öğretmeni tarafından daha önce paylaşılan öğrenci günlüğü formunun linki deney ve kontrol grubu öğrencilerine ulaştırılmıştır. Her dersin sonunda öğrencilerin bu linke tıklayarak günlüğü doldurmaları sağlanmıştır.

Odak Grup Görüşme

Araştırmada deney ve kontrol grupları ile ayrı ayrı odak grup görüşmesi yapılmıştır. 10 haftalık ve 20 saatlik aldıkları öğretim programları hakkında daha detaylı bilgiye sahip olmak için, her iki gruptaki öğrencilerden odak grup görüşme yöntemiyle fikirleri alınmıştır. Ekiz (2003) göre odak grup görüşmelerinin amacı, belirlenmiş bir konu ile ilgili katılımcıların görüşlerine, yaşanmışlıklarına, tecrübelerine, ilgilerine, eğilimlerine, algılarına, tutum ve alışkanlıklarına ve duygularına dair detaylı ve çok yönlü nitel bilgi edinmektir. Bu doğrultuda, her odak grup için öğrenci sayısı 6 olarak belirlenmiştir.

Veri Toplama Süreci

Proje Hazırlama dersinde 10 haftalık (20 saatlik) uygulama sürecine yönelik hazırlanmış kontrol ve deney gruplarının etkinlik planları doğrultusunda, eğitim öğretim haftasının 3. Hafta- 13. Hafta aralığında kontrol ve deney gruplarına deneysel çalışma yapılmıştır. Bu çalışmada ortaöğretim öğrencilerinin Arduino ile robotik kodlama ve Android ile mobil programlama uygulamalarını yaparak öğrenmeleri amaçlanmıştır. Her iki grup öğrencilerine her hafta bitiminde günlük doldurmaları istenmiştir.

Araştırma esnasında geliştirilen Programlama Öz yeterlilik ölçeği (Cesur Özkara & Yanpar Yelken, 2020) ön test olarak öğretim tasarımlarına başlamadan önce öğrencilere uygulanmıştır. 10 haftalık eğitim sürecinin tamamlanmasından sonra son test olarak programlama öz yeterliliği ölçeği aynı şekilde öğrencilere uygulanmıştır. Ayrıca, uygulamaların tamamlanması sonucunda ders öğretmeni kontrolünde her öğrenci tarafından performans değerlendirme formu doldurulmuştur. Doldurulan form sonucunda öğrencilerin uygulamalara yönelik performansları görülmektedirler.

Ayrıca öğrenci günlüklerinin yanı sıra odak grup görüşme sonucunda her iki grubun görüşleri alınmıştır. Böylece, nitel veri olarak, öğrenci günlükleri ve odak grup görüşme sorularının yanıtları elde edilmiştir.

Araştırmada, deneysel çalışmanın gerçekleştirilmesi için, öncelikli olarak gruplar belirlendikten sonra, 10 hafta – 20 saatlik deneysel işlem uygulama takvimi oluşturulmuştur. Bu takvim doğrultusunda deneysel çalışma gerçekleştirilmiştir. 10 haftalık uygulama süreci içerisinde, deney grubu Android-mobil programlama ile ilgili etkinlikler, kontrol grubu robotik kodlama ile ilgili etkinlikler öğrenciler tarafından yapılmıştır.

Uygulama sürecinde, ders anlatımına geçmeden önce hem kontrol hem de deney grubuna, programlama kavramı ve Arduino ile Android programlama ile ilgili sahip oldukları tüm bilgiler sorulmuştur. Bu bilgiler doğrultusunda, öğrencilerin çoğunun robotik kodlama ve mobil programlama ile ilgili pek bilgiye sahip olmadıkları görülmüştür. Öğrencilerin hazır bulunuşluk bilgilerinin belirlenmesi sonucunda, 10 haftalık eğitimler başlamıştır.

Öğrencilerin hazır bulunuşluk düzeyleri belirlenirken, bir ders saati içerisinde her iki gruba da aynı uygulamalar yaptırılmıştır. Bu uygulamalar tamamen algoritma kavramını öğretmeyi içermektedir. Algoritmanın tanımı, algoritma şeması oluşturma ve algoritmik sembollerin hepsi iki veya üç örnek ile soru cevap yöntemiyle öğrencilerle birlikte çözülmüştür.

Algoritma ile ilgili her iki gruptaki öğrencilere ödev verilmiştir ve ödevi dersin ilk haftasında öğrencilerden kâğıda algoritma şemasının hazırlanmış şekilde getirmeleri istenmiştir. Aşağıdaki başlıklarda deneysel çalışmanın eğitim öğretim sürecinin kontrol ve deney gruplarının haftalık olarak detaylı anlatımları yer almaktadır.

Verilerin Analizi

Araştırmada ortaya çıkan nicel ve nitel verilerin analizleri aşağıda ayrıntılı bir şekilde belirtilmiştir.

Nicel Verilerin Analizi

Ortaöğretim öğrencileri ile yapılmış olan deneysel araştırmada, programlama öz-yeterlilik ölçeğinin (Ö-T) ve (S-T) için kullanılması uygun görülmüştür. Deney(D) ve kontrol(K) gruplarına ölçeğin ayrı ayrı uygulaması yapılmıştır ve elde edilen verilerin dağılımının normalliğine bakılmıştır. Normallik için her grubun örneklem sayısının 33 olmasından dolayı Shapiro-Wilks normallik testinden faydalanılmıştır. Shapiro-Wilks testlerinde ortaya çıkan $p>0,05$ değerinden daha büyük bir sayısal veriye sahip olması ile birlikte verilerin normal dağılıma uygun olduğu görülmektedir (Kalaycı, 2014). Dağılımın normalliği testi sonuçları Tablo 3.'te verilmiştir.

Tablo 3

Normallik Testi Sonuçları

Değişkenler	Gruplar	Shapiro-Wilk		
		İstatistik	sd	p
Programlama Öz yeterliliği Ölçeği Ön test	DG	0,939	33	0,630
	KG	0,967	33	0,398
Programlama Öz yeterliliği Ölçeği Son test	DG	0,974	33	0,583
	KG	0,980	33	0,790
Programlama Öz yeterliliği Ölçeği	Pilot Çalışma (DG ve KG)	0,993	367	0,073
Performans Değerlendirme	DG	0,974	33	0,583
	KG	0,967	33	0,398

Tablo 3. incelendiğinde, araştırmada kullanılan programlama öz yeterliliği ölçeğinin ön-test ve son-testi ile performans değerlendirme testinin toplam puanlarının ve ölçeğin tüm alt boyutlarının normal dağılım şeklinde görüldüğü ($p<0,05$) belirlenmiştir. Bu durumda araştırmada iki farklı örneklem grubu puan ortalamalarının karşılaştırılması için bağımsız gruplar t-testi (Kalaycı, 2014) uygulanmıştır. Araştırmada analizi yapılan nicel veriler SPSS 24.0 paket programında çözümlenmiştir ve çalışmanın tüm verileri 0,95 puanında ($p<0,05$) olarak test edilmiştir.

Nitel Verilerin Analizi

Öğrencilerden her dersin sonunda Arduino ve Android uygulamaları hakkında görüşlerini almak için öğrenci günlüklerini yazmaları istenmiştir. Ayrıca, 10 haftalık etkinlik uygulaması hakkında öğrencilerin görüşlerinin alınması için odak grup görüşmeler yapılmıştır. Her haftanın sonunda, iki gruptaki öğrencilerden günlüklerin yazılması istenmiştir. Odak grup görüşmelerde ise, iki gruptaki 6'şar kişilik öğrencilerin ayrı ayrı bir araya getirilerek görüşme sorularının sorulması, ses kayıtlarının alınması sağlanmış ve daha sonra bu kayıtlar, metin haline dönüştürülmüştür. Yapılmış olan tüm çalışmalar sonucunda belirlenen veriler, içerik analizi ile değerlendirilmiştir. İçerik analizinin yapılması ile ortaya çıkan verilerin her biri için farklı uzmanlardan analiz yapılması istenmiştir. Uzmanlardan gelen içerik analizleri sonucunda, güvenilirlik formülü kullanılmıştır (Güvenirlik = Görüş Birliği / (Görüş Birliği + Görüş Ayrılığı)) (Miles ve Huberman, 1994). Aşağıda bulunan Tablo 4.' te nitel içerik analizi yapılan nitel verilerin temaları, alt temaları ile Miles ve Huberman katsayı değerleri verilmiştir.

Tablo 4

Nitel Veri Analizi Sonuçları

Nitel Veri İçerikleri		Miles ve Huberman Güvenirlik Katsayısı
1	Öğrenci Günlüğü (Arduino)	
	-Uygulamalar hakkında bilgi	0,85
	-Ders hakkında görüşler	0,75
	-Ders ile ilgili tutum	0,83
2	Öğrenci Günlüğü (Android)	
	-Uygulamalar hakkında bilgi	0,92
	-Ders hakkında görüş	0,75
	-Ders ile ilgili tutum	0,75
3	Odak Grup Görüşme Formu (Arduino)	
	-Programlamaya karşı tutum	0,88
	-Dersin kazanımları	0,81
	-Dersin olumlu yönleri	0,85
	-Dersin olumsuz yönleri	0,85
	-Ders için öneriler	0,80
4	Odak Grup Görüşme Formu (Android)	
	-Programlamaya karşı tutum	0,80
	-Dersin kazanımları	0,80
	-Dersin olumlu yönleri	0,85
	-Dersin olumsuz yönleri	0,83
	-Ders için öneriler	0,80

Tablo 4. incelendiğinde, içerik analizinde elde edilen tema ve alt temaların güvenilirlik sonucuna bakıldığında, öğrenci günlüğü güvenilirliği %81, odak grup görüşme kayıtları güvenilirliği %83 olduğu görülmektedir. Arduino öğrenci günlüğü temalarının güvenilirliği %81, Android öğrenci günlüğü temalarının güvenilirliği %81, Arduino odak grup görüşme temalarının güvenilirliği %84 ve Android odak grup görüşme temalarının güvenilirliği %82 olarak hesaplanmıştır.

BULGULAR

Araştırmanın alt problemlere ilişkin elde edilen istatistiki analizler ve bulgular incelenmektedir.

Lise Öğrencilerine Yönelik Programlama Öz Yeterliliği Ölçeği ile ilgili Bulgular

Araştırmada geliştirilmiş olan Programlama Öz yeterliliği Ölçeği ortaöğretim öğrencilerine ön test olarak 10 haftalık eğitimin başında ve son test olarak da 10 haftalık eğitimin sonunda uygulanmıştır. Çalışma grubu, Arduino robotik kodlama eğitimi alan kontrol grubu, Android mobil programlama eğitimi alan deney grubu olarak iki ayrı gruptan oluşmaktadır. Birinci alt probleme göre, deney(D) ve kontrol(K) gruplarının programlama erişimi (son programlama yeterlilikleri-ön programlama yeterlilikleri) sonuçlarında anlamlı bir farkın olup olmadığı incelenmiştir. (D) ve (K) gruplarının programlama erişimi (S-T programlama yeterlilikleri-Ö-T programlama yeterlilikleri) puanlarında anlamlı bir farkın olup olmadığına ilişkin verilerin analizinde bağımsız t-testi uygulanmıştır. 2018-2019 eğitim öğretim yılı güz döneminin başlangıcında ve sonucunda (D) ve (K) gruplarında yer alan toplam 66 öğrenene ön test(Ö-T) – son test(S-T) uygulanmıştır. Aşağıda sıra ile, gruplarının programlama erişimi puan ortalamalarının karşılaştırılması yapılmıştır.

Bu aşamada yapılan ön teste göre gruplara bakılmıştır. Programlama öz yeterliliği ölçeğinin Ö-T puan ortalamasının gruplarda anlamlı bir farkı gösterip göstermediğini ortaya çıkarmak amacıyla yapılan t-testi sonuçları Tablo 5 'te verilmiştir.

Tablo 5

Deney ve Kontrol Grupları Programlama Öz yeterliliği ölçeği Ön teste Ait T -Testi Sonuç Tablosu

Gruplar	N	\bar{X}	S	sd	t	p
KG	33	3,35	,51121	63	1,262	0,212
DG	33	3,17	,59289			

$p < 0,05$

Proje Hazırlama dersinde yapılan uygulamalar sonucunda, deney grubu öğrencilerinin Programlama Öz yeterliliği ölçeği aritmetik puan ortalamaları ($\bar{X} = 3,17$) olarak hesaplanırken, kontrol grubu öğrencilerinin Programlama Öz yeterliliği ölçeği aritmetik puan ortalamaları ($\bar{X} = 3,35$) olarak hesaplanmıştır. Tablo 5. incelendiğinde, kontrol grubundaki 33 öğrencinin ortalaması $\bar{X} = 3,35$ ve deney grubundaki 33 öğrencinin ortalaması $\bar{X} = 3,17$ olarak sonuç belirlenmiştir. Tablo 5' teki bağımsız gruplarda t testi sonuçlarına göre grupların Programlama Öz yeterliliği ölçeğinden aldıkları puan ortalamaları arasında %5' lik anlamlılık düzeyinde bir fark görülmemektedir, $p = 0,21$, $p > 0,05$. Grupların ortalamalarında görülen farkın anlamlı çıkmaması (D) ve (K) gruplarının Ö-T puan ortalamalarına göre birbiriyle eş-değer olduğu şeklinde açıklanabilir. Her iki grubunda programlama öz yeterliliğinin benzer düzeyde olduğu söylenebilir.

Öğrencilere uygulanan programlama öz yeterliliği ölçeği son testine göre grupların puan ortalamaları arasındaki ilişkiye bakılmıştır. Son test puan ortalamalarının gruplarda anlamlı bir farkı gösterip göstermediğini belirlemek amacıyla yapılan t-testi sonuçları Tablo 6 'da verilmiştir.

Tablo 6

Deney ve Kontrol Grupları Programlama Öz yeterliliği ölçeğinin Son Teste Ait T -Testi Sonuçları

Gruplar	N	\bar{X}	S	sd	t	p
KG	33	3,14	,45243	64	-0,325	0,746
DG	33	3,18	,55190			

$p < 0,05$.

Proje Hazırlama dersinde gerçekleştirilen uygulamalar sonucunda, kontrol(K) grubu öğrencilerine uygulanan programlama öz yeterliliği ölçeğinin son test aritmetik puan ortalamaları ($\bar{X} = 3,14$) şeklinde hesaplanırken, deney(D) grubu öğrencilerine uygulanan programlama öz yeterliliği ölçeğinin son test aritmetik puan ortalamaları ($\bar{X} = 3,18$) şeklinde hesaplanmıştır. Tablo 6' da görüldüğü üzere bağımsız gruplarda t testi sonuçlarına göre %5'lik anlamlılık seviyesinde grupların Programlama Öz yeterliliği ölçeğinden aldıkları puan ortalamalarında anlamlı bir fark görülmektedir, ($p = 0,75$, $p > 0,05$). Kontrol grubundaki 33 öğrencinin ortalaması $\bar{X} = 3,14$ ve deney grubundaki 33 öğrencinin ortalaması $\bar{X} = 3,18$ dir. Grupların aldıkları 10 haftalık

programlama derslerinin sonucunda, gruplarda programlama öz yeterliliklerinin eş değer düzeyde olduğu söylenebilir.

10 Haftalık verilen programlama derslerinin sonunda, kontrol ve deney gruplarına son test olarak uygulanmış olan programlama öz yeterliliği ölçeğinin analizleri sonucunda, kontrol grubunun programlama öz yeterliliği ile deney grubunun programlama öz yeterliliği ile eş değer olduğu görülmüştür. Bunun yanında, son test olarak gruplara uygulanan programlama öz yeterliliği ölçeğinin analizleri sonucunda, gruplarda programlama öz yeterliliğinde anlamlı farklılık olmadığı görülmüştür. Sonuç olarak, 10 haftalık programlama dersinin kontrol ve deney grupları öz yeterlilik düzeyleri üzerindeki etkisinde anlamlı bir fark olmadığı görülmüştür.

Grupların erişim puanları arasındaki anlamlı farka bakılmıştır. “Android programlamanın” uygulandığı (D) grubu öğrencileriyle, “robotik-Arduino programlamanın” uygulandığı (K) grubu öğrencilerinin programlama öz yeterliliği toplam erişim puanları hesaplanmıştır. Bu puanlar ile iki grubun aritmetik ortalamaları ve standart sapmaları bulunmuştur. Erişim puan ortalamalarının gruplar arasında anlamlı bir fark gösterip göstermediğini saptamak gerekçesiyle yapılan t-testi sonuçları Tablo 7’de verilmiştir.

Tablo 7

D ve K Gruplarının Erişim Puanlarının T testi Sonuç Tablosu

Gruplar	N	Erişim \bar{X}	S	Sd	t	p
DG	33	0,0885	0,63402	64	-2,686	0,009
KG	33	-0,3294	0,62997			

Tablo 7. göz önüne alındığında, Proje Hazırlama dersinde, robotik kodlamanın, Android programlama becerisinde anlamlı bir etkisini gösterip göstermediğini ortaya çıkaran yapılan ilişkisiz örneklem için t testinde, dersteki robotik kodlama (K) grubunun öğrencilerinin S-T ve Ö-T erişim puan ortalaması ile ($\bar{X}_K = 0,33$), Android programlama (D) grubu öğrencilerinin S-T ve Ö-T erişim puan ortalaması ($\bar{X}_D = 0,09$) arasında anlamlı bir fark görülmüştür [$t(64) = 0,009, p > 0,05$]. Bu bağlamda grupların erişim puan ortalamaları arasında anlamlı bir fark olduğu tespit edilmiştir.

Performans Değerlendirme Formu ile İlgili Bulgular

Proje Hazırlama dersinde yapılan uygulamalar sonucunda, “(D) ve (K) gruplarının uygulamadan sonraki performans değerlendirme puan ortalamaları arasında anlamlı bir farklılık var mıdır?” sorusuna cevap olarak gruplarının performans değerlendirme puan ortalamalarının karşılaştırılması yapılmıştır ve grupların performans değerlendirme puan ortalamaları arasındaki ilişkiyi ortaya çıkaran analizler ve sonuçları aşağıda belirtilmektedir.

Proje Hazırlama dersinde yapılan 10 haftalık programlama etkinliklerinin sonucunda, öğrencilerin performans durumlarını tespit etmek için deney ve kontrol grubuna performans değerlendirmesi yapılmıştır. Önceden hazırlanmış olan performans değerlendirme formu, kontrol ve deney grubundaki öğrencilere etkinlikler sonucunda puan verilerek oluşturulur. Bu bölümde, kontrol ve deney grubunun performans değerlendirme puan ortalamaları arasındaki ilişkiye bakılmıştır. Performans değerlendirme puan ortalamalarının gruplarda anlamlı bir farkın oluşup oluşmadığını belirlemek amacıyla yapılan t-testi sonuçları Tablo 8’de verilmiştir.

Tablo 8

Kontrol ve Deney Grubu Performans Değerlendirme puan Ortalamalarına İlişkin T Testi Sonuç Tablosu

Gruplar	N	\bar{X}	S	sd	t	p
(K)	33	89,0909	6,78401	64	-0,761	0,449
(D)	33	90,4091	7,27093			

$p=0,00, p < 0,05$.

T-testi sonuç tablosunun incelenmesi sonucunda, Proje Hazırlama dersinde, robotik kodlama konusunda etkinlik yapan kontrol grubu ile Android programlama konusunda etkinlik yapan deney grubu arasında performans değerlendirmesinde anlamlı bir farkın var olup olmadığını görmek için ilişkisiz örneklem için t testi yapılmıştır. Derste robotik kodlama kontrol grubunun öğrencilerinin performans değerlendirme puan ortalaması ile ($\bar{X} K = 89,09$), Android programlama deney grubu öğrencilerinin performans değerlendirme puan ortalaması ($\bar{X} D = 90,41$) arasında anlamlı bir fark bulunmamıştır [$t(64) = 0,449, p > 0,05$]. Bu şekilde, grupların performans değerlendirme puan ortalamaları incelendiğinde anlamlı bir fark olmadığı görülmüştür. 10 haftalık programlama dersi sonucunda, grupların etkinliklerden öğrendikleri doğrultusunda programlama bilgilerini artırdıkları söylenebilir.

Öğrencilerin Android Programlama ve Robotik Programlamanın Öğretme-Öğrenme Sürecine Yönelik Görüşleri ile ilgili Bulgular

Bu kısımda, araştırmanın üçüncü alt problemi olan öğrencilerin Android programlama ve Robotik programlamanın öğretme-öğrenme sürecine yönelik görüşleri hakkında yapılan analizler ve sonuçları yer almaktadır. Öğrenci günlüğü ile odak grup görüşme kayıtlarının analizi yapılmıştır. Elde edilen veriler sonucunda kategoriler elde edilmiştir ve bu doğrultuda ortaya çıkan tema ve alt temalar aşağıda belirtilmektedir.

Öğrenci Günlükleri Üzerine Kontrol ve Deney Gruplarının Görüşleri

Bu bölümde hem kontrol hem deney grubu öğrencileri, programlama eğitim-öğretim sürecindeki görüşlerini her dersin bitiminde günlüklere yazmışlardır. Yapılan içerik analizi sonucunda, robotik (Arduino) kodlama ile ilgili öğrenci günlükleri üzerine görüşler üç ana temadan oluşmaktadır. Her ana temanın alt temaları frekans değerlerine göre Tablo 9’da verilmiştir. Her iki grubunda ana temaları ‘uygulamalar hakkında bilgiler’, ‘ders hakkında görüşler’ ve ‘ders ile ilgili tutumlar’ olarak belirlenmiştir.

Tablo 9

Öğrenci Günlükleri Üzerine Kontrol Grubu Öğrencilerinin Görüşleri

Robotik (Arduino) Kodlama ile ilgili Öğrenci Günlükleri Üzerine Görüşleri	f
Ders ile ilgili tutumlar	81
Dersin güzel olması	24
Dersin verimli olması	18
Uygulamaların eğlenceli olması	15
Robot proje uygulamasının güzel olması	10
Dersin çok verimli geçmesi	9
Dersin biraz sıkıcı olması	5
Uygulamalar hakkında bilgiler	64
Led yakılması	17
Trafik lambası yapılması	17
Potansiyometre ile LED uygulaması yapılması	9
Kapı zili uygulaması yapılması	7
LDR uygulaması yapılması	7
Gece lambası yapılması	7
Ders hakkında görüşler	33
Arduino ile uygulamalar yapma	27
Arduino ve çeşitlerini öğrenme	6

Tablo 9. incelendiğinde, analizler sonucunda elde edilen veriler doğrultusunda, “derse karşı tutumlar” temasının frekans değerinin ($f=81$) daha yüksek olduğu görülmektedir. Bu temanın alt temalarına bakıldığında, kontrol grubu öğrencilerinin ders hakkında olumlu görüşlere sahip oldukları görülmektedir. Öğrencilere göre,

dersin güzel olması (f=24), dersin verimli olması (f=18), dersin eğlenceli olması (f=15), ders ile ilgili olarak genellikle olumlu tutumlara sahip olduklarını göstermektedir.

Aynı şekilde mobil (Android) programlama ile ilgili öğrenci günlükleri üzerine görüşler üç ana temadan oluşmaktadır ve her ana temanın alt temaları frekans değerlerine göre Tablo 10' da verilmiştir. Her iki grubunda ana temaları 'uygulamalar hakkında bilgiler', 'ders hakkında görüşler' ve 'ders ile ilgili tutumlar' olarak belirlenmiştir.

Tablo 10

Öğrenci Günlükleri Üzerine Deney Grubu Öğrencilerinin Görüşleri

Mobil (Android) programlama ile ilgili Öğrenci Günlükleri Üzerine Görüşleri	f
Uygulamalar hakkında bilgiler	297
Frame layout kavramının öğrenilmesi	34
Textview kullanımının öğrenilmesi	34
Linear layout kavramının öğrenilmesi	29
SeekBar kullanımının öğrenilmesi	29
Progressbar kullanımının öğrenilmesi	27
Tablo 10 (Devam)	
Uygulamalar hakkında bilgiler	f
Edittext kullanımının öğrenilmesi	25
Grid layout kavramının öğrenilmesi	23
Screen orientation kavramının öğrenilmesi	21
ImageView kullanımının öğrenilmesi	21
Radiobutton kullanımının öğrenilmesi	18
WebView kullanımının öğrenilmesi	15
Checkbox kullanımının öğrenilmesi	14
Appinventor uygulamasının yapılması	7
Ders ile ilgili tutumlar	80
Dersin güzel geçmesi	33
Dersin eğlenceli geçmesi	27
Konuların anlaşılması	12
Dersin biraz zor gelmesi	8
Ders hakkında görüşler	50
Android studio programının kullanılması	20
Konuların biraz karışık olması	17
Ders öğretmenin gönderdiği iletilerinin incelenmesi	13

Tablo 10. incelendiğinde, analizler sonucunda elde edilen veriler doğrultusunda, “uygulamalar hakkında bilgiler” temasının frekans değerinin (f=297) daha yüksek olduğu görülmektedir. Bu temanın alt temalarına bakıldığında, deney grubu öğrencilerinin derste yaptıkları uygulamaların tümü hakkında bilgiye sahip oldukları görülmektedir. Frame layout kavramının ve textview kullanımının (f=34), linear layout kavramının ve seekbar kullanımının (f=29), progressbar kullanımının (f=27), edittext kullanımının (f=25), grid layout kavramının (f=23), Screen orientation kavramının ve ImageView kullanımının (f=21), radiobutton kullanımının (f=18), WebView kullanımının (f=15), checkbox kullanımının (f=14) öğrenilmesi ve Appinventor uygulamasının yapılması (f=7) derste yapılan ve öğrencilerin en çok ilgili gösterdikleri uygulamalar olduğu görülmektedir.

Odak Grup Görüşme Kayıtları Üzerine Kontrol ve Deney Gruplarının Görüşleri

Bu bölümde hem kontrol hem deney grubu öğrencileri, programlama eğitim-öğretim sürecindeki görüşlerini, 10 haftalık derslerin bitiminde önceden oluşturulmuş olan 6'şar kişilik gruplarla odak grup

görüşme yapılmıştır. Hem deney hem kontrol grubu öğrencileriyle yapılan görüşmeler ses kaydına alınmıştır. Yapılan içerik analizi sonucunda, robotik (Arduino) kodlama ders süreci ile ilgili görüşler beş ana temadan oluşmaktadır. Her ana temanın alt temaları frekans değerlerine göre Tablo 11’ de verilmiştir. Her iki grubunda ana temaları ‘ders öncesi programlamaya karşı tutum’, ‘dersin kazandırdıkları’, ‘ders hakkında olumlu görüşler’, ‘ders hakkında olumsuz görüşler’ ve ‘ders için öneriler’ olarak belirlenmiştir.

Tablo 11*Odak Grup Görüşme Kayıtları Üzerine Kontrol Grubu Öğrencilerinin Görüşleri*

Robotik (Arduino) programlama ile ilgili Öğrenci Günlükleri Üzerine Görüşleri	f
Ders öncesi programlamaya karşı tutum	29
Programlama hakkında bilgiye sahip olmama	12
Programlamayı sevme	11
İlgi çekici olması	2
Daha önce oyun yapılması	2
Hiç programlama dersi alınmaması	2
Tablo 11 (Devam)	
Dersin kazandırdıkları	32
Programlamanın öğrenilmesi	13
Kod yazmanın öğrenilmesi	11
Programlamanın zor olduğunun fark edilmesi	4
Kod yazmada zorlanması	2
Kodlamaya yönelik temel kavramların öğrenilmesi	2
Dersin olumlu yönleri	26
İyi düşünülmesi	20
Çok güzel bir ders olması	6
Dersin olumsuz yönleri	14
Sınıfın çok kalabalık olması	8
Kılavuz olmasının daha iyi olması	6
Ders için Öneriler	15
Malzemelerin eksiksiz olması	4
Az kişili grupların olması	4
Başka bir sınıf ortamı yaratılması	4
Sınıfın ikiye ayrılması	3

Tablo 11. incelendiğinde, analizler sonucunda elde edilen veriler doğrultusunda, “ders öncesi programlamaya karşı tutum” temasının frekans değerinin (f=29) olduğu görülmektedir. 10 haftalık eğitim öğretim sürecinin başlamasından önce öğrencilerin programlamaya karşı tutumları, olumlu ve olumsuz görüşleri içermektedir. Sevilmesi (f=11), ilgi çekici olması (f=2) ve daha önce oyun yapılması (f=2) gibi olumlu ifadelerin yanı sıra olumsuz ifadeler de şu şekildedir; Programlamanın hiç görülmemesi (f=12) ve hiç programlama dersi alınmaması (f=2)’ dir.

Aynı şekilde mobil (Android) programlama ders süreci ile ilgili görüşler beş ana temadan oluşmaktadır ve her ana temanın alt temalarının frekans değerleri Tablo 12’ de verilmiştir. Her iki grubunda ana temaları ‘ders öncesi programlamaya karşı tutum’, ‘dersin kazandırdıkları’, ‘ders hakkında olumlu görüşler’, ‘ders hakkında olumsuz görüşler’ ve ‘ders için öneriler’ olarak belirlenmiştir.

Tablo 12*Odak Grup Görüşme Kayıtları Üzerine Deney Grubu Öğrencilerinin Görüşleri*

Mobil (Android) programlama ile ilgili Odak Grup Görüşme Kayıtları Üzerine Görüşleri	f
Ders öncesi programlamaya karşı tutum	51
Daha önce hiç görülmemiş olması	15
Hiç sevilmemiş olması	13
İlginin olmaması	5
İlgi çekici olmaması	4
İlgi çekici olması	3
Sıkıcı gelmesi	3
Materyallerin hiç kullanılmaması	3
Uygulamaların güzel olması	2
İlgilenilmesi ve sevilmesi	2
Sevip sevmediğini bilmemek	1
Dersin kazandırdıkları	65
Kodlamanın öğrenilmesi	31
Temel kavramların öğrenilmesi	4
Son uygulamanın daha kolay olması	4
Tüm kavramların öğrenilmesi	4
Çalışma kağıtları ile çalışılması	4
Mobil programlamanın zor olduğunun öğrenilmesi	3
Bir şey kazandırmaması	3
EBA platformunun aktif kullanılması	3
Apk yapılmasının öğrenilmesi	3
Ders hakkında olumlu görüşler	56
Algoritmanın öğrenilmesi	13
Güzel bir ders olması	11
Dersin eğlenceli olması	7
Programlama yapmanın öğrenilmesi	6
Dersten sonra daha çok bilgiye sahip olunması	5
Ders hakkında olumlu görüşler	14
Uygulama yapılmasının öğrenilmesi	5
Yeniliklerin öğrenilmesi	4
Son uygulama(proje) ile öğrenilmesi	3
Programlama hakkında bilgi sahibi olunması	2
Ders hakkında olumsuz görüşler	19
Sınıfın gürültülü olması	7
Dersin bir ders saati olması	4
Derse karşı ilgisiz olunması	4
Bilgisayar sayısının yetersiz olması	4

Tablo 12 (Devamı)

Ders için Öneriler	63
Bilgisayar sayısının artırılması	21
Uygulamaların kolaylaştırılması	13
Ders saatinin artırılması	8
Uygulamanın bilgisayarda yapılması	6
Uygulamanın derste yapılma imkanı	6
Anlatım tekniğinden sonra uygulama yapılması	6
Dersin daha eğlenceli hale getirilmesi	3

Tablo 12. incelendiğinde, analizler sonucunda elde edilen veriler doğrultusunda, “ders öncesi programlamaya karşı tutum” temasının frekans değerinin (f=51) olduğu görülmektedir. Tablo 12’ de öğrencilerin 10 haftalık eğitim öğretim sürecinin başlamasından önce programlamaya karşı tutumların, olumlu ve olumsuz görüşleri içerdiği görülmektedir. “Daha önce hiç görülmemiş olması (f=15)”, “Hiç sevilmemiş olması (f=13)”, “İlginin olmaması (f=5)”, “İlgi çekici olmaması (f=4)”, “sıkıcı gelmesi (f=3)” ve “materyallerin hiç kullanılmaması (f=3)” gibi olumsuz ifadelerin yanı sıra olumlu ifadeler de şu şekildedir; “İlgi çekici olması (f=3)”, “Uygulamaların güzel olması (f=2)” ve “İlgilenilmesi ve sevilmesi (f=2)” dir. Öğrencilerin 10 haftalık eğitim öğretim sürecinin başlamasından önce programlamaya karşı tutumlarının pek olumlu olmadığı frekans değerlerine bakılarak söylenebilir. 10 haftalık eğitim-öğretim süreci sonunda, öğrencilerin mobil programlamaya yönelik görüşlerinin daha olumlu olduğu frekans değerleri doğrultusunda görülmektedir.

TARTIŞMA VE SONUÇ

Deney ve Kontrol gruplarının programlama erişimi (son programlama yeterlilikleri-ön programlama yeterlilikleri) puanlarında anlamlı bir farkın var olup olmadığı incelenmiştir. Şu aşamada gerçekleştirilen ön teste göre gruplar karşılaştırılmıştır. Sonuç olarak, ön test verileri analizi sonucunda, grupların programlama öz yeterliliği ölçeğinden ortaya çıkan puan ortalamalarında anlamlı bir fark bulunmadığı görülmüştür. Grupların ön-test programlama öz yeterlilik verilerinin karşılaştırılması sonucunda alanyazında benzer bulgularda araştırmalar bulunduğu görülmüştür. Alanyazında anlamlı sonuç çıkan çalışmaların anlam farklılığı çıkan çalışmalardan daha az olduğu görülmektedir. Programlama öz yeterliliğinin öğrencilere uygulanan ön testlerin puanları arasında anlamlı bir fark olmaması, genellikle grupların eğitim düzeylerindeki eşit değerdeki bilgiye, yeterliliğe ve tutuma sahip olmalarından kaynaklanmaktadır.

Araştırmada, öğrencilere uygulanan programlama öz yeterliliği ölçeği son testine göre grupların puan ortalamaları arasındaki ilişkiye bakılmıştır. S-T puan ortalamalarının gruplar arasında anlamlı bir farkın var olup olmadığı incelenmiştir ve grupların Programlama Öz yeterliliği ölçeğinden aldıkları puan ortalamalarında anlamlı bir fark bulunmadığı görülmüştür. Bu doğrultuda literatürde benzer çalışmalara rastlanmıştır. Alanyazında anlamlı sonuç çıkan çalışmaların anlam farklılığı çıkan çalışmalardan daha az olduğu görülmektedir. Programlama öz yeterliliğinin öğrencilere uygulanan son testlerin puanları arasında anlamlı bir fark olmaması, genellikle grupların aldıkları eğitimlerden sonra eğitim düzeyleri sonucunda eşit değerdeki bilgiye, yeterliliğe ve tutuma sahip olmalarıdır.

Araştırmada, grupların programlama öz yeterliliğine yönelik erişimi puanları arasındaki anlamlı farkın sonucuna göre literatür incelendiğinde, Altın (2021)’ nin “Ortaokul öğrencilerinin kodlama ve bilgi işlemsel düşünme becerileri; programlama öğretiminde geleneksel ve disiplinler arası yaklaşım” isimli çalışmasında, farklı disiplinler ile ilişkilendirilmiş matematik ile kodlama öğretiminin hem öğrencilerin kodlama hem de bilgi işlemsel düşünme becerilerini artırdığını görülmektedir. Bununla birlikte, programlama eğitiminin akademik başarıya etkisinin olumlu olduğu ortaya çıkmıştır. Araştırmada, programlama öz yeterliliği ile ilgili olarak grupların erişimi puan ortalamaları arasında anlamlı bir fark olması, akademik başarıya etkisinin olduğunu göstermektedir. Ayrıca, programlama öğreniminin mantıksal yönü ile sayısal derslerde özellikle matematik dersinde öğrenciler üzerinde daha fazla olumlu etkisi olduğu söylenebilir.

Araştırmada, robotik kodlama kontrol(K) grubunun öğrencilerinin performans değerlendirme puan ortalaması ile, Android programlama deney(D) grubu öğrencilerinin performans değerlendirme puan ortalaması arasında anlamlı bir fark görülmemiştir. Literatür taraması sonucunda elde edilen sonuçlar doğrultusunda, öğrenenlerin programlamaya yönelik öz-yeterlik algısı puanlarının, programlama performanslarını etkilediği görülmektedir (Ramalingam, LaBelle & Wiedenbeck, 2004; Aşkar & Davenport, 2009; Davidson, Larzon & Ljunggren, 2010; Altun & Mazman, 2013).

Alagöz (2020)' nin yüksek lisans tezinde, 5 haftalık “Metaforlarla programlama öğreniyorum” etkinliği sonucunda öğrencilerin akademik başarılarında ve bilgi sayımsal düşünme becerilerinde anlamlı bir değişim olduğu gözlenmiştir. Şimşek (2019)' un çalışmasında, robotik kodlama etkinliklerinin yapıldığı (D) grubundan (K) grubuna göre, (D) ve (K) gruplarının Ö-T puanlarına göre düzeltilmiş olup S-T puanlarında anlamlı bir fark görülmüştür. Bilimsel süreç becerileri puanlarındaki değişimlerde ise anlamlı bir fark oluşturduğu sonucuna varılamamıştır.

Araştırmada verilen eğitimin sonunda, kontrol grubunun robotik kodlama ile deney grubunun mobil kodlama becerileri konusunda üretken ve yapılması gereken yazılımları başarı ile gerçekleştireceklerine inanan yetenekte bireyler yetiştirildiği söylenebilir. Robotik kodlama ve mobil programlama öğrenimi ile uygulamalarının tümü ortaöğretim öğrencilerinin derslerdeki performanslarını olumlu düzeyde etkilemiştir. Dolayısıyla, programlama öğrenimi ile öğrencilerin akademik başarıları da performanslarının artması ile artmıştır.

Araştırmada öğrenci günlükleri ve odak grup görüşmelerinden elde edilen sonuç kontrol grubu öğrenci günlükleri ile ilgili olarak literatür incelemesi yapıldığında, araştırmada, programlama öğreniminin ve uygulamalarının etkisi öğrenciler üzerinde 21. Yüzyıl yeteneklerinden problem çözme, eleştirel düşünme, yaratıcı düşünme ve analitik düşünme yeteneklerinin tümünü pekiştirmede çok büyük etkisinin olduğu görülmektedir. Bu olumlu etki sayesinde, öğrencilerin derslerdeki performansları artmış ve akademik başarılarına etkisinin diğer disiplinlere nitelikli faydasının olduğu belirgin bir şekilde gözlenmiştir.

(D) ve (K) gruplarının (Ö-T) puanlarının karşılaştırılması sonucunda, iki grubun ön test(Ö-T) puan ortalamalarının arasındaki farkın anlamlı olmaması, grupların programlama öz yeterliliği ile ilgili olarak eş değer düzeyde oldukları söylenebilir. Grupların aldıkları 10 haftalık programlama derslerinin sonucunda, gruplarda aldıkları ortalama puanlar eş değer düzeyde olduğu görülmektedir. Son test olarak gruplara uygulanan programlama öz yeterliliği ölçeğinin analizleri sonucunda, gruplarda programlama öz yeterliliğinde anlamlı farklılık olmadığı görülmüştür. Sonuç olarak, 10 haftalık programlama dersinin öğrenciler üzerindeki öz yeterlilik düzeyleri, başlangıca göre yüksek çıkmıştır. Eğitim sonunda, kontrol grubunun robotik kodlama ile deney grubunun mobil kodlama becerileri konusunda üretken ve yapılması gereken yazılımları başarı ile gerçekleştireceklerine inanan yetenekte bireyler yetiştirildiği söylenebilir.

Robotik kodlama (K) grubu öğrenenlerinin (S-T) ve (Ö-T) erişim puan ortalaması ile Android programlama (D) grubu öğrencilerinin (S-T) ve (Ö-T) erişim puan ortalamasında anlamlı bir fark görülmüştür. Böyle bir durumda grupların erişim puan ortalamalarında anlamlı bir farkın olduğu tespit edilmiştir. Bu sonuç doğrultusunda, eğitim alan öğrencilerin kodlamaya ilgilerinin arttığı ve programlamaya karşı meraklarının oluşması sonucunda ileriye yönelik çalışmalara yönelmek istedikleri görülmektedir.

Proje Hazırlama dersinde, robotik kodlama konusunda etkinlik yapan kontrol grubu ile Android programlama konusunda etkinlik yapan deney grubu arasındaki performans değerlendirmesinde anlamlı bir fark görülmemiştir.

Öğrenci günlüklerindeki görüşlerden elde edilen veriler doğrultusunda, “derse karşı tutumlar” temasının en yüksek frekans değerine sahip olduğu görülmektedir. Bu temanın alt temalarına bakıldığında, kontrol grubu öğrencilerinin ders hakkında olumlu görüşlere sahip oldukları görülmektedir. Görüşlerden elde edilen veriler doğrultusunda, “uygulamalar hakkında bilgiler” temasının en yüksek frekans değeri olduğu görülmektedir. Bu temanın alt temalarına bakıldığında, deney grubu öğrencilerinin derste yaptıkları uygulamaların tümü hakkında bilgiye sahip oldukları görülmektedir.

Kontrol grubu ve deney grubu odak grup görüşme kayıtlarından elde edilen veriler doğrultusunda, “ders öncesi programlamaya karşı tutum” temasının en yüksek frekans değeri olduğu görülmektedir. Öğrencilerin 10 haftalık eğitim öğretim sürecinin başlamasından önce programlamaya karşı tutumlar olumlu ve olumsuz görüşleri içermektedir. Sevilmesi, ilgi çekici olması ve daha önce oyun yapılması gibi olumlu ifadelerin yanı sıra olumsuz ifadeler olarak programlamanın hiç görülmemesi ve hiç programlama dersi alınmaması şeklindedir. Bu araştırmanın sonucunda, verilen eğitimler ile ortaöğretim öğrencilerinin programlamaya karşı ilgilerinin arttığı görülmektedir. Robotik ve mobil programlamaya yönelik uygulamalar sonucunda, öğrencilerin programlama öz yeterliliklerinin yüksek oranda olumlu düzeyde olduğu görülmüştür. Bununla birlikte, öğrencilerin geleceğe yönelik hedeflerinin şekillenmesine yardımcı olduğu sonucuna varılmıştır.

Araştırmada, öğrencilerin yeniliğe açık olmaları, güncel teknolojiye meraklı ve yenilikçi tüm bilgileri öğrenmeye yönelik davranış sergilemeleri her düzeydeki okulda programlama ya da bilişim teknolojileri alanının zorunlu ders haline getirilmesi gerekliliğini ortaya çıkarmıştır.

Ana sınıfı düzeyinden itibaren etkinliklerle çocuklara programlama eğitimlerinin oyunlaştırılarak verilmesi için öğretim planlamalarının yapılması gerekir. Çocukların geleceklerini isteklerine göre planlayarak yapmaları bu şekilde sağlanmalıdır. Programlama eğitimi, sadece bilişsel alan düzeyinde değil, duyuşsal alan düzeyinde de öğrencilere yön veren ve yaşantılarını yaratıcılık ve özgünlük üzerine ve kendilerini keşfeden bireyler olarak yetişmesini sağlayacak şekilde planlanmalıdır. Programlama içeren derslerin disiplinler arası bağı oluşturan bir disiplin olması söz konusudur. Bulduğumuz çağın koşulları doğrultusunda, programlama içeren dersin, temel derslerin içerisinde yer alması gerekmektedir. Millî Eğitim Bakanlığına bağlı tüm okullarında programlama ile yapılan ve yapılacak olan tüm öğrenci ve öğretmen etkinliklerinin EBA platformunda paylaşılması sağlanmalıdır. Etkinlikler için toplanan beğeniler doğrultusunda ödüllendirme yapılması programlamaya ilginin artmasını sağlayacaktır. Kodlama eğitimi ile öğrencilerden alınan görüş ve izlenimlere göre yeni bir eğitim modeli oluşturulması gereği öngörülmüştür. Ortaöğretim öğrencilerinin daha soyut öğretiler üzerinden öğrenimlerini gerçekleştirmelerinden dolayı hem bireysel hem iş birliği içerisinde yaşantılarını gerçekleştirebilecekleri ve akademik kaygı yaşamadan istedikleri kalitede yaşam sürdürmelerini sağlayacak eğitim politikalarının oluşturulması gerekmektedir.

Zorunlu ders olarak Bilişim Teknolojileri ve Yazılım dersinin ortaokul ve lise düzeyi okullarda verilmesi, akademik başarıya katkı sağlayacağından daha ileri düzeyde öğretim programlarının planlanması yerinde olacaktır. Yapay zeka ile makine öğrenme ve derin öğrenme kavramlarının lise düzeyindeki öğrencilere entegre edilecek şekilde öğrenmelerini sağlayacak yeni bir oluşumun gerçekleştirilmesi geleceğe büyük katkı sağlayacak eğitsel bir yatırım olacaktır.

Okullarda atölye çalışmalarının ders dışı egzersiz haricinde yapılabilmesi için farklı platformlarda (yarışma ya da projeler) yapılabilmesi sağlanmalıdır. Her okulun proje çalışması yapabilmesi için gerekli üst makamların okullara yeterli ödenek ayırabilmesi gerekmektedir.

KAYNAKÇA

- Akkoyunlu, B. & Kurbanoglu, S. (2003). Öğretmen Adaylarının Bilgi Okuryazarlığı ve Bilgisayar Özyeterlik Algıları Üzerine Bir Çalışma. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 24 (24), 1-10.
- Akpınar, Y. & Altun, A. (2014). Bilgi Toplumu Okullarında Programlama Eğitimi Gereksinimi. *İlköğretim Online*, 13 (1), 1-4.
- Alagöz, N. B. (2020). *Metaforlarla Programlama Öğreniyorum: Metaforların Programlama Öğretimi Üzerine Etkisi*, (Yüksek lisans tezi). Ondokuz Mayıs Üniversitesi, Samsun.
- Altın, R. (2021). *Secondary School Students' Programming And Computational Thinking Skills: Traditional And Interdisciplinary Approaches To Teaching Programming*, (Yüksek lisans tezi). Orta Doğu Teknik Üniversitesi, Ankara.
- Altun, A. & Mazman, S.G. (2012). Programlamaya İlişkin Öz Yeterlilik Algısı Ölçeğinin Türkçe Formunun Geçerlilik ve Güvenirlik Çalışması. *Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi*, 3 (2), 297- 308.
- Arabacıoğlu, T. (2006). *İnternet Destekli Programlama Mantığı Öğretimi*, (Yüksek lisans tezi). Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Arslan, B. & Gülnar, S. (2014). Java-Android Yazılım Mimarisi: Bir Masaüstü İle Çoklu Tablet Bilgisayar Haberleşme Uygulaması, Akademik Bilişim '14 - XVI. Akademik Bilişim Konferansı Bildirileri, Mersin Üniversitesi.
- Aşkar, P. & Umay, A. (2001). İlköğretim Matematik Öğretmenliği Öğrencilerinin Bilgisayarla İlgili Özyeterlilik Algısı. *Hacettepe Üniversitesi, Eğitim Fakültesi Dergisi*, 21 (21), 1-8.
- Aytekin, A., Çakır, F.S., Yücel, Y.B. & Kulaöz, İ. (2018). Algoritmaların Hayatımızdaki Yeri ve Önemi. *Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi (ASEAD)*, 5 (7), 151-162.
- Bandura, A. (1994). Self-Efficacy, In Ramachandran (Ed.), *The Encyclopedia Of Human Behavior* (pp. 71-81), Volume: 4, R-Z And Index, New York: Academic Press.
- Bozkurt, A. (2015). Mobil Öğrenme: Her Zaman, Her Yerde Kesintisiz Öğrenme Deneyimi. *Açık öğretim Uygulamaları ve Araştırmaları Dergisi*, 1 (2), 65-81.
- Bybee, R. W. (2010). *The Teaching Of Science: 21st Century Perspectives*. NSTA Press.
- Cesur Özkara, E. & Yanpar Yelken, T. (2020). Ortaöğretim Öğrencilerine Yönelik Programlama Öz Yeterlilik Ölçeğinin Geliştirilmesi: Geçerlilik ve Güvenirlik Çalışması. *Eğitim Teknolojisi Kuram ve Uygulama*, 10(2), 345-365. <https://doi.org/10.17943/etku.632606>
- Ceylan, V.K. (2020). *Senaryo Temelli Scratch Öğretim Programının Öğrencilerin Bilgi İşlemsel Düşünme Becerilerine, Problem Çözme ve Programlama Ünitesi Erişilerine Etkisi*, (Doktora Tezi). Aydın Adnan Menderes Üniversitesi, Aydın.
- Christian, B. & Griffiths, T. (2017). *Hayatımızdaki Algoritmalar*. 2. Baskı; Buzdağı Yayınevi: Ankara, s. 14-15.
- Creswell, J. W. & Clark, V. L. P. (2015). *Pesquisa de Métodos Mistos-: Série Métodos de Pesquisa*. Penso Editora.
- Çakıroğlu, Ü. & Kiliç, S. (2020). Assessing teachers' PCK to Teach Computational Thinking Via Robotic Programming. *Interactive Learning Environments*. 31(2), 818-835. <https://doi.org/10.1080/10494820.2020.1811734>
- Davidsson, K., Larzon, L. & Ljunggren, K. (2010). *Self-Efficacy In Programming Among STS Students*. <http://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports>. (Erişim Tarihi:25/08/2025).
- Demirer, V. & Sak, N. (2016). Programming Education and New Approaches Around The World And In Turkey. *Journal Of Theory And Practice In Education*, 12(3), 521-546.
- Eguchi, A. (2014). Educational Robotics For Promoting 21st Century Skills. *Journal of Automation Mobile Robotics and Intelligent Systems*, 8(1), 5-11. DOI: [10.14313/JAMRIS_1-2014/1](https://doi.org/10.14313/JAMRIS_1-2014/1)
- Ekiz, D. (2003). *Eğitimde Araştırma Yöntem ve Metotlarına Giriş*. Ankara: Anı Yayıncılık.
- Elçiçek, M. (2019). *Programlama Öğretimine Yönelik Video Destekli Çevrimiçi Bir Öğrenme Ortamının Tasarımı ve Değerlendirilmesi*, (Doktora Tezi). Trabzon Üniversitesi Lisansüstü Eğitim Enstitüsü, Trabzon.
- Estonya'da Bilgisayar (2012). *Estonya'da Bilgisayar Dili 1'inci Sınıfa Girdi*. *Hürriyet Gazetesi*. <http://www.hurriyet.com.tr/planet/21405007.asp> (Erişim Tarihi:25/08/2025).
- Gerják, I. (2017). Image Processing Algorithms In the Secondary School Programming Education, *Acta Didactica Napocensia*, 10(3), 69-76.

- Gomes, A. & Mendes, A. (2007). Learning To Program- Difficulties And Solutions, *Proceedings Of The International Conference On Engineering Education- ICEE*, 3- 7 Eylül 2007, Coimbra.
- Göksoy, S. & Yılmaz, İ. (2018). Bilişim Teknolojileri Öğretmenleri ve Öğrencilerinin Robotik ve Kodlama Dersine İlişkin Görüşleri. *Düzce Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 8(1), 178-196.
- Grafinger, D. J. (1988). Basics Of Instructional Systems Development. INFO-LINE Issue 8803. Alexandria: American Society for Training and Development.
- Gustafson, K.L. & Branch, R.M. (2002). *Survey Of Instructional Models*. New York: Eric Clearinghouse on Information & Technology.
- Gülbahar, Y. & Kalelioğlu, F. (2018). Bilişim Teknolojileri ve Bilgisayar Bilimi: Öğretim Programı Güncelleme Süreci. *Milli Eğitim dergisi*, 47 (217), 5-23.
- Günüç, S., Odabaşı, H. & Kuzu, A. (2013). 21. Yüzyıl Öğrenci Özelliklerinin Öğretmen Adayları Tarafından Tanımlanması: Bir Twitter Uygulaması / The Defining Characteristics of Students of The 21st Century By Student Teachers: A Twitter Activity. *Eğitimde Kuram ve Uygulama*, 9 (4), 436-455.
- Johnson, R. B. & Onwuegbuzie, A J. (2004). Mixed Method Research. A Research Paradigm Whose Time Has Come. *Educational Researcher*, 33(7), 14-26. DOI:[10.3102/0013189X033007014](https://doi.org/10.3102/0013189X033007014)
- Kalaycı, Ş. (2014). *SPSS Uygulamalı Çok Değişkenli İstatistik Teknikleri* (6. Baskı). Ankara: Asil Yayın Dağıtım.
- Kalelioğlu, F. (2015). A New Way Of Teaching Programming Skills To K-12 Students: Code.org. *Computers in Human Behavior*. 52, 200-210. <https://doi.org/10.1016/j.chb.2015.05.047>
- Karabak, D. & Güneş, A. (2013). Ortaokul Birinci Sınıf Öğrencileri İçin Yazılım Geliştirme Alanında Müfredat Önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(3), 21.
- Kert, S. B. & Uğraş, T. (2009). Programlama Eğitiminde Sadelik ve Eğlence: Scratch Örneği. In *The First International Congress of Educational Research, Çanakkale, Turkey*.
- Kılınç, A. (2014). *Robotik Teknolojisinin 7. Sınıf Işık Ünitesi Öğretiminde Kullanımı*, (Yüksek lisans tezi). Erciyes Üniversitesi, Kayseri.
- Konvalina, J., Stephens, L. & Wileman, S. (1983). Identifying Factors Influencing Computer Science Aptitude And Achievement. *AEDS I*, 16 (2), 106-112.
- Korkmaz, Ö. (2013). Prospective CITE Teachers' Self-Efficacy Perceptions On Programming. *Procedia-Social and Behavioral Sciences*, 83, 639-643. <https://doi.org/10.1016/j.sbspro.2013.06.121>
- Köse, U. & Tüfekçi, A. (2015). Algoritma ve Akış Şeması Kavramlarının Öğretiminde Akıllı Bir Yazılım Sistemi Kullanımı. *Pegem Eğitim ve Öğretim Dergisi*, 5(5), 569-586.
- Kurşuncu E. (2016). *İki Tekerlekli Dengede Durabilen Robotlarda Kontrol Metotları*, (Yayınlanmamış yüksek lisans tezi). Karabük Üniversitesi, Karabük.
- Maja J Matari'c (1990). *A Distributed Model For Mobile Robot Environment-Learning And Navigation*, (Yüksek lisans tezi). Massachusetts Institue of Technology, Cambridge, MA.
- Mayring, P. (2007). Introduction: Arguments For Mixed Methodology. In P. Mayring, G. L. Huber, L. Gurtler, M. Kiegelmann (Eds.), *Mixed Methodology In Psychological Research*, Brill Sense, 1-4. https://doi.org/10.1163/9789087903503_002
- MEB. (2019). *Bilişimle Üretim Öğretim Programı*. Millî Eğitim Bakanlığı. <http://www.meb.gov.tr> . (Erişim Tarihi:25/08/2025).
- MEB-TTKB (Millî Eğitim Bakanlığı Talim Terbiye Kurulu Başkanlığı), (2012). *Bilişim teknolojileri ve yazılım dersi öğretim programı*, [Çevrimiçi] <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>. (Erişim Tarihi:25/08/2025).
- Meisalo, V., Suhonen, J., Torvinen, S., & Sutinen, E. (2002). Formative Evaluation Scheme For A Web-Based Course Design. In *ACM SIGCSE Bulletin*, 34(3), 130-134. <https://doi.org/10.1145/637610.544454>
- Miles, M. B. & Huberman, A. M. (1994). *Qualitative data analysis: An Expanded Sourcebook* Sage.
- Newsted, P. R. (1975). Grade And Ability Predictions In An Introductory Programming Course. *ACM SIGCSE Bulletin*, 7 (2), 87-91. <https://doi.org/10.1145/382205.38289>
- Numanoğlu, M. & Keser, H. (2017). Programlama Öğretiminde Robot Kullanımı- Mbot Örneği. *Bartın Üniversitesi Eğitim Fakültesi Dergisi*, 6(2), 497. Doi: 10.14686/buefad.306198
- Okuyucu, M. O. (2019). *Robotik Kodlama Eğitiminin Lise Öğrencilerinin Üst biliş ve Yansıtıcı Düşünme Düzeyleri Üzerindeki Etkisinin İncelenmesi*, (Yüksek Lisans Tezi), Binali Yıldırım Üniversitesi Fen Bilimleri Enstitüsü, Erzincan.
- Özçakmak, Ş. (2014). *Bilgisayar Kullanımı Çocukta Bağımlılık Yapar mı?* [http://www.haberturk.com/polemik/haber/973204-bilgisayar-kullanimi-cocukta-bagimlilik-yapar-mi?](http://www.haberturk.com/polemik/haber/973204-bilgisayar-kullanimi-cocukta-bagimlilik-yapar-mi) (Erişim Tarihi:25/08/2025).

- Özdemir, B. (2020). *Okul Öncesi Öğretmenlerinin Erken Matematik Eğitimine İlişkin Pedagojik Alan Bilgileri İle Öz Yeterlilikleri Arasındaki İlişkinin İncelenmesi*, (Yüksek Lisans Tezi), İstanbul Sabahattin Zaim Üniversitesi, İstanbul.
- Resnick, M. (2013). *Learn To Code- Code To Learn*. <https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn>. (Erişim Tarihi:25/08/2025).
- Robins, A., Rountree, J. & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172. <https://doi.org/10.1076/csed.13.2.137.14200>
- Rogers, C. B., Wendell, K. & Foster, J. (2010). The Academic Bookshelf: A Review Of The Nae Report. Engineering in k-12 Education. *Journal of Engineering Education*, 99(2), 179-181.
- Ruzzenente, M., Koo, M., Nielsen, K., Grespan, L. & Fiorini, P. (2012). *A Review Of Robotics Kits For Tertiary Education*. In Proceedings of International Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum (s. 153-162).
- Salter, J. (2013). *Coding For Kids: Schoolchildren Learn Computer Programming, The Telegraph*, <http://www.telegraph.co.uk/technology/10468460/Coding-for-kidsschoolchildren-learn-computer-programming.html>. (Erişim Tarihi:25/08/2025).
- Sayın, Z. & Seferoğlu, S. S. (2016). Yeni Bir 21. Yüzyıl Becerisi Olarak Kodlama Eğitimi ve Kodlamanın Eğitim Politikalarına Etkisi. Akademik Bilişim Konferansı, Aydın, s.3-5.
- Schillaci, M. A., Waitzkin, H., Carson, E. A., Lopez, C. M., Boehm, D. A., Lopez, L. A., et al. (2004). Immunization Coverage And Medicaidmanaged Care In M,New Mexico: A Multimethod Assessment. *Annals of Family Medicine*, 2(1), 13-21. DOI: [10.1370/afm.100](https://doi.org/10.1370/afm.100)
- Stallman, R. (2021). *Android ve Kullanıcıların Özgürlüğü. Özgür Yazılım Vakfı*. <https://www.gnu.org/philosophy/android-and-users-freedom.tr.html>. (Erişim Tarihi:25/08/2025).
- Sullivan, A. & Bers, M.U. (2016). Robotics In The Early Childhood Classroom: Learning Outcomes From An 8-Week Robotics Curriculum In Pre-Kindergarten Through Second Grade. *International Journal of Technology and Design Education*, 26(1), 3-20.
- Şimşek, K. (2019). *Fen Bilimleri Dersi Madde Ve Isı Ünitesinde Robotik Kodlama Uygulamalarının 6. Sınıf Öğrencilerinin Akademik Başarı Ve Bilimsel Süreç Becerileri Üzerine Etkisinin İncelenmesi*, (Yüksek lisans tezi). Marmara Üniversitesi, İstanbul.
- Tashakkori, A. & Creswell, J. W. (2007). The New Era Of Mixed Methods [Editorial]. *Journal Of Mixed Methods Research*, 1(1), 3-7. <https://doi.org/10.1177/2345678906293042>
- Tashakkori, A. & Teddlie, C. (Eds.) (2003a). *Handbook Of Mixed Methods In Social & Behavioral Research*. Thousand Oaks, CA:Sage.
- Williams, L. & Cernochova, M. (2013). Literacy From Scratch. In *Proceedings Of The 10th IFIP World Conference On Computers In Education, WCCE*, s.17-27.
- Yauney, J., Bartholomew, S. R., & Rich, P. (2023). A Systematic Review Of “Hour of Code” Research. *Computer Science Education*, 33(4), 512-544. <https://doi.org/10.1080/08993408.2021.2022362>.
- Yıldırım, A. & Şimşek, H. (2003). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri*. Seçkin Yayıncılık (11.Baskı). Ankara.



The Effects of Android and Robotic Programming Practices on The Students' Programming Self Efficacy and Performance in the Secondary School Project Preparation Lesson

Esra Cesur^{1*}
Tuğba Yanpar Yelken²

¹ Mersin University, Department of Educational Sciences, Mersin, Turkey
esrasesur@gmail.com

* Corresponding Author

² Mersin University, Department of Educational Sciences, Mersin, Turkey
tyanpar@gmail.com

Received: 22.05.2024
Accepted: 14.12.2024
Available Online: 21.10.2025

Abstract: The aim of the research is to determine the impression of Android programming and robotic programming practices on the students' programming self efficacy and performance in the high school Project Preparation Lesson. The name of the method is that used in the research is Mixed method. Convergent parallel design was used in this research. In the experimental part of research, as it is a semi-experimental research, sample of the research consists of 33 learners of control group, 33 learners of experimental group and 66 learners in total. Curriculum planning of Android programming for experimental group, curriculum planning of robotic programming for control group were prepared in the research. Firstly, the scale of programming self efficacy was developed. Afterwards, a performance evaluation form was developed. After the scale of the programming self efficacy for high school students was carried out as a pre test to the control group and experimental group, the activities within ten weeks and twenty hours of curriculum have been done and then the programming self efficacy scale for high school learners as post test was carried out. The programming self-efficacy scale for high school learners, performance evaluation forms, diaries of lesson and focus group interviews are the data collection tools. As a result of the 10 week programming training, programming self-efficacy levels of students were higher than the beginning. It was determined that there was a significant difference between the gain puaans of the programming self-efficacy in the control and experimental group. As a result of the focus group interview data analysis, it was found out that the thoughts of both the control group and the experimental group students about programming included both positive and negative opinions.

Keywords: Programming, Android, Arduino, Self- efficacy, High school students.

INTRODUCTION

The aim of programming or coding education, which seeks to advance 21st-century skills of learners, is to prepare individuals who solve problems, generate knowledge, actively use information, think critically, are entrepreneurial and are empathetic for society. International coding events and programming instruction offered in massive open online learning environments have gained more importance and popularity after computational thinking skills were recognized as part of 21st-century skills (Yauney, Bartholomew & Rich, 2021). There are studies that demonstrate that one of the methods that should be applied for learners to acquire and utilize these skills is teaching programming or coding (Williams & Cernochova, 2013:2; Günüş, Odabaşı & Kuzu, 2013:436).

In the rapidly advancing life, the aim is to show the emerging young generation that they can create new applications instead of merely consuming existing programs. In order to become a productive country in a science and technology-centered world, importance is given to the coding competencies of individuals. Thus, programming and coding skills have become a fundamental competency that individuals must possess in the fields of science and activity, as well as in the processes of learning and teaching (Köse & Tüfekçi, 2015).

Programming education, which is one of the fundamental principles of information technology, emerges as an important field of instruction that enables the existence and sustainability of software applications (Kert & Uğraş, 2009). Introducing students to concepts such as algorithms, coding, and programming as part of compulsory foundational education from an early age will add value to their ability to process information in the later years of their lives (Bybee, 2010; Çepni, 2017; Okuyucu, 2019). The first crucial step in teaching programming or coding is the instruction of programming logic (Arabacıoğlu, 2006).

Programming not only enhances learners' mathematical-computational skills (such as variables and conditional statements), but also teaches them problem-solving learning strategies, how to design projects, and

Cite as (APA 7): Cesur, E. & Yanpar Yelken, T. (2025). The effects of android and robotic programming practices on the students' programming self efficacy and performance in the secondary school project preparation lesson. *Trakya Journal of Education*, 15(4), 1503–1548. <https://doi.org/10.24315/tred.1488016>

how to make connections among ideas while writing code. Skills are necessary not only for expert programmers but also for individuals of all ages who are engaged in the field and the profession (Resnick, 2013).

In the present times, programming skills are one of the key competencies expected to be possessed in the 21st century. Particularly with well-known individuals expressing this concept, it has been discussed on various platforms, and projects have been initiated and implemented to incorporate it into educational systems, while the private sector is also seen to support these initiatives with great enthusiasm. As a result of this situation, certain programming languages, domains, and application software have been developed to teach programming to preschool learners. With these programming language tools, most of which are free, it has become possible to create applications without using code, similar to drag-and-drop or playing puzzles. These environments are referred to as Block Programming Environments. Applications like Alice, Blockly, App Inventor, Code.org, Code Studio, Scratch, KoduLab, and Snap can be characterized as innovations in this field. (Numanoğlu & Keser, 2017).

When research conducted according to student levels is examined, it is observed that programming has a positive effect on students' performance. In another study, it was found that it is rare for high school age individuals to successfully learn a programming lesson within the scope of a specific curriculum, and it was stated that the situation that it is boring for an individual at that age to complete an education process within a bounded program is also seen in programming education. (Gerják, 2017). For this reason, it is emphasized that concepts such as algorithms, programming and databases should be accepted as knowledge and skills that should be acquired within basic education as of the 5th grade of secondary school (Akpınar & Altun, 2014). In our country, coding education and information technologies education have been gradually introduced into the curriculum starting from the 5th grade after 2012 (Sayın & Seferoğlu, 2016).

It is seen in the literature that students' advanced programming learning increases their academic success in secondary education and university education. Konvalina, Stephens, and Wileman (1983) argued in their research that success and performance in coding lessons at university are related to the year-end grade point average in high school, the year-end grade point average in high school mathematics, and the information technology lessons given in secondary education. Newsted (1975) investigated the factors affecting success in the introductory programming lesson at the University of Wisconsin and found that there was a significant relationship between success in introductory programming and year-end grade point average (general academic achievement).

The use of the classical teaching method in programming lessons leads to inadequate learning of the programming language and underdevelopment of problem solving skills and future analysis and design skills. For this reason, the success of programming education is measured by the level of applicability to the problems that arise in today's life (Karabak & Güneş, 2013). In this way, it can be said that with programming education, countries can reach the top in the field of technology by increasing their level of development (Demirer ve Sak, 2016).

In Rountree, Robins, and Rountree's (2003) research, it is stated that the obstacles and difficulties encountered in programming teaching are due to the complexity of programming and coding teaching, the necessity of teaching knowledge about what has been learned, developing new methods related to this learning, and the ability to write programs in applications. When the reason for this situation was investigated, the students stated that the theory and application studies of programming and coding learning were very difficult and that they had anxiety about this issue (Meisalo, Suhonen, Torvinen & Sutinen, 2002).

Programming teaching is carried out with different studies at each level of students. In addition, research shows that the effects of programming learning on achievement and performance in other lessons are positive. The details of the beginning and development processes of programming education in Turkey and around the world are given below.

Programming Education of Turkey

In Turkey, research on the development of programming and coding education and training in schools is as follows; In 2006, in the Information Technologies (IT) lesson in force in the primary education program, learning areas for advanced applications were established. These learning areas include the concept of programming, object oriented programming and web design. (Gülbahar & Kalelioğlu, 2018). In line with Decree No. 69 of 2012, the name of the ICT lesson was changed to “Information Technologies and Software”, units related to programming and software teaching were added to the curriculum, and learners started to learn basic programming as of the 5th grade. In this way, with the curriculum of the IT and Software lesson published by the Ministry of National Education, it is aimed to raise individuals who use the basics of information technologies in the best and most effective way (Talim Terbiye Kurulu Başkanlığı, 2012). The lesson, which was renamed Computer Science as a result of the renewal made in 2016, started to be implemented as Lesson 1- Lesson 2 in secondary education. At the secondary level, Computer Science continues to be offered as an elective lesson with the latest update of the curriculum. As of 2025, the name of the Computer Science lesson has been changed to the elective Information Technology and Software lesson and the lesson remains within the same curriculum.

Programming Education of The World

Estonia, as of October 2012, children and young people as young as 7 years old started to receive programming instruction (Demirer & Sak, 2016). In Canada, programming education and training is compulsory in Grade 6 and as of 2017, teaching is ongoing (Şimşek, 2019). The Ministry of Science and Future Planning of the country of South Korea announced that programming lessons will be compulsory starting from the primary school level, and that primary school students in 2017 and secondary school students in 2018 will gradually receive programming education (Özçakmak, 2014). As of November 2013, Basic Computer Programming education and training has started in all schools in the UK (Salter, 2013). Prior to the UK government's renewal of the secondary education curriculum, 2014 was declared the “year of coding” and in this way, progress was made in the European Union (EU) in terms of raising awareness of “teaching children to code”. With the progress made, the EU organized various workshops, applications and events in November 2013 under the name of European Code Week celebrations (Öymen, 2014; Demirer & Sak, 2016). In addition, while in Norway, Finland and other countries of Northern Europe, IT teaching starts at an early age, in Italy, Spain and France it starts in the 6th grade (Ceylan, 2020).

In Turkey and around the world, children and adults are provided with 21st century skills to improve their programming abilities. In addition, there has been a need to organize various activities and organizations that realize the codes written to create programs according to the levels of individuals. A few of these organizations, events and platforms are as follows: “Code Org”, “Wise Beaver Event”, “Let's Coding Girls”, “IT Garage Academy”, “Coding Hour”, “Code Dejo”, “Code Club” and “Khan Academy” (Elçiçek, 2019).

In the world, as in our country, programming education standards are carried out in a similar process. For this reason, all events and activities in the international IT platform are aimed at increasing learners' programming knowledge and self-efficacy.

Android (Mobile) Programming

New features are being added to everyday mobile devices, making them a powerful and dynamic learning tool. The features in the application programs developed for mobile devices have attracted the attention of educators and the academic community, and with these innovations, mobile learning has started to take place in education and training (Bozkurt, 2015).

Android is a Linux-based open source operating system being developed by Google for all its mobile devices. The reason why the system is open source is that it has pretty much the same goal as all other open source projects. The Google Play game and app store runs on the Android operating system. Android applications can also be accessed from different sites on the internet (Arslan & Gülnar, 2014).

The application (working file) extension supported by Google Play is “.apk”. Android has a large and diverse group of developers who write applications that are used on devices. The middleware, libraries and API in the Android operating system are written in C programming language. Application software (.apk) runs on an application framework built on Apache harmony containing Java-compatible libraries (Stallman, 2021).

Robotics Programming

The word robot first appeared in Rossum's *Universal Robots*, written by Karel Čapek (1920), and then the word started to be used all over the world. In 1990, in Maja Mataric's definition of the word robot, it is stated as a machine that can act logically in accordance with the working principle by blending the information it sees from its environment with the information available in the robot (Kurşuncu, 2016). Later, the word robotics, derived from the word robot, was used in 1941 by Isaac Asimov in 1941 (Kılınc, 2014).

The concept of robotics is a broad field that includes motors, mechanical materials, sensors and concepts related to coding. Today, with the increasing interest in robotics, the concept of educational robotics has been mentioned for the first time. Educational robotics kits used in robotics studies have become widespread at all levels from primary to secondary education (Rogers, Wendell & Foster, 2010).

Educational robot kits through simulation are real active and active learning objects that demonstrate cognitive adaptation (Ruzzenente vd, 2012). Educational robotics creates a system suitable for learners to encounter problems in life and produce solutions, and learners learn through the robots they create and develop (Eguchi, 2014).

Robotic coding is considered as a difficult process that requires an intense mental effort where hardware and coding should be put to work together by the students (Çakıroğlu & Kiliç, 2020). Education and training systems are updated to provide young learners with “Coding and Robotics” training and many skills to perform robotic works based on coding (Göksoy & Yılmaz, 2018). When the MEB (2019) curriculum is examined, it is seen that the Robot Programming unit has started to be included in the curriculum, even though the lesson name has not changed. Learning approaches that enhance learning with robotics are project-based, problem-based, learning by design, student-centered and constructivist learning approaches that focus on the learning process rather than the final product (Çakıroğlu & Kiliç, 2020). When we look at developed countries, it is seen that countries teach coding up to the age group where kindergarten education is given and they provide science, technology and engineering education from an early age thanks to educational robot kits (Sullivan & Bers, 2016).

It is evident that robotic programming training should be in the areas of interest of individuals and that developing this area of interest will increase the level of success of programming knowledge and skills in our country. For this reason, the development of programming with both hardware and software can be realized with robotic coding trainings. Likewise, developing mobile programming and seeing it as a field of interest can increase the development of programming and training in our country. In this way, competent and qualified projects can gain a place in international platforms.

When the researches were examined, it was seen that there were very few programming teaching studies conducted with secondary school students compared to students at other educational levels. The research provides information about the programming self-efficacy and academic achievement levels of secondary school students.

In the study, it was aimed to examine the effect of Android Programming and Robotic Programming on students' programming self-efficacy and performance in Secondary Project Preparation Lesson.

METHOD

Research Design

In the mixed method study, qualitative and quantitative data were collected simultaneously. In this context, it is a convergent parallel design research model. In the research model approach, the researcher collects qualitative and quantitative data at the same time, analyzes them separately and compares the results of all the data obtained and examines whether the findings confirm the results of the data (Creswell ve Plano Clark, 2015). Semi-experimental research, including research using a semi-experimental design, is research in

which the sample is not randomly selected, there are experimental (E) and control (C) groups, and the experimental application is performed (Yıldırım ve Şimşek, 2003). For this reason, two classes, one experimental (E) and one control (C) group, were randomly assigned among the classes with similar characteristics. Android mobile programming was taught to the experimental group and Arduino robotic programming was taught to the control group. The results of the programming self-efficacy scale used to collect the quantitative data of the pre-test and post-test and the performance evaluation scores of the students were calculated. The data obtained through focus group interviews and diaries are qualitative data. İlk başta, nicel veriler toplanmıştır. Qualitative data were then collected and analyzed in the same period of time. The results of the analysis of quantitative and qualitative data obtained from E and C groups were compared. It was observed whether the results of qualitative and quantitative data supported each other.

Determination of programming self-efficacy of first level secondary school students was carried out with the "Programming Self-efficacy Scale" and students' performance evaluation scores as pre-test (T-T) and post-test (S-T). At the same time, qualitative data were collected from the students through diaries to get their opinions at the end of the lesson and focus group interviews at the end of the curriculum. In this way, it is aimed to better understand whether there is a correspondence between the comparison of qualitative and quantitative analyses to determine the programming self-efficacy and performance of secondary school students. The number of secondary school students (N=66) determined as the experimental and control groups and the number of secondary school students (N=362) in the programming self-efficacy scale construction study were selected as the sample. In this study, data obtained from 66 students were used to combine qualitative and quantitative data. In the study, a total of three measurements were made; two measurements related to the students' programming self-efficacy to collect quantitative data and one measurement to determine their performance. In the first measurement, the programming self-efficacy scale was administered to the students as a pretest at the beginning of the 20-hour instructional program on programming learning. As the second measurement, the programming self-efficacy scale was administered to the students as a post-test at the end of the programming learning curriculum. The third measure is the students' performance evaluation scores on the programming language they have learned. While analyzing the quantitative data, it was concluded whether the pre-test (T-T) and post-test (S-T) scores were compatible, however, the same statistical operations were performed for the students' performance evaluation scores. Then, descriptive statistics were presented and comparisons were made between statistical groups to determine a significant difference between E and C groups in terms of other variables.

In the study, semi-structured interview questions were developed about students' programming self-efficacy and focus group interviews were conducted in both the control and experimental groups. Interviews were transcribed and content analyzed. The same content analysis was conducted for the diaries. This analysis is independent of quantitative analysis. Thus, qualitative and quantitative data sets were identified and analyzed differently and simultaneously in order to link both qualitative and quantitative information. Accordingly, both types of data were deemed equally important to reveal the purpose of the study. Figure 1 shows the details of the convergent parallel pattern flowchart in the research.

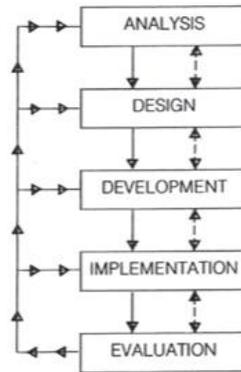
Figure 1

Convergent Parallel Pattern Flowchart of Research

STEP 1	<p>Designing the Quantitative Part Programming self-efficacy scale and performance evaluation form were created. Permissions were obtained for quantitative data. Two classes were identified and a semi-experimental design approach was applied.</p>	<p>Designing the Qualitative Part Focus group interviews and a diary were conducted. Permissions were obtained for qualitative data and data were collected.</p>
STEP 2	<p>Quantitative Data Analysis The pretest of the Programming Self-Efficacy Scale was administered to the Experimental and Control groups at the beginning of the course. At the end of the lesson, the scale is applied as a post-test. Students' performance scores are determined with the performance evaluation form. Pre-test and post-test and other variables are analyzed. Performance evaluation scores are also analyzed. Descriptive statistics are made.</p>	<p>Qualitative Data Analysis Qualitative data analysis was conducted using theme development procedures. Focus group interviews and diaries were content analyzed.</p>
STEP 3	<p>Using Strategies to Combine Control and Experimental Group Results Control and experimental group analysis results are compared. Differences in the control and experimental groups are determined based on other variables and the differences are examined in an organized image as a result of the dimensions obtained. The content analysis is made comparable for the control and experimental groups. The themes are converted into numbers and compared and related.</p>	
STEP 4	<p>Interpretation of Combined Results The results for the control and experimental groups are summarized and interpreted independently. T-T and S-T results are interpreted for the groups. Performance evaluation scores are interpreted for each group. Odak grup görüşmeleri ve günlükler her grup için yorumlanır. Explain the points of convergence of the results of the control and experimental group data, the ways in which they differ from each other and whether they are related to each other.</p>	

ADDIE Model

ADDIE design model was used in the study. The name of the model is a combination of the first letters of the words: Analysis(A), Design(D), Development(D), Implementation(I), Evaluation(E). A diagram representation of the ADDIE Design Model is given below in Figure 2.

Figure 2*ADDIE Model Diagram*

Source: (Grafinger, 1988)

The ADDIE design model is an instructional design model for education that encompasses planning, development, implementation, and finally evaluation, and includes the instructional domain, the teacher, the learner, assessment and evaluation, and external factors (Gustafson & Branch, 2002).

According to the ADDIE model, in the analysis phase of the research, a specific study was planned to enable secondary school students to have more comfortable lives all over the world by combining hardware and software technologies, and for this study, it was thought that the education and training environment would be more effective and permanent in line with the concept of programming self-efficacy. In the design phase, activities were planned and then implemented. With the convergent parallel design method, a strategy is determined to combine the results of the experimental and control groups as a result of first collecting and analyzing quantitative and then qualitative data. First, the analysis results of the control and experimental groups are compared. The differences that emerge are identified based on other variables and the differences are examined in an organized image as a result of the themes obtained. In the development phase, it was tried to prepare the mobile programming activities of the experimental group and the robotic programming activities of the control group in accordance with the classroom environment. An appropriate learning environment was tried to be created. 20-hour robotics and mobile programming curriculum were created and worksheets were created for both groups. It was deemed appropriate to use lecture and demonstration methods as teaching methods. During the evaluation process, performance evaluation forms were designed for both groups. The implementation phase culminates with the realization of the 20-hour curriculum in appropriate classroom environments, with worksheets, lecture and demonstration methods, and the completion of performance evaluation forms according to students' performance in the activities. In the evaluation phase, quantitative analyses of the results of the analysis of the secondary level programming self-efficacy scale and the results of the performance evaluation form, which were applied to Control-G and Experimental-G on behalf of the pre-test (T-T) and post-test (P-T), were conducted. After the lesson, the content analysis of the diaries written by the students and the focus group interviews were conducted. Content analysis of the diaries and interviews was carried out by creating themes. Accordingly, as a result of the analyses, the points of convergence of the experimental and control groups, their points of divergence, and whether they are related to each other are explained.

Participants and Sampling

This research was applied as a semi-experimental design in two classes taking the Project Preparation lesson in a secondary school because it is an applied study. An experimental study was conducted with 66 10th grade students in an Anatolian high school in Yenişehir district of Mersin. There were 33 learners (mobile programming) in the experimental (E-G) group and 33 learners (robotic programming) in the control (C-G) group. The gender distribution of the learners in the semi-experimental study group of the research is shown in Table 1.

Table 1

Distribution of Students in the Semi-Experimental Application Study Group According to Gender (G) Variable

(G)	(EG)		(CG)		(T)	
	f	%	f	%	f	%
Female	16	48	19	58	35	53
Male	17	52	14	42	31	47
Total	33	100	33	100	66	100

When Table 1 is examined, the number of learners in the study group is 35 female (53%) and 31 male (47%). There were 33 (50%) learners in the experimental group and 33 (50%) in the control group. The number of students in the experimental group was 16 girls (48%) and 17 boys (52%). The number of students in the control group was 19 girls (58%) and 14 boys (42%).

In the study, the study group, in which qualitative focus group interview questions and diary form were created and opinions were taken, included 66 students who were applied. The study group for the focus group interview questions consisted of 6 groups of 5-6 students each for the control group and 6 groups of 5-6 students each for the experimental group, totaling 12 groups. The number of students in the group was the same as that of the study group and included 66 students in total.

Table 2

Distribution of Students in the Focus Group Interview Study Group According to Gender Variables

Groups	EG		CG		T	
	F	M	F	M	F	M
Group1	5	-	3	3	8	3
Group2	3	3	2	4	5	7
Group3	5	-	4	2	9	2
Group4	-	5	5	-	5	5
Group5	-	6	3	2	3	8
Group6	3	3	2	3	5	6
T	16	17	19	14	35	31

As seen in Table 2, the number of students in the focus group interview study group in the order of the experimental group was 5 female (F) students, 3 female (F) 3 male (M) students, 5 female (F) students, 5 male (M) students, 6 male (M) students and 3 female (F) 3 male (M) students, totaling (T) 33 students. According to the order of the control group, the number of students was 3 female(F) 3 male(M) students, 2 female(F) 4 male(M) students, 4 female(F) 2 male(M) students, 5 female(F) students, 3 female(F) 2 male(M) students and 2 female(F) 3 male(M) students, totaling (T) 33 students.

Data Collection Instruments

The data collection tools in the study are the programming self-efficacy scale for secondary school students, performance evaluation form, student diary and focus group interviews. All data collection tools are explained in detail below.

Programming Self-Efficacy Scale for Secondary School Students

Determining the programming self-efficacy of high school students who have little programming or coding knowledge and who choose Computer Science lessons makes an effective and positive contribution to the teaching process. Considering the level of readiness of younger students who have knowledge about programming, a self-efficacy study is necessary since high school students are in a transition period between

primary, secondary and university students. Therefore, it was deemed necessary to conduct a validity and reliability study of the “Programming self-efficacy scale for high school students”.

All analyses related to the development of the scale were conducted and as a result, it was seen that the reliability and validity of the scale were ensured.

Performance Evaluation Form

In the study, the performance evaluation forms prepared to be used for end-of-lesson evaluation during the implementation of the curriculum in both the experimental and control groups contain scores for the products created by each student and the gains achieved. The 29-item performance evaluation forms, which consisted of separate outcomes for the experimental and control groups, were examined by three field experts and found suitable for use for measurement and evaluation purposes.

Performance evaluation forms consisting of 29 items each for the experimental and control groups included outcome statements. At the end of each lesson, it is filled in by the teacher and performance evaluation scores are obtained for each group of students at the end of 10 weeks of lessons. In this way, the performance evaluations of the students in both the experimental and control groups were calculated.

Student Diary

In the study, a diary form was created for the experimental and control groups at the end of each lesson to collect the opinions of the students. The students who will fill out the form were determined as experimental group students in the 10-E class (Android- Mobile programming) and control group students in the 10-D class (Arduino- Robotic coding). Accordingly, the students of each group were asked to write down their ideas and feelings about what they had learned in the lesson each week. The student diary form created in Google Forms includes the diary name (lesson topic), student name, surname, number, class, thoughts and date. The link to the student diary form previously shared by the lesson teacher on the Education Informatics Network (EBA) platform, which was determined as the information sharing platform of the Ministry of National Education, was delivered to the experimental and control group students. At the end of each lesson, students were encouraged to fill in the diary by clicking on this link.

Focus Group Interview

In the study, focus group interviews were conducted separately with the experimental and control groups. In order to have more detailed information about the 10-week and 20-hour curriculum, the students in both groups were asked for their opinions through focus group interviews. According to Ekiz (2003), the purpose of focus group interviews is to obtain detailed and multifaceted qualitative information about the views, experiences, experiences, interests, tendencies, perceptions, attitudes, habits and emotions of the participants on a given topic. Accordingly, the number of students for each focus group was determined as 6.

Data Collection Procedure

In line with the activity plans of the control and experimental groups prepared for the 10-week (20-hour) implementation process in the Project Preparation lesson, an experimental study was conducted in the control and experimental groups between Week 3 and Week 13 of the education week. This study aims to teach secondary school students robotic coding with Arduino and mobile programming with Android through practical applications. Both groups of students were asked to fill out a daily log at the end of each week.

The Programming Self-Efficacy Scale (Cesur Özkara & Yanpar Yelken, 2020) developed during the research was administered to students as a pre-test before starting instructional designs. After completing the 10-week training process, the programming self-efficacy scale was administered to the students in the same manner as a post-test. In addition, upon completion of the applications, the lesson instructor filled out a performance evaluation form for each student. The completed form shows the students' performance in the applications.

In addition to student diaries, the opinions of both groups were obtained through focus group interviews. Thus, qualitative data was obtained from student diaries and responses to focus group interview questions.

In the study, after the groups were determined, a 10-week, 20-hour experimental treatment schedule was created to carry out the experimental study. An experimental study was conducted in accordance with this schedule. During the 10-week application period, activities related to Android-mobile programming were carried out by the students in the experimental group, and activities related to robotic coding were carried out by the students in the control group.

During the application process, before proceeding with the lesson, both the control and experimental groups were asked about their knowledge of programming concepts and Android programming with Arduino. Based on this information, it was observed that most students did not have much knowledge about robotic coding and mobile programming. As a result of determining the readiness information of the students, 10-week training lessons have begun.

When determining the readiness levels of the students, the same exercises were given to both groups during one class period. These applications are entirely focused on teaching the concept of algorithms. The definition of the algorithm, the creation of the algorithm diagram, and all algorithmic symbols were solved together with the students using the question-and-answer method with two or three examples.

Both groups of students were given an assignment related to algorithms, and they were asked to bring a paper with the algorithm diagram prepared during the first week of the lesson. The following headings contain detailed weekly descriptions of the experimental study's control and experimental groups in the education and training process.

Data Analysis

The analyses of the quantitative and qualitative data obtained in the study are described in detail below.

Analysis of Quantitative Data

In the experimental study conducted with secondary school students, it was deemed appropriate to use the programming self-efficacy scale (T-T) and (S-T). The scale was administered separately to the experimental (E) and control (C) groups, and the normality of the distribution of the obtained data was examined. Since each group had a sample size of 33, the Shapiro-Wilks normality test was used to assess normality. The data appears to be normally distributed, with a numerical value greater than $p > 0.05$ in the Shapiro-Wilks tests (Kalaycı, 2014). The results of the normality test of the distribution are presented in Table 3.

Table 3

Normality Test Results

Variables	Groups	Shapiro-Wilk		
		Statistics	sd	p
Programming Self-Efficacy Scale Pre-test	EG	0,939	33	0,630
	CG	0,967	33	0,398
Programming Self-Efficacy Scale Post-test	EG	0,974	33	0,583
	CG	0,980	33	0,790
Programming Self-Efficacy Scale	Pilot Study (EG and CG)	0,993	367	0,073
Performance Evaluation	EG	0,974	33	0,583
	CG	0,967	33	0,398

Table 3 shows that the pre-test and post-test scores of the programming self-efficacy scale used in the study and the total scores of the performance assessment test and all sub-dimensions of the scale were found to be normally distributed ($p < 0.05$). In this case, an independent groups t-test (Kalaycı, 2014) was applied to

compare the mean scores of two different sample groups in the study. The quantitative data analyzed in the study were analyzed using the SPSS 24.0 software package, and all data in the study were tested at a significance level of 0.95 ($p < 0.05$).

Analysis of Qualitative Data

At the end of each lesson, students were asked to write in their diaries to share their thoughts on Arduino and Android applications. In addition, focus group interviews were conducted to gather students' opinions on the 10-week activity program. At the end of each week, students in both groups were asked to write their diaries. In the focus group interviews, six students from each of the two groups were brought together separately, asked interview questions, and audio recordings were made, which were later transcribed into text. The data determined as a result of all the work carried out has been evaluated through content analysis. It was requested that each piece of data revealed by the content analysis be analyzed by different experts. Based on content analyses provided by experts, a reliability formula was used (Reliability = Agreement / (Agreement + Disagreement)) (Miles ve Huberman, 1994). Table 4 below presents the themes and sub-themes of the qualitative data analyzed using qualitative content analysis, along with the Miles and Huberman coefficient values.

Table 4

Qualitative Data Analysis Results

	Qualitative Data Contents	Miles and Huberman Reliability Coefficient
1	Diary of Student (Arduino)	
	-Information about the Applications	0,85
	-Opinions about the lesson	0,75
	- Attitude towards the lesson	0,83
2	Diary of Student (Android)	
	-Information about the Applications	0,92
	-Opinions about the lesson	0,75
	- Attitude towards the lesson	0,75
3	Focus Group Interview Form (Arduino)	
	-Attitude towards programming	0,88
	-Lesson outcomes	0,81
	-Positive aspects of the lesson	0,85
	-Negative aspects of the lesson	0,85
	-Suggestions for the lesson	0,80
4	Focus Group Interview Form (Android)	
	-Attitude towards programming	0,80
	-Lesson outcomes	0,80
	-Positive aspects of the lesson	0,85
	-Negative aspects of the lesson	0,83
	-Suggestions for the lesson	0,80

When Table 4 is examined, it can be seen that the reliability of the themes and sub-themes obtained in the content analysis is 81% for student diaries and 83% for focus group interview records. The reliability of Arduino student diary themes was calculated as 81%, the reliability of Android student diary themes was calculated as 81%, the reliability of Arduino focus group interview themes was calculated as 84%, and the reliability of Android focus group interview themes was calculated as 82%.

RESULTS

The statistical analyses and findings obtained regarding the sub-problems of the research are examined.

Results Related to the Programming Self-Efficacy Scale for High School Students

The Programming Self-Efficacy Scale developed in the study was administered to secondary school students as a pre-test at the beginning of the 10-week training and as a post-test at the end of the 10-week training. The study group consists of two separate groups: a control group receiving Arduino robotics coding training and an experimental group receiving Android mobile programming training. According to the first sub-problem, it was examined whether there was a significant difference between the experimental (E) and control (C) groups in terms of programming proficiency (final programming proficiency-pre-programming proficiency) results. An independent t-test was applied to analyze the data regarding whether there was a significant difference in the programming access scores (P-T programming proficiency-T-T programming proficiency) of groups (E) and (C). At the beginning and end of the fall semester of the 2018-2019 academic year, a pre-test (T-T) and post-test (P-T) were administered to a total of 66 students in groups (E) and (C). Below, the average programming access scores of the groups are compared in order.

At this stage, the groups were examined based on the preliminary test results. The results of the t-test conducted to determine whether there was a significant difference in the mean T-T scores of the programming self-efficacy scale between the groups are presented in Table 5.

Table 5

Experimental and Control Groups Programming Self-Efficacy Scale Pre-test T-Test Results Table

Groups	N	\bar{X}	S	sd	t	p
CG	33	3,35	,51121	63	1,262	0,212
EG	33	3,17	,59289			

$p < 0,05$

As a result of the applications made in the Project Preparation lesson, the arithmetic mean scores (is $\bar{X} = 3.17$) of the experimental group students on the Programming Self-Efficacy Scale were calculated, while the arithmetic mean scores (is $\bar{X} = 3.35$) of the control group students on the Programming Self-Efficacy Scale were calculated. When Table 5 is examined, the average of the 33 students in the control group is determined to be is $\bar{X} = 3.35$, and the average of the 33 students in the experimental group is determined to be is $\bar{X} = 3.17$. According to the t-test results in Table 5, there is no significant difference at the 5% significance level between the mean scores obtained by the independent groups on the Programming Self-Efficacy Scale, $p = 0.21$, $p > 0.05$. The fact that the difference observed in the averages of the groups is not significant can be explained by the fact that groups (E) and (C) are equivalent to each other in terms of their T-T score averages. It can be said that both groups have a similar level of programming self-efficacy.

The relationship between the average scores of the groups was examined based on the post-test of the programming self-efficacy scale applied to the students. The results of the t-test conducted to determine whether there was a significant difference in the post-test score averages between the groups are presented in Table 6.

Table 6

Experimental and Control Groups Programming Self-Efficacy Scale Results of the T-Test for the Post-Test

Groups	N	\bar{X}	S	sd	t	p
CG	33	3,14	,45243	64	-0,325	0,746
EG	33	3,18	,55190			

$p < 0,05$.

As a result of the applications carried out in the Project Preparation lesson, the post-test arithmetic mean scores (is $\bar{X}= 3.14$) of the programming self-efficacy scale applied to the control (C) group students were calculated, while the arithmetic mean scores (is $\bar{X}= 3.18$) of the post-test administered to the experimental (E) group students using the programming self-efficacy scale were calculated as follows. As shown in Table 6, according to the results of the t-test in independent groups, there is a significant difference in the mean scores obtained by the groups on the Programming Self-Efficacy Scale at a 5% significance level ($p=0.75$, $p>0.05$). The average score for the 33 students in the control group is $\bar{X}=3,14$ and the average score for the 33 students in the experimental group is $\bar{X}=3,18$. As a result of the 10-week programming lesson taken by the groups, it can be said that the groups have achieved an equivalent level of programming self-efficacy.

At the end of the 10-week programming lesson, the results of the analysis of the programming self-efficacy scale, which was administered as a final test to the control and experimental groups, showed that the programming self-efficacy of the control group was equivalent to that of the experimental group. In addition, the analysis of the programming self-efficacy scale administered to the groups as a post-test showed that there was no significant difference in programming self-efficacy between the groups. As a result, no significant difference was observed between the control and experimental groups in terms of the effect of the 10-week programming lesson on self-efficacy levels.

A significant difference between the access scores of the groups was observed. The total programming self-efficacy achievement scores of the students in the (E) group, to whom "Android programming" was applied, and the (C) group, to whom "robotics-Arduino programming" was applied, were calculated. The arithmetic means and standard deviations of the two groups were calculated using these scores. The results of the t-test conducted to determine whether there was a significant difference between the groups in terms of access scores are presented in Table 7.

Tablo 7

T-test Results Table for Access Scores of Groups Experimental(E) and Control(C)

Groups	N	Access \bar{X}	S	Sd	t	p
EG	33	0,0885	0,63402	64	-2,686	0,009
CG	33	-0,3294	0,62997			

Considering Table 7, in the Project Preparation lesson, the t-test conducted for independent samples to determine whether robotic coding had a significant effect on Android programming skills revealed that the average S-T and T-T achievement scores of students in the robotic coding (C) group in the lesson ($\bar{X} C = 0,33$), while the average S-T and T-T achievement scores of students in the Android programming (E) group ($\bar{X} E = 0,09$) showed no significant difference [$t(64) = 0,009$, $p>0,05$]. In this context, a significant difference was found between the average scores of the groups.

Results Related to the Performance Evaluation Form

As a result of the applications made in the Project Preparation lesson, the performance evaluation score averages of groups (D) and (K) were compared to answer the question, "Is there a significant difference between the performance evaluation score averages of groups (D) and (K) after the application?" and The analyses and results revealing the relationship between the average performance evaluation scores of the groups are presented below.

As a result of the 10-week programming activities carried out in the Project Preparation lesson, performance assessments were conducted on the experimental and control groups to determine the students' performance levels. The pre-prepared performance evaluation form is created by giving points to students in the control and experimental groups based on the results of the activities. In this section, the relationship between the performance evaluation scores of the control and experimental groups was examined. The results of the t-test conducted to determine whether there was a significant difference in the average performance evaluation scores between the groups are presented in Table 8.

Table 8*T-Test Results Table For Control And Experimental Group Performance Evaluation Scores*

Groups	N	\bar{X}	S	sd	t	p
(C)	33	89,0909	6,78401	64	-0,761	0,449
(E)	33	90,4091	7,27093			

p=0,00, p<0,05.

As a result of examining the t-test result table, a t-test for independent samples was conducted to see whether there was a significant difference in performance evaluation between the control group that did activities on robotic coding and the experimental group that did activities on Android programming in the Project Preparation lesson. There was no significant difference between the performance assessment score average of students in the robotics coding control group ($\bar{X}_K = 89,09$) and the performance assessment score average of students in the Android programming experimental group ($\bar{X}_D = 90,41$) [$t(64) = 0,449$, $p > 0,05$]. Thus, when the average performance evaluation scores of the groups were examined, no significant difference was found. It can be said that the groups increased their programming knowledge based on what they learned from the activities as a result of the 10-week programming lesson.

Results Related to Students' Views on the Teaching-Learning Process of Android Programming and Robotics Programming

This section contains analyses and results regarding students' views on the teaching-learning process of Android programming and robotics programming, which is the third sub-problem of the research. The student diary and focus group interview records were analyzed. Categories were obtained as a result of the data obtained and The themes and sub-themes that emerged in this regard are listed below.

Control and Experimental Groups' Opinions on Student Diaries

In this section, both control and experimental group students wrote their opinions about the programming education and training process in their diaries at the end of each lesson. As a result of the content analysis, opinions on student diaries related to robotic (Arduino) coding consist of three main themes. The sub-themes of each main theme are presented in Table 9 according to their frequency values. The main themes of both groups were determined as 'information about applications', 'opinions about the lesson' and 'attitudes towards the lesson'.

Table 9*Control Group Students' Opinions on Student Diaries*

Opinions on Student Diaries Related to Robotics (Arduino) Coding	f
Attitudes towards the lesson	81
The lesson is enjoyable.	24
The effectiveness of the lesson	18
Applications should be fun	15
The robot project application is nice	10
The lesson was very productive	9
The lesson is a little boring	5
Information about applications	64
LED lighting	17
Traffic light construction	17
Making an LED application with a potentiometer	9
Doorbell application to be implemented	7
Implementation of LDR application	7
A night light construction	7

Table 9 (Continued)

Opinions about the lesson	33
Creating applications with Arduino	27
Learning Arduino and its variants	6

When Table 9 is examined, it is seen that the frequency value of the theme "attitudes towards the lesson" is higher ($f=81$) in line with the data obtained as a result of the analysis. When the subthemes of this theme are examined, it is seen that the control group students have positive opinions about the lesson. According to the students, the fact that the lesson is enjoyable ($f=24$), productive ($f=18$), and entertaining ($f=15$) indicates that they generally have positive attitudes toward the lesson.

Similarly, opinions on student diaries related to mobile (Android) programming consist of three main themes and The sub-themes of each main theme are presented in Table 10 according to their frequency values. The main themes of both groups were determined as 'information about applications', 'opinions about the lesson' and 'attitudes towards the lesson'.

Table 10

Experimental Group Students' Opinions on Student Diaries

Opinions on Student Diaries Related to Mobile (Android) Programming	<i>f</i>
Information about applications	297
Learning the concept of Frame layout	34
Learning the concept of Textview	34
Learning the concept of Linear layout	29
Learning how to use the seekbar	29
Learning how to use the Progressbar	27
Learning how to use the Edittext	25
Learning how to use the Grid layout	23
Learning the concept of Screen orientation	21
Learning how to use the Imageview	21
Learning how to use the Radiobutton	18
Learning how to use the Webview	15
Learning how to use the Checkbox	14
Creating an Appinventor application	7
Attitudes towards the lesson	80
Having a good lesson	33
The lesson was enjoyable	27
Understanding the topics	12
The lesson was a little difficult	8
Suggestions for the lesson	50
Using the Android Studio program	20
The topics are a bit confusing	17
Review of messages sent by the teacher	13

When Table 10 is examined, it is seen that the frequency value of the theme "information about applications" is higher ($f=297$) in line with the data obtained as a result of the analysis. When the subthemes of this theme are examined, it is seen that the experimental group students have knowledge about all the practices they do in the lesson. It was observed that the most interesting applications for the students were learning the frame layout concept and using textview ($f=34$), linear layout concept and using seekbar ($f=29$),

progressbar (f=27), edittext (f=25), grid layout concept (f=23), screen orientation concept and using Imageview (f=21), radiobutton use (f=18), Webview use (f=15), checkbox use (f=14) and using Appinventor (f=7).

Control and Experimental Groups' Opinions on Focus Group Discussion Records

In this section, both control and experimental group students shared their views on the programming education and training process in focus group interviews conducted with groups of six students formed in advance at the end of the 10-week lesson. Interviews with both the experimental and control group students were recorded. As a result of the content analysis, opinions regarding the robotic (Arduino) coding lesson process consist of five main themes. The sub-themes of each main theme are presented in Table 11 according to their frequency values. The main themes of both groups were determined as “attitudes toward pre-lesson programming,” “what the lesson taught,” “positive opinions about the lesson,” “negative opinions about the lesson,” and “suggestions for the lesson”.

Table 11

Control Group Students' Opinions on Focus Group Discussion Records

Opinions on Student Diaries Related to Robotics (Arduino) Programming	<i>f</i>
Attitude towards pre-lesson programming	29
Lack of knowledge about programming	12
Liking of programming	11
Being interesting	2
Previously made game	2
Never having taken a programming class	2
Lesson outcomes	32
Learning programming	13
Learning to code	11
Realizing that programming is difficult	4
Difficulty in writing code	2
Learning basic concepts related to coding	2
Positive aspects of the lesson	26
Well thought out	20
It was a very good lesson	6
negative aspects of the lesson	14
The class is very crowded	8
It would be better to have a guide	6
Suggestions for the lesson	15
Completeness of materials	4
The presence of small groups of people	4
Creating a different kind of classroom environment	4
Division of the class into two groups	3

When Table 11 is examined, it can be seen that the frequency value (f=29) of the theme “attitude towards pre-lesson programming” is obtained based on the data obtained from the analyses. Before the start of the 10-week education and training process, students' attitudes toward programming include both positive and negative opinions. In addition to positive expressions such as liking of programming (f=11), being interesting (f=2) and previously made game (f=2), negative expressions are also as follows; Programming has never been seen (f=12) and no programming lessons have been taken (f=2).

Similarly, opinions regarding the mobile (Android) programming lesson process consist of five main themes and The frequency values of the sub-themes of each main theme are given in Table 12. The main themes of both groups were determined as “attitudes toward pre-lesson programming” “lesson outcomes” “positive aspects about the lesson” “negative aspects about the lesson” and “suggestions for the lesson”.

Table 12*Experimental Group Students' Opinions on Focus Group Interview Records*

Opinions on Focus Group Interview Records Related to Mobile (Android) Programming	<i>f</i>
Attitudes towards pre-lesson programming	51
Never seen before	15
Never having been liked	13
Lack of interest	5
Not being interesting	4
Being interesting	3
Seeming boring	3
Not using the materials at all	3
The applications are good	2
Caring for and liking	2
Not knowing whether you like someone or not	1
Lesson outcomes	65
Learning to code	31
Making the application	6
Learning basic concepts	4
Easier final implementation	4
Learning all concepts	4
Working with worksheets	4
Learning that mobile programming is difficult	3
Not gaining anything	3
Active use of the EBA platform	3
Learning how to create an APK	3
Positive feedback about the lesson	56
Learning the algorithm	13
A nice lesson	11
The lesson is fun	7
Learning to program	6
Gaining more knowledge after the lesson	5
Learning to apply	5
Learning innovations	4
Learning through the latest application (project)	3
Knowledge of programming	2

Tablo 12 (Continued)

Negative feedback about the lesson	19
The class is noisy	7
The lesson is one lesson hour	4
Lack of interest in the lesson	4
Insufficient number of computers	4
Suggestions for the lesson	63
Increasing the number of computers	21
Facilitating applications	13
Increasing lesson hours	8
Performing the application on a computer	6
The possibility of implementing the application in lesson	6
Practical application after narrative technique	6
Making the lesson more enjoyable	3

When Table 12 is examined, it can be seen that the frequency value ($f=51$) of the theme “attitude towards pre-lesson programming” is obtained based on the data obtained from the analyses. Table 12 includes the students' attitudes towards programming and their positive and negative views before the start of the 10-week education process. In addition to negative expressions such as “never seen before ($f=15$)”, “Never having been liked ($f=13$)”, “lack of interest ($f=5$)”, “Not being interesting ($f=4$)”, “Seeming boring ($f=3$)” and “Not using the materials at all ($f=3$)”, positive expressions are as follows; “The applications are good ($f=3$)”, “Caring for and liking ($f=2$)” and “Not knowing whether you like someone or not ($f=1$)”.

DISCUSSION AND CONCLUSION

The study investigated whether there was a significant difference in programming proficiency scores (post-programming competencies minus pre-programming competencies) between the experimental and control groups. A comparison of the groups was made according to the pre-test administered at this stage. In conclusion, the pre-test data analysis showed that there was no significant difference in the mean scores derived from the programming self-efficacy scale between the groups. Upon comparing the groups' pre-test programming self-efficacy data, it was observed that similar findings exist in the literature. A review of the literature indicates that the number of studies yielding statistically significant results is smaller compared to those reporting non-significant differences. The absence of a significant difference in pre-test scores measuring students' programming self-efficacy typically stems from the groups having equivalent levels of knowledge, competence, and attitudes in their educational background.

In the study, the relationship between the groups' mean scores was examined based on the programming self-efficacy scale post-test administered to students. The study examined whether there was a significant difference in post-test mean scores between the groups, and the results revealed no significant difference in the groups' average scores on the Programming Self-Efficacy Scale. In line with this, comparable studies have been found in the existing literature. Significant findings appear less frequently than non-significant ones in the existing literature. The non-significant differences observed in post-test measures of programming self-efficacy may be attributed to the groups achieving comparable competency levels, knowledge acquisition, and attitudinal development through their training.

The analysis of literature, based on the significant between-group differences in programming self-efficacy achievement scores, aligns with Altın's (2021) findings in 'Middle School Students' Coding and Computational Thinking Skills: Traditional and Interdisciplinary Approaches in Programming Instruction,' which revealed that interdisciplinary mathematics-coding instruction improves both coding proficiency and computational thinking. Furthermore, research has revealed that programming instruction positively impacts

academic performance. In the study, the significant difference between groups' mean achievement scores regarding programming self-efficacy indicates its impact on academic performance. Furthermore, it can be argued that the logical aspects of programming instruction have particularly positive effects on students in quantitative subjects, especially mathematics.

The study found no significant difference between the mean performance assessment scores of students in the robotics coding control group (C) and those in the Android programming experimental group (E). In line with results from the literature review, learners' self-efficacy perception scores in programming appear to influence their programming performance (Ramalingam, LaBelle & Wiedenbeck, 2004; Aşkar & Davenport, 2009; Davidson, Larzon & Ljunggren, 2010; Altun & Mazman, 2013).

In the master's thesis by Alagöz (2020), a significant improvement was observed in both academic achievement and computational thinking skills following a 5-week 'Learning Programming Through Metaphors' intervention. Şimşek's (2019) study revealed statistically significant differences in adjusted post-test scores between the experimental group (robotics coding activities) and control group (E and C groups), when controlling for pre-test scores. However, the analysis showed no significant effect on scientific process skills development.

The study findings indicate that the training program effectively developed competent individuals in both the robotics coding control group and mobile programming experimental group, with students demonstrating self-efficacy in completing required software tasks. The implementation of both robotics and mobile programming curricula showed significant positive effects on secondary students' lesson performance. This programming instruction consequently led to improved academic outcomes through enhanced student performance metrics.

The study analyzed student journals and focus group interviews, comparing results with control group data and existing literature. Findings indicate that programming education substantially reinforces 21st-century skills—problem-solving, critical thinking, creativity, and analytical reasoning—among learners. These outcomes correlate with enhanced classroom performance and academic achievement, with interdisciplinary benefits evident in other subject areas.

No significant difference was found between the control and experimental groups' pre-test mean scores indicating equivalent baseline levels of programming self-efficacy. After the 10-week programming lesson, both groups demonstrated comparable mean scores, with no statistically significant divergence in performance. Analysis of the Programming Self-Efficacy Scale revealed no significant between-group differences in post-test scores suggesting the intervention similarly affected both groups' self-perceived competence. Post-intervention analysis revealed that the 10-week programming instruction led to statistically significant improvements in students' self-efficacy levels relative to pre-lesson assessments. At the conclusion of the intervention, both the control group (trained in robotics coding) and experimental group (trained in mobile programming) had developed competent individuals who demonstrated productive software development capabilities and confidence in successfully executing required programming tasks.

A statistically significant difference was observed between the pre-test and post-test mean achievement scores of both the robotics coding control group and the Android programming experimental group. In this case, a statistically significant difference was identified between the groups' mean achievement scores. Consistent with these findings, the intervention appears to have enhanced students' interest in coding, fostered programming curiosity, and increased their willingness to engage in advanced computational studies.

No significant difference was found in performance outcomes between the control group (engaged in robotics coding activities) and the experimental group (focused on Android programming) in the Project Development Lesson.

"Attitudes toward the lesson" appeared as the highest-frequency theme in student dairy analysis. When examining subordinate themes, the data indicate favorable attitudes toward the lesson among control group participants. Analysis of participant responses showed "application-related information" to be the most

prevalent theme based on frequency counts. When examining subordinate themes, findings indicate that participants in the experimental condition possessed complete knowledge about each practical activity conducted during instruction. “Pre-lesson programming attitudes” was identified as the dominant theme in both groups' focus group analyses. Pre-lesson programming attitudes among students showed dichotomous positive/negative orientations before the 10-week instruction. Participants reported favorable views (enjoyment, interest, game-making experience) alongside unfavorable ones (no prior programming exposure or education). Findings indicate that the educational intervention effectively enhanced programming interest among secondary education students. The implementation of robotics and mobile programming applications resulted in students demonstrating significantly high levels of positive programming self-efficacy. Additionally, results indicate the intervention played a formative role in influencing participants' long-term educational and professional goals. Findings demonstrate that students' innovative mindset, technological curiosity and proactive learning orientation toward emerging knowledge collectively support the policy imperative for universal programming/IT education. Education systems should design developmentally appropriate, play-based programming curriculum beginning in preschool (ages 3-6), thereby empowering children to actively design their future pathways. Programming instruction must transcend cognitive development to nurture affective learning, enabling students to construct original solutions while engaging in creative self-exploration and identity formation. Programming-based lessons constitute an inherently interdisciplinary discipline that establishes connections across multiple fields of study. In alignment with 21st-century requirements, programming must be established as a fundamental discipline among basic educational lessons. The Ministry of National Education should establish protocols for documenting and disseminating all current and future programming initiatives (student projects and teacher trainings) through the EBA digital platform. Implementing a reward system based on activity engagement metrics will significantly enhance student interest in programming. Analysis of learner perspectives in programming education indicates a critical need for restructuring instructional approaches through an evidence-based educational model. The introduction of mandatory IT lessons at secondary education levels should be accompanied by progressively sophisticated curricula, given their demonstrated positive impact on scholastic performance. The creation of an innovative learning structure incorporating artificial intelligence/machine learning/deep learning (AI, ML and DL) fundamentals at the secondary level represents a strategic educational investment with substantial long-term returns. Workshop programs in schools require diversified delivery mechanisms - including competitive events and collaborative projects - to transcend traditional extracurricular constraints. To ensure equitable access to project work opportunities, governing bodies should provide adequate funding allocations to every educational institution.

REFERENCES

- Akkoyunlu, B. & Kurbanoglu, S. (2003). A Study On Teacher Candidates' Perceived Information Literacy Self-Efficacy And Perceived Computer Self-Efficacy. *Hacettepe University Journal of Education*, 24 (24), 1-10.
- Akpınar, Y. & Altun, A. (2014). The Requirement for Programming Instruction in Schools of the Information Society. *Elementary Education Online*, 13 (1), 1-4.
- Alagöz, N. B. (2020). *Metaphors In The Early Age Programming Teaching: Learning Programming With Metaphors*, (Master's Thesis). Ondokuz Mayıs University, Samsun.
- Altın, R. (2021). *Secondary School Students' Programming And Computational Thinking Skills: Traditional And Interdisciplinary Approaches To Teaching Programming*, (Master's thesis). Middle East Technical University, Ankara.
- Altun, A. & Mazman, S.G. (2012). Programlamaya İlişkin Öz Yeterlilik Algısı Ölçeğinin Türkçe Formunun Geçerlilik ve Güvenirlilik Çalışması. *Journal of Measurement And Evaluation In Education and Psychology*, 3 (2), 297- 308.
- Arabacıoğlu, T. (2006). *Internet Supported Programming Logic Instruction*, (Master's thesis). Gazi University Institute of science, Ankara.
- Arslan, B. & Gülnar, S. (2014). Java-Android Yazılım Mimarisi: Bir Masaüstü İle Çoklu Tablet Bilgisayar Haberleşme Uygulaması, Akademik Bilişim' 14 - XVI. Akademik Bilişim Konferansı Bildirileri, Mersin University.
- Aşkar, P. & Umay, A. (2001). Perceived Computer Self-Efficacy Of The Students In The Elementary Mathematics Teaching Programme. *Hacettepe University Journal of Education*, 21 (21), 1-8.

- Aytekin, A., Çakır, F.S., Yücel, Y.B. & Kulaözü, I. (2018). The Importance Of Algorithms In Human Life. *Eurasian Journal of Researches in Social and Economics (EJRSE)*, 5 (7), 151-162.
- Bandura, A. (1994). Self-Efficacy, In Ramachandran (Ed.), *The Encyclopedia Of Human Behavior* (pp. 71-81), Volume: 4, R-Z and Index, New York: Academic Press.
- Bozkurt, A. (2015). Mobil Öğrenme: Her Zaman, Her Yerde Kesintisiz Öğrenme Deneyimi. *Açık Öğretim Uygulamaları ve Araştırmaları Dergisi*, 1 (2), 65-81.
- Bybee, R. W. (2010). *The Teaching of Science: 21st Century Perspectives*. NSTA press.
- Cesur Özkara, E. & Yanpar Yelken, T. (2020). Programming Self-Efficacy Scale For High School Students: Development, Validation And Reliability. *Educational Technology Theory And Practice*, 10(2), 345-365. <https://doi.org/10.17943/etku.632606>
- Ceylan, V.K. (2020). *The Effect Of Scenario Based Learning Scratch Curriculum On Students' Computational Thinking Skills And Learning Outcomes Of Problem Solving And Programming Unit*, (Doctorate's thesis). Aydın Adnan Menderes University, Aydın.
- Christian, B. & Griffiths, T. (2017). *Algorithms to Live By: The Computer Science of Human Decisions*. 2nd Edition; Buzdağı Publishing: Ankara, 14-15.
- Creswell, J. W. & Clark, V. L. P. (2015). *Pesquisa de Métodos Mistos-: Série Métodos de Pesquisa*. Penso Editora.
- Çakıroğlu, Ü. & Kiliç, S. (2020). Assessing Teachers' PCK to Teach Computational Thinking Via Robotic Programming. *Interactive Learning Environments*. 31(2), 818-835. <https://doi.org/10.1080/10494820.2020.1811734>
- Davidsson, K., Larzon, L. & Ljunggren, K. (2010). *Self-Efficacy In Programming Among STS Students. Technical reports from Computer Science Education course of Uppsala University. Retrieved from <http://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports> (Access date:25/08/2025)*.
- Demirer, V. & Sak, N. (2016). Programming Education and New Approaches Around The World And In Turkey. *Journal of Theory and Practice in Education*, 12(3), 521-546.
- Eguchi, A. (2014). Educational Robotics For Promoting 21st Century Skills. *Journal of Automation Mobile Robotics and Intelligent Systems*, 8(1), 5-11. DOI:[10.14313/JAMRIS_1-2014/1](https://doi.org/10.14313/JAMRIS_1-2014/1)
- Ekiz, D. (2003). *Eğitimde Araştırma Yöntem ve Metotlarına Giriş*. Ankara: Anı Publishing.
- Elçiçek, M. (2019). *The Design and Evaluation of a Video-supported Online Learning Environment For Programming Instruction*, (Doctorate's thesis). Trabzon University Postgraduate Education Institute, Trabzon.
- Estonya'da Bilgisayar (2012). *Estonya'da Bilgisayar Dili 1'inci Sınıfa Girdi*. *Hürriyet Newspaper*. <http://www.hurriyet.com.tr/planet/21405007.asp> (Access date:25/08/2025).
- Gerják, I. (2017). Image Processing Algorithms In the Secondary School Programming Education, *Acta Didactica Napocensia*, 10(3), p.69-76.
- Gomes, A. & Mendes, A. (2007). Learning To Program- Difficulties And Solutions, *Proceedings Of The International Conference On Engineering Education- ICEE*, 3- 7 September 2007, Coimbra.
- Göksoy, S. & Yılmaz, I. (2018). The Opinions Of Information Relations Teacher And Their Students With Regard To Lessons Of Robots And Decoding. *Duzce University Journal of Social Sciences*, 8(1), 178-196.
- Grafinger, D. J. (1988). *Basics Of Instructional Systems Development*. INFO-LINE Issue 8803. Alexandria: American Society for Training and Development.
- Gustafson, K.L. & Branch, R.M. (2002). *Survey Of Instructional Models*. New York: Eric Clearinghouse on Information & Technology.
- Gülbahar, Y. & Kalelioğlu, F. (2018). Information And Communication Technologies And Computer Science: The Process Of Curriculum Development. *National Education Journal*, 47 (217), p.5-23.
- Günüç, S., Odabaşı, H. & Kuzu, A. (2013). The Defining Characteristics of Students of The 21st Century By Student Teachers: A Twitter Activity. *Journal of Theory & Practice In Education*, 9 (4), 436-455.
- Johnson, R. B. & Onwuegbuzie, A J. (2004). Mixed Method Research. A Research Paradigm Whose Time Has Come. *Educational Researcher*, 33(7), 14-26. DOI:[10.3102/0013189X033007014](https://doi.org/10.3102/0013189X033007014)
- Kalaycı, Ş. (2014). *SPSS Uygulamalı Çok Değişkenli İstatistik Teknikleri* (6th Edition). Ankara: Asil Publishing.
- Kalelioğlu, F. (2015). A New Way Of Teaching Programming Skills To K-12 Students: Code.org. *Computers in Human Behavior*. 52, p.200-210. <https://doi.org/10.1016/j.chb.2015.05.047>
- Karabak, D. & Güneş, A. (2013). Curriculum Proposal For First Class Secondary School Students In The Field Of Software Development. *Journal of Research in Education and Teaching*, 2(3), 21.

- Kert, S. B. & Uğraş, T. (2009). Simplicity and Fun in Learning to Program: The Case of Scratch. In *The First International Congress of Educational Research, Çanakkale, Turkey*.
- Kılınç, A. (2014). *The Using Of Robotic Technology In Teaching Of Light Unit 7th Grade*, (Master's thesis). Erciyes University, Kayseri.
- Konvalina, J., Stephens, L. & Wileman, S. (1983). Identifying Factors Influencing Computer Science Aptitude And Achievement. *AEDS I*, 16 (2), 106-112.
- Korkmaz, Ö. (2013). Prospective CITE Teachers' Self-efficacy Perceptions on Programming. *Procedia-Social and Behavioral Sciences*, 83, 639-643. <https://doi.org/10.1016/j.sbspro.2013.06.121>
- Köse, U. & Tüfekçi, A. (2015). Usage of An Intelligent Software System In Teaching Algorithm And Flowchart Concepts. *Pegem Journal of Education and Instruction*, 5(5), 569-586.
- Kurşuncu E. (2016). *Control Techniques for Two-Wheeled Balance Robots*, (Unpublished master's thesis). Karabuk University, Karabuk.
- Maja J Matari'c (1990). *A Distributed Model For Mobile Robot Environment-Learning And Navigation*, (Master's thesis). Massachusetts Institue of Technology, Cambridge, MA.
- Mayring, P. (2007). Introduction: Arguments for mixed methodology. In P. Mayring, G. L. Huber, L. Gurtler, ve M. Kieglmann (Eds.), *Mixed methodology in psychological research*, Brill Sense, 1-4. https://doi.org/10.1163/9789087903503_002
- MEB. (2019). *Information Technology Production Curriculum*. Ministry of Education. <http://www.meb.gov.tr>. (Access date:25/08/2025).
- MEB-TTKB (Millî Eğitim Bakanlığı Talim Terbiye Kurulu Başkanlığı), (2012). *Bilişim Teknolojileri ve Yazılım Dersi Öğretim Programı*, [Çevrimiçi] <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>. (Access date:25/08/2025).
- Meisalo, V., Suhonen, J., Torvinen, S., & Sutinen, E. (2002). Formative Evaluation Scheme For A Web-based course design. In *ACM SIGCSE Bulletin*, 34(3), 130-134). <https://doi.org/10.1145/637610.544454>
- Miles, M. B. & Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook* Sage.
- Newsted, P. R. (1975). Grade And Ability Predictions In An Introductory Programming Course. *ACM SIGCSE Bulletin*, 7 (2), p.87-91. <https://doi.org/10.1145/382205.38289>
- Numanoğlu, M. & Keser, H. (2017). Robot Usage in Programming Teaching - Mbot Example. *Bartın University Journal of Faculty of Education*, 6(2), p.497. Doi: 10.14686/buefad.306198
- Okuyucu, M. O. (2019). *A Study On The Effect Of Robotic Coding Training On The High School Students' Metacognitive And Reflective Thinking Skills*, (Master's thesis), Binali Yıldırım University Institute of Science, Erzincan.
- Özçakmak, Ş. (2014). *Bilgisayar Kullanımı Çocukta Bağımlılık Yapar mı?* <http://www.haberturk.com/polemik/haber/973204-bilgisayar-kullanimi-cocukta-bagimlilik-yapar-mi?>. (Access date:25/08/2025).
- Özdemir, B. (2020). *Investigation Of The Relationship Between The Pedagogical Content Knowledge And Self Efficacy Of Preschool Teachers In Early Mathematics Education*, (Master's thesis), Istanbul Sabahattin Zaim University, Istanbul.
- Resnick, M. (2013). *Learn To Code- Code To Learn*. <https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn>. (Access date:25/08/2025).
- Robins, A., Rountree, J. & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172. <https://doi.org/10.1076/csed.13.2.137.14200>
- Rogers, C. B., Wendell, K. & Foster, J. (2010). The Academic Bookshelf: A Review Of The Nae Report. Engineering In K-12 Education. *Journal of Engineering Education*, 99(2), 179-181.
- Ruzzenente, M., Koo, M., Nielsen, K., Grespan, L. & Fiorini, P. (2012). *A Review Of Robotics Kits For Tertiary Education*. In Proceedings of International Workshop Teaching Robotics Teaching with Robotics: Integrating Robotics in School Curriculum, p.153-162.
- Salter, J. (2013). *Coding For Kids: Schoolchildren Learn Computer Programming*, *The Telegraph*, <http://www.telegraph.co.uk/technology/10468460/Coding-for-kidsschoolchildren-learn-computer-programming.html>. (Access date:25/08/2025).
- Sayın, Z. & Seferoğlu, S. S. (2016). Coding Education as a new 21st Century Skill and its Effect on Educational Policies. Academic Informatics Conference, Aydın, s. 3-5.

- Schillaci, M. A., Waitzkin, H., Carson, E. A., Lopez, C. M., Boehm, D. A., Lopez, L. A., et al. (2004). Immunization Coverage And Medicaidmanaged Care In M,New Mexico: A Multimethod Assessment. *Annals of Family Medicine*, 2(1), 13-21. DOI: [10.1370/afm.100](https://doi.org/10.1370/afm.100)
- Stallman, R. (2021). *Android ve Kullanıcıların Özgürlüğü*. Özgür Yazılım Vakfı. <https://www.gnu.org/philosophy/android-and-users-freedom.tr.html>. (Access date:25/08/2025).
- Sullivan, A. & Bers, M.U. (2016). Robotics In The Early Childhood Classroom: Learning Outcomes From An 8-Week Robotics Curriculum In Pre-Kindergarten Through Second Grade. *International Journal of Technology and Design Education*, 26(1), 3-20.
- Şimşek, K. (2019). *Investigation Of The Effects Of Robotic Coding Application On Science Achievement And Scientific Process Skills Of 6th Grade Students In Science Course Matter And Heat Unit*, (Master's thesis). Marmara University, Istanbul.
- Tashakkori, A. & Creswell, J. W. (2007). The new era of mixed methods [Editorial]. *Journal of mixed methods research*, 1(1), 3-7. <https://doi.org/10.1177/2345678906293042>
- Tashakkori, A. & Teddlie, C. (Eds.) (2003a). *Handbook of mixed methods in social & behavioral research*. Thousand Oaks, CA:Sage.
- Williams, L. & Cernochova, M. (2013). Literacy from scratch. In *Proceedings of the 10th IFIP World Conference on Computers in Education, WCCE* (pp. 17-27).
- Yauney, J., Bartholomew, S. R., & Rich, P. (2023). A Systematic Review of “Hour of Code” Research. *Computer Science Education*, 33(4), 512-544.
- Yıldırım, A. & Şimşek, H. (2003). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri*. Seçkin Publishing (11th Edition). Ankara.