



e-ISSN: 2630-631X

Article Type
Research ArticleSubject Area
Computer and
Instructional Technologies
EducationVol: 8 Issue: 63
Year: 2022 September
Pp: 1639-1650Arrival
31 July 2022
Published
30 September 2022
Article ID 64199Doi Number
<http://dx.doi.org/10.29228/smrj.64199>

How to Cite This Article
Mutlu, F.; Akkuş, E.,
Beysun, H., Şahin, Y.,
Kurtuluş, M.Ö. & Dursun,
H. (2022). "Bilişim
Teknolojileri Alanının
2018 ve 2020 Çerçeve
Öğretim Programındaki
Programlama Temelleri
Dersinin Karşılaştırmalı
İçerik Analizi",
International Social
Mentality and Researcher
Thinkers Journal,
(Issn:2630-631X) 8(63):
1639-1650



Social Mentality And Researcher
Thinkers is licensed under a
Creative Commons Attribution-
NonCommercial 4.0 International
License.

GİRİŞ







Bilişim sistemlerinin etrafımızı hızlıca sardığı günümüzde geçmişte hâkim olan sanayi toplumu yerini bilgi toplumuna bırakmıştır. Gelişen teknolojiyle günümüz dünyasında bireylerde aranılan nitelikler değişiklik göstermiştir. Bu nitelikler arasında programlama bilgisi birçok araştırmacı tarafından önemli olarak görülmektedir. Çamcan (2019) programlama bilgisinin ekonomiye olan katkısına yönelik çalışmasında, dijital dünyada küreselleşen ekonomide var olmanın aracı olarak programlama sektörünü göstermiştir. Kukul ve Gökçearsan (2014) programlama eğitiminin bireylerin problem çözme becerilerini olumlu yönde etkilediğini belirtmişlerdir. Yıldız vd. (2017) programlama sürecinin kişinin bilimsel düşünme becerilerini geliştirdiğini belirtmişlerdir. Altunçekiç vd. (2017) hayat boyu öğrenme kapsamında bilişim kurslarının bireyde sağladığı değişikliklere yönelik çalışmasında, programlama dili öğreniminin kişisel, sosyal ve toplumsal alanlarda kazançlar sağladığını belirtmişlerdir.

Hızla gelişen dünyada yeni yetişen genç nesillere mevcut programları tüketmekten çok, onlara yeni programları nasıl üretebileceklerini göstermek gerekmektedir. Teknoloji odaklı bir dünyada üretken bir ülke olabilmek adına bireylerin programlama becerileri önem kazanmaktadır. Günümüzde yazılım alanında yetişmiş insanlara duyulan ihtiyaç artmakta, bu konuda üretken ve yaratıcı bireylerin yetiştirilmesi ülkemizin geleceği için önemli hale gelmektedir (Demirer ve Sak, 2016).

Dijital ve elektronik aygıtların istenilen amaçları gerçekleştirebilmesi için programlanmaları gerekir. Programlar veya uygulamalar programlama dili aracılığıyla yazılırlar. Programlama dili özel kelime ve sembollerden oluşan komutlar bütünüdür. Farklı bilgisayar sistemleri ve teknolojik ortamlar için oluşturulmuş birçok programlama dili mevcuttur. Her programlama dilinin hedeflediği dijital ortamlar ve cihazlar farklılık göstermektedir. Her bir dilin kendine özgü deyim ve yazım kuralları mevcut olup bu kurallar çerçevesinde komutlar oluşturulur ve çalıştırılır. Bu komutların yazılması süreci kodlama ya da programlama, ortaya çıkan

Bilişim Teknolojileri Alanının 2018 ve 2020 Çerçeve Öğretim Programındaki Programlama Temelleri Dersinin Karşılaştırmalı İçerik Analizi

Comparative Content Analysis of Programming Primitives Course in 2018 and 2020 Framework Curriculum of Information Technologies

Fırat Mutlu¹  Enver Akkuş²  Hasan Beysun³  Yusuf Şahin⁴  M.Özgür Kurtuluş⁵  Hacı Dursun⁶ 
¹⁻²⁻³⁻⁴⁻⁵⁻⁶ M.E.B., Öğretmen, Diyarbakır, Türkiye

ÖZET

Mesleki ve Teknik Anadolu Lisesi Bilişim Teknolojileri Alanı 2018 ve 2020 Çerçeve Öğretim Programında yer alan Programlama Temelleri Dersinin karşılaştırmalı içerik analizine tutulduğu bu çalışmada nitel araştırma yaklaşımı benimsenmiştir. Araştırma yöntemi olarak doküman inceleme yönteminin kullanıldığı bu çalışmada veri analizinde içerik analizi kullanılmıştır. Programlama Temelleri Dersi, Bilişim Teknolojileri Alanında programlamayla ilk tanışma ve sonraki programlama öğrenimlerinde de önemli bir ders olarak görülmektedir. Analiz sonucunda 2018 öğretim programında yirmi kazanım olduğu görülürken 2020 öğretim programında ise yirmi dokuz kazanım olduğu görülmüştür. 2018 öğretim programında bilişim etiği ve bilgi güvenliği konuları yer alırken 2020 öğretim programında bu konulara yer verilmediği görülmüştür. Blok tabanlı programlama 2018 öğretim programında bulunmazken, 2020 öğretim programında yer aldığı görülmüştür. Yapılan değerlendirmeler sonucunda sonuç kısmında önerilere yer verilmiştir.

Anahtar Sözcükler: Çerçeve öğretim programı, programlama temelleri dersi, meslek lisesi

ABSTRACT

Qualitative research approach was adopted in this study, in which the Programming Primitives Course in the Vocational and Technical Anatolian High School Information Technologies Field 2018 and 2020 Framework Curriculum was subjected to comparative content analysis. In this study, in which document analysis method was used as a research method, content analysis was used in data analysis. Programming Fundamentals Course is seen as an important course in the first meeting with programming in the field of Information Technologies and in later programming learning. As a result of the analysis, it was seen that there were twenty acquisitions in the 2018 curriculum, while there were twenty-nine acquisitions in the 2020 curriculum. While IT ethics and information security issues were included in the 2018 curriculum, it was observed that these topics were not included in the 2020 curriculum. While block-based programming was not included in the 2018 curriculum, it was seen in the 2020 curriculum. As a result of the evaluations, suggestions are given in the conclusion part.

Keywords: Framework curriculum, programming primitives course, vocational high school

son ürün ise program ya da uygulama olarak adlandırılır (Ersoy vd., 2011). Programlamanın temelinde problem çözme yer almaktadır. Bu problem çözülürken kullanılan programlama dili ve problem çözme stratejisi önemli görülmektedir. Birey programlama dili bilgisini ve problem çözme stratejilerini kullanarak probleme çözüm geliştirmek için yollar arar. Bu süreçte birey, problem çözme basamaklarını programlama ortamı içerisinde uygulamaya çalışır. Papert (1993) programlama aşamasını problemi analiz etmek, problem çözümü için yollar bulmak, çözümü uygularken olası hataları ayıklamak ve çözümü test etmek olarak sıralamıştır. Programlama yaparken, problemlerin çözümü için gerekli stratejinin belirlenmesi, stratejinin uygulanması, çözüm için farklı yollar denenmesi ve en önemlisi de çözüm için en kestirme yolu belirlemek programlamada önemli görülmektedir (Aydın, 2022).

Programlamanın Tarihi

Bilinen en eski "bilgisayar" genellikle Charles Babbage'ın Analitik Motor tasarımı olarak kabul edilir. 1834'te icat edilen bu cihaz değişken bir sayısal işlemler dizisi ve verilerin dahili depolanmasını sağlayacak şekilde tasarlanmıştır. Günün mevcut teknolojisinin çok ötesinde olan cihaz Lady Augusta Ada tarafından programlanmış ve bu durum tarihte ilk programlama olarak görülmektedir (Rusch, 1969; Eddins, 1981; Friedman, 1992; Bergin, 2007). Analitik Motor Sipariş Kodu olarak adlandırılan programlama çalışmaları, günümüz elektronik bilgisayarların atası sayılan ENIAC makinası için geliştirilen ENIAC Coding System diliyle gelişimine devam etmiştir (Kabakçı, 2018). Programlamada kullanılan dil makine dili olarak adlandırılmış öğrenilmesi ve uygulanması özel bir kılavuz kitap aracılığıyla gerçekleştiriliyordu (War, 1996). Literatürde yapısal anlamda ilk programlama dili olarak kabul gören makine dili assembly dili olarak isimlendirilmiştir (Hyde, 2003; Irvine, 2003; Hohl ve Hinds, 2014). Assembly dilinde program yazmak iyi bir hafıza ve odaklanma gerektiriyordu. Bu durum dilin evrensel boyutta öğrenilmesine engel teşkil etse de günümüzde mikrodenetleyicili sistemlerde hala kullanılmaktadır (Çalışkan, 2019). 1950-1960 yılları arası programlama dilleri açısından verimli bir dönem olmuştur. Fortran, COBOL, LISP, BASIC ve Algol bu dönemde geliştirilen diller arasında yer almaktadır. Fortran ve COBOL 20 yıla kadar aktif olarak kullanılmış ve günümüzde de bilimsel çalışmalarda hala kullanılmaktadır (Ece, 2021). 1970 yılında geliştirilen pascal yazılımcılara daha özgür ortamlar sağlıyordu ve öğrenilmesi nispeten önceki dillere göre daha kolaydı. 1972 yılında ise tüm zamanların en çok kullanılan programlama dili olan C geliştirilmiştir. Yapılandırılmış programlama için tasarlanmış genel amaçlı bir dil olan C dili konuşma diline yakın olması en önemli artışı olarak görülmektedir (thussong, 2015). 1983'te C dilinin nesneye yönelik yüksek seviyeli dili olan C++ geliştirildi. C++ dili; oyun, video düzenleme, grafik işlemleri ve işletim sistemleri gibi birçok alanda kullanıldı ve günümüzde hala kullanılmaya devam edilmektedir. Hazır kütüphaneler içermesi, hızlı işleme ve derlemeye sahip olması en avantajlı özellikleri olarak görülmektedir (Staff, 2021). 1991 yılında Phyton, 1995 yılında Java ve 1995 yılında PHP dili geliştirilmiş. Geliştirilen bu dillerin en önemli ortak özellikleri lisans ücretlerinin olmayışı ve komutların günlük konuşma diline eş seviyede yer alması olarak gösterilebilir (Lestal, 2020). 2000 yılında Microsoft tarafından geliştirilen C# , C++'ın bilgi işlem yeteneğini ve Visual Basic'in sadeliğini içeren bu dil günümüz Microsoft ürünlerinin masaüstü uygulamalarını geliştirmek için kullanılmakta ve yazılımcılar arasında en popüler programlama dilleri arasında yer almaktadır (Beyer, 2019). Günümüzde kullanılan programlama dillerine bakıldığında, yeni programlama dillerinin eski dillerde kullanılan kavramlar üzerine inşa edildiği anlaşılmaktadır. Sonradan geliştirilen programlama dilleri genelde programcıların işini basitleştirmeyi ve programlama dili öğrenimini kolaylaştırmayı amaçlar. Programlanması gereken aygıt sayısının giderek artması programlamayı bilen insan kaynağına olan ihtiyacı da arttırmaktadır. Bu durum programlama öğrenimi ve öğretiminin günümüzde olduğu gibi gelecekte de önemini koruyacağı düşünülmektedir.

Programlama Öğretimi

Programlama eğitimi, hem küresel ekonomi üzerindeki olumlu etkisi hem de yeni ve gelişmiş teknolojik gelişmelerin bir sonucu olarak dünya çapında hızla trend olmuştur. Programlama, günümüz eğitim programlarının, özellikle K-12 eğitiminin en zorlayıcı bileşenlerinden biri olarak kabul edilmiştir. Okullarda programlama öğretiminin uygun pedagojik bakış açısı ve yaklaşıma uygun olarak yapılması için eğitim programcılarının ve uygulayıcı olarak öğretmenlerin dikkatli olması gerekmektedir (Altın, 2021). Öğrencilerin programlamayı zor buldukları kabul edilen bir gerçektir (Milne ve Rowe, 2002; Robins vd., 2003; Costelloe, 2004; Dasso vd., 2005; Saeki vd., 2011; Kalelioglu ve Gülbahar, 2014; Kalelioğlu, 2015) ve bu durumu değiştirmek amacıyla programlama öğretiminin geleneksel yaklaşımdan öğrencilerin daha aktif olduğu ve ilgi çekici bir öğretime geçmek için birçok çalışma yapılmaktadır.

Vihavainen vd. (2011) , programlama öğrenmenin zor bir süreç olduğunu, usta-çırak ilişkisiyle, birlikte yaparak ve akran etkileşimiyle öğrenmenin sağlanabileceğini belirtmişlerdir. Tillmann vd. (2012) bilgisayar programlamanın ve dolayısıyla programlama öğretiminin günümüzün her anında kullandığımız mobil

cihazlarla yapılması gerektiğini belirtmişlerdir. Araştırmacılar yeni bir mobil programlama ortamı olan TouchDevelop'u tanıttıkları çalışmalarında mobil öğrenme ortamının yarattığı fırsatlara değinmişlerdir. Arawjo vd. (2017) kodların soyut olmasının programlama öğretimi ve öğreniminde engeller teşkil ettiğini, oyun tabanlı programlama etkinlikleriyle kodların somutlaştırılarak programlanma öğreniminin sağlanabileceğini belirtmişlerdir. O'Kelly ve Gibson (2006) programlama öğreniminin bireyin problem çözme becerilerini geliştirdiğine yönelik çalışmalarında programlama öğreniminin dijital robotlar aracılığıyla gerçekleştiğinde olumlu dönütler aldıklarını belirtmişlerdir. Programlama öğretimine yönelik araç, yöntem ve teknik geliştirmeye yönelik çalışmaların görselleştirme üzerine yoğunlaştığı görülmektedir. Myers (1986); Baldwin ve Kuljis (2001); Lahtinen (2006); Rudder vd. (2007); Çisar vd. (2011); Ben-Ari (2013); Ergashev (2021); Mladenović vd. (2021) programlamada görsel araçların ve efektlerin kullanımının dikkat çekme, anlama ve öğrenmeyi kolaylaştırdıklarını belirtmişlerdir.

Alan yazında erken yaşlarda programlama öğretimi veya geç yaşlarda programlama öğretiminde yatay geçişlerin sağlanması ve öğrenime yönelik algı ve tutumların pozitif yönde gelişmesinde blok tabanlı programlama araçlarının kullanıldığı görülmektedir. Mohamad vd. (2011); Charalampos Kyfonidis vd. (2017); Weintrop ve Wilensky (2017); Moors vd. (2018); Weintrop ve Wilensky (2018); Weintrop (2019); Xu vd. (2019) çalışmalarında, blok tabanlı programlamanın öğrenenlerin programlama mantığına odaklanmalarına rehberlik ederek programlamayı öğrenmelerini kolaylaştırdıklarını belirtmişlerdir. Programlama öğretiminde 3 boyutlu etkileşimli platformlar kullanmak bir diğer yöntem olarak görülmektedir. Alice bu platformlar arasından en öne çıkanlardan biri olarak görülmektedir (Dann vd., 2006; Sykes, 2007; Wang vd., 2009; Daly, 2011). Öğrencilere animasyon filmler ve basit video oyunları geliştirmelerine olanak sağlayan Alice temel programlama kavramlarını öğrenmelerini kolaylaştıran yenilikçi bir 3B programlama ortamıdır (Zhang vd., 2014).

Araştırmanın amacı

Bu çalışmanın amacı; Mesleki ve Teknik Anadolu Lisesi (MTAL) Bilişim Teknolojileri Alanının (BTA) 2018 ve 2020 Çerçeve Öğretim Programında (ÇEÖP) yer alan Programlama Temelleri (PT) dersinin analiz edilip karşılaştırmalı olarak değerlendirilmesidir. Amaç doğrultusunda aşağıdaki sorulara cevap aranmıştır.

PT dersinin 2018 ve 2020 ÇEÖP 'teki;

- ✓ Ders saati sayısı, Sınıf Seviyesi ve toplam ders süresinin meslek dersleri içindeki oranı nedir?
- ✓ Konu dağılımı ve konulara ayrılan süre ağırlıkları nedir
- ✓ Kazanım sayılarının öğrenme birimlerine göre dağılımı nedir?
- ✓ Genel amacındaki değişim nedir
- ✓ Öğrenim kazanımlarındaki değişim nedir
- ✓ Derste kullanılan yardımcı programlar ve programlama dilindeki değişim nedir?

Araştırmanın Önemi ve Gereksesi

Günümüzde programlama bilgisinin sadece öğrenciler için değil iş dünyasındaki diğer çalışanlar anahtar bir yetkinlik haline geldiği görülmektedir. Dijital ekonomiyle beraber ülkelerin gelişmişlik seviyesinin değişkenlik gösterdiği günümüzde (Parlak, 2017) programlama konusundaki nitelikli bireylerin ülkelerin ekonomik gelişmişlik seviyesini etkilediği görülmektedir (Kieran vd., 2020). Ülkemizde programlamayla ilgili nitelikli ara eleman meslek liseleri aracılığıyla sağlanmaya çalışılmaktadır.

Programlamanın ortaya çıkış tarihinden günümüze programlama öğrenim yönteminden öğrenilen programlama dillerine değin birçok alanda değişimin olduğu görülmektedir. Programlama, yardımcı araç gereç ve uygun yöntemler kullanılmadığında öğrenciler tarafından zor olarak görülmektedir (Dasso vd., 2005; Kak, 2009; Busjahn ve Schulte, 2013). Programlama temelleri dersi meslek liselerinde programlamayla tanışma dersi olarak görülebilir. İlk defa programlamayla tanışan öğrencinin kazanacağı beceriler dersin öğrenme durumlarıyla bağlantılıdır. Öğretilen programlama dili, harcanan süre, yardımcı araçlar, kazanım, dersin amacı ve öğrenme birimlerinin güncelliği öğrenme sürecini etkilemektedir. Bu bağlamda meslek lisesi bilişim teknolojileri alanının 2018 ve 2020 çerçeve öğretim programında yer alan programlama temelleri dersinin analiz edilerek karşılaştırmalı değerlendirilmesi önemli görülmektedir.

YÖNTEM

Araştırmanın modeli

Bu çalışmada nitel araştırma yaklaşımı benimsenmiştir. Bir araştırma alanı olarak nitel araştırma, birçok disiplini ve konuyu aşar. Nitel araştırma, açık, düşünsel ve yorumlayıcı yöntemlerle keşif ve açıklamaya vurgu yaparak anlamı anlamaya ve aydınlatmaya odaklanır (Rafuls, 1997). Gözlenen örüntülerin, özellikle de

görünmez veya şaşırtıcı olanların nedenlerini keşfetmek için nitel araştırmalar önemli görülmektedir (Busetto vd., 2020). Yöntem olarak doküman inceleme yönteminin kullanılmıştır. Doküman inceleme, hem basılı hem de elektronik (bilgisayar tabanlı ve İnternet üzerinden iletilen) materyalleri gözden geçirmek veya değerlendirmek için sistematik bir yöntemdir (Bowen, 2009). Dokümanların tarafsız ve tutarlı analizini kolaylaştırması için içeriklerin titizlikle ve sistematik olarak analiz edilmesinde sık sık tercih edilen bir yöntemdir (Wach ve Ward, 2013). Doküman incelemesi veya analizi diğer nitel yöntemlerin kullanıldığı durumlarda ek bilgi kaynağı olarak da işe yarayabilir (Yıldırım ve Şimsek, 2018).

Evren ve Örneklem

Araştırmanın evreni Millî Eğitim Bakanlığı tarafından yayınlanmış bilişim teknolojileri alanına ait tüm çerçeve öğretim programlarından oluşmaktadır. Araştırmanın örneklemini ise 2018 ve 2020 yıllarında yayınlanmış çerçeve öğretim programından oluşmaktadır. Amaçlı örnekleme yöntemlerinden ölçüt örnekleminin kullanıldığı bu çalışmada ölçüt ise programların yakın tarihte uygulanmış ve hâlâ uygulamada olan programlar olmasıdır. Amaçlı örnekleme çalışmanın amacı bağlamında bilgi açısından zengin durumların seçilmesi durumudur (Büyüköztürk vd., 2012). Ölçüt örneklemede temel anlayış önceden belirlenmiş ölçütleri sağlayan tüm durumların çalışılmasıdır (Yıldırım ve Şimsek, 2018).

Verilerin Toplanması

Veri toplama sürecinde doküman inceleme yönteminin aşamaları takip edilmiştir. Bu aşamalar Yıldırım ve Şimsek (2018) tarafından şöyle sıralanmıştır: Dokümanlara ulaşma, özgünlüğü kontrol etme, dokümanları anlama, veriyi analiz etme, veriyi kullanma.

Bu adımlar dikkate alınarak Mesleki ve Teknik Anadolu Lisesi Bilişim Teknolojileri Alanı 2018 ve 2020 Çerçeve Öğretim Programında yer alan Programlama Temelleri Dersine ulaşılmıştır (MEB, 2022a). Çerçeve öğretim programlarında gerekli bilgiler edinildikten sonra ilgili dersin Ders Bilgi Formları incelenmiştir (MEB, 2022b). İncelenen veriler derlenmesinde araştırmacılar tarafından hazırlanan program karşılaştırma ve değerlendirme formu kullanılmıştır.

Verilerin Analizi

Karşılaştırma ve değerlendirme formlarından elde edilen veriler araştırmacılar tarafından hazırlanan karşılaştırma tablolarında içeriksel olarak analiz edilmiştir. İçerik analizinde kaynaklardan doğrudan alıntılar yapmak yerine verilerin derinlemesine analizi ve yorumlanarak okuyucuya aktarılması söz konusudur (Günbayı, 2019). İçerik analizinde benzerliği tespit edilen veriler belirli kavramlar ve temalar çerçevesinde bir araya getirilerek yorumlanır (Yıldırım ve Şimsek, 2018). Çalışmanın amacı doğrultusunda elde edilen bulgular tablolar halinde sunulmuştur.

BULGULAR

PT dersinin 2018 ve 2020 ÇEÖP'teki ders süreleri, sınıf seviyeleri ve meslek dersleri içindeki oranları Tablo 1'de karşılaştırmalı olarak gösterilmiştir.

Tablo 1. PT Dersinin Süreleri, Sınıf Seviyeleri ve Meslek Dersleri İçindeki Oranları

ÇERÇEVE ÖĞRETİM PROGRAMI	Haftalık Ders Saati	Sınıf Seviyesi	Meslek Dersleri İçindeki Oranı
2018	5	10	%35
2020	4	9	%36

Tablo 1'deki veriler incelendiğinde Programlama Temelleri dersinin 2018 ÇEÖP'te sınıf seviyesinin 10.sınıf, haftalık ders saatinin 5 saat ve meslek dersleri içindeki oranı %35 olarak görülürken 2020 ÇEÖP'te ise sınıf seviyesinin 9.sınıf, haftalık ders saatinin 4 saat ve meslek dersleri içindeki oranı %36 olarak görülmüştür. 2018 programında sınıf seviyesinin 10.sınıf olması programlama eğitimiyle erken tanışma bakımından dezavantaj oluşturduğu söylenebilirken, haftalık ders saatinin 5 saat oluşu 2020 programına göre avantajlı tarafı olarak görülebilir.

Programlama Temelleri Dersinin 2018 ve 2020 ÇEÖP'teki konu ve süre ağırlıklarının dağılım oranı Tablo 2'de gösterilmiştir.

Tablo 2. PT Dersinin Konu ve Süre Ağırlıklarının Dağılım Oranı

KONULAR	2018		2020	
	Ders Süresi	Oran	Ders Süresi	Oran
Bilişim etiği ve bilgi güvenliği	15	8,0	0	0,0
Problem çözme ve algoritma	15	8,0	16	11,0
Programlama dili temelleri	20	11,0	8	5,0
Veri yapıları ve operatörler	50	28,0	24	17,0

Karar ve döngü yapıları	45	25,0	24	17,0
Metotlar (Fonksiyonlar)	25	14,0	24	17,0
Hazır metotlar	10	6,0	12	8,0
Hata yakalama işlemleri	0	0,0	8	6,0
Blok tabanlı programlama	0	0,0	16	11,0
Dosya İşlemleri	0	0,0	12	8,0
TOPLAM	180	100,0	144	100,0

Tablo 2 incelendiğinde 2018 öğretim programında veri yapıları ve operatörler konusuna 50 saat (%28) ayrıldığı görülürken 2020 öğretim programında ise 24 (%17) saat ayrıldığı görülmektedir. Karar ve döngü yapıları konusuna 2018 öğretim programında 45 (%25) saat ayrılırken 2020 öğretim programında ise 24 (%17) saat ayrıldığı görülmektedir. Problem çözme ve algoritma konusuna 2018 öğretim programında 15 (%8) saat ayrılırken 2020 öğretim programında 16 (%11) saat ayrıldığı görülmektedir. Metotlar (Fonksiyonlar) konusuna 2018 öğretim programında 25 (%14) saat ayrılırken 2020 öğretim programında 24 (%17) saat ayrıldığı görülmektedir. Bilişim eğiti ve bilgi güvenliği konusuna 2018 öğretim programında 15 (%8) saat ayrılırken 2020 öğretim programında bu konuya yer verilmemiştir. 2018 öğretim programında Hata Yakalama İşlemleri, Blok Tabanlı Programlama ve Dosya İşlemleri konularına yer verilmezken 2020 öğretim programında sırasıyla 8(%6),16 (%11) ve 8(%6) saat yer verildiği görülmektedir.

Programlama Temelleri Dersinin 2018 ve 2020 Kazanım sayılarının öğrenme birimlerine göre dağılımı Tablo 3'te gösterilmiştir.

Tablo 3. PT dersinin kazanım sayılarının öğrenme birimlerine göre dağılımı

ÖĞRENME BİRİMİ	2018		2020	
	Kazanım Sayısı	Oran	Kazanım Sayısı	Oran
Bilişim etiği ve bilgi güvenliği	6	30,0	0	0,0
Kodlamaya hazırlık	4	20,0	9	31,0
Basit kodlar	3	15,0	4	14,0
Kontrol deyimleri	4	20,0	3	10,0
Metotlar	3	15,0	5	18,0
Blok programlama	0	0,0	3	10,0
Dosya İşlemleri	0	0,0	2	7,0
Hata yakalama işlemleri	0	0,0	3	10,0
TOPLAM	20	100,0	29	100,0

Programlama Temelleri Dersinin 2018 ÇEÖP'te kazanım sayısının en fazla 6 (%30) kazanım ile bilişim eğiti ve bilgi güvenliği konusunda olduğu görülürken 2020 ÇEÖP'te ise bu konuda kazanıma rastlanılmamıştır. 2020 Çerçeve öğretim programında en fazla kazanımın 9(%31) kazanım ile Kodlamaya Hazırlık konusunda olduğu görülürken aynı konuda 2018 ÇEÖP'te ise 4(%20) kazanım olduğu görülmüştür. 2018 ve 2020 ÇEÖP'te yer verilen ortak konuların (Basit Kodlar, Kontrol Deyimleri ve Metotlar) kazanım sayılarında neredeyse sayıca eşitlik olduğu görülmüştür.

Programlama Temelleri Dersinin 2018 ve 2020 ÇEÖP'te belirtilen amaçları Tablo 4'te gösterilmiştir.

Tablo 4. PT Dersinin 2018 ve 2020 ÇEÖP'te Belirtilen Amacı

ÇERÇEVE ÖĞRETİM PROGRAMI	Dersin Amacı
2018	İş sağlığı ve güvenliği tedbirlerini alarak temel programlama işlemlerini yapma ile ilgili bilgi ve becerilerin kazandırılması amaçlanmaktadır.
2020	Temel algoritma, görsel blok programlama ve programlama işlemlerini yapma ile ilgili bilgi ve becerilerin kazandırılması amaçlanmaktadır.

Tablo 4 incelendiğinde Programlama Temelleri Dersinin amacı 2018 ÇEÖP'te "İş sağlığı ve güvenliği tedbirlerini alarak temel programlama işlemlerini yapma ile ilgili bilgi ve becerilerin kazandırılması amaçlanmaktadır." diye belirtilirken 2020 ÇEÖP'te ise "Temel algoritma, görsel blok programlama ve programlama işlemlerini yapma ile ilgili bilgi ve becerilerin kazandırılması amaçlanmaktadır." şeklinde ifade edilmiştir. 2018 programında iş sağlığı ve güvenlik tedbirlerinin vurgulandığı görülürken 2020 öğretim programında ise böyle amacın benimsendiği görüşmemiştir. 2018 ve 2020 ÇEÖP'te ortak amacın "temel programlama işlemlerini yapmak" olduğu görülürken 2018 ÇEÖP'ten farklı olarak 2020 ÇEÖP'te görsel blok programlama ve temel algoritma ile ilgili bilgi ve becerilerin kazandırılmasının amaçlandığı görülmektedir.

Programlama Temelleri Dersinin 2018 ve 2020 ÇEÖP'te belirtilen kazanımları Tablo 5'te gösterilmiştir.

Tablo 5. PT Dersinin 2018 ve 2020 ÇEÖP'te Belirtilen Kazanımları

ÇERÇEVE ÖĞRETİM PROGRAMI	Dersin Öğrenme Kazanımları
2018	<ol style="list-style-type: none"> 1. Bilişim etiği ve bilgi güvenliği kavramlarını açıklar. 2. İş sağlığı ve güvenliği tedbirlerini alarak programlama diline uygun kodlama öncesi hazırlık işlemlerini yapar. 3. İş sağlığı ve güvenliği tedbirlerini alarak programlama diline uygun basit kodlar yazar. 4. İş sağlığı ve güvenliği tedbirlerini alarak probleme uygun kontrol deyimlerini kullanır. 5. İş sağlığı ve güvenliği tedbirlerini alarak probleme uygun metodlar ile çalışır.
2020	<ol style="list-style-type: none"> 1. Problem ve problem çözme ile ilgili kavramları açıklar ve bir problemi çözebilmek için gerekli becerileri kazanarak algoritma ve akış şemaları hazırlar. 2. Blok tabanlı programın ara yüzünü ve özelliklerini tanıtır ve programda yer alan görsel blokları kullanarak etkileşimli hikayeler, oyunlar ve animasyonlar programlamak amacıyla kodlama yapar. 3. Çeşitli alanlarda uygulama geliştirebilmek amacıyla kullanılan programlama dili yazılımını kurar ve açıklar. 4. Değişken, sabit ve operatörleri kullanarak farklı veri tipleriyle listeler oluşturur ve basit kod parçaları yazar. 5. Karar yapılarını açıklayıp, karar-kontrol yapılarını kullanarak basit kodları yazar. Döngü kavramını açıklayarak döngü türlerini karşılaştırır ve döngü işlemlerini yapar. 6. Fonksiyon kavramını ve programlama dili içerisinde nasıl tanımlanıp kullanılacağını öğrenir. Kendi tanımladığı fonksiyonlara parametre gönderip geri dönüşleri işler. 7. Programlama dili içerisinde tarih ve metin veri tipleri ile çalışmayı öğrenir. Metin veri tipi değişkenler içinde değişiklik yapmayı öğrenir. 8. Hata ve hata yakalama kavramlarını bilir, programda karşılaşılabileceği hatalara uygun çözümler üretir. 9. Programlama dili ile dosya oluşturma, var olan dosyaya erişme ve okuma becerileri kazanır. Dosya silme ve yedekleme konularında bilgi sahibi olur.

Tablo 5 incelende 2018 çerçeve öğretim programında bilişim etiği ve güvenliği ile iş sağlığı ve güvenliği kazanımlarının varlığı görülürken, 2020 çerçeve öğretim programında ise bu kazanımların olmadığı görülmektedir. 2020 çerçeve öğretim programında kazanımların daha detaylı olduğu görülürken kazanım sayısı bakımından da 2018 öğretim programından ileride olduğu görülmektedir.

Programlama Temelleri Dersinin 2018 ve 2020 ÇEÖP'te öğretim amaçlı belirtilen programlama dili ve öğretime yardımcı olması için kullanılan yardımcı program ve araçlar Tablo 6'da gösterilmiştir.

Tablo 6. PT Dersinde Kullanılan Yardımcı Programlar ve Öğretilen Programlama Dili

ÇERÇEVE ÖĞRETİM PROGRAMI	Programlama Dili	Akış Diyagramı	Blok Tabanlı Programlama
2018	C#(C Sharp)	FlowChart	
2020	Phyton	FlowChart	Scratch

Tablo 6 incelendiğinde Programlama Temelleri Dersi için 2018 ÇEÖP'te Visual C# dili öğretimi amaçlanırken 2020 ÇEÖP'te ise Phyton programlama dili öğretimi amaçlandığı görülmektedir. Her iki programda da akış diyagramları için yardımcı program olarak FlowChart programının kullanıldığı görülmektedir. 2018 ÇEÖP'te blok tabanlı programlama yer almazken 2020 ÇEÖP'te ise blok tabanlı programlama için Scratch kullanıldığı görülmektedir.

SONUÇ VE ÖNERİLER

Bilgisayar ve programlanabilir cihazlar günümüzün baskın teknolojileridir. Ekonomik, sosyal ve kişisel yaşamın neredeyse tüm yönleri artık onlarsız düşünülemez hale gelmiştir. Bilgisayar donanımı yazılım tarafından, yani programlama dillerinde yazılmış kodlarla kontrol edilir. Programlama veya kodlama bu nedenden dolayı dünya çapında milyonlarca insanın uğraştığı bir alan haline gelmiştir. Programlama öğretimi günümüzde birçok ülkenin eğitim programında farklı seviyelerden başlayarak yer almıştır. Programlama öğreniminin öğrenciler açısından zor olarak görüldüğü bir gerçek haline gelmiştir. Programlama öğretim dersinde öğretilen dilin güncelliği, amaç, kazanım, öğretimde kullanılan yöntem, teknik, yardımcı programlar ve araç gereçler verimliliği arttırmaktadır (Milne ve Rowe, 2002).

Ülkemizde ortaöğretim düzeyinde programlama eğitimi meslek liselerinin bilişim teknolojileri alanında yapılmaktadır. Bu çalışmada Bilişim Teknolojileri Alanının 2018 ve 2020 ÇEÖP'te yer alan Programlama Temelleri Dersi karşılaştırmalı içerik analizine tutulmuştur. Analiz sonucu aşağıdaki sonuçlara ulaşılmıştır.

2018 ÇEÖP'te sınıf seviyesi 10.sınıf olarak görülürken 2020 ÇEÖP'te ise 9.sınıf olarak görülmektedir. Programlama dili öğretiminin erken yaşlarda başlaması gerektiğiyle ilgili çalışmalar (Papert, 1993; Burton ve Bruhn, 2003; Kalelioğlu, 2015) göz önüne alındığında 2020 çerçeve öğretim programının hazırlanırken bu durumun dikkate alındığı söylenebilir. Programlama Temelleri Dersi 2018 ÇEÖP'te haftalık 5 saat iken 2020 ÇEÖP'te 4 saat olarak görülmektedir. Yıldız ve Kaya (2013) öğretmen görüşlerine başvurdukları

çalışmalarında bilişim teknolojileri öğretmenlerinin programlama öğretiminde haftalık kullanılan sürenin yetersiz olduklarını vurgulamışlardır. 2018 ÇEÖP'teki süreyle kıyaslandığında 2020 ÇEÖP'teki sürenin yetersiz olduğu söylenilebilir. Her iki Çerçeve öğretim programında programlama temelleri dersinin meslek dersleri içindeki oranının değişmediği görülmektedir. Ancak 2018 ÇEÖP'te programlama temelleri dersi başarılması gereken ders olarak kabul edilmezken 2020 ÇEÖP'te ise tersi bir durum söz konusudur. Bu durum programlama öğretimine karşı hassasiyetin arttığına yönelik bir değişiklik olarak görülebilir.

Programlama giriş dersi olarak kabul gören Programlama Temelleri Dersinde işlenen konular ve konulara ayrılan süreler önem görülmektedir. 2018 ÇEÖP'te bilişim etiği ve bilgi güvenliği konusuna 15 (%8) saat ayrılırken 2020 ÇEÖP'te ise bu konuya yer verilmediği görülmektedir. Literatürde Bilişim etiği ve bilgi güvenliği kazanımlarının öğrencilere kazandırılması gerektiğine yönelik birçok çalışma (Banerjee vd., 1998; Darcan ve Aydoğan, 2014; Fidan, 2016; Özdemir, 2019) yapılmıştır. Blok tabanlı programlama konusunun 2018 ÇEÖP'te olmadığı görülürken 2020 ÇEÖP'te yer almıştır. Blok tabanlı programlamanın özellikle giriş aşamasında programlama dili öğrenimini kolaylaştırdığı ve algoritma oluşturma aşamalarına katkı sağladığına yönelik çalışmalar (Turto, 2008; Wyeth, 2008; Patel vd., 2010; Wang vd., 2012; Dasgupta, 2015; Kurihara vd., 2015; Hermans vd., 2016; Kiesler, 2016; Koracharkornradt, 2017; C. Kyfonidis vd., 2017; Lopez ve Gutierrez, 2017; Price ve Barnes, 2017) yürütülmüştür. 2020 çerçeve öğretim programında blok tabanlı programlama konusunun yer alması yeni çerçeve öğretim programının güncel yönelimlere dönük olarak hazırlandığı söylenilebilir. Benzer konuların ders süreleri ve süre dağılımlarına bakıldığında Problem Çözme ve Algoritma Konusuna 2018 ÇEÖP'te 15 (%8) saat ayrıldığı görülürken 2020 ÇEÖP'te ise 16 (%11) saat ayrıldığı görülmektedir. İlgili konu programlama öğreniminde önemli bir aşama olarak değerlendirilmektedir (Namlı ve Şahin, 2017). Programlama Dili Temelleri konusunda 2018 ÇEÖP'te 20 (%11) saat ayrıldığı görülürken 2020 ÇEÖP'te ise 8 (%5) saat ayrıldığı görülmektedir.

Kazanım sayılarına göre karşılaştırma yapıldığında 2018 ÇEÖP'te toplam 20 kazanım olduğu görülürken 2020 ÇEÖP'te ise 29 kazanım görülmektedir. Kodlamaya Hazırlık öğrenme birinde 2018 ÇEÖP'te 4 (%20) kazanım varken 2020 ÇEÖP'te ise 9 kazanım olduğu görülmektedir. 2018 ÇEÖP'te en fazla kazanım Bilişim Etiği ve bilgi güvenliği öğrenme birimindeyken 2020 ÇEÖP'te ise 9 kazanım ile Kodlamaya Hazırlık öğrenme biriminde olduğu görülmüştür. Programlama öğreniminde hazırlık süreci ve hazırlık sürecinden öğrenciden beklenen performans verimliliği artırılabilir. Genel olarak 2020 ÇEÖP'te kazanım sayısının artırımına bağlı olarak öğrenciden beklenen performans sayısının arttığı söylenebilir.

Dersin amacındaki değişim durumuna bakıldığında 2018 ÇEÖP'te iş sağlığı ve güvenliği tedbirlerinin göz önünde bulundurulması temel programlama işlemlerinin amaçlandığını görmekteyiz. İş sağlığı ve işçi güvenliği dünya genelinde neredeyse tüm işyerleri ve kuruluşlar için uyulması gereken kurallar ve tedbirler öngörür. 2020 ÇEÖP'te iş sağlığı ve işçi güvenliği tedbirlerine vurgu yapılmazken görsel blok programlamayla ilgili amaç belirtimi önemli görülmektedir.

Öğrenme kazanımları bakımından karşılaştırmalı değerlendirme yapıldığında 2018 ÇEÖP'te iş sağlığı ve güvenliği tedbirlerinin tüm kazanımlarda yer aldığı görülmektedir. Basit kod, uygun kontrol deyimleri ve uygun metot kullanımı kazanımlarının belirtildiği ÇEÖP'te toplam 5 öğrenme kazanımı görülmektedir. Toplam dokuz öğrenme kazanımının olduğu 2020 ÇEÖP'te iş sağlığı ve güvenliği kazanımlarının olmadığı görülürken güncel programlama becerilerine yönelik kazanımların belirlendiği değerlendirilmektedir. Özellikle Blok tabanlı programlama kazanımlarının yer alması, öğrenciler tarafından zor olarak kabul edilen metot, karar kontrol yapıları, döngü ve veri yapılarına yönelik kazanımların yer alması önemli görülmektedir.

Programlama öğretiminde yardımcı programlar, öğretilen programlama dili ve platformlar önemli görülmektedir. 2018 ÇEÖP'te Visual C# programlama dilinin olduğu görülürken 2020 ÇEÖP'te ise Phyton programlama dilinin yer aldığı görülmektedir. Swacha ve Muszyńska (2011) her iki programlama dilini öğrenci görüşlerine göre değerlendirdikleri çalışmalarında öğrenciler açısından Visual C# dilinin öğrenilmesinin daha kolay olduğunu belirtmişlerdir. C# dili 2001 yılından itibaren popülerliğini korusa da son yıllarda phyton dili hız ve esnekliğinden dolayı daha çok tercih edilen bir dil haline gelmiştir (Malkoç, 2012). Next Technology şirketinin “öğrenme ve uygulama” kriterlerine göre belirledikleri en popüler programlama dili sıralamasında Phyton birinci olurken aynı sıralamada C# ise altıncı olmuştur (Technology, 2022). Berkeler Üniversitesinin “11 Most In-Demand Programming Languages in 2022” isimli çalışmasında Phyton dili ikinci olurken C# ise sekizinci olmuştur (Camps, 2022). Her iki çerçeve öğretim programında akış diyagramı çizimi için FlowChart programının olduğu görülürken, sadece 2020 ÇEÖP'te blok tabanlı programlama için program yer aldığı görülmüştür.

Bilişim Teknolojileri Alanı 2018 ve 2020 ÇEÖP'te yer alan Programlama Temelleri Dersinin karşılaştırmalı analizinin yapıldığı bu çalışmanın sonucunda aşağıdaki öneriler sıralanabilir:

- ✓ 2020 ÇEÖP'te yer alan dersin öğretmen ve öğrenci görüşlerine göre değerlendirildiği çalışmalar yapılabilir,
- ✓ 2018 ve 2020 ÇEÖP'te Programlama Temelleri Dersinin değerlendirme faaliyetlerinin karşılaştırıldığı çalışma yapılabilir,
- ✓ 2020 ÇEÖP'le uygulanmaya başlanılan blok tabanlı programlama yönelik öğretmen görüşlerine başvuru yapılan çalışma yapılabilir,
- ✓ 2020 ÇEÖP'te yer alan Programlama Temelleri Dersine yönelik başarı testi geliştirme çalışması yürütülebilir.

KAYNAKÇA

1. Altın, R. (2021). Secondary school students' programming and computational thinking skills: Traditional and interdisciplinary approaches to teaching programming [Ortadoğu Teknik Üniversitesi]. Ankara.
2. Altunçekiç, A., Yılmaz, Y., ve Üstündağ, M. (2017). Hayat boyu öğrenme kapsamında bilişim kurslarındaki kursiyerlerin eğilimleri. 11.Uluslararası Bilgisayar ve Öğretim Teknolojileri Eğitimi Sempozyumu (ICITS 2017), Malatya.
3. Arawjo, I., Wang, C.-Y., Myers, A. C., Andersen, E., ve Guimbretière, F. (2017). Teaching programming with gamified semantics. Proceedings of the 2017 CHI conference on human factors in computing systems,
4. Aydın, M. (2022). Programlama öğretimi için artırılmış gerçeklik tabanlı editörün geliştirilmesi, programlama ve transfer becerilerine etkisinin incelenmesi [Trabzon Üniversitesi]. Trabzon.
5. Baldwin, L. P., ve Kuljis, J. (2001). Learning programming using program visualization techniques. Proceedings of the 34th Annual Hawaii International Conference on System Sciences,
6. Banerjee, D., Cronan, T. P., ve Jones, T. W. (1998). Modeling it ethics: A study in situational ethics. *Mis Quarterly*, 22(1), 31-60.
7. Ben-Ari, M. (2013). Visualization of programming (1. ed.) <https://www.taylorfrancis.com/chapters/edit/10.4324/9780203078723-14/visualization-programming-mordechai-ben-ari>
8. Bergin, T. J. (2007). A history of the history of programming languages. *Communications of the ACM*, 50(5), 69-74. <https://dl.acm.org/doi/fullHtml/10.1145/1230819.1230841>
9. Beyer, C. (2019, 09.06.2022). A brief totally accurate history of programming languages. <https://medium.com/commitlog/a-brief-totally-accurate-history-of-programming-languages-d2e2b09553f8>
10. Bowen, G. A. (2009). Document analysis as a qualitative research method. *Qualitative research journal*.
11. Burton, P. J., ve Bruhn, R. E. (2003). Teaching programming in the oop era. *ACM SIGCSE Bulletin*, 35(2), 111-114.
12. Busetto, L., Wick, W., ve Gumbinger, C. (2020). How to use and assess qualitative research methods. *Neurological Research and practice*, 2(1), 1-10.
13. Busjahn, T., ve Schulte, C. (2013). The use of code reading in teaching programming. Proceedings of the 13th Koli Calling international conference on computing education research,
14. Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö., Karadeniz, Ş., ve Demirel, F. (2012). Örneklemeye yöntemleri. In.
15. Camps, B. B. (2022, 17.07.2022). 11 most in-demand programming languages in 2022. <https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages/>
16. Çisar, S. M., Pinter, R., ve Radosav, D. (2011). Effectiveness of program visualization in learning java: A case study with jeliot 3. *International Journal of Computers Communications & Control*, 6(4), 668-680.
17. Costelloe, E. (2004). Teaching programming the state of the art. The Center for Research in IT in Education. Dept. of Computer Science Education. Dublin: Trinity College. Recuperado de http://www.scss.tcd.ie/disciplines/information_systems/crite/crite_web/publications/sources/programmingv1.pdf.
18. Çalışkan, B. (2019, 08.06.2022). Kısaca programlama dillerinin tarihi (yeniden paylaşım). <https://blog.metu.edu.tr/e224336/2019/03/10/kisaca-programlama-dillerinin-tarihi-yeniden-paylasim/>

19. Çamcan, A. (2019). Yazılım sektörünün ekonomiye katkısı: Türkiye örneği Sosyal Bilimler Enstitüsü].
20. Daly, T. (2011). Minimizing to maximize: An initial attempt at teaching introductory programming using alice. *Journal of Computing Sciences in Colleges*, 26(5), 23-30.
21. Dann, W. P., Cooper, S., ve Pausch, R. (2006). *Learning to program with alice, brief edition*. Prentice-Hall, Inc.
22. Darcan, E., ve Aydoğan, H. (2014). Bilişim etiği. In *Siber etik* (pp. 137-151). Adalet Yayınevi. https://www.researchgate.net/profile/Emirhan-Darcan/publication/338533589_Cyber_Ethics/links/5e19ca5492851c8364c41ff9/Cyber-Ethics.pdf
23. Dasgupta, S. (2015). Block-based programming with scratch community data: A position paper. 2015 Ieee Blocks and Beyond Workshop (Blocks and Beyond), 97-98. <Go to ISI>://WOS:000380879300024
24. Dasso, A., Funes, A., Riesco, D. E., Montejano, G. A., Peralta, M., ve Salgado, C. H. (2005). Teaching programming. *I Jornadas de Educación en Informática y TICs en Argentina*,
25. Demirer, V., ve Sak, N. (2016). Dünyada ve türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama*, 12(3), 521-546.
26. Ece, N. (2021, 09.06.2022). Programlama dili: İlk programlama dilleri ve tarihi. <https://www.superprof.com.tr/blog/programlamanin-gecmisi-ve-dilleri/>
27. Eddins, J. M. (1981). A brief history of computer-assisted instruction in music. *College Music Symposium*,
28. Ergashev, N. (2021). Methods of using visualized educational materials in teaching programming languages in technical universities. *INNOVATION IN THE MODERN EDUCATION SYSTEM*.
29. Ersoy, H., Madran, R. O., ve Gülbahar, Y. (2011). Programlama dilleri öğretimine bir model önerisi: Robot programlama. *Akademik Bilişim*, 11, 731-736.
30. Fidan, M. (2016). Bilişim etiği boyutlarına göre bilişim teknolojileri ve yazılım dersi öğretim programı kazanımlarının incelenmesi. *Kastamonu Eğitim Dergisi*, 24(4), 1641-1654.
31. Friedman, L. W. (1992). From babbage to babel and beyond: A brief history of programming languages. *Computer Languages*, 17(1), 1-17.
32. Günbayı, İ. (2019). Nitel araştırmada veri analizi: Tema analizi, betimsel analiz, İçerik analizi ve analitik genelleme. Retrieved 02.07.2022 from <http://www.nirvanasosyal.com/h-392-nitel-arastirmada-veri-analizi-tema-analizi-betimsel-analiz-icerik-analizi-ve-analitik-genelleme.html>
33. Hermans, F., Stolee, K. T., ve Hoepelman, D. (2016). Smells in block-based programming languages. 2016 Ieee Symposium on Visual Languages and Human-Centric Computing (VI/Hcc), 68-72. <Go to ISI>://WOS:000392158000012
34. Hohl, W., ve Hinds, C. (2014). *Arm assembly language: Fundamentals and techniques*. Crc Press.
35. Hyde, R. (2003). *The art of assembly language* (Vol. 75). No Starch Press San Francisco, CA, USA.
36. Irvine, K. R. (2003). *Assembly language for intel-based computers*. Citeseer.
37. Kabakçı, E. (2018, 08.06.2022). Programlama dillerinin tarihçesi ve gelişim süreci. <https://emirhankabakci.com/programlama-dillerinin-tarihcesi/>
38. Kak, A. (2009). Teaching programming. In.
39. Kalelioglu, F., ve Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in education*, 13(1), 33-50.
40. Kalelioğlu, F. (2015). A new way of teaching programming skills to k-12 students: Code. Org. *Computers in Human Behavior*, 52, 200-210.
41. Kieran, R., Garcia, D., Medina, M., Lewis, D., ve Herrera, L. (2020). Overcoming the lack of qualified computer programming teachers: A field experiment in belize. *Journal of Learning and Teaching in Digital Age*, 5(2), 1-9.
42. Kiesler, N. (2016). Teaching programming 201 with visual code blocks instead of vi, eclipse or visual studio - experiences and potential use cases for higher education. *Edulearn16: 8th International*

- Conference on Education and New Learning Technologies, 3171-3179. <Go to ISI>://WOS:000402955903032
43. Koracharkornrad, C. (2017). Tuk tuk: A block-based programming game. Proceedings of the 2017 Acm Conference on Interaction Design and Children (Idc 2017), 725-728. <https://doi.org/10.1145/3078072.3091990>
 44. Kukul, V., ve Gökçearslan, Ş. (2014). Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi investigating the problem solving skills of students attended scratch programming course.
 45. Kurihara, A., Sasaki, A., Wakita, K., ve Hosobe, H. (2015). A programming environment for visual block-based domain-specific languages. Proceedings of the 2015 International Conference on Soft Computing and Software Engineering (Scse'15), 62, 287-296. <https://doi.org/10.1016/j.procs.2015.08.452>
 46. Kyfonidis, C., Moumoutzis, N., ve Christodoulakis, S. (2017). Block-c: A block-based programming teaching tool to facilitate introductory c programming courses. Proceedings of 2017 Ieee Global Engineering Education Conference (Educon2017), 570-579. <Go to ISI>://WOS:000405192300088
 47. Kyfonidis, C., Moumoutzis, N., ve Christodoulakis, S. (2017). Block-c: A block-based programming teaching tool to facilitate introductory c programming courses. 2017 IEEE Global Engineering Education Conference (EDUCON),
 48. Lahtinen, E. (2006). Integrating the use of visualizations to teaching programming. Proceedings of the conference Methods, Materials and Tools for Programming Education,
 49. Lestal, J. (2020, 09.06.2022). History of programming languages. <https://devskiller.com/history-of-programming-languages/>
 50. Lopez, J. M. S., ve Gutierrez, R. C. (2017). Computational thinking and visual programming through blocks in the elementary school classroom. *Educar*, 53(1), 129-146. <https://doi.org/10.5565/rev/educar.841>
 51. Malkoç, B. (2012). Temel bilimler ve mühendislik eğitiminde programlama dili olarak python. XIV. Akademik Bilişim Konferansı Bildirileri, 201.
 52. MEB. (2022a). Çerçeve öğretim programları. Retrieved 06.06.2022 from <http://meslek.eba.gov.tr/?p=Ogretim-Programi&tur=mtal>
 53. MEB. (2022b). Ders bilgi formları. Retrieved 06.06.2022 from <http://meslek.eba.gov.tr/?p=Ders-Bilgi-Formu&tur=mtal>
 54. Milne, I., ve Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, 7(1), 55-66.
 55. Mladenović, M., Žanko, Ž., ve Aglič Čuvić, M. (2021). The impact of using program visualization techniques on learning basic programming concepts at the k-12 level. *Computer Applications in Engineering Education*, 29(1), 145-159.
 56. Mohamad, S. N. H., Patel, A., Latih, R., Qassim, Q., Na, L., ve Tew, Y. (2011). Block-based programming approach: Challenges and benefits. Proceedings of the 2011 International Conference on Electrical Engineering and Informatics,
 57. Moors, L., Luxton-Reilly, A., ve Denny, P. (2018). Transitioning from block-based to text-based programming languages. 2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE),
 58. Myers, B. A. (1986). Visual programming, programming by example, and program visualization: A taxonomy. *ACM sigchi bulletin*, 17(4), 59-66.
 59. Namlı, N. A., ve Şahin, M. C. (2017). Algoritma eğitiminin problem çözme becerisi üzerine etkisi. *Recep Tayyip Erdoğan Üniversitesi Sosyal Bilimler Dergisi*, 3(5), 135-153.
 60. O'Kelly, J., ve Gibson, J. P. (2006). Robocode & problem-based learning: A non-prescriptive approach to teaching programming. Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education,

61. Özdemir, S. (2019). Lisans öğrencilerinin bilişim etiği konusundaki tutumlarının incelenmesi Ufuk Üniversitesi]. İstanbul.
62. Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. Basic Books, Inc.
63. Parlak, B. (2017). Dijital çağda eğitim: Olanaklar ve uygulamalar üzerine bir analiz. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 22(Kayfor 15 Özel Sayısı), 1741-1759.
64. Patel, K., Duval, M. A., Shipp, A. R., Knoll, A., ve Scott, P. H. (2010). Building blocks of cf: An education program for newly diagnosed cf patients and families. *Pediatric Pulmonology*, 457-457. <Go to ISI>://WOS:000282988800736
65. Price, T. W., ve Barnes, T. (2017). Position paper: Block-based programming should offer intelligent support for learners. 2017 Ieee Blocks and Beyond Workshop (B&B), 65-68. <Go to ISI>://WOS:000425946800014
66. Rafuls, S. E. (1997). Qualitative research methods. *Research in counseling & therapy*, 65-67.
67. Robins, A., Rountree, J., ve Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2), 137-172.
68. Rudder, A., Bernard, M., ve Mohammed, S. (2007). Teaching programming using visualization. *Proceedings of the Sixth IASTED International Conference on Web-Based Education*,
69. Rusch, R. B. (1969). *Computers: Their history and how they work*. Simon & Schuster
70. Saeli, M., Perrenet, J., Jochems, W. M., ve Zwaneveld, B. (2011). Teaching programming in secondary school: A pedagogical content knowledge perspective. *Informatics in education*, 10(1), 73-88.
71. Staff. (2021, 09.06.2022). Computer programming languages. <https://www.computerscience.org/resources/computer-programming-languages/#c-plus-plus>
72. Swacha, J., ve Muszyńska, K. (2011). Python and c#: A comparative analysis from students' perspective. *Annales Universitatis Mariae Curie-Skłodowska. Sectio AI, Informatica*, 11(1).
73. Sykes, E. R. (2007). Determining the effectiveness of the 3d alice programming environment at the computer science i level. *Journal of Educational Computing Research*, 36(2), 223-244.
74. Technology, N. (2022). Top 10 programming languages in 2022. <https://nexttechnology.io/top-10-programming-languages-for-2022/>
75. thussong. (2015, 09.06.2022). Programming languages through the years. <https://www.thesoftwareguild.com/blog/history-of-programming-languages/>
76. Tillmann, N., Moskal, M., De Halleux, J., Fahndrich, M., Bishop, J., Samuel, A., ve Xie, T. (2012). The future of teaching programming is on mobile devices. *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*,
77. Turto, T. (2008). Building blocks for a web programming language. *Proceedings of the 34th Euromicro Conference on Software Engineering and Advanced Applications*, 310-317. <https://doi.org/10.1109/Seaa.2008.15>
78. Vihavainen, A., Paksula, M., ve Luukkainen, M. (2011). Extreme apprenticeship method in teaching programming for beginners. *Proceedings of the 42nd ACM technical symposium on Computer science education*,
79. Wach, E., ve Ward, R. (2013). Learning about qualitative document analysis. *Institute of Development Studies*. <https://opendocs.ids.ac.uk/opendocs/handle/20.500.12413/2989>
80. Wang, D. L., Zhang, Y., Gu, T. Y., He, L., ve Wang, H. A. (2012). E-block: A tangible programming tool for children. *Adjunct Proceedings of the 25th Annual Acm Symposium on User Interface Software and Technology*, 71-72. <Go to ISI>://WOS:000325072100028
81. Wang, T., Mei, W., Lin, S., Chiu, S., ve Lin, J. M. (2009, 18-21 Oct. 2009). Teaching programming concepts to high school students with alice. 2009 39th IEEE Frontiers in Education Conference,
82. War, I. (1996). *History of programming languages*. In.

83. Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM*, 62(8), 22-25.
84. Weintrop, D., ve Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1-25.
85. Weintrop, D., ve Wilensky, U. (2018). How block-based, text-based, and hybrid block/text modalities shape novice programming practices. *International Journal of Child-Computer Interaction*, 17, 83-92.
86. Wyeth, P. (2008). How young children learn to program with sensor, action, and logic blocks. *Journal of the Learning Sciences*, 17(4), 517-550. <https://doi.org/10.1080/10508400802395069>
87. Xu, Z., Ritzhaupt, A. D., Tian, F., ve Umaphy, K. (2019). Block-based versus text-based programming environments on novice student learning outcomes: A meta-analysis study. *Computer science education*, 29(2-3), 177-204.
88. Yıldırım, A., ve Şimsek, H. (2018). Sosyal bilimlerde nitel araştırma yöntemleri Seçkin Yayıncılık.
89. Yıldız, M., Çiftçi, E., ve Karal, H. (2017). Bilişimsel düşünme ve programlama. *Eğitim teknolojileri okumaları*, 75-86.
90. Yıldız, M., ve Kaya, Z. (2013). Meslek liselerindeki programlama temelleri dersi programının değerlendirilmesi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 2(2), 359-368.
91. Zhang, J. X., Liu, L., de Pablos, P. O., ve She, J. (2014). The auxiliary role of information technology in teaching: Enhancing programming course using alice. *The International journal of engineering education*, 30(3), 560-565.