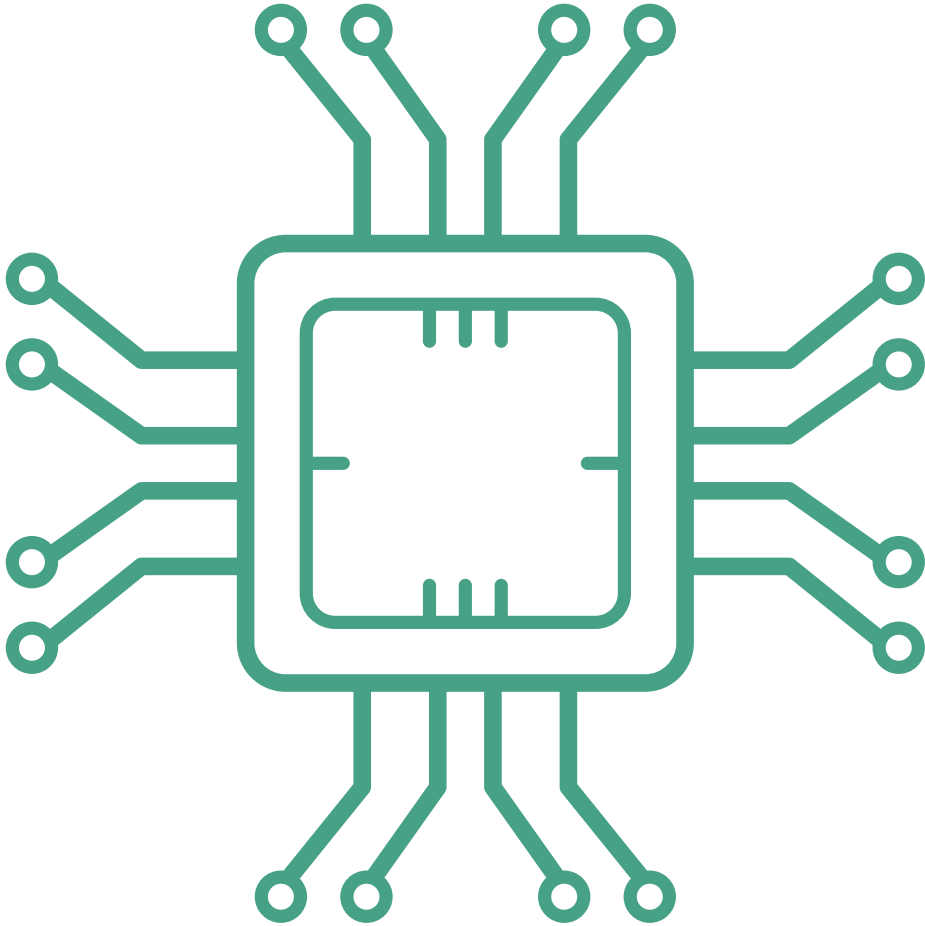




ÇEVİRİM İÇİ YAZILIM TEKNOLOJİLERİ

ORTAOKUL



TÜBİTAK Deneyap Kitapları 18

**Çevrim İçi Yazılım
Teknolojileri**
ORTAOKUL

Doç. Dr. Caner ÖZCAN
Doç. Dr. Rafet DURGUT
Dr. Öğr. Üyesi Sevil ORHAN ÖZEN
Bilişim Teknolojileri Öğrt. Sercan ÖZEN

© Türkiye Bilimsel ve Teknolojik Araştırma Kurumu, 2023

Bu kitabın bütün hakları saklıdır.
Yazılar ve görsel malzemeler, TÜBİTAK'tan yazılı izin alınmadan
tümüyle veya kısmen çoğaltılamaz ve yayımlanamaz.
Kitabın PDF formatındaki elektronik nüshasına
"https://yayinlar.tubitak.gov.tr/deneyap-atolyesi" adresinden ulaşılabilir.
TÜBİTAK Deneyap Kitapları *DENEYAP TÜRKİYE* Projesi kapsamında hazırlanmıştır.

ISBN: 978-605-312-532-7
Yayıncı Sertifika No: 47703

Yayın Tarihi: Ekim 2023

TÜBİTAK Başkanı: Prof. Dr. Hasan MANDAL
Bilim ve Toplum Başkanı: Ömer KÖKÇAM
Genel Yayın Yönetmeni: Fatma BAŞAR
Editör: Dr. İpek PİRİROĞLU GENCER
Düzeltili: Dr. Mustafa ORHAN
Telif İşleri Sorumlusu: Havva Hilal KAÇAR

TÜBİTAK Bilim ve Toplum Başkanlığı
Tunus Caddesi No: 80 Kavaklıdere 06680 Ankara
Tel: (312) 298 96 50
e-posta: deneyap@tubitak.gov.tr
https://yayinlar.tubitak.gov.tr/deneyap-atolyesi

DENEYAP

Teknoloji Atölyeleri

ÇEVİRİM İÇİ YAZILIM TEKNOLOJİLERİ

ORTAOKUL

Doç. Dr. Caner ÖZCAN

Doç. Dr. Rafet DURGUT

Dr. Öğr. Üyesi Sevil ORHAN ÖZEN

Bilişim Teknolojileri Öğrt. Sercan ÖZEN



İçindekiler

Yazarlar	i
İçindekiler	ii
Çevrim İçi Yazılım Teknolojileri Dersi Öğretim Planı Uygulama Kılavuzu	1
Çevrim İçi Yazılım Teknolojileri Dersi Bilgi Paketi	1
Dersin Amacı.....	1
Dersin Çıktıları.....	1
Neden C++ Programlama Dili	2
Ders Haftalık Planlaması	4
Dersin İşleniş Biçimi	5
Eğitmenin Çevrim İçi Yazılım Teknolojileri Dersinin Öğretim Süreçlerinde Kullanabileceği Öğretim Metotları ve Teknikleri	6
Eğitmenin Çevrim İçi Yazılım Teknolojileri Dersinin Süreçlerinde Kullanabileceği Değerlendirme Teknikleri.....	8
Eğitmenin Çevrim İçi Yazılım Teknolojilerinde Kullanacağı Programların Tanıtımı	10
Eğitmenin Çevrim İçi Yazılım Teknolojilerinde Kullanacağı Diğer Teknolojik Araçların Tanıtımı	11
Kaynakça	14
Hafta 1. Programlamaya Giriş	15
A. Giriş	16
B. Gelişme	18
Bilgisayarın Çalışması Neye Benzer?	18
Bilgisayar Verileri Nasıl Saklar?	20
Beni Bul ve Değiştir.....	22
Farklı Atama Türlerini Tanıyalım.....	26
Sayı Sistemlerini Keşfet.....	27
C. Sonuç.....	30
Hafta 1. Ders Materyalleri	33
Hafta 2. Algoritma Tasarımı	38
A. Giriş	39
B. Gelişme	40
Algoritmayı Tanıyorum	40
Algoritmaları Eşleştiriyorum.....	41
Algoritma Yazıyorum.....	42

Algoritma Sıralıyorum	43
Akış Diyagramı Çiziyorum.....	44
C. Sonuç.....	45
Hafta 2. Ders Materyalleri	51
Hafta 3. Algoritmada Değişkenler ve Değerleri	63
A. Giriş	64
B. Gelişme	64
Algoritma Terimlerini Keşfedelim	64
Otomatik Park Edelim.....	66
Değişkenler Değerlidir	67
Algoritmayı Adım Adım Çalıştırma.....	68
C. Sonuç.....	71
Hafta 3. Ders Materyalleri	75
Hafta 4. C++ Dilinde Değişken ve Veri Tipleri	76
A. Giriş	77
B. Gelişme	77
Değişkenlere İsim Verelim	77
Sabitleri Ayıralım	80
Örnekleri Çoğaltalım	81
Çıktıları Karşılaştıralım	81
Operatörlerle Yüzleşelim	82
C. Sonuç.....	84
Hafta 4. Ders Materyalleri	87
Hafta 5. Karar Mantık Yapıları	100
A. Giriş	101
B. Gelişme	102
Karar Yapılarını Tanıyalım.....	102
Görevleri Kodlayalım	104
İç İç Koşula Farklı Bir Bakış (Switch)	106
C. Sonuç.....	109
Hafta 5. Ders Materyalleri	111
Hafta 6. Döngü Yapıları.....	118
A. Giriş	119
B. Gelişme	119

Döngüleri Tanıyalım	119
Döngüleri Ne İçin Kullanıldığını Keşfetme.....	120
Döngü Görevlerini Kodlayalım	121
C. Sonuç.....	126
Hafta 6. Ders Materyalleri	129
Hafta 7. Diziler ve Katarlar	136
A. Giriş	137
B. Gelişme	137
Dizileri Tanıyalım	137
Dizilere Değer Verelim.....	139
Döngülerle Diziler	139
Etkinlik: Dizilerle Kodlayalım	141
Kodlama Ekibi	145
Farkı Keşfedelim.....	148
C. Sonuç.....	149
Hafta 7. Ders Materyalleri	153
Hafta 8. Fonksiyonlar	158
A. Giriş	159
B. Gelişme	159
Fonksiyonları Tanıyalım.....	159
Fonksiyonların Nasıl Kullanıldığını Keşfediyorum	161
Fonksiyonları Kullanarak Kodlama Yapalım	162
C. Sonuç.....	168
Hafta 8. Ders Materyalleri	170
Hafta 9. Nesneler	174
A. Giriş	175
B. Gelişme	175
Nesneni Çiz.....	175
Sınıfının Özelliklerini Tanı	179
Afişi Yeniden Tasarla	186
C. Sonuç.....	187
Hafta 9. Ders Materyalleri	190
Hafta 10. Nesne Yönelimli Programlama	191
A. Giriş	192

B. Gelişme	192
Yarış Benimle	192
Kod Satırlarını Tamamla	196
El Kaldır	201
C. Sonuçlar	205
Hafta 10. Ders Materyalleri	208
Hafta 11. C++ Programlama Dilinde Kütüphane Kullanımı ve Dosyalama İşlemleri.....	215
A. Giriş	216
B. Gelişme	216
C++ Programlama Dilinde Yerleşik Kütüphaneleri Keşfediyorum.....	216
Kütüphanelerdeki Bazı Fonksiyonları Kullanarak Kodluyorum	218
Neden Dosyalama İşlemleri Yaparız?	221
C++ Dilinde Dosyalama İşlemleri Yapıyorum.....	224
Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum	226
C. Sonuç.....	230
Hafta 11. Ders Materyalleri	233
Hafta 12. Proje Yarışması	243
Görevlerin Değerlendirilmesi	248
Görevlerin Çözümleri.....	249

Çevrim İçi Yazılım Teknolojileri Dersi Öğretim Planı Uygulama Kılavuzu

Bu kılavuzda Çevrim İçi Yazılım Teknolojileri dersinin öğretimi hakkında aşağıdaki başlıklara yer verilmektedir:

- Çevrim İçi Yazılım Teknolojileri dersinin hangi öğretim tasarım zemininde oluşturulduğu,
- Belirlenen öğretim tasarımının aşamalarının ne olduğu ve bu öğretim tasarımının aşamalarında eğitmenin süreçte nasıl rol alacağı,
- Eğitmenin Çevrim İçi Yazılım Teknolojileri dersinin öğretim süreçlerinde kullanabileceği öğretim yöntem ve teknikleri,
- Eğitmenin Çevrim İçi Yazılım Teknolojileri dersinin süreçlerinde kullanabileceği değerlendirme teknikleri,
- Eğitmenin Çevrim İçi Yazılım Teknolojileri dersinde kullanacağı programların tanıtımı,
- Eğitmenin Çevrim İçi Yazılım Teknolojileri dersinde kullanacağı dijital araçların tanıtımı.

Çevrim İçi Yazılım Teknolojileri Dersi Bilgi Paketi

Dersin Amacı

Bu dersin amacı, öğrencilere temel programlama mantığının öğretilmesi, problem çözümüne yönelik algoritmaların tasarlanması, tasarlanan algoritmaların C++ programlama dili kullanılarak kodlanmasıdır. Bu amaç doğrultusunda hedeflerimiz,

- Programlama temellerinin öğretilmesi,
- Verilen probleme yönelik uygun algoritma tasarımlarının geliştirilmesi,
- Akış diyagramından kodlamaya geçiş yapılması,
- C++ programlama dili kullanılarak problem çözümü,
- Nesne yönelimli programlama mantığının geliştirilmesi,
- C++ programlama dili ile proje geliştirilebilmesidir.

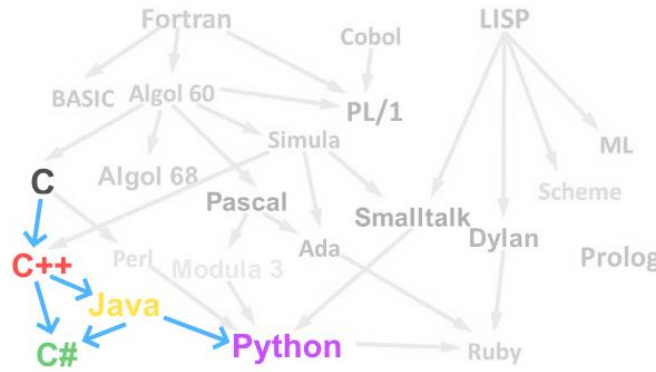
Dersin Çıktıları

Bu dersi alan öğrenciler;

- Gerçek hayat problemlerini algoritmaya dönüştürebilir ve akış diyagramları oluşturabilir.
- C++ programlama dilini kullanarak akış diyagramını koda dönüştürebilir.
- C++ veri tipleri, programlama komutları ve fonksiyonların kullanımını sağlayabilir.
- Nesne yönelimli programlama ile kodlama gerçekleştirebilir.
- İşaretçi ve dosyalama kavramlarını kullanarak proje tasarlayabilir.

Neden C++ Programlama Dili

C++, 1979 yılında Bjarne Stroustrup tarafından Bell Labs'da geliştirilen nesne yönelimli ve yüksek seviyeli, genel maksatlı programlama dilidir. C++ dili kullanılarak sistem yazılımları, özel yazılımlar, uygulamalar, sürücü yazılımları, kullanıcı taraflı yazılımlar ve gömülü firmware yazılımlar üretilmektedir. C++ dilinin orta seviyeli bir dil olmasından dolayı diğer yüksek seviyeli programlama dillerinden gerekli optimizasyon yapıldığında daha performanslı olduğu söylenebilir. C++ dilini öğrenir, mantığını anlarsak bu dilden etkilenerek oluşmuş programlama dillerini de temel seviyede öğrenmemiz kolay olur. Birçok üniversitede programlamaya giriş dersi olarak C++ eğitimi verilmektedir. Günümüzün en başarılı programcılarının çoğu C veya C ++ ile kod yazmayı öğrenmeye başlamıştır. Resim 1'de verilen programlama dillerine ait aile ağacı grafiğinde özellikle C#, Java ve Python dillerinin atasının C++ olduğunu görebilirsiniz.



Resim 1. Programlama dillerine ait aile ağacı

C++ programlama dilinin, farklı birçok uygulamada tercih edilen bir dil olmasını sağlayan iki temel özellik hız ve donanıma yakınlıktır. C++ programlama dili nesnelerin kullanımını sağlayan popüler bir dildir. Programcı için işleri kolaylaştıran yerleşik işlevlerle dolu bir kütüphane sunar. Java ve Python programlama dillerinden farklı olarak yorumlayıcı tabanlı değil derleyici tabanlı bir dildir ve bu nedenle bu dillerden nispeten daha hızlıdır. Basit içeriği ile yeni bir programlama dili öğrenmek isteyen programcılara hitap eder. Resim 2'de C++ programlamanın üstün özelliklerini görebilirsiniz.



Resim 2. C++ programlama dilinin üstün özellikleri

C++ önemli ve güçlü bir kullanım alanına sahiptir. Bu alanlardan bazılarını örnek verecek olursak;

- Gömülü Sistemler (Robotik Programlama) ve Elektronik Kartlar
- Masaüstü ve Hesaplama Uygulamaları
- Web Tarayıcı Oluşturma, Oyun Programlama, Derleyici Geliştirme
- Yeni Programlama Dili ve Yeni İşletim Sistemi Geliştirme

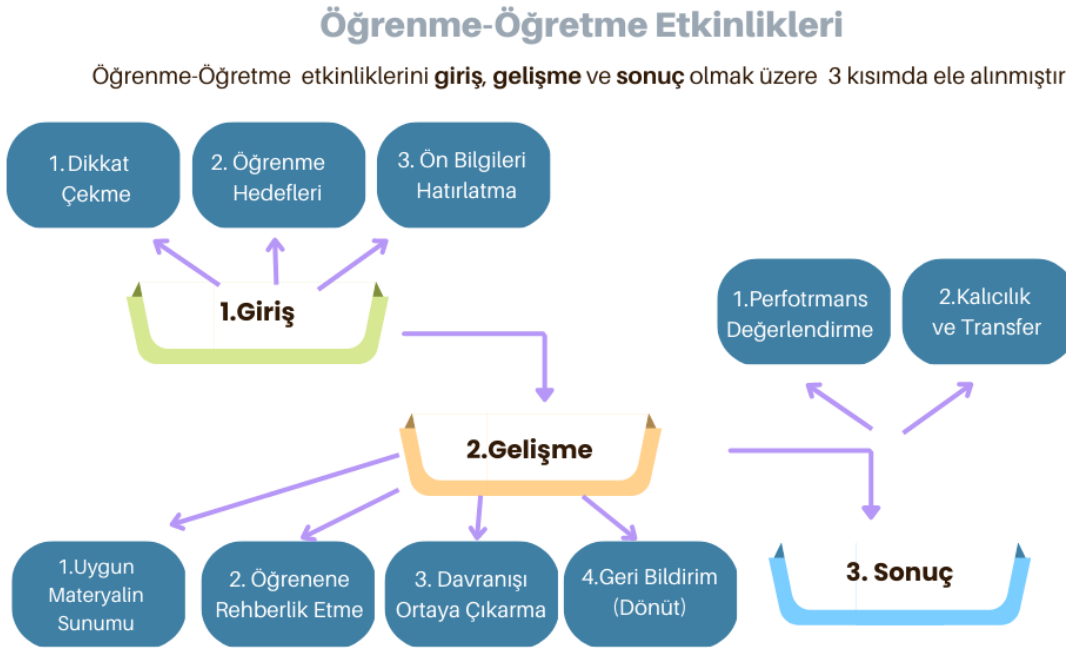
YouTube, Google, Amazon, Twitter, Facebook gibi uygulamaların yapımında C++ programlama dili de kullanılmıştır. Dünya çapında popülerliğini korumaktadır. Yüksek performanslı uygulamalar, oyunlar ve karmaşık araçlar yazmak için ve yazılan uygulamanın direkt olarak donanım ile haberleşmesini için C++ dili kullanmak gereklidir.

Ders Haftalık Planlaması

- Hafta 1: Programlamaya Giriş, Programlama Terimleri, Matematiksel İşlemler, Karşılaştırma İşlemleri, Mantıksal İşlemler, Sayı Sistemleri
- Hafta 2: Algoritma Tasarımı, Operatörler ve Kavramlar, Akış Şemalarına Giriş
- Hafta 3: Örnek Akış Şemaları, Kodlamaya Geçiş, Program Yazmanın Adımları (Hafta İçi Uygulama: C++ Programlama Dilinin Özellikleri, IDE (Derleyici) Kurulumu ve İlk Programlama)
- Hafta 4: C++ Tanımlamalar ve Operatörler, Değişken Tanımlama, Veri Tipleri, Sabitlerin Tanımlanması, Operatörlerin Kullanımı
- Hafta 5: Karar Verme Komutları (if-else, switch), If ifadesi, Switch İfadesi, Mantıksal Operatörler, Operatör Öncelikleri
- Hafta 6: Döngü Komutları (for, while, do-while), İç İç Döngüler, Rastgele Sayılar
- Hafta 7: Diziler, Karakter Katarları, Örnek Problem Çözümleri
- Hafta 8: Fonksiyonlar, Fonksiyon Sözdizimi, Fonksiyon Çağırma
- Hafta 9: Nesne Yönelimli Programlama, Sınıf Tanımlama, Metod Tanımlama, Nesne Değişkenleri Oluşturma, Yapıcı Yöntemler
- Hafta 10: Nesne Yönelimli Programlama, Nesne Değişkenleri Oluşturma, Yapıcı ve Yıkıcı Metotlar
- Hafta 11: Kütüphane Kullanımı ve Dosyalama İşlemleri
- Hafta 12: Proje Yarışması

Dersin İşleniş Biçimi

Çevrim İçi Yazılım teknolojileri dersinin öğrenme öğretme etkinlikleri için Gagne'nin dokuz aşamalı öğretim etkinlikleri modelinden ilham alınmıştır (Gagne, Wager, Golas ve Keller, 2005). Gagne'nin öğretim durumlarına ilişkin temel varsayımları arasında öğrenciler kendi deneyimleriyle yaparak yaşayarak öğrenerek öğretmenden daha etkin olmalıdır. Çevrim İçi Yazılım Teknolojileri dersi için bu varsayımdan yola çıkarak öğrencinin daha etkin olduğu ve ürettiği öğrenme sürecinin planlanması sağlanmıştır. Buna göre Çevrim İçi Yazılım Teknolojileri dersinin işleniş girişi, gelişme ve sonuç olmak üzere üç kısımda ele alınmıştır. Dersin işleniş biçimi Resim 3'teki gibi özetlenmiştir.



Resim 3. Dersin İşleniş Biçimi

Dersin giriş kısmında, öğrencilerin derse yönelik motivasyonlarını ve dikkatlerini artırabilmek için buz kırma tekniğiyle kurgulanmış oyunlar etkileşimli dijital araçlar ile kullanılır. Ardından derste öğrenilecek içerik ve kazanımlar hakkında bilgilendirme yapılır. Öğrenciler öğrenme hedefleri konusunda bilgilendirilir. Böylece öğrencilerin derse olan ilgisi ve motivasyonu artırılmaya çalışılır. Giriş kısmının son bölümünde ise bir hafta önceki konularla ilgili çeşitli sorular eşliğinde ön bilgileri hatırlatılarak, öğrencilerin o haftaki konuya yönelik hazırbulunuşluk düzeylerinin ve dikkatlerinin artırılması sağlanır.

Gelişme bölümünde ilgili haftanın eğitim öğretim faaliyetleri için hazırlanmış ve öğretmenler ile önceden paylaşılmış materyallerin uygulanması sağlanır. Gelişme bölümünde öğretmenlerden paylaşılan materyalleri öğrencilerin nasıl kullanacağına yönelik rehberlik yapması beklenir. Ardından, öğrencilerin beklenen kazanımlara yönelik davranışları bir ürün, geri bildirim ya da paylaşım yaparak göstermesi beklenir. Bunun için öğrencilere o haftanın konusu ile ilgili belirli algoritmik ya da kodlama üzerine görevler verilir. Öğrencilerin ilgili haftaki ders kazanımları ile

ilgili öğrenme görevlerini yerine getirirken, eğitimler dijital araçlardan faydalanarak öğrencilerin performanslarını yansıtmalarını ister. Eğitimler öğrencilere çevrim içi ortamda performanslarını yansıtmaları için gerekli ortamı oluşturur. Oluşturulan ortamlar için çevrim içi topluluk ortamı sunan dijital tartışma panoları, anlık paylaşım imkânı sunan dijital araçlar kullanılır. Böylece tüm katılımcıların bireysel olarak birbirlerinin paylaşımlarından haberdar olmaları sağlanır. Öğrenciler sohbet, anlık anket, çevrim içi kelime bulutu oluşturma gibi diğer sanal sınıf araçlarını kullanarak görevler üzerine performans ya da fikir paylaşımında bulunur. Gelişme bölümünün son kısmında ise öğrenme görevlerine yönelik öğrenci performans yansımaları ya da ürünleri üzerinden konu özetlenir. Çevrim içi ortamda öğrenciden gelen ürünler üzerinden konunun özetlenmesi önemlidir. Böylece eğitmenin öğrencilerin yanlış ve eksik öğrenmelerini görmeleri, bu eksik bilginin kapatılması ya da yanlış bilginin düzeltilmesi yoluyla yeni bilginin öğrenci tarafından anlamlandırılması sağlanır.

Sonuç kısmının ilk aşamasında gelişme bölümünde verilen görevler sonunda öğrencilerin performansına yönelik değerlendirme kriterleri ya da görev doğru cevabı öğrencilerle paylaşılır. Bunun üzerine öğrencilerin kriterleri ya da doğru yanıtları gelen paylaşımlar içinde incelemeleri istenerek çevrim içi oylama yapmaları sağlanır. Akran değerlendirmenin önem kazandığı bu süreçte, diğer bir değerlendirme yöntemi ise öz değerlendirmedir. Öğrenciler ilgili kriterlerin sunulması ya da doğru yanıtın iletilmesi üzerine, bireysel paylaşımlarında neyin eksik neyin doğru yaptıklarını sorabilir. Ortak hatalar ve doğru yapılanlar üzerine benzerlikler özetlenmesi sağlanır. Sonuç kısmının ikinci bölümünde ise öğrencilerden istenen davranışların kalıcılığını ve bir sonraki derslere yönelik bilgi transferinin sağlanması açısından süreli ödevler verilir. Süreli ödevlerin öğretmenler tarafından değerlendirilerek öğrencilerin eksikliklerinin belirlenip, bu eksikliklerin giderilmesi sürecin devamlılığı açısından oldukça önem arz etmektedir. Öğretmen geri bildirimlerinin önem kazandığı bu aşamada yapılan değerlendirmeler sonucunda belli becerilere göre kategorilere ayrılmış olan görevlere bağlı olarak öğrencilere rozet tanımlanır. Bu şekilde öğrencilerin becerilerle eşleştirilmiş rozetlere sahip olma durumuna göre portfolyo oluşturmaları desteklenir.

Eğitmenin Çevrim İçi Yazılım Teknolojileri Dersinin Öğretim Süreçlerinde Kullanabileceği Öğretim Metotları ve Teknikleri

Eleştirel Düşünme Teknikleri

Çevrim içi öğrenme görevlerinin öğrenciye sunumunda eleştirel düşünme tekniklerinden yararlanılmaktadır. Bu tekniklerden bazıları örnekleriyle şu şekildedir:

- Sıralama: Algoritma ya da kod satırlarını düzene koyma ya da sıraya dizme.
- Gruplama: Ortak özelliklere sahip nesneleri ya da dizileri gruplama.
- Talimat Verme: İki sayıyı toplama ile ilgili kod yazarı ekranına sözde kod yöntemiyle talimat verme.
- Tahmin Etme ve Çıkarımda Bulunma: Ekran çıktısı ya da kodun çözüm getirdiği problemi tahmin etme.
- Analiz ve Sentez: Diziler içinden eleman çıkarma, iki yazıyı birleştirme.

SCAMPER Tekniği

Çevrim içi öğrenme görevlerinin tasarımında kullanılan bir diğer teknik ise SCAMPER yönlendirilmiş beyin fırtınası tekniğidir (Çilci, 2019; İslim, 2011; Yağcı, 2012; Yiğitalp, 2014). Yaratıcı düşünme tekniği (Özyaprak, 2016) olarak da bilinen SCAMPER tekniği ile bir nesne ya da fikri farklı açılardan düşünmeyi sağlayacak sorular yöneltilir. Buna göre her bir harf, temelde farklı soru kalıplarına işaret etmektedir (Özyaprak, 2016). SCAMPER akrostişini oluşturan İngilizce kelimeler ve kullanım örnekleri şu şekildedir:

- S: Substitute (Yer değiştirme): Buradaki dizilerin yerlerine başka ne gelebilir?
- C: Combine (Birleştirme): Hangi kodlar birleştirilirse, problemi çözebiliriz? Ekran çıktısında hangi iki sayıyı bir araya getirip toplanmıştır?
- A: Adapt (Uyarlama): Ortamın sıcaklık değeri 10 derece azalsaydı, kodun ekran çıktısında tahmin edilen hava durumu nasıl olurdu?
- M: Modify, Minify, Magnify (Değiştirme, küçültme, büyütme): Bu kodu daha hızlı çalışır hâle getirmek için nasıl bir değişiklik yapabilirim? Nesne değişkeninin değerlerini daha büyük veririm, ekran çıktısı nasıl olur?
- P: Put to other uses (Başka amaçlarla kullanma): “Bu değişkeni başka hangi amaçla kullanabilirim?” ya da “bir nesneyi polimorfizm yoluyla başka hangi amaçlarla kullanabilirim?”.
- E: Eliminate (Yok etme, çıkarma): “Bir sınıf tanımlama içerisinden method çıkarsanız, ekran çıktısı nasıl olur?” ya da “diziden eleman çıkarsanız, ne değişir?”
- R: Reverse, Rearrange (Tersine çevirme ya da yeniden düzenleme): Bir dizideki elemanların dizilimini tersine çevirirseniz, ekran çıktısı nasıl değişir?

Çevrim İçi İş Birlikli Öğrenme

Öğrenciler çevrim içi öğrenme görevlerini tamamlarken çalışma odaları (breakout rooms) aracılığıyla küçük gruplar hâlinde çalışmaktadır. Burada bahsi geçen iş birlikli öğrenme sıradan bir grup çalışması değildir. Bunun nedenleri “Her Küçük Grup Çalışması İş Birlikli Öğrenme Değildir” alt başlığı altında ayrıntılı olarak açıklanmaktadır. Grup çalışmalarının iş birlikli öğrenme yapan öğrencilerin hem kendilerini hem de arkadaşlarını kapasitelerinin sonuna kadar geliştirmeye çalışmalarıdır. Bu, tek tek her öğrencinin öğretilenleri tam olarak öğrenmesinden farklı bir durumdur. Grup çalışması sırasında öğrenciler tek başlarına edinemeyecekleri, ancak başka biriyle etkileşerek kazanacakları öğrenme yaşantılarını deneyimler. Örneğin soru sorma, açıklama yapma, eleştirme, örnek verme gibi.

Çevrim içi iş birlikli yöntemin kullanıldığı grup çalışmalarında dijital tartışma panolarının grup temelli şablonları kullanılır. Süreçte grup temelli öğrenme görevlerinin yanı sıra, iş birlikli çalışma tekniklerinden çevrim içi istasyon tekniği ve çevrim içi ayrılıp birleşme (jigsaw) tekniği de kullanılır. İstasyon tekniği bütün katılımcıların her aşamada (her istasyonda) çalışarak bir önceki grubun yaptıklarına katkı sağladığı, yanıtı bir basamak ileri götürmeyi, yarım kalan işi tamamlamayı öğreten bir yöntemdir. İstasyonlar öğrencilerin eş zamanlı olarak çeşitli öğrenme aktivitelerini gerçekleştirebilecekleri merkezlerdir. Çevrim içi istasyon tekniğinde bu merkezler, dijital tartışma panolarında grup sütunlarındaki görevlerden oluşmaktadır. Buna göre çalışma odaları aracılığıyla birbirinden ayrılan öğrenciler örneğin birinci grup olarak panodaki birinci sütunu yaparken, birkaç dakika sonra, dijital panodaki ikinci sütundaki göreve yönlendirilir. Böylece ikinci sütunda yapılanlar yeni grup tarafından incelenir ya da tamamlanır.

Benzer şekilde yönetilen diğer bir iş birlikli teknik ayrılıp birleşmedir. Bu teknik ile öğrenciler 5-7 kişilik takımları çalışma odalarına dağıtarak oluştururlar. Akademik materyal (ünite) ya da konu gruplardaki öğrenci sayısına bölümlere (konuya) ayrılır. Her takıma aynı ünite (konu) verilir ve takımlardaki üyelerden ünite parçalarından (konulardan) birini seçmeleri istenir. Ünite parçaları dijital tartışma panosunda grup teması kullanılarak sütunlara dağıtılır. Her üye kendi konusunu inceleyip, öğrenir. Daha sonra farklı takımlarda aynı konuyu alan üyeler -gruplarından ayrılarak uzmanlık gruplarında bir araya gelirler; konu üzerinde tartışır. Burada eğitmenin her gruptan birer kişi alarak yeni çalışma odalarına dağıtması gerekir. Eğitmeninin başlangıçta oluşturulan takımları tekrar kurması için her takımı ve üyelerini kaydetmesi kolaylık olacaktır. Bu şekilde öğrencilerin kendi takımlarıyla geri birleşerek, takım arkadaşlarını, kendi konularıyla ilgili olarak bilgilendirmeleri istenir.

Çevrim İçi Yazılım Teknolojileri dersi için geliştirilen görevler iş birlikli çalışmayı gerektirirken, aynı zamanda istasyon ya da jigsaw tekniğine uygun olarak aşamalı görevler ya da grup görevleri şeklindedir. Çalışma odaları aracılığıyla rastgele odalara dağılma ya da eğitmen tarafından grupların oluşturulması, bir sonraki görevde grup değişimini kolaylaştırır. Dijital tartışma panolarında grup şablonu ise, yeni gelen grubun bir önceki grup yanıtlarını inceleyerek yeni görevde nasıl ilerleyeceklerini anlamaları için kullanışlıdır.

Eğitmenin Çevrim İçi Yazılım Teknolojileri Dersinin Süreçlerinde Kullanabileceği Değerlendirme Teknikleri

Oyun Temelli Düşünme, günlük veya dönemsel yaşam deneyimlerini, yarışma, keşif, senaryolaştırma veya iş birliğine dayalı eylemlere dönüştürmeye dayalı felsefe-düşünme biçimidir. Bu derste de birtakım oyunlaştırma öğelerinden yararlanılmaktadır. Bunlar her ders sonunda öğrenciler tarafından tamamlanan süreli görevlerinden alınacak beceri rozetlerinin kullanımı ile sağlanacaktır. Süreli görevleri, öğrencilerin görev içinde bir rutin oluşturmak için gerekli becerileri otomatikleştirmeye kadar tamamladıkları alıştırmadır. Her haftanın son saatinde bu öğrenme görevleri öğrenciye yeterli tekrarı sunmak ve süreç becerilerini değerlendirmek amacıyla süreli olarak kullanılır. Süre kullanılmasının temelinde oyunlaştırma öğeleri ile motivasyonun sağlanması hedeflenmektedir. Ayrıca öğrenciler bu süre içinde süreli görevlerden kendi seçimiyle istedikleri sayıda görevi tamamlama esnekliğine sahiptir. Diğer bir ifadeyle, öğrenciler haftanın sonunda 5 süreli görev varsa bunlardan 3'ünü istediği sıralamada yapmak isteyebilir. Eğitmenlerin bu noktada onları tüm süreli görevleri tamamlama konusunda teşvik etmesi beklenir. Tamamlanan her görevin ardından öğrenci bir beceri rozeti kazanmaktadır. Beceri rozetlerinin tanımlanması için ÖYS ortamı kullanılabilir ya da eğitmenlerin bir dijital ortamda öğrencilerin edindiği beceri rozetlerini kaydetmesi beklenir. Bu anlamda beceri rozetleri öğrencilere süreç sonunda, öğrenilen becerileri pratikleştirme imkânı veren yeni proje görevleri sunmak için kullanılmaktadır. Tüm haftalarda süreli görevlerin içeriğiyle uyumlu ve göreve tanımlı dört farklı rozet bulunmaktadır. Bunlar:

1. **Analizci Rozet:** Verilen problem için üretilen çözümlerin uygunluğunu kontrol eder ve varsa mantık hataların giderilmesini sağlar.



2. **Kodlayıcı Rozet:** Probleme uygun çözümlerin uygulamaya geçebilmesi için kodlanmasını sağlar.



3. **Tasarlayıcı Rozet:** Verilen probleme uygun çözümün nasıl olabileceği ile ilgili ön hazırlıkları yaparak gerekli algoritma ve akış diyagramlarının hazırlanmasını sağlar.



4. **Denetleyici Rozet:** Verilen problem için üretilen kodlamaların uygunluğunu kontrol eder ve varsa derleyici hatalarının giderilmesini sağlar.



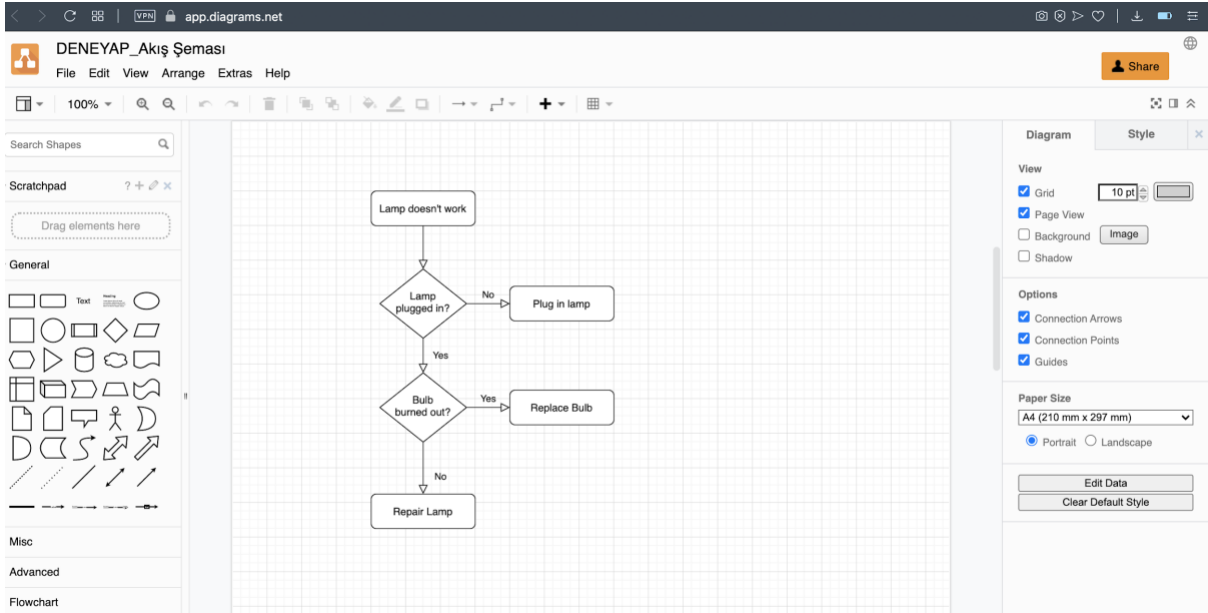
Beceri rozetleri verilen süreli görevleri süreçte değerlendirmeyi sağlamak amacıyla kullanılmıştır. Bu şekilde öğrencilerin etkinliklere katılımını artırmak hedeflenmektedir. Ayrıca öğrenciler tüm hafta boyunca birden fazla analizci ya da kodlayıcı rozeti kazanabilir. Örneğin dönem sonunda edindikleri analizci rozet sayıları öğrencinin portfolyosunda ya da özgeçmişinde yer alacaktır. Bu noktada öğrenciler dönem boyunca edindikleri rozet tür ve sayılarına göre dönem sonunda yer

alacakları proje görevlerinde ekip arkadaşlarını bulacaktır. Bu projeler öğrencilerin edindikleri becerilerin tamamına hitap eden bütüncül görevlerdir. Buradaki amaç ise oluşturulacak ekiplerin niteliğini artırmak adına birtakım yeterliliklere gelmiş öğrencilerin bir araya gelmesini sağlamak, ekip üyelerinin birbirlerini beceri anlamında tamamlamalarına imkân vermektir.

Eğitmenin Çevrim İçi Yazılım Teknolojilerinde Kullanacağı Programların Tanıtımı

Code::Blocks: Öğrencilerin, Çevrim İçi Yazılım Teknolojileri dersi kapsamında öğrenecekleri C++ programlama dilindeki programlarını yazmalarını ve derlemelerini kolay bir şekilde gerçekleştirebilmesi için Entegre Geliştirme Ortamı (Integrated Development Environment-IDE) ile derleyici özelliği olan ve her platform tarafından desteklenen Code::Blocks geliştirme ortamı tercih edilmiştir. Code::Blocks, geliştiriciler için işlevsel araçlarla tamamen yapılandırılabilir ve genişletilebilir açık kaynaklı ve ücretsiz bir IDE çözümüdür. Açık kaynaklı tasarımı sayesinde işlevlerinin büyük kısmı eklentilerle genişletilebilmektedir. Var olan gelişmiş araçlar sayesinde çok başarılı bir hata yakalama çerçevesi sağlamaktadır. Yazılımın derleyici eklentisi, yazılımcıların çok sayıda görevi birleştirmesini kolaylaştıran çalışma alanları sunmaktadır. Yazılım, platformlar arası bir çözümdür ve Mac, Windows ve Linux dahil olmak üzere farklı işletim sistemlerinde çalışır. C++ ile yazılmıştır ve çalışması için kısıtlayıcı kütüphaneler veya çevrilmiş bir dil gerektirmez. Code::Blocks kurulumu ders içeriğinde öğrencilere verilecek materyaller arasında mevcuttur.

Diagram Çizim Uygulaması: Dersin önemli teknoloji ortamlarından biri ücretsiz çevrim içi diyagram çizim uygulaması olan app.diagrams.net'tir. Bu uygulama Moodle sistemi içinde entegre şekilde kullanılmakta olup, öğrenciler tarafından ayrı bir kullanıcı kaydı ya da kurulum gerektirmemektedir. Öğrenciler uygulamaya farklı bir link üzerinden erişim sağlamaksızın, Moodle üzerinden aktif şekilde kullanabilmektedir. Bu uygulama ile öğrenciler Yazılım Teknolojileri dersindeki projeler için algoritmaların akış şemalarını bu dijital ortamda kolaylıkla çizebilecektir. Resim 4'te app.diagrams.net uygulamasının arayüzü verilmektedir.



Resim 4. App.diagrams.net uygulamasının arayüzü

Öğrenci çizimleri Google Drive, Dropbox, Onedrive, Github gibi dijital ortamlarla mevcut cihaz ya da bilgisayarda kaydedilebilir. Bu şekilde eğitmenin öğrenci çizimlerini takip etmesi daha kolay hâle gelecektir. Yazılım Teknolojileri dersinde öğrenci çizimlerinin kayıt ortamları için çoğunlukla Github kullanılmaktadır.

Eğitmenin Çevrim İçi Yazılım Teknolojilerinde Kullanacağı Diğer Teknolojik Araçların Tanıtımı

Öğrenme Yönetim Sistemi (ÖYS): Derste aktif şekilde kullanılacak olan temel ortamlardan biri öğrencilerin ilgili derse kullanıcı kaydı yapacakları bir Öğrenme Yönetim Sistemidir. Çevrim İçi Yazılım Teknolojileri dersleri öğrenme görevleri şeklinde öğrenciler tarafından tamamlanan ya da üretilen ürünlerle ilerlemektedir. Bu nedenle öğrenme görevleri ÖYS ortamında modüler bir yapıda hafta hafta sunulmaktadır.

Çevrim İçi Yazılım Teknolojileri dersi için ÖYS ortamının aşağıdaki amaçlara hizmet etmesi beklenir:

- Dersle ilgili genel duyuruların iletilmesi
- Öğrencilere sunulacak öğrenme materyallerinin paylaşılması
- Öğrenme görevi sırasında kullanılacak diğer teknolojik araçların iletilmesi
- Süreli görevler gibi bireysel öğrenme etkinliklerinin teslimi
- Grup olarak oluşturulacak ürün ya da tamamlanan görevlerin teslimi
- Eğitmenin öğrenci ürünlerini takip etmesi ve geri bildirim sağlaması
- Öğrencilerin kazandığı beceri rozetlerinin iletilmesi, kaydı ve raporlanması
- Öğrenci e-portfolyosunun raporlanması
- Öğrenci-eğitmen, öğrenci-öğrenci ve eğitmen-eğitmen etkileşiminin atölye dışında da sürdürülmesi

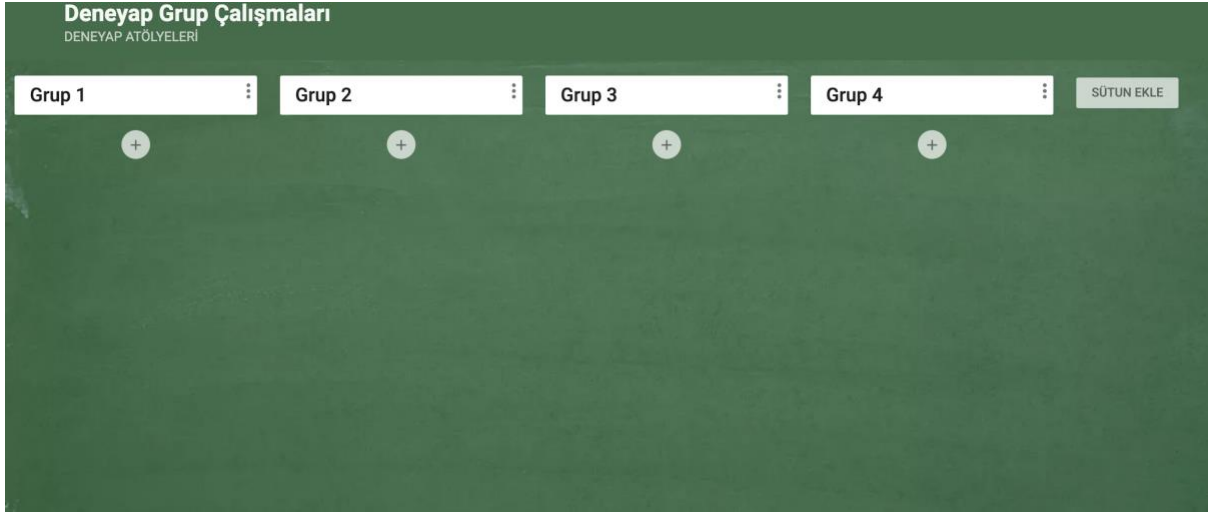
- Gerekğinde senkron (canlı) ders ortamının oluşturulması (Oluşturulan canlı ders çalışma odaları destekli olup grup çalışmalarına imkân vermelidir.)
- Eğitimci eğitimleri sürecinde gerekli kaynakların, sunumların ve dokümanların eğitimcilerle paylaşılabilmesi
- Eğitimcilerin fikir alışverişi yapabilmesi için tartışma ortamlarının kurulabilmesi

GitHub: Derste aktif şekilde kullanılacak olan arşivleme portfolyo oluşturma ortamlarından biri Github'tır. GitHub açık kaynaklı projeler tarafından tercih edilen ve yazılım geliştirme projeleri için kullanılan web tabanlı popüler bir depolama servisi. GitHub ile dünya çapında herkes tarafından görüntülenebilen bir projenize, farklı ekip üyeleri ekleyerek takım çalışmaları düzenlenebilir. Ayrıca GitHub üzerinde paylaşılan kodlar ile kişisel gelişim sağlanabilir. Bu anlamda öğrencilerin GitHub ortamını kullanma deneyimini Yazılım Teknolojileri dersinde edinmesi önemli bir beceridir.

Çevrim İçi Yazılım Teknolojileri dersi için GitHub aşağıdaki amaçlara hizmet etmektedir:

- Her öğrencinin GitHub kullanıcı hesabı olması
- GitHub Branches ile küçük gruplar hâlinde küçük projeler geliştirilmesi
- GitHub Repository ile öğrencilerin kendi depolarını oluşturarak portfolio hazırlanması
- Öğrencilerin takıldıkları noktalarda yardımlaşması için GitHub görev yönetimi (Issues) ile düzenlemeler önerilmesi, yapılacaklar listesinin oluşturulması ve görev atamalarının yapılması
- Tartışma başlıkları açılması

Dijital Tartışma Panosu: Derste aktif şekilde kullanılacak olan temel ortamlardan bir diğeri dijital tartışma pano uygulamalarından kullanımı kolay ve popüler olan Padlet'tir. Bu uygulama Türkçe destekli pek çok özelliği ile 3 panoya kadar ücretsiz olan bir ortamdır. Bu uygulama boş bir duvarı içeriklerle doldurma imkânı veren dijital bir panodur. Öğrenciler giriş yapmaksızın padlet ortamında ücretsiz şekilde görsel, video ya da yazı ekleyebileceği bir panoyu birlikte doldurabilmektedir. Ayrıca padlet ortamının raf stili grup çalışmalarını desteklemektedir. Raf stili ile dijital pano üzerinde sütun şeklinde her grubun kendine ait bir alanı bulunur. Gruplar kendine ait sütun altında yer alan + işaretine tıklayarak ürünlerini paylaşabilir. Çevrim İçi Yazılım Teknolojileri dersinde bu stilden oldukça fazla yararlanılmaktadır. Resim 5'te örnek bir padlet raf stilinin görselini inceleyebilirsiniz.



Resim 5. Padlet ortamı raf stili örneği

Çevrim İçi Yazılım Teknolojileri dersinde padlet ortamı aşağıdaki amaçlara hizmet etmektedir:

- Gruplar ya da öğrenciler için çevrim içi ortak çalışma alanı kurma
- Çevrim içi tartışma ya da beyin fırtınası oluşturma
- Grup ürünlerini paylaşma
- Akranların yaptıklarını çevrim içi inceleme
- Hem senkron hem de asenkron olarak iş birlikli çalışmayı destekleme
- Dersler sonunda üretilen dijital panoyu çıktı hâlinde öğrencilerle paylaşma

Çevrim İçi Yazılım Teknolojileri dersinde padlet kullanımının avantajları aşağıda verilmektedir.

- Ders aktivitelerinde çevrim içi grup ya da bireysel öğrenci paylaşımlarının kaydını tutma
- Tüm öğrencilere birbirlerinin paylaşımlarını çevrim içi anlık inceleme fırsatı sunma
- Çevrim içi akran öğrenimini destekleme
- Ders kaynak ya da materyallerini dijitalleştirme, eğitim maliyetini düşürme
- Öğrencinin multimedya kaynaklarına ders sürecinde erişimini artırma
- Canlı ders yürütürken grup çalışmalarının ürünlerini dijital tartışma panosu aracılığıyla takip etme

Kaynakça

- Costa, J. M., & Miranda, G. L. (2019). Using Alice Software with 4C-ID Model: Effects in Programming Knowledge and Logical Reasoning. *Informatics in Education*, 18(1), 1-15.
- Çilci, N., & Aydın, İ. (2019). *Scamper (Yönlendirilmiş Beyin Fırtınası) Tekniğinin 5 ve 6. Sınıf Öğrencilerinin Yaratıcı Yazıları Üzerindeki Etkisi* (Master's thesis). Ordu Üniversitesi, Ordu.
- Demirel, Ö. (2005). *Eğitimde Program Geliştirme*. Ankara: Pegem A Yayınları.
- Fox, J. (2004). *Rotate, differentiate, and motivate: "how a blend of learning stations and multiple intelligences theory can boost motivation and enhance learning in the middle school classroom* (Unpublished master's thesis). USA, Virginia: College of William & Mary.
- Frerejean, J., van Merriënboer, J. J., Kirschner, P. A., Roex, A., Aertgeerts, B., & Marcellis, M. (2019). Designing instruction for complex learning: 4C/ID in higher education. *European Journal of Education*, 54(4), 513-524.
- Gagne, R. M., Wager, W. W., Golas, K. C., & Keller, J. M. (2005). *Principles of Instructional Design* (4 b.). Belmont: CA: Wadsworth/Thompson Learning.
- İslim, Ö. F. (2011). Scamper (Yönlendirilmiş Beyin Fırtınası Tekniği). *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, 22-24.
- İşman, A. (2015). *Öğretim Teknolojileri ve Materyal Tasarımı*. Ankara: Pegem Akademi.
- Mager, R.F. (1984). *Measuring Instructional Results*. Belmont, CA: David S. Lake Publishers.
- Özyaprak, M. (2016). Yaratıcı Düşünme Eğitimi: SCAMPER Örneği. *Journal of Gifted Education and Creativity*, 3 (1), 67-81. <https://dergipark.org.tr/tr/pub/jgedc/issue/38681/449375>
- Senemoğlu, N. (2007). *Gelişim öğrenme ve öğretim. (Düzenlenmiş Yeni Baskı)*. Ankara: Gönül Yayıncılık.
- Van Merriënboer, J. J., & Kester, L. (2014). The four-component instructional design model: Multimedia principles in environments for complex learning. Maastricht University.
- Van Merriënboer, J. (2016). *How people learn*. The Wiley handbook of learning technology, 15-34.
- Van Merriënboer, J. J., Kirschner, P. A., & Kester, L. (2003). Taking the load off a learner's mind: Instructional design for complex learning. *Educational psychologist*, 38(1), 5-13.
- Yağcı, E. (2012). Yönlendirilmiş beyin fırtınası Tekniği: Scamper konusunda veli görüşleri üzerine bir çalışma. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 2012(43), 485-494.
- Yiğitalp, N. (2014). *Yönlendirilmiş beyin fırtınası (Scamper) tekniğine dayalı eğitimin beş yaş çocuklarının problem çözme becerilerine etkisinin incelenmesi* (Yüksek Lisans Tezi). Hacettepe Üniversitesi, Ankara.

Hafta 1. Programlamaya Giriş

Kazanımlar

- K1. Temel programlama terimlerini öğrenir.
- K2. Bilgisayarın temel birimlerini kavrar.
- K3. Veri saklama birimlerini anlar.
- K4. Programlamadaki aritmetik işleçlerin işlevini kavrar.
- K5. Programlamadaki karşılaştırma işleçlerinin işlevini kavrar.
- K6. Programlamadaki mantıksal işleçlerin işlevini kavrar.
- K7. Değişkenlere değer atama türlerini tanır.
- K8. İkili ve onlu sayı sistemlerini bilir.

Haftanın Amacı ve Materyaller

Bu haftanın amacı programlama dünyasına giriş yapılarak, öğrencilerin temel düzeyde matematiksel, karşılaştırma ve mantıksal işleçleri kullanmalarını sağlamaktır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız.](#)

Önerilen Ders Akışı (50+50+50+50 dk.)

GİRİŞ

Ritmini Keşfet (15 dk.)

Aklına Ne Geliyor? (15 dk.)

GELİŞME

Bilgisayarın Çalışması Neye Benzer? (20 dk.)

Ders Arası (10 dk.)

Bilgisayar Verileri Nasıl Saklar? (20 dk.)

Beni Bul ve Değiştir (30 dk.)

EK 1. İşleç bilgi kartları

EK 2. Örnek Olay sunum kartı

Ders Arası (10 dk.)

Farklı Türleri Tanıyalım (20 dk.)

Sayı Sistemlerini Keşfet (20 dk.)

EK 3. Görev Kartları

SONUÇ (10 dk.)

EK 4. Hafta 1 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 1 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır.

Dikkat Çekme: Ritmini Keşfet

Süre: 15 dk.

Materyaller: Hafta 1 Sunum

Uygulama: Eğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 1 sunumunu ekranda paylaşmak üzere yükler. Sunumun ilk slaytına ilerleyen eğitmen, ilk olarak kendi ismini ve isminin baş harfi ile bir özelliğini açıklar. Örneğin **Sevil**. Özelliğim ise “Sessizlikten hoşlanmaz.” Ardından eğitmen sözlü olarak “Şimdi topu Caner’e atıyorum.” diyerek, konuşmak için öğrencilerden birine söz verir. Bu sefer topu alan Caner kendi ismini ve isminin baş harfi ile bir özelliğini söyler. Daha sonra topu sınıftan bir başka arkadaşına atar ve tüm öğrencilerin bu şekilde konuşmaları sağlanır. Oyun bu şekilde devam eder.

Eğitmen bu oyuna hareketlilik de katabilir. Örneğin isminin özelliğini söyledikten sonra kendine özgü bir hareket gösterir. Bu hareket öğrencinin sınıftaki arkadaşlarını selamlama şekli de olabilir. Topu attığı öğrencilerin de bu şekilde yapmasını ister. En son herkesin söz hakkı bittikten sonra rastgele bir öğrencinin adını söyler. Herkesin söylediği kişinin hareketini birlikte tekrar etmesini ister. Bu şekilde öğrencilerin birbirlerini takip etmelerini de sağlar. Eğitmen sınıftaki herkesin konuşmasını sağlamak için takipte olmalıdır. Öğrencilerin birbirlerine topu atma aşamalarında konuşmayanları onlara hatırlatmalıdır.

Hedeften Haberdar Etme: Eğitmen girişteki bu kaynaşma oyunu ile dikkat çekmeye çalışır. Eğitmen oyun sonunda haftanın konu ve kazanımlarını paylaşarak öğrencileri hedeften haberdar eder. Daha sonra ön bilgileri hatırlatmak için “Aklına Ne Geliyor?” etkinliğini yürütür.

Ön Bilgileri Hatırlatma: Aklına Ne Geliyor?**Süre:** 15 dk.**Kazanımlar:** K1. Temel programlama terimlerini öğrenir.**Materyaller:** Hafta 1 Sunum**Uygulama:** Bu görevde eğitmen öğrencilerden sanal sınıfta “sohbet” kısmını kullanmalarını ister. Eğitmen öğrencilere;

“Programlama dediğimde, aklınıza gelen ilk kelimeyi sohbet kısmından herkese açık olacak şekilde yazın.” der. Tüm öğrenciler sohbet kısmına paylaşımlarını yazarken, 1 dk. süre ile bir müzik açılır. Müzik sonunda, eğitmen kelimeyi bu sefer değiştirir ve öğrencilere şunu söyler.

“Peki Programlama yerine, Programlama Dili deseydim, aklınıza gelen kelimeler ne olurdu? Lütfen tekrar sohbetten hızlıca yazın.”

Eğitmen tekrar 1 dk.’lık bir müzik açar. Müzik sonunda eğitmen sohbet kısmında önemli olan anahtar kelimeleri yakalar ve sunumda yer alan Tablo 1 üzerinden öğrencilerin yazdıklarına geri bildirim verir. Bu şekilde Programlama ve Programlama dili kavramları beyin fırtınası yoluyla özetlenir. Sonuç olarak öğrencilerin ekranda aşağıdaki içeriğe görmeleri sağlanır.

Tablo 1. Programlama ve Programlama Dili Arasındaki Farklar

Programlama	Programlama Dili
Algoritma, Akış diyagramı	Bilgisayar ile konuşma
Yazılım, Kodlama	Bilgisayarla iletişim dili
Talimat verme	Kod dizisi
Bilgisayar kontrolü	Sözdizimi
Bilgisayar görevi	C, C++, Python, Java

Eğitmene Öneriler: Eğitmen yukarıdaki tabloyu oluştururken öğrencilere açıklamalarda bulunur. Bu açıklamalar için aşağıda yer alan konu içeriğinden faydalanabilirsiniz.

Programlamanın tanımı, “Çeşitli görevleri ya da işlemleri gerçekleştirmek için bilgisayara talimat verme.” olarak verilebilir. Bilgisayarlar güçlü ve hızlı sistemler olmalarına rağmen insanlar gibi akıllı varlıklar olmadıkları için ne yapacaklarını bildiren bazı talimatlara ihtiyaç duyarlar. İşte programlama bu talimatları yazma işlemidir. Bunu yapmak için de bir programlama dili kullanılır.

Tıpkı biz insanların birkaç dili (İngilizce, Almanca, Çince, vb.) anlayabilmesi gibi, bilgisayarlarda da durum böyledir. Bilgisayarlar, programlama dili olarak adlandırılan belirli bir sözdizimi biçiminde yazılan talimatları anlar. Görevler ise “iki sayıyı toplama”, “sayının karesini alma” gibi basit görevler olabileceği gibi bir dizi çoklu talimat içeren karmaşık görevler de olabilir. Özet olarak, programlama bilgisayarlara belirli bir görevi yerine getirmelerini söylemenin bir yoludur.



Resim 6. Programlama Süreci

Örneğin İngilizce en popüler ve tanınmış insan dillerinden biridir. İngilizcenin, doğru bir şekilde yazılması için kendi içerisinde uyulması gereken dilbilgisi kuralları seti vardır. Tıpkı insan dilleri gibi, programlama dilleri de sözdizimi adı verilen dilbilgisini takip eder. Programlama dilleri, belirli bir sözdizimi biçiminde yazılır ve bilgisayarın anlayabileceği okunabilir bir formata sahiptir. Hazırlanan programlar daha sonra bilgisayar tarafından yürütülerek istenilen görev ya da işlemlerin gerçekleştirilmesi sağlanır.

Yukarıda belirtildiği üzere bilgisayarlar, programlama dili olarak adlandırılan belirli bir sözdizimi biçiminde yazılan talimatları anlar. Programcının istediği bir görevin bilgisayar tarafından anlaşılması ve yürütülmesi için ifade etmesini sağlar. Popüler programlama dillerinden bazıları C, C++, Python, Java olarak verilebilir. Bu ders kapsamında C++ programlama dili kullanılarak çalışmalar gerçekleştirilecektir. Bu dile ait bilgiler ilerleyen haftalarda detaylı olarak verilecektir.

B. Gelişme

Bilgisayarın Çalışması Neye Benzer?

Süre: 20 dk

Kazanımlar: K2. Bilgisayarın temel birimlerini kavrar.

Materyaller: Hafta 1 Sunum

Uygulama: Bu etkinlikte bilgisayarın temel bileşenlerinin insan organları ile eşleştirilmesine çalışılır. Öğretmen sunumun anket slaytında bekler. Uyarıcı materyal olarak anket yoluyla *yeni bilginin sunulması* sağlanacaktır. Öğretmen sözlü olarak anketteki birinci soruyu aşağıdaki gibi yöneltir ve sanal sınıf aracının anket özelliğini başlatır.

“Sizlere söyleyeceğim ifadeyi lütfen dikkatlice dinleyin. İlk kelimemiz **Girdi: Bilgisayarlar girdi ile beslenir.** Bilgisayarın çalışmasına yönelik verdiğim bu ifadeyi ekranda gördüğünüz insan organlarından hangisi ile eşleştirebilirsiniz? Anketi cevaplayınız.”

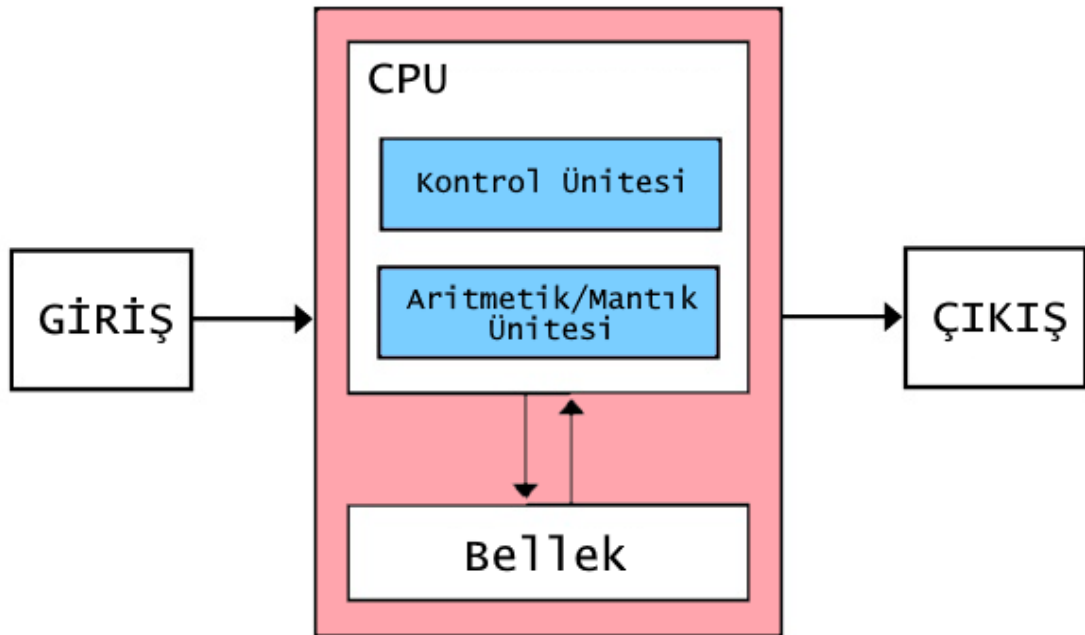
Daha sonra öğrencilerin yanıtlarını anket aracılığıyla iletmesini ister. Bu şekilde öğrencilerin performansını yansıtmayı beklenir. Öğrencilerin her bir anket sorusuna verdiği yanıttan sonra eğitmen öğrenci yanıtlarından yararlanarak yapılan eşleşme hakkında öğrenmeye kılavuzluk eder. Eğitmen birinci soru ile ilgili anketi ve açıklamalarını bitirdikten sonra sırasıyla aşağıdaki anket sorularını da yöneltir ve süreci aynı şekilde devam ettirir.

Anket Soruları:

1. **Girdi:** Bilgisayarlar Girdi ile beslenir.
2. **Kontrol Birimi:** İşlemci kontrol birimi tarafından yönlendirilir
3. **Aritmetik Birimi** Bilgisayardaki işlevleri gerçekleştiren merkezi birimdir
4. **Bellek:** Bilgisayarın belleği tüm verileri kalıcı olarak kaydeder
5. **Çıktı:** Bilgisayar sonuçları çıktı ile üretir

Eğitmen tüm soruları yanıtladıktan sonra bir sonraki slayta ilerleyerek tüm eşleşmeyi tek bir tabloda gösterip geri bildirim verir. Öğrenmeye kılavuzluk etmek ve geri bildirim vermek için aşağıdaki içerik kullanılabilir.

Konu İçeriği: Bu bölümde öğrencilere temel bilgisayar mimarisinin öğretilmesi amaçlanmaktadır. Bilgisayarların birkaç bileşenden oluştuğu bilinmektedir. Bu bileşenlerden olan klavye, fare veya monitörü tanımlamak çok kolaydır. Bunların dışında CPU (merkezi işlem birimi) ve bellek gibi daha kompleks bileşenler vardır. CPU, bilgisayarın beyni olarak tüm kararların alındığı, veri işleyen ve yazılım komutlarını gerçekleştiren bölümdür. Genellikle RAM yani Bellek (rastgele erişimli bellek) olarak adlandırılan bellek ise verileri ve talimatları saklamak için kullanılan ve herhangi bir zamanda okunabilen veya değiştirilebilen bir tür veri deposudur.



Resim 7. Bilgisayar Mimarisi

Kontrol ünitesi, işlemcinin çalışmasını yöneten bir bilgisayarın CPU biriminin bileşenidir. Bilgisayarın bellek, aritmetik ve mantık ile giriş ve çıkış birimlerinin bir programdan alınan talimatlara nasıl yanıt vereceğini bilmesini sağlar. Bir kontrol ünitesi, kontrol sinyallerine dönüştürdüğü ve daha sonra merkezi işlemciye gönderilen giriş bilgilerini alarak çalışır.

Bilgisayarın işlemcisi daha sonra bağlı donanıma hangi işlemlerin gerçekleştirileceğini söyler. Bu süreçte verilerin birimler arasında iletilmesi veri akışı olarak tanımlanır. Bilgisayar sisteminin sağlıklı iletişimi için veri akışı çok önemlidir.

Aritmetik/mantık ünitesi, karmaşık bir dijital devredir ve veriler üzerinde aritmetik ve mantık işlemleri gerçekleştirir. İkili sayılar üzerinde hem bit düzeyinde hem de matematiksel işlemler gerçekleştirir ve işlemcide hesaplamaları gerçekleştiren son bileşendir. Kendisine giriş verileri için hangi işlemleri gerçekleştireceğini söyleyen işlenenleri ve kodu kullanır. Bilgiler aritmetik/mantık ünitesi tarafından işlendikten sonra bilgisayarın belleğine gönderilir.

Anakart üzerindeki bileşenlerin birbirlerine bilgi iletmek için kullandıkları yola veri yolu adı verilir. Girdi bir veri yolu üzerinden CPU'ya gelirken, çıktı CPU'dan veri yolu üzerinden çıkar. Veriler, klavye girişi, dosya içeriği, web sunucusu istekleri veya bilgisayar tarafından kullanılan herhangi bir bilgi parçası gibi birçok şeyi ifade edebilir. Talimatlar ise bu iki sayıyı ekleyin, bu verileri buraya taşıyın, daha sonra bu talimata atlayın gibi CPU'ya bir sonraki adımda ne yapacağını belirten özel bir veri türüdür.

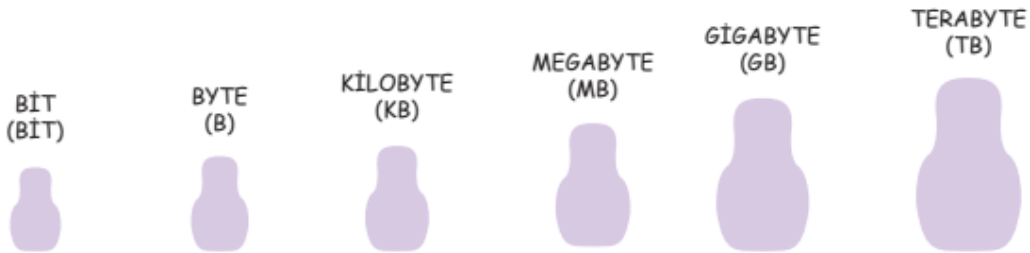
Bilgisayar Verileri Nasıl Saklar?

Süre: 20 dk.

Kazanımlar: K3. Veri saklama birimlerini anlar.

Materyaller: Oyun: [Sıralama oyunu](#)

Uygulama: Eğitimci, öğrencilere bir soru yöneltir. “Nasıl olur da sayılarla çalışan bilgisayar benim resimlerimi depolayabilir?” Eğitimci bu soruyu öğrencilerine sorarak, bir iki öğrenciden söz hakkı ile kısa yanıtlar ister. Daha sonra uyarıcı materyal olarak bir sıralama oyunu oynayacaklarını söyler. Öğrenciler bu oyunu bireysel oynayacaklardır. Oyunun süresini belirtmek için eğitimci 5 dk.’lık sözsüz bir odaklanma müziği açar. Müzik sonunda oyunun sona ereceğini söyler. Bu şekilde *öğrencilerin performanslarını yansıtmaları beklenir*. Öğrencilere oyun linki paylaşılan notlar kısmından da iletir. Oyun sonunda öğrencilerin aşağıdaki sıralamaya ulaşması beklenir. Eğitimci doğru sıralamaya kaç kişinin ulaştığını öğrenmek için öğrencilere kullanıcı durumlarını gülen yüz ya da ağlayan yüz olarak değiştirmelerini ister.

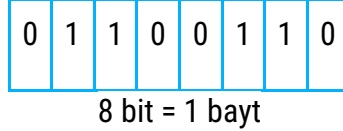


Resim 8. Sıralama çizgisi

Eğitmen oyun sonunda slaytı ilerletir ve yukarıdaki şekli öğrencilere slaytta sunar. Bu sıralama üzerinden konuyu aşağıdaki içeriğe uygun olacak şekilde açıklar. Bu şekilde *öğrenmeye kılavuzluk eder* ve öğrencilere *geri bildirim verir*.

Konu içeriği:

Bilgisayar sistemlerindeki bütün bilgiler ikili sistemde “0” ve “1” ile temsil edilerek saklanır. İkili sistemdeki her bir basamağa bit denir. Bit nicelik ifade edebilmek için yeterli bir birim olmadığı için en yaygın dijital veri depolama birimi olarak 8 bitlik bayt kullanılır.



Bilgisayar sisteminde her bir karakter 8 bit’ten oluşur. Örneğin: A karakteri bilgisayar sisteminde “00100001” ikili sayısı ile ifade edilir. İşte bu sayının her basamağına 1 bit denir. Depolama bayt cinsinden ifade edildiği için tüm büyük birimlerin kısaltılmış adları kullanılır. Bilgisayar verileri metrik sistemde olduğu gibi bayt (B), kilobayt (KB), megabayt (MB), gigabayt (GB), terabayt (TB), ve petabayt (PB) olarak ifade edilir ve 1 KB 1024 bayttır.

Tablo 2. Birimler

Birim	Kısaltma	Kapasite
Bit	b	1 ya da 0
Bayt	B	8 bit
Kilobayt	KB	1024 bayt
Megabayt	MB	1024 kilobayt
Gigabayt	GB	1024 megabayt
Terabayt	TB	1024 gigabayt
Petabayt	PB	1024 terabayt

Tıpkı alfabeyi, konuşulan dilleri ve sayıları anlamak gibi dijital depolama birimlerinde de bilgili olmak önemlidir. Yeni dijital dünya bu yeni dijital depolama sistemlerini öğrenmeyi önemli hale getirdi. Bu tür bir dil ve ölçümün anlaşılmasının faydaları sürekli artmaktadır. Bu dil sisteminde ne kadar akıcı olursak, dijital dünyayı o kadar hızlı anlarız. Zaman içinde veri hacmi ile daha fazla depolama ihtiyacı arttıkça, kaçınılmaz olarak gerekli yeni kelimeleri de geliştireceğiz.

Örnekler:	
10 MB	= 10240 KB
2048 MB	= 2 GB
3 GB	= 3072 MB
5 MB	= 5120 KB
4096 MB	= 4 GB
2048 PB	= 2147483648 GB
10 TB	= 10485760 MB

Beni Bul ve Değiştir

Süre: 30 dk

Kazanımlar: K4. Programlamadaki aritmetik işleçlerin işlevini kavrar.

K5. Programlamadaki karşılaştırma işleçlerinin işlevini kavrar.

K6. Programlamadaki mantıksal işleçlerin işlevini kavrar.

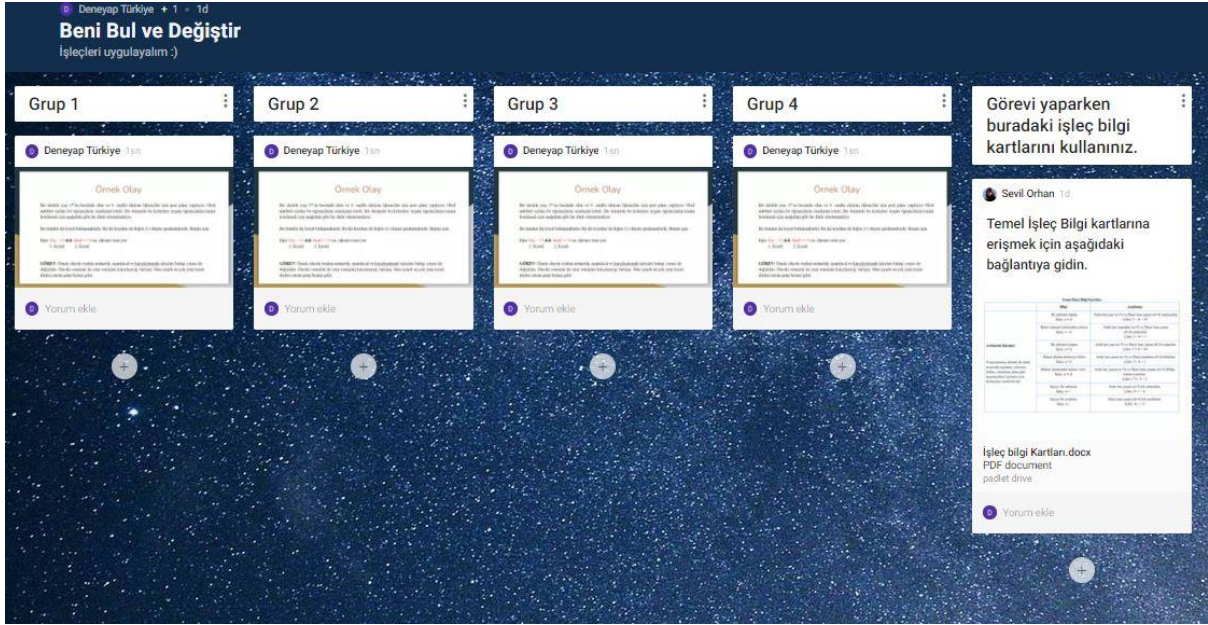
Materyaller: Oyun: [Beni Bul ve Değiştir](#)

EK 1. İşleç bilgi kartları

EK 2. Örnek olay sunum kartı

Dijital Tartışma Panosu: Grup Çalışması

Hazırlık: Örnek linkteki gibi materyalleri barındıran raf temalı ve materyallerin erişilebilen padlet linki oluşturulur. Örnek bir tasarım Resim 9'da gösterilmiştir. EK1 ve EK2'deki materyaller görev ve materyal sütununda ayrı olacak şekilde yüklenecektir. Örnek bir tasarım Resim 9'da gösterilmiştir.



Resim 9. Beni Bul ve Değiştir Örnek Padlet Panosu

Uygulama: Bu öğrenme görevi iki aşamadan oluşmaktadır. İlk aşamada eğitimci işleç kavramı hakkında kısa bir giriş yapar.

Programlama dillerinde kendi başlarına kullanımlarında herhangi bir anlam ifade etmeyen fakat program akışına katkıda bulunan sembol veya sembol topluluklarına işleç denir. Buna örnek olarak toplama ya da çıkarma işleci verilebilir. + sembolü tek başına bir anlam ifade etmezken iki sayı arasında yer aldığına toplama işlemi yapılmasını sağlar.

İşleçler aritmetik, karşılaştırmalı ve mantıksal olmak üzere 3 gruba ayrılır.

1. *Aritmetik işleçler: Programlama dilinde iki ifade arasında toplama, çıkarma, bölme, yüzdesini alma gibi matematiksel işlemler için kullanılan sembollerdir.*
2. *Karşılaştırmalı işleçler: Programlama dilinde aynı türdeki değişkenler arasında belli bir koşula göre karşılaştırma yaptıran sembollerdir. Koşul doğru ise 1, yanlış ise 0 olarak program sonuçlanır.*
3. *Mantıksal işleçler: Programlama dilinde iki ya da daha fazla koşulun birlikte değerlendirilmesinde kullanılan sembollerdir.*

Bu kısa bilgiden sonra, eğitimci öğrencilere bir oyun oynayacaklarını belirtir. Oyun ile *uyarıcı materyal kullanılarak yeni bilgi* sunulur. Oyun linki ([Beni Bul ve Eşleştir](#)) paylaşılan notlar kısmında öğrencilere iletilir. Öğrenciler 5 dk. süre içinde oyunu bireysel olarak oynarlar. Öğrenciler oyun oynarken eğitimci 5 dk.'lık sözsüz bir odaklanma müziği açar. Müzik bitince oyun sona erecektir. Oyun sonunda öğrencilerden ekranlarındaki yanıtları kapatmamaları istenir. Sınıfta rastgele bir öğrenci seçilir ve bu öğrenciden oyun sonundaki ekranı sanal sınıfta paylaşması istenir. Bu şekilde öğrenciler performansı yansıtmış olurlar. Öğrencinin paylaşım ekranında yaptığı gruplamanın doğru yanıtlarını göstermesi beklenir ve bu yanıtlar üzerinden eğitimci konuyu özetleyerek *geri bildirim* verir.

İkinci görevde ise, eğitimci slaytta ilerleyerek öğrencilere bir örnek olay verir. Bu örnek olay üzerinde öğrencilerin yapacakları bir görev bulunmaktadır. Eğitimci öğrencilere paylaşılan notlar kısmında raf temasıyla oluşturulmuş padlet linkini atar. Öğrenciler çalışma odalarına

gönderilmek üzere beşerli dört gruba ayrılır. Her grup padlet ortamında bir görev sütunu ile karşılaşır. Örneğin birinci çalışma odasında yer alan öğrenciler birinci sütundaki görevi tamamlayacaktır. Onlardan örnek olaydaki görevi 10 dk. süresince yapmaları ve padletteki grup sütununa yazmaları istenir. Ayrıca padlet ortamında görev altında link halinde işleç bilgi kartları bulunmaktadır (EK 1). Bu kartlarda örnek olayda yer alan aritmetik, karşılaştırma ve mantıksal işleçleri bulup değiştirecekleri yeni işleç hakkında bilgi ve örnek kullanımlar yer alır. Gruplar bu linke tıkladığında açılan ekrandan bir tane aritmetik, bir tane mantıksal bir tane de karşılaştırmalı işleç seçerek örnek olaydaki ile değiştirecektir. Öğrenciler grup olarak padlette ilgili grup sütunu altında işleç bilgi kartlarını kullanarak yeni koşul ifadeleri yazacaktır. 10 dk. süre içinde en çok koşul ifadesi yazan grup birinci seçilir. Öğrenciler bu şekilde *performanslarını yansıtmaktadır*. Görevlerin tamamlanmasının ardından, çalışma odalarından dönen öğrenciler için eğitmen padleti ekran paylaşımı yapar. Yanıtlar üzerinden *öğrenmeye kılavuzluk etmek* üzere *geri bildirimlerde* bulunur. Bunun için öğrencilerin padlet ortamında yazdıkları yeni koşullar arasındaki benzer ve farklı işlemler, beyin fırtınası ile tartışılır. Bu şekilde eğitmen öğrencilerin aşağıdaki bilgilere erişmesi sağlanır. Ders sonunda eğitmen padlet etkinliğinin pdf baskısını ÖYS üzerinde ilgili hafta altına ekleyerek öğrencilerin ders sonrası erişimine açar.

Konu içeriği:

Aritmetik İşleçler

Aşağıdaki tabloda, bilgisayar programlamasında kullanılan önemli aritmetik işleçler listelenmiştir. Tabloda x ve y bir değişken olarak kabul edilmiştir.

Tablo 3. Aritmetik İşleçler ve Kullanımları

İşleç	Açıklama	Kullanım
+	İki işleneni birbirine ekler.	$x + y$
-	İkinci işleneni birinciden çıkarır.	$x - y$
*	İki işleneni çarpar.	$x * y$
/	Birinci işlenenin ikinci ile bölümünden elde edilen tam kısımdır (bölme işlemindeki bölüm).	x / y
%	Birinci işlenenin ikinci ile bölümünden elde edilen kalan kısımdır.	$x \% y$
++	Sayıyı bir arttırma	$x++$ veya $++x$
--	Sayıyı bir azaltma	$x--$ veya $--x$

Not: Bölme işlemlerinde elde edilen bölüm kısmı kesirli sayı olarak elde edilebilir. Fakat yukarıdaki tabloda verilen bölme işleminde “/” işleci ile bölme işlemi sonucunda bölümün tam kısmı alınmaktadır.

Karşılaştırma İşleçleri

Bu işleçlerin amacı iki değişken ya da değişken grubunu belirtilen şarta göre karşılaştırmaktır. Bu karşılaştırmalar aynı türdeki değişkenler arasında olmalıdır. Bu işleçleri özellikle ilerleyen haftalarda göreceğimiz koşul yapıları ve döngülerde kullanılır. Yapılan karşılaştırmalar doğruysa "1" yanlışsa "0" sonucunu döndürür.

Tablo 4. Karşılaştırma işleçleri

İşleç	Açıklama	Kullanım
<	Küçüktür	$x < y$
>	Büyüktür	$x > y$
<=	Küçük eşittir	$x \leq y$
>=	Büyük eşittir	$x \geq y$
==	Eşittir	$x == y$
!=	Eşit değildir	$x != y$

Mantıksal İşleçler

Mantıksal işleçler programlama dillerinde çok önemli bir yere sahiptir ve belirli koşullara göre karar vermemize yardımcı olurlar. İki koşulun sonucunu birleştirmek istediğimizde mantıksal VE ve VEYA istediğimiz sonucu üretmemize yardımcı olur. Bütün şartların sağlanması için koşullar arasına VE, herhangi birinin sağlanması isteniyorsa koşullar arasına VEYA ve koşulu sağlamayanlar isteniyorsa DEĞİL işleci kullanılmalıdır.

Tablo 5. Mantıksal işleçler

İşleç	Açıklama	Kullanım
&&	Mantıksal VE	$x \&\& y$
	Mantıksal VEYA	$x y$
!	Mantıksal DEĞİL	$!x$

Mantıksal işleçlerin sonucu şu şekilde belirlenmektedir. Eğer $x\&\&y$ kullanılıyor ise her iki değişkenin değeri "1" olursa sonuç "1" olur, aksi hâlde sonuç "0" olur. Eğer $x||y$ kullanılıyor ise her iki değişkenin değeri "0" olursa sonuç "0" olur, aksi hâlde sonuç "1" olur. Eğer $!x$ kullanılıyor ise değişkenin değeri "1" ise sonuç "0", "0" ise sonuç "1" olacaktır. Mantıksal VE için aşağıdaki tabloyu inceleyip çalışma prensibini kontrol edebilirsiniz.

Tablo 6. Mantıksal VE kullanımı

İşlenen 1	İşleç	İşlenen 2	Sonuç
0	&&	Sıfırdan farklı	0
Sıfırdan farklı	&&	0	0
0	&&	0	0
Sıfırdan farklı	&&	Sıfırdan farklı	1

Mantıksal VEYA için aşağıdaki tabloyu inceleyip çalışma prensibini kontrol edebilirsiniz.

Tablo 7. Mantıksal VEYA kullanımı

İşlenen 1	İşleç	İşlenen 2	Sonuç
0		Sıfırdan farklı	1
Sıfırdan farklı		0	1
0		0	0
Sıfırdan farklı		Sıfırdan farklı	1

Mantıksal DEĞİL için aşağıdaki tabloyu inceleyip çalışma prensibini kontrol edebilirsiniz.

Tablo 8. Mantıksal DEĞİL kullanımı

İşleç	İşlenen 2	Sonuç
!	Sıfırdan farklı	0
!	0	1

Farklı Atama Türlerini Tanıyalım

Süre: 20 dk

Kazanımlar: K7. Değişkenlere değer atama türlerini tanır.

Hazırlık: Hafta 1 Sunusunda “Farklı atama türlerini tanıyalım” başlıklı slayt açılır.

Uygulama: Eğitimci hafta 1 sunusunda “Farklı atama türlerini tanıyalım” başlıklı slaytı açar. Aşağıdaki örnek olay üzerinden uyarıcı materyal yoluyla bilgiyi açıklar.

Bir değişkene değer atamak için kullanılır. Atama operatörünün sol taraftaki işleneni değiştendirir ve atama operatörünün sağ taraftaki işleneni bir değerdir.

x = 5

x değişkenine 5 değeri atanıyor.

Eğitmen yukarıdaki örnek üzerinde yapacakları görevler üzerinden farklı tür atama işleçlerini birlikte keşfedeceklerini belirtir. Slaytta örnek olay altındaki ilk görevi öğretmen yapar. Daha sonra slaytı ilerletir ve diğer görevleri de öğrencilerin tahmin etmesini ister. Öğrencilerden göreve ilişkin tahminlerini sohbet ortamında yazmalarını ister. Bu şekilde öğrencilerin performansı yansıtmaları beklenir. Yazılan yanıtlar üzerinden eğitmen her bir göreve verilen yanıtlar üzerinden geri bildirimlerde bulunur. Eğitmen gerektiğinde öğrencilere söz hakkı verir. Son olarak eğitmen farklı tür atama işleçlerini aşağıdaki gibi tablo halinde ekran paylaşımı şeklinde özetler.

Konu İçeriği:

Farklı tür atama operatörleri aşağıda tabloda verilmektedir.

Tablo 9. Atama operatörleri

İşleç	Açıklama	Kullanım
=	Sağdaki değeri soldaki değişkene atamak için kullanılır.	$x = y$ $z = 10$
+=	Bu işleç, '+' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değere ekler ve ardından sonucu soldaki değişkene atar.	$x += y$ $(x = x + y)$
-=	Bu işleç, '-' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değerden çıkarır ve ardından sonucu soldaki değişkene atar.	$x -= y$ $(x = x - y)$
=	Bu işleç '' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değerle çarpar ve ardından sonucu soldaki değişkene atar.	$x *= y$ $(x = x * y)$
/=	Bu işleç '/' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değere böler ve ardından sonucu soldaki değişkene atar.	$x /= y$ $(x = x / y)$
%=	Bu işleç '%' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değere göre modunu alır ve ardından sonucu soldaki değişkene atar.	$x \% = y$ $(x = x \% y)$

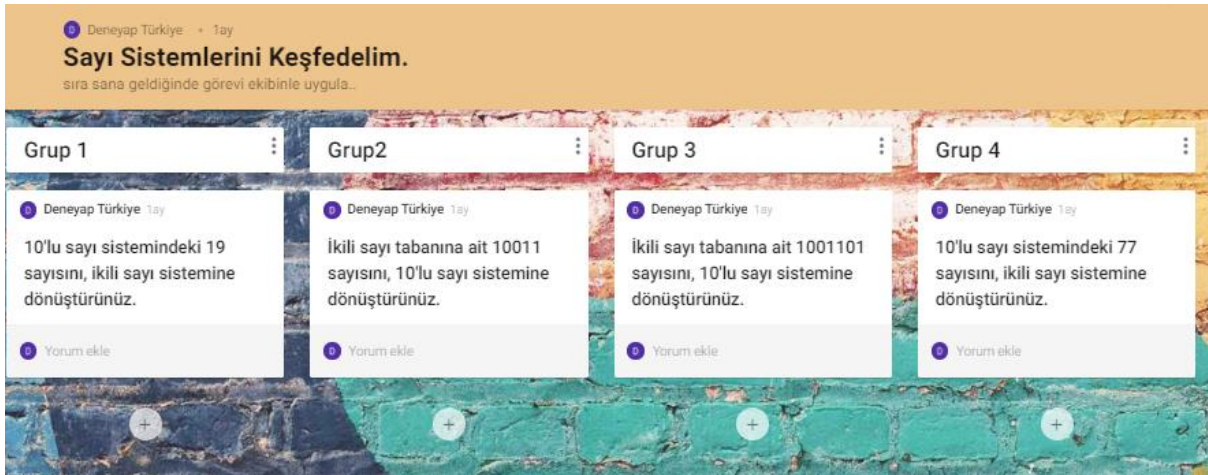
Sayı Sistemlerini Keşfet

Süre: 20 dk.

Kazanımlar: K8. İkili ve onlu sayı sistemlerini bilir.

Materyaller: EK 3. Görev Kartları

Hazırlık: Öğrenci ürünlerinin paylaşımı için raf temalı bir padlet ortamı oluşturulur. Örnek bir tasarım Resim 10'da gösterilmiştir. Ek 3'teki materyalde yer alan her bir örnek olay kartı gruplara yüklenecektir. Örnek bir tasarım Resim 10'da gösterilmiştir.



Resim 10. Sayı Sistemlerini Dönüştürelim Örnek Padlet Panosu

Uygulama: Eğitimci Hafta 1 sunusunda Sayı Sistemleri başlıklı slayta gelir. Bu ve devamındaki slaytları kullanarak sayı sistemlerinde basamak, basamak değeri, basamak ağırlığı ve sayı tabanı kavramlarını özetler. İkili ve onlu sayı tabanlarını tanıtır, veri dönüşümü örneklerini açıklar. Bu şekilde uyarıcı materyal kullanarak bilgi verilir. Daha sonra eğitimci öğrencileri beşer kişilik dört gruba ayırarak 10 dk.'lık çalışma odasına gönderecektir. Çalışma odalarına ayrılmadan önce öğrencilere padlet linki verilir. Padlet grup çalışmalarına izin veren raf temalı olup, her grubun ilgili sütun altındaki "Sıra Sende" görevini ekip olarak tamamlaması istenir. Gruplar tamamladıkları görevleri padlet üzerinden birbiriyle paylaşarak *performanslarını yansıtır*. Çalışma odalarının sonunda ana odaya dönen öğrenciler için padlet linki ekran paylaşımına açılır. Padlet görevlerinde grup 1 ve 2 ile grup 3 ve 4 yanıtları birbirini doğrulamaktadır. Eğitimci bu özelliği kullanarak padlet ortamında verilen yanıtlar üzerinden benzerlikleri ve varsa yapılan hatalı çözümleri sınıfta tartışarak konuyu özetler, öğrencilere yanıtları üzerinden *geri bildirim verir*.

Konu İçeriği:

Onlu Sayı Sistemi

Onlu sayı sisteminde 0 ile 9 arasında yalnızca on rakam vardır. Bu sayı sisteminde her sayı 0,1,2,3,4,5,6,7,8 ve 9 ile temsil edilir. Onlu sayı sisteminin tabanı 10'dur, çünkü yalnızca 10 rakam kullanılır. Bu sistemde her bir basamak ağırlığı şu şekilde gösterilebilir:

10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0
--------	--------	--------	--------	--------	--------	--------	--------

Örnek: Aşağıdaki onlu sayıların basamak analizini inceleyiniz.

$$328 = 3 \times 10^2 + 2 \times 10^1 + 8 \times 10^0$$

$$59 = 5 \times 10^1 + 9 \times 10^0$$

$$7401 = 7 \times 10^3 + 4 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$$

İkili Sayı Sistemi

İkili sayı sisteminde yalnızca 0 ve 1 olmak üzere iki rakam bulunur. Her sayı, bu sayı sisteminde 0 ve 1 ile gösterilir. İkili sayı sisteminin tabanı 2'dir, çünkü sadece iki rakam kullanılır. Her ikili basamak ayrıca bit olarak ifade edilir. Bu sistemde her bir basamak ağırlığı şu şekilde gösterilebilir:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
-------	-------	-------	-------	-------	-------	-------	-------

Herhangi bir ikili sayıda, en sağdaki basamağa en az anlamlı bit (LSB) ve en soldaki basamağa en anlamlı bit (MSB) denir.

1	0	1	0
---	---	---	---

MSB

LSB

Verilen bu sayının onlu eşdeğeri, basamak değeri ile her bir basamak ağırlığının çarpımının toplamıdır.

Örnek: Aşağıdaki ikili sayının basamak analizini inceleyiniz.

$$(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 8 + 0 + 2 + 0$$

$$= (10)_{10}$$

Örnek: Aşağıdaki ikili sayının basamak analizini inceleyiniz.

$$(1001101)_2 = 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 64 + 0 + 0 + 8 + 4 + 0 + 1$$

$$= (77)_{10}$$

Onlu Sayı Sisteminden İkili Sayı Sistemine Dönüştürme

İkili sayı sisteminden onlu sayı sistemine ve onlu sayı sisteminden ikili sayı sistemine dönüştürme işlemleri tüm bilgisayar sistemlerinin temelini oluşturduğundan önemli bir kavramdır. Yukarıdaki bölümde ikili sayı sisteminden onlu sayıya dönüştürmenin nasıl yapıldığını örneklerle gösterdik. Onlu sayı sisteminden ikili sayı sistemine dönüştürme için aşağıdaki adımlar kullanılır:

1. Sayıyı 2 ile bölünüz.
2. Sonraki yineleme için tamsayı bölümünü alın.
3. İkili sayının oluşturulması için kalanı alın.
4. Bölüm 0'a eşit olana kadar adımları tekrarlayın.

Örnek: $(19)_{10}$ sayısını ikili sayı sistemine dönüştürelim.

2 ile Bölme	Bölüm	Kalan	Bit Numarası
19/2	9	1	0
9/2	4	1	1
4/2	2	0	2
2/2	1	0	3
1/2	0	1	4

Sonuç olarak $(19)_{10} = (10011)_2$ olmaktadır.

C. Sonuç

Süre: 10 dk.

Materyal: EK 4 Hafta 1 Süreli Ödev

Hazırlık: Rozet görevleri afişi öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta Algoritma konusuna giriş yapacakları bildirilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. EK 4 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler ileildikten sonra eğitimci görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar: Görevler, öğrenciye verilecek beceri rozetleri ile isimlendirilmiştir. Her bir rozetin altında ilgili görevin yanıtları verilmektedir. Eğitimci bu yanıtları uygulama süresinin sonunda GitHub ya da ÖYS üzerinden öğrencilerle paylaşır.

Tasarlayıcı: Tasarlayıcı rozeti 3 farklı görevden oluşmaktadır. Bu görevlerden birinin tamamlanması Tasarlayıcı rozetinin kazanılması için yeterlidir. Üç görevi de yapan öğrenciler üç adet Tasarlayıcı rozetini almış olur.

1. Kişiye özel bir oyuncak tasarlamalısın. Düğmeye bastığımızda şarkı söyleyen, adımızı hatırlayan, saate göre günaydın/iyi akşamlar/iyi geceler dileyen bir oyuncak için giriş-çıkış, CPU, Bellek görevleri neler olmalıdır? Çizerek anlatır mısın?

Cevap: Bu soru tamamen öğrencinin hayal yeteneğine bağlı olup basitçe; Düğme: Giriş, Hoparlör: Çıkış, Adın hatırlanması: Bellek, Saatin hesaplanması: CPU olarak düşünülebilir.

2. Aşağıdaki ifadeyi bilgisayar diline çeviriniz.

$$3x-2xy+4y$$

Cevap: $3*x - 2*x*y + 4*y$

3. Arda öğretmen sınav sorularının puan değerini ikili tabana göre aşağıdaki gibi belirledi. Sınavdan en yüksek kaç puan alabilirsin? Cevap anahtarına göre doğruların 1001011 şeklinde ise sınavdan kaç alırsın?

Soru 1 2 3 4 5 6 7

Puan 64 32 16 8 4 2 1

Cevap: Sınavdan en yüksek 127 alabilirsin.

Sınavdan alacağın not: $64+8+2+1 = 75$

Analizci: Ayşe'nin parka gitmesi için; Ödevlerini bitirmiş olması (A); Hava sıcaklığının 20-30 derece arasında olması (B); Karnının tok olması (C) gerekiyor. Buna göre Ayşe'nin parka gitmesini mantıksal operatörler ile yazın.

Cevap:

A = Ödevini bitirdi mi? (0 ise bitirmedi 1 ise bitirdi.)

B = $20 < \text{Hava sıcaklığı} < 30$ (Doğru/Yanlış)

C = Karnı tok mu? (0 ise aç1 ise tok.)

(A==1 && B ==1 && C==1) Kısaca: (A&&B&&C)

Kodlayıcı: Ali bilgisayara 1'den 10'a kadar olan sayıları aşağıdaki koşullara uygun olarak yazdırmak istiyor.

Eğer sayı ≤ 5 ise sayı = sayı * 2

Eğer sayı > 5 ise sayı = sayı / 2

Buna göre bilgisayar ekranında ne yazacaktır?

Cevap:

2
4
6
8
10
0
1
0
1

Hafta 1. Ders Materyalleri

EK 1. İşleç Bilgi Kartları

	Bilgi	Açıklama
Aritmetik İşlemler <i>Programlama dilinde iki ifade arasında toplama, çıkarma, bölme, yüzdesini alma gibi matematiksel işlemler için kullanılan sembollerdir.</i>	İki işleneni toplar. İşleç: $a + d$	Arda'nın yaşı ($a=5$) ve Duru'nun yaşını ($d=4$) toplayalım. Çıktı: $5 + 4 = 9$
	İkinci işleneni birinciden çıkarır. İşleç: $a - d$	Arda'nın yaşından ($a=5$) ve Duru'nun yaşını ($d=4$) çıkaralım. Çıktı: $5 - 4 = 1$
	İki işleneni çarpar. İşleç: $a * d$	Arda'nın yaşı ($a=5$) ve Duru'nun yaşını ($d=4$) çarpalım. Çıktı: $5 * 4 = 20$
	İkinci işlenen birinciye böler İşleç: a / d	Arda'nın yaşını ($a=5$) ve Duru'nunkine ($d=4$) bölelim. Çıktı: $5 / 4 = 1$
	Bölme işleminden kalanı verir. İşleç: $a \% d$	Arda'nın yaşını ($a=5$) ve Duru'nun yaşına ($d=4$) bölüp, kalanı yazalım. Çıktı: $5 \% 4 = 1$
	Sayıyı bir arttırma İşleç: $a++$	Arda'nın yaşını ($a=5$) bir arttıralım. Çıktı: $5++ = 6$
	Sayıyı bir azaltma İşleç: $a--$	Duru'nun yaşını ($d=4$) bir azaltalım. Çıktı: $4-- = 3$

Karşılaştırmalı İşlemler <i>Programlama dilinde aynı türdeki değişkenler arasında belli bir koşula göre karşılaştırma yaptıran sembollerdir. Koşul doğru ise 1, yanlış ise 0 olarak program sonuçlanır.</i>	Büyüktür. İşleç: $a > d$	Arda'nın yaşı ($a=5$), Duru'nun ($d=4$) yaşından büyüktür. Çıktı: 1 (Doğru)
	Küçüktür. İşleç: $<$	Arda'nın yaşı ($a=5$), Duru'nun ($d=4$) yaşından küçüktür. $a < d$ Çıktı: 0 (Yanlış)
	Küçük eşittir İşleç: $<=$	Arda'nın yaşı ($a=5$), Duru'nun ($d=4$) yaşından küçük eşittir. $a <= d$ Çıktı: 0 (Yanlış)
	Büyük eşittir İşleç: $>=$	Arda'nın yaşı ($a=5$), Duru'nun ($d=4$) yaşından büyük eşittir. $a >= d$ Çıktı: 1 (Doğru)
	Eşittir İşleç: $==$	Arda'nın ($a=5$) ve Duru'nun ($d=4$) yaşı eşittir. $a == d$ Çıktı: 0 (Yanlış)
	Eşit değildir. İşleç: $!=$	Arda'nın ($a=5$) ve Duru'nun ($d=4$) yaşı eşit değildir. $a != d$ Çıktı: 1 (Doğru)

Mantıksal İşlemler <i>Programlama dilinde iki ya da daha fazla koşulun birlikte değerlendirilmesinde kullanılan sembollerdir.</i>	VE İşleç: &&	Arda'nın yaşı a=5 ve Duru'nun yaşı d=4'tür. $((a==5) \ \&\& \ (d>5))$ Çıktı: 1 && 0 = 0 (Yanlış) $((a!=5) \ \&\& \ (d>5))$ Çıktı: 0 && 0 = 0 (Yanlış) $((a==5) \ \&\& \ (d<5))$ Çıktı: 1 && 1 = 1 (Doğru)
	VEYA İşleç:	Arda'nın yaşı a=5 ve Duru'nun yaşı d=4'tür. $((a==5) \ \ (d>5))$ Çıktı: 1 0 = 1 (Doğru) $((a!=5) \ \ (d>5))$ Çıktı: 0 0 = 0 (Yanlış) $((a==5) \ \ (d<5))$ Çıktı: 1 1 = 1 (Doğru)
	DEĞİL İşleç: !	Arda'nın yaşı a=5 ve Duru'nun yaşı d=4'tür. $! (a==4)$ Çıktı: 1 (Doğru) $! (d!=5)$ Çıktı: 0 (Yanlış)

EK 2. Örnek Olay Sunum Kartı

Örnek Olay

Bir okulda yaşı 15'in üzerinde olan ve 9. sınıfta okuyan öğrenciler için gezi planı yapılıyor. Okul müdürü sizden bu öğrencilerin isimlerini istedi. Bu durumda bu kriterlere uygun öğrencilerin ismini listelemek için aşağıdaki gibi bir ifade oluşturmalıyız.

Bu örnekte iki koşul bulunmaktadır. Bu iki koşulun da doğru (1) olması gerekmektedir. Bunun için,

Eğer $Yaş > 15$ && $Sınıf == 9$ ise, öğrenci ismi yaz

1. Koşul 2. Koşul

GÖREV: Örnek olayda verilen aritmetik, mantıksal ve karşılaştırmalı işlemleri bulup, yenisi ile değiştirin. Önceki sonuçlar ile yeni sonuçları karşılaştırıp, tartışın. Süre içinde en çok yeni koşul ifadesi yazan grup birinci gelir.

EK 3. Görev Kartları

Grup 1 10'lu sayı sistemindeki 19 sayısını, ikili sayı sistemine dönüştürünüz.
Grup 2 İkili sayı tabanına ait 10011 sayısını, 10'lu sayı sistemine dönüştürünüz.
Grup 3 İkili sayı tabanına ait 1001101 sayısını, 10'lu sayı sistemine dönüştürünüz.
Grup 4 10'lu sayı sistemindeki 77 sayısını, ikili sayı sistemine dönüştürünüz.

EK 4. Hafta 1 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 2. Algoritma Tasarımı

Kazanımlar

- K1. Verilen algoritmanın temel özelliklerini analiz eder.
- K2. Bir problemin çözümüne uygun farklı algoritmalar tasarlar.
- K3. Bir problemin çözümüne uygun algoritmanın akış şemasını oluşturur.
- K4. Bir problemin çözümüne uygun algoritmanın akış şemasını bilgisayar ortamında çizer.

Haftanın Amacı ve Materyaller

Bu haftanın amacı bir problemi algoritma ve akış diyagramları kullanarak çözmek ve problemin çözümünü bilgisayar kullanarak aktarmaktır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız](#).

Önerilen Ders Akışı (50 + 50 + 50 + 50 dk.)

GİRİŞ

Aklına Ne Geliyor? (10 dk.)

GELİŞME

Algoritmayı Tanıyorum (20 dk.)

Algoritmaları Eşleştirim (20 dk.)

EK 1. Algoritmaları Eşleştirelim Çalışma Kâğıdı

Ders Arası (10 dk.)

Algoritma Yazıyorum (30 dk.)

EK 2. Akış Diyagramları Bilgi Afişi

EK 3. Nereye Gidelim Haritası ve Görevler

Algoritma Sıralıyorum (20 dk.)

EK 4. Çıkartma Algoritması Eşleştirme Kâğıdı

Ders Arası (10 dk.)

Akış Diyagramı Çiziyorum (40 dk.)

EK 5. Grup Görevleri

EK 6. App.diagrams.net Kullanım Kılavuzu

SONUÇ (10 dk.)

EK 7. Hafta 2 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Çizgi Nokta Oyunu

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 2 Sunum). Sunu aşağıdaki içeriğe uygun şekilde süreci yönetmek için kullanılır.

Dikkat Çekme: Aklına Ne Geliyor?

Süre: 10 dk.

Kazanımlar: K1. Algoritma temel kavramlarını bilir.

Materyaller: Hafta 2 Sunum

Uygulama: Eğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 2 sunumunu ekran paylaşımına açar. Öğrencilerin derse girişi tamamlandıktan sonra, haftanın konu ve kazanımlarını paylaşarak öğrencileri hedeften haberdar eder. Daha sonra ön bilgileri hatırlatmak için Aklına Ne Geliyor? etkinliğini yürütür. Bu etkinlikte eğitmen öğrencilerden sanal sınıfta “sohbet” kısmını kullanmalarını ister. Eğitmen öğrencilere;

“Algoritma dediğimde aklınıza gelen ilk kelimeyi bana sohbet kısmından herkese açık olacak şekilde yazın.” der.

Tüm öğrenciler sohbet kısmına paylaşımlarını yazarken, bir dk. süre ile bir müzik açılır. Müzik sonunda, eğitmen algoritmanın ne olduğu ile ilgili konuya dikkat çeker ve aşağıdaki gibi kısa bir giriş yapar.

Algoritma, bir problemi ya da sorunu çözmek için izlenecek mantıksal kural ve adımların listesidir. Diğer bir ifadeyle, bir problemi çözmek için takip edilecek sonlu sayıda adımdan oluşan bir çözüm yoludur. Mevcut var olan bilgilerden istenilenlere erişmemizi sağlar. Bir algoritmayı anlamanın en iyi yolu onu bir tarif (ilaç, yemek vb.) olarak düşünmektir. Günlük hayattaki yaşantı şeklimizde düzenli olarak birtakım işlemlerin sıra ile yapılması şeklindedir. Yani bir işi yapabilmek için bir takım alt işleri peş peşe gerçekleştiririz.

B. Gelişme

Algoritmayı Tanıyorum

Süre: 20 dk.

Kazanımlar: K1. Algoritma temel kavramlarını bilir.

Materyaller: Hafta 2 Sunum

Hazırlık: Öğitmen sunumun “Doğru algoritma hangisi?” başlıklı slaytında bekler.

Uygulama: Bu etkinlikte uyarıcı materyal olarak anket yoluyla *yeni bilginin sunulması* sağlanacaktır. Slaytta birden fazla algoritma örneği verilmiştir. Ancak bunlardan sadece biri doğru yazılmıştır. Öğitmen sözlü olarak anketteki birinci soruyu aşağıdaki gibi yöneltir ve sanal sınıf aracının anket özelliğini başlatır.

Verilen algoritmalar arasındaki benzerlik ve farklılıkları düşünün. En iyi yazılan kek algoritması hangisidir? Anketime yanıt vererek cevabınızı iletin.

Ankete verilen yanıtlar ile performansı yansıtmaya aşamasıdır. Bundan sonra öğrencilere, öğretmen doğru algoritma hakkında kılavuzluk eder, yanıtlar üzerinden geri bildirim verir. Devamında öğrenciler beşer kişilik 10 dk. süre ile çalışma odalarına gönderilir. Bu sefer öğrencilere aşağıdaki talimat verilir.

Çalışma odasına gittiğinizde verilen doğru algoritmayı diğerleri ile karşılaştırarak algoritma yazma özellikleri hakkında grup arkadaşınızla tartışınız.

Bu görevleri yaparken öğrencilerden algoritmanın özellikleri üzerine düşünmeleri ve performanslarını yansıtmaları beklenir. Öğitmen görev sonunda grupların düşüncelerini sohbet ortamından yazmalarını ister ve öğrenci yanıtları üzerinden beyin fırtınası yaptırır. Bu şekilde öğrencilerin algoritmanın özelliklerini aşağıdaki gibi keşfetmelerini sağlar.

- *Bir başlangıç noktası olmalı*
- *Basit olmalı*
- *Algoritma içindeki ifadeler herkes tarafından aynı şekilde anlaşılabilir olmalı*
- *Yorum gerektirmemeli ve belirsiz ifadelere sahip olmamalı*
- *Her adımda tek bir iş yapılmalı*
- *Adımların gerçekleşme sırası doğru olmalı*
- *Mümkün olan en az adım ile en kısa sürede gerçekleşmeli*
- *Sonsuz döngüye girmemeli*
- *Bir bitiş noktası olmalı*

Eğitmene Öneriler: Görev sonunda öğretmen öğrencilere en iyi yazılmış algoritmanın diğerinden farklı ve benzer olan yanları üzerine sorular sorabilir. Bunlar algoritmanın özellikleri için ipuçları verecektir. Örneğin “Diğer algoritmalara bakarak, doğru algoritma olduğuna nasıl karar verdiniz?”

Algoritmaları Eşleştiriyorum

Süre: 20 dk.

Kazanımlar: K1. Algoritma temel kavramlarını bilir.

Materyaller: EK 1. “Algoritmaları Eşleştirelim” çalışma kâğıdı

Hazırlık: Eğitimci derse girmeden önce EK 1 materyalinin bağlantı linkini paylaşmak üzere hazırda tutar. Materyal aynı zamanda ÖYS ortamında da öğrenci erişimine açılabilir.

Uygulama: Eğitimci paylaşılan notlar kısmından “Algoritmaları Eşleştirelim” materyalinin linkini verir (EK 1). Materyalde iki farklı problemin metin, sözde kod ve akış diyagramı şeklinde hazırlanmış algoritması vardır. Öğrencilerden problemi ve yazım şekillerini eşleştirerek tahmin etmeleri istenir. Eğitimci çalışma kâğıdını ekran paylaşımı yapar ve öğrencilerle birlikte doldurmaya başlar. Bunu yapmak için öğrencilere her bir eşleştirme için anket özelliğini kullanarak sorular yöneltir ve onlardan performanslarını yansıtmasını bekler. Örnek anket sorusu öğrencilere aşağıdaki gibi yöneltilir.

A hücresi soru 1. Çalışma kâğıdında A hücresinde yer alan algoritma hangi probleme aittir?

P1. Çemberin alanını hesaplama

P2. İki sayıyı toplama

A hücresi soru 2. Çalışma kâğıdında A hücresinde yer alan algoritma hangi ifade şekli ile oluşturulmuştur?

MY: Metinsel Yazım

SK: Sözde Kod

AD: Akış Diyagramı

Öğrencilerden gelen yanıtlar üzerine geri bildirimlerde bulunan eğitimci çalışma kâğıdında ilgili hücreyi doldurur. Bu şekilde tüm eşleştirmelere ilişkin benzer sorular yöneltilir. Bu etkinlik ile öğrencilerden bir problemin algoritmasının üç farklı şekilde oluşturulabildiğini keşfetmeleri sağlanır. Eğitimci öğrenciler tarafından verilen anket yanıtlarını aldıkça aşağıdaki içerik hakkında *geri bildirimlerde* bulunur.

Eğitmene Öneriler: Bu etkinlikte öğrencilerin edinmesi gereken içerik için aşağıdaki bilgiler incelenebilir.

a) *Metin olarak yazım:* Problemin çözüm adımları, düz metin olarak açık cümlelerle ifade edilir. Algoritmadaki her bir satıra numara verilir. “Başla” ile başlayıp “Bitir” ile bitirilir.

b) *Sözde kod (pseudocode) yazım:* Problemin çözüm adımları komut benzeri anlaşılır metinlerle ifade edilir. Sözde kod konuşma dilinde ve programlama mantığı altında,

“eğer”, “iken” gibi koşul kelimeleri ve ‘>’, ‘=’, ‘<’ gibi ifadeler ile birlikte yazılır. Algoritma yazma ile kod yazma arasında kalan bir kısımdır ama kodlamaya daha yakındır. İyi yazılmış sözde koddan bir programlama diline kolaylıkla geçiş yapabilirsiniz.

c) *Akış diyagramlarının çizilmesi*: Problemin çözüm adımları, görsel olarak simge ya da sembollerle gösterilir. Akış şemalarının algoritmadan farkı, algoritma adımlarının semboller şeklinde kutulara yazılması ve adımlar arasındaki ilişki ve yönün oklar ile gösterilmesidir.

Algoritma Yazıyorum

Süre: 30 dk.

Kazanımlar: K2. İhtiyaç duyulan algoritmanın tasarımını sağlar.

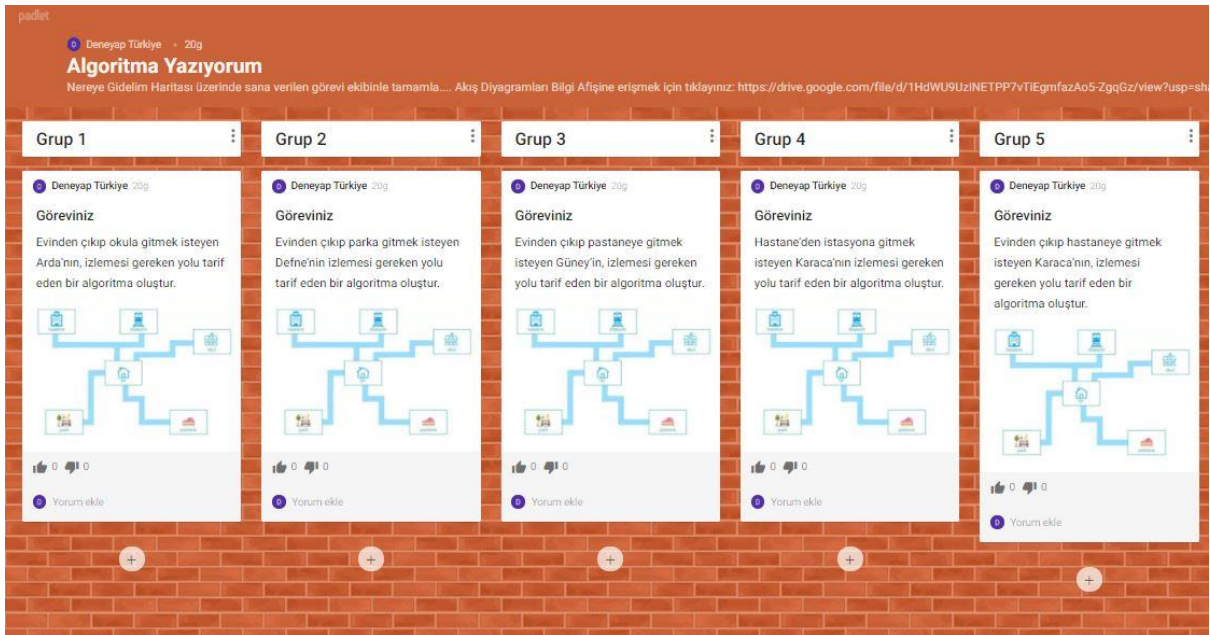
K3. Bir algoritmayı akış şemasında gösterir.

Materyaller: Dijital Tartışma Panosu: Grup Çalışması

EK 2. Akış Diyagramları Bilgi Afişi

EK 3. Nereye Gidelim Haritası ve Görevler

Hazırlık: Eğitimci derse girmeden önce grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletacaktır. Eğitimci padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. Örnek bir tasarım Resim 11’de gösterilmiştir. EK 2 ve EK 3 materyalleri dijital tartışma panosuna yüklenecektir. Örnek bir tasarım Resim 11’de gösterilmiştir.



Resim 11. Algoritma Yazıyorum Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci Hafta 2 sunusunda Akış Diyagramları slaytına devam edilir. Akış diyagramları hakkında bilgi verildikten sonra, öğrenciler dörderli gruplar halinde 10 dk. süre ile çalışma odalarına dağılır. Gruplar dağılmadan önce paylaşılan notlar kısmından etkinliğin padlet linki paylaşılır. Padlet ortamında her grup sütununda “EK 3. Nereye Gidelim Haritası” ve görev tanımı verilmiştir. Eğitimci çalışma odalarına tek tek katılarak gruplara bilgi vermelidir. Görevler haritada bir yerden diğerine ulaşmak için izlenmesi gereken talimatların algoritmasını sözde kod ile yazmayı gerektirir. Görev sonunda ana odaya geri dönen öğrenciler tekrar rastgele olacak şekilde odalara dağıtılır. İlk gruplardan farklı yeni gruplar oluşturulmuştur. Bu sefer odalara geçmeden önce öğrencilerden bir önceki grubun oluşturduğu algoritmanın akış diyagramını çizmeleri istenir. Eğitimci bu aşamada öğrencilerden “Akış diyagramları bilgi afişi”ni kullanmalarını ister. Bu afişe padlet linkinde yer alan pano başlığının altından erişebileceklerini belirtir. Eğitimci öğrencileri tekrar gruplara dağıtmadan önce bu afişin linkini paylaşılan notlar kısmından da iletebilir (EK 2). Öğrencilerden biri ekip lideri olarak grup çizimini bir kâğıtta ya da bilgisayar ortamında oluşturabilir. Süre sonunda ekip lideri çizimin ekran görüntüsünü padlet ortamına yükler. Bu etkinlikte de süre 10 dk. olarak belirtilir. Eğitimci süre sonunda görevlere ait doğru akış diyagramlarını grupların altına yükler. Öğrencilerin ürünleri doğru olanları ile karşılaştırmalarını ister ve varsa hatalar üzerinden her grup ürününü sınıfça tartışmaya açar. Öğrenci sorularına göre yanlışları düzelterek konuyu özetler. Bu şekilde tüm gruplara geri bildirim verilir.

Eğitmene Öneriler: Eğitimci bu etkinlikte akış diyagramlarına giriş yapmaktadır. App.diagrams.net programını kullanarak öğrencilerin akış diyagramlarını bilgisayar ortamında oluşturabileceklerini söyleyebilir. Etkinlik süresine bağlı olarak kısaca program öğrencilere tanıtılabilir. Programın kullanım kılavuzu ders öncesi öğrencilere ÖYS üzerinden gönderilir.

Algoritma Sıralıyorum

Süre: 20 dk.

Kazanımlar: K2. İhtiyaç duyulan algoritmanın tasarımını sağlar.

K3. Bir algoritmayı akış şemasında gösterir.

K4. Bir algoritmanın akış şemasını bilgisayar ortamında çizer.

Materyaller: EK 4. Çıkartma Algoritması Eşleştirme Kâğıdı

Hazırlık: Eğitimci derse girmeden önce EK 4 materyalinin bağlantı linkini paylaşmak üzere hazırda tutar. Materyal aynı zamanda ÖYS ortamında da öğrenci erişimine açılabilir.

Uygulama: Eğitimci bu etkinlikte Hafta 2 sunumuna geri döner ve Çıkartma Algoritması slaytını açar. Eğitimci paylaşılan notlar kısmından “EK 4” materyalinin linkini öğrencilere verir. Aynı zamanda eğitimci materyali ekran paylaşımı yapar ve öğrencilerle birlikte eşleştirmeleri yapmak üzere çalışma kâğıdını doldurur. Bunu yapmak için öğrencilere her bir eşleştirme için sorular yöneltir ve onlardan sohbeti kullanarak yanıtlarını iletmelerini ister. Bu şekilde öğrencilerden performanslarını yansıtmalarını beklenir. Örnek sorular:

- A satırı kaç numaralı şekle girmelidir?
- B satırı kaç numaralı şekil ile eşleşir?

Eğitmen öğrenci yanıtlarından yararlanarak eşleştirme kâğıdını herkesin göreceği şekilde ekranda doldurur. Bu işlem sırasında eğitmen şekillere ilişkin bilgi vererek, geri bildirimde bulunur. Slaytta devam edilir ve algoritmanın bitmiş hali üzerinden eğitmen tekrar yapar. Bu şekilde doğru algoritmaya ait akış diyagramının öğrenciler tarafından keşfedilmesi sağlanır.

Akış Diyagramı Çiziyorum

Süre: 40 dk.

Kazanımlar: K2. İhtiyaç duyulan algoritmanın tasarımını sağlar.

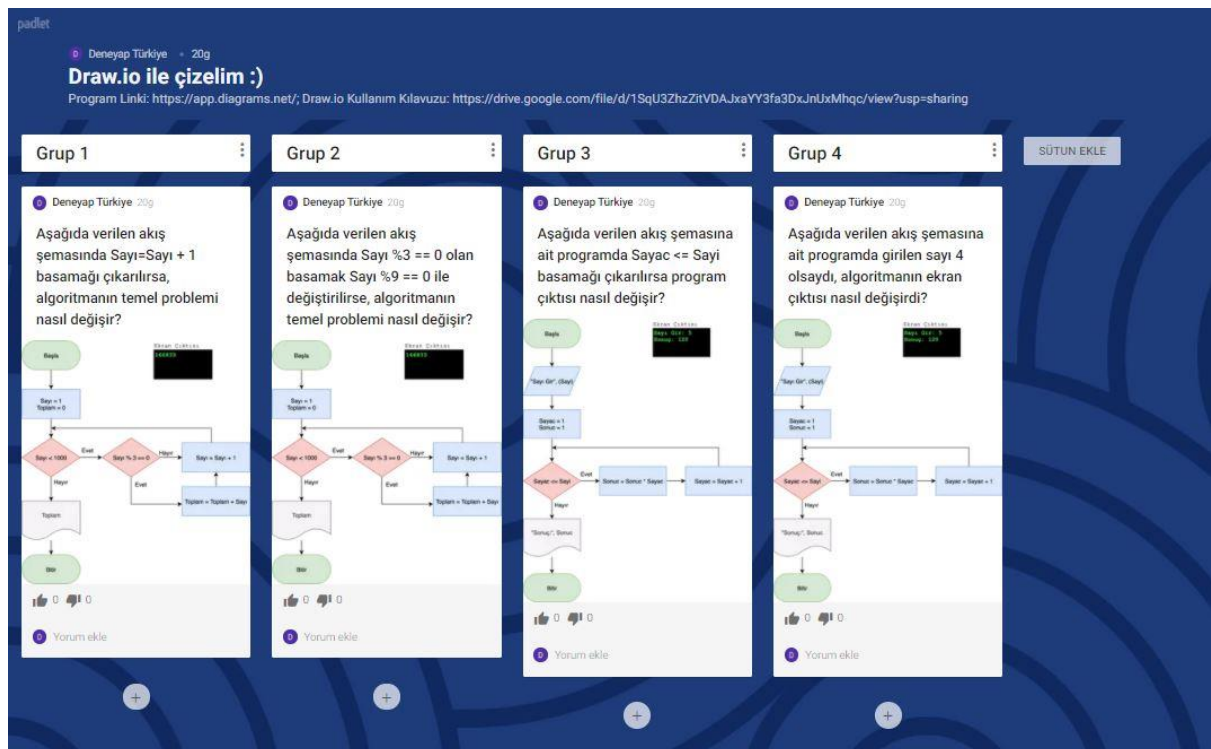
K3. Bir algoritmayı akış şemasında gösterir.

Materyaller: Dijital Tartışma Panosu: Grup Çalışması

EK 5. Grup Görevleri

EK 6. App.diagrams.net Kullanım Kılavuzu

Hazırlık: Eğitmen derse girmeden önce grup çalışması için duvar temalı bir dijital pano oluşturur. Link ders sırasında öğrencilere paylaşılan notlar kısmından iletecektir. Eğitmen padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. Örnek bir tasarım Resim 12’de gösterilmiştir. EK 5 materyali gruplar altına ayrı ayrı yüklenecektir. EK 6 materyali dijital tartışma panosunda bağlantı olarak verilir. Örnek bir tasarım Resim 12’de gösterilmiştir.



Resim 12. Akış Diyagramını Sen Çiz Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci öğrencilere grup görevleri için padlet linkini paylaşılan notlar kısmından atar. Öğrenciler bu etkinlikte çalışma odalarında 10 dk. süre ile dört grup halinde çalışır. Her gruba tartışma panosunda bir görev atanmıştır. Görevi çözdükten sonra, her gruptan bir kişi grup cevabını padlette ilgili grup sütunu altına yazar. Süre sonunda öğrenciler ana sayfaya dönerler. Eğitimci öğrencilere padlet ortamında yer alan app.diagrams.net kullanım kılavuzunu açtırır. Bu kılavuza erişim linkini paylaşılan notlar kısmından da ayrıca gönderir. Kılavuz ÖYS ortamında sunumla birlikte paylaşılmalıdır. Eğitimci kılavuzu öğrencilere açıklar ve programı gösterir. Bu aşamadan sonra öğrencileri tekrar rastgele dört gruba ayırır ve padlet ortamına geri gönderir. Bu sefer 15 dk. süreleri vardır. İkinci görevde gruplar, bir önceki grubun göreve ilişkin yanıtını kontrol eder ve düzeltme gerekirse yeni yanıtlarını padlet ortamında yazarlar. Şimdi tüm grup üyelerinin yeni yanıtın üzerinden yeniden belirlenen görevin akış diyagramını app.diagrams.net kullanım kılavuzunu kullanarak çizmeleri istenir. Burada tüm öğrenciler ilgili görevin yeni halinin akış diyagramını kendi bilgisayarlarında çizecektir. Bir göreve ait beş kişinin akış diyagramını çizmesi ve padlette ilgili grup sütunu altında paylaşması istenir. Süre sonunda bir görevin altında beş farklı kişiden gelen akış diyagramı ekran görüntüsü bulunmalıdır.

Ana sayfaya dönen öğrenciler grup içinde diğer arkadaşlarının akış diyagramlarını inceler. Bunun için eğitimci 1 dk. süre ile müzik açar ve grup üyelerinin kendi görevlerine ilişkin diğer akış diyagramlarından en doğru olanı oylamalarını ister. Eğitimci her bir görevde en çok oy alan yanıt üzerinden konuyu özetler.

C. Sonuç

Süre: 10 dk.

Materyal: EK 7 Hafta 2 Süreli Ödev

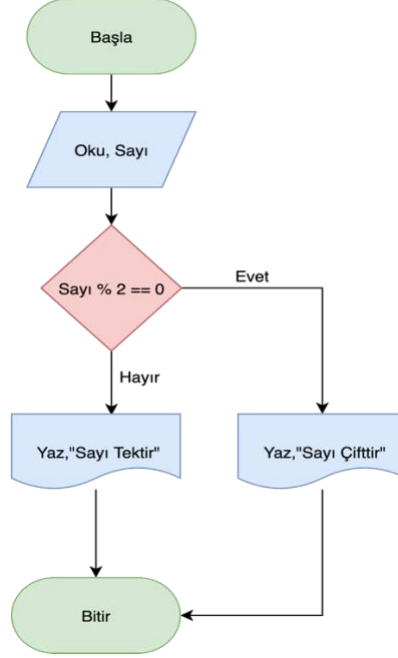
Hazırlık: Rozet görevleri afişi öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl gönderileceği açıklanır ve gelecek hafta Algoritmada Değişkenler ve Değerleri konusunu işleyecekleri bildirilir.

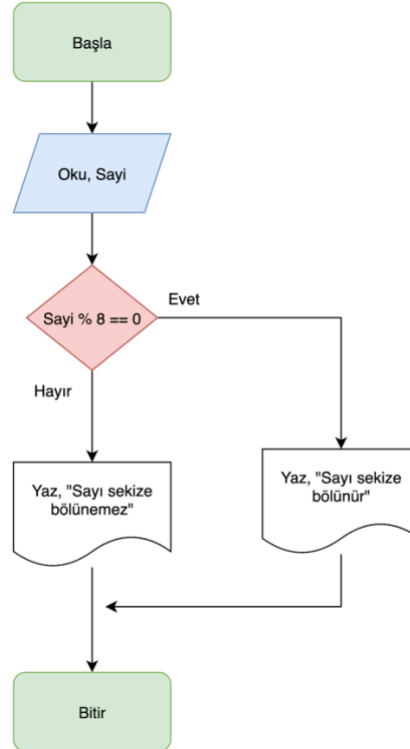
Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. EK 6 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler ileildikten sonra eğitimci görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar: Görevler, öğrenciye verilecek beceri rozetleri ile isimlendirilmiştir. Her bir rozetin altında ilgili görevin yanıtları verilmektedir. Eğitimci bu yanıtları uygulama süresinin sonunda GitHub üzerinden öğrencilerle paylaşır.

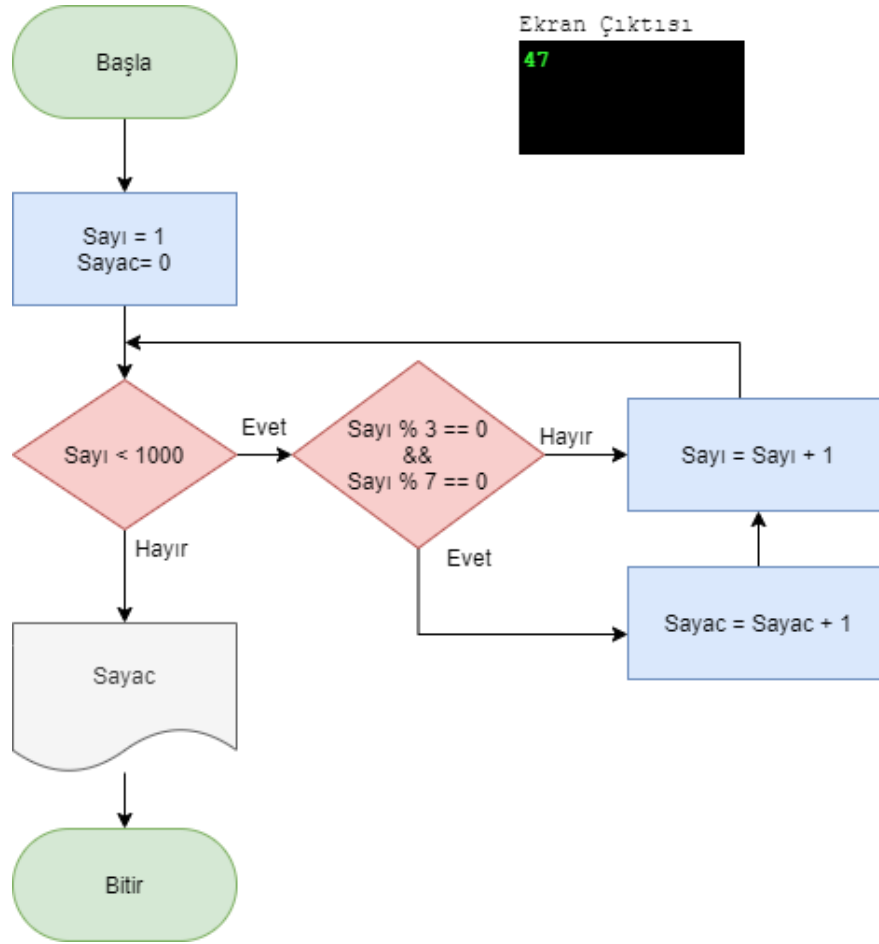
Tasarlayıcı 1: Aşağıda klavyeden girilen sayının çift olduğunu belirleyen programın akış şeması verilmiştir. Bu algoritma girilen sayının 8 ile bölünüp bölünmediğini gösterseydi, şema nasıl değişirdi?



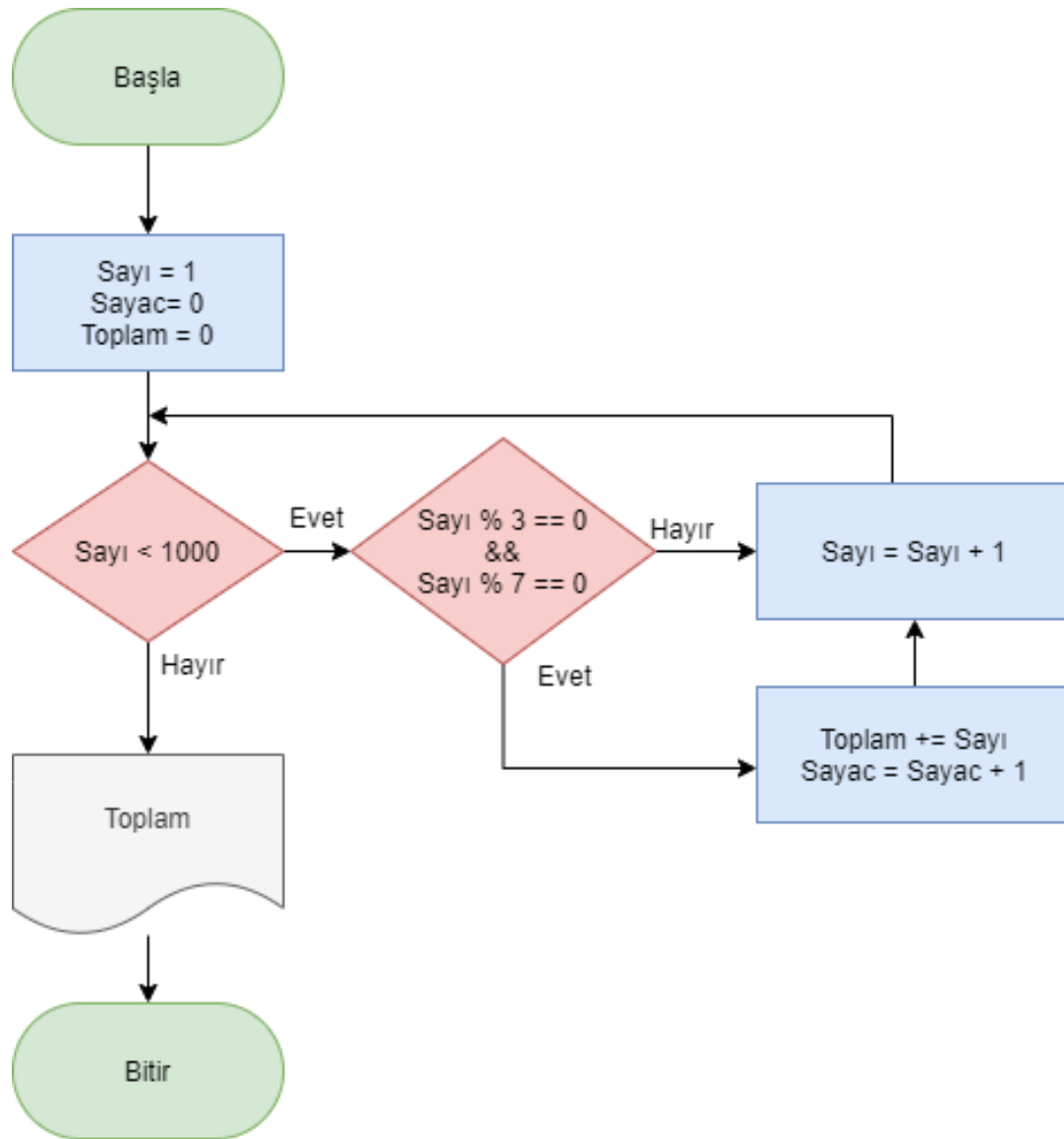
Yanıt:



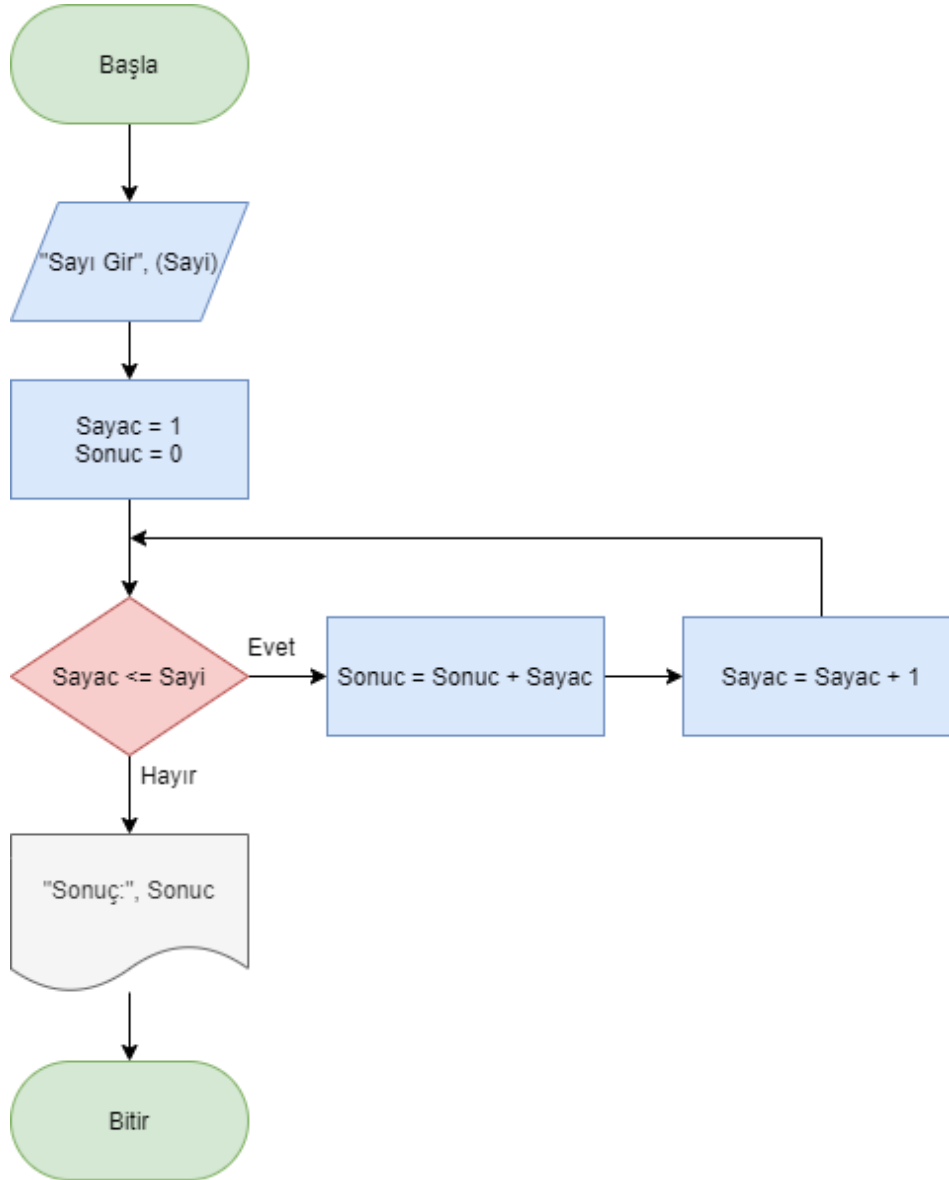
Tasarlayıcı 2: Aşağıdaki algoritma üç ve yedi ile tam bölünen 1000'den küçük sayıların toplamını yazdırıyor olsaydı, şema nasıl değişirdi?



Yanıt:

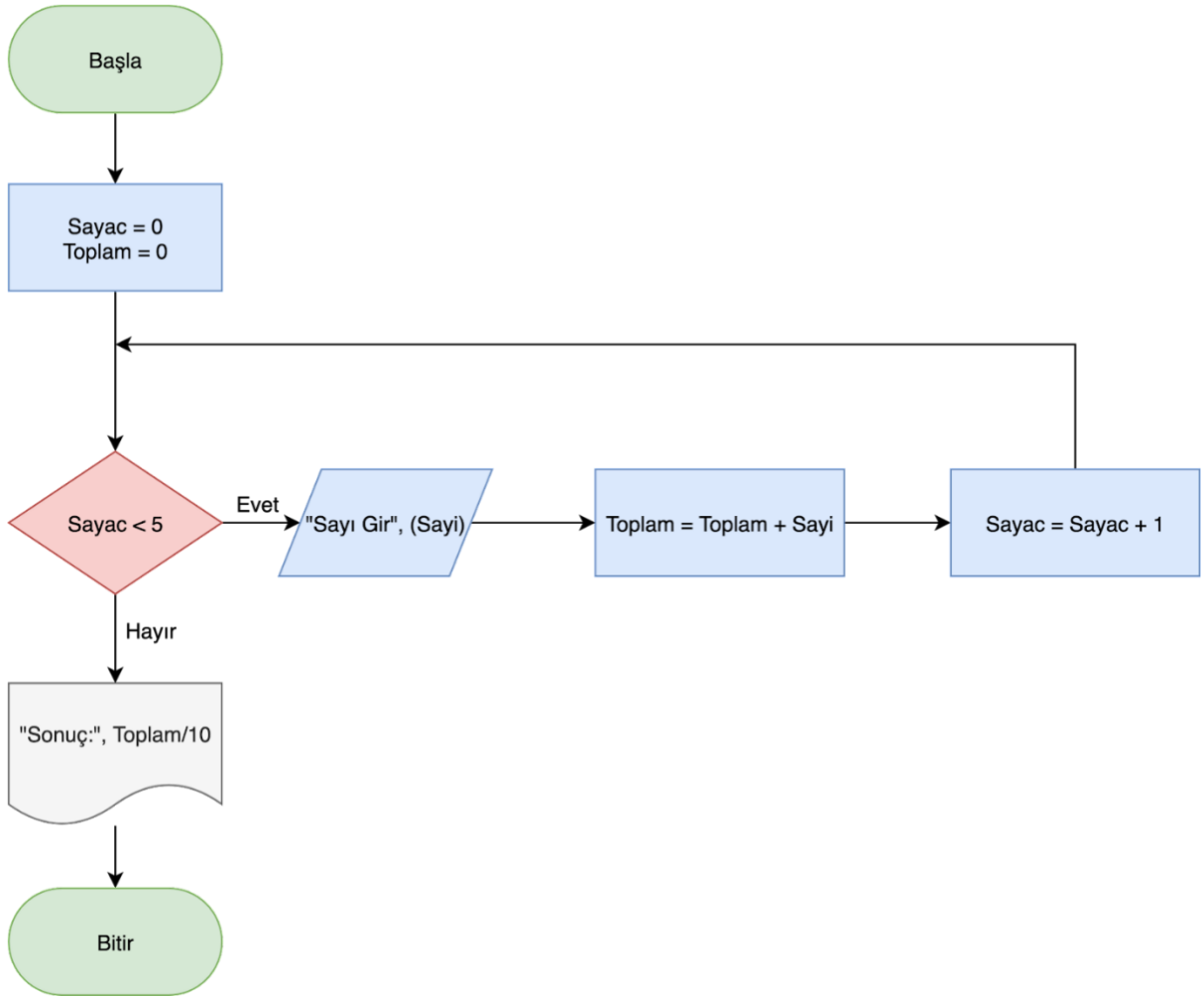


Analizci: Akış şeması verilen algoritmanın temel problemini tahmin edin. Sayı=5 için çıktı nedir?



Yanıt: 1'den Klavyeden girilen sayıya kadar olan tüm sayıları toplar. Sayı 5 için, Sonuc = 1 + 2 + 3 + 4 + 5 = 15 olur.

Denetleyici: Akış şeması verilen algoritmanın ekran çıktısını (4,6,8,4,2) için tahmin edin.



Yanıt: $4+6+8+4+2 = 24$

Çıktı = $24/10 = 2.4$ olur.

Hafta 2. Ders Materyalleri

EK 1. Algoritmaları Eşleştirelim Çalışma Kâğıdı

Aşağıda metin şeklinde yazımı, sözde kod ve akış diyagramı olmak üzere bir probleme üç farklı şekilde algoritma yazılmıştır. Hangi algoritma hangi probleme aittir? lütfen eşleştiriniz. Eşleştirme için aşağıdaki kısaltmalardan uygun olanı kutucuk altında ilgili alana yazınız.

Problemler

P1. Çemberin alanını hesaplama

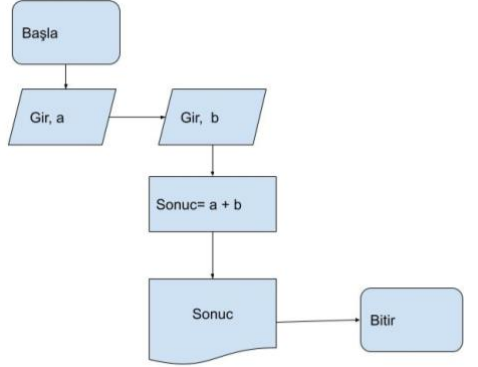
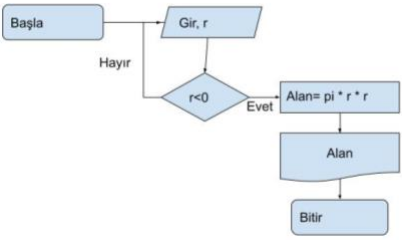
P2. İki sayıyı toplama

Algoritma İfade Şekli

MY: Metinsel Yazım

SK: Sözde Kod

AD: Akış Diyagramı

1. Başla 2. Birinci sayıyı gir (a) 3. İkinci sayıyı gir (b) 4. İşlemi yap. (sonuc = a+b) 5. Ekrana yaz (sonuc) 6. Bitir.	1. Başla. 2. Gir, r 3. Eğer $r < 0$ ikinci adıma git. 4. alan= πr^2 5. Yaz, Sonuc 6. Bitir.	
Problem: Algoritma İfadesi:	Problem: Algoritma İfadesi:	Problem: Algoritma İfadesi:
	Başla. 1. Yarıçap gir (r). 2. Eğer yarıçap sıfırdan küçükse ikinci adıma git. 3. Alanı hesapla (alan= πr^2) 4. Sonucu ekrana yazdır. 5. Bitir.	1. Başla 2. Gir, a 3. Gir, b 4. Sonuc = a+b 5. Yaz, Sonuc 6. Bitir.
Problem: Algoritma İfadesi:	Problem: Algoritma İfadesi:	Problem: Algoritma İfadesi:

EK 2. Akış Diyagramları Bilgi Afişi

BAŞLA

Başla ve Bitir adımları yandaki şekildeki gibi gösterilir. Algoritmamızı meydana getirecek olan diğer bütün adımlar bu iki şekil arasına eklenir.

BİTİR

İŞLEMLER

(+ - / *)

Toplama, çıkarma, aritmetik işlemlerin yapılması gerektiği durumlarda dikdörtgen kullanılır. Örneğin; üçgenin alanını hesaplama

Kullanıcının
Gireceği
Değerler

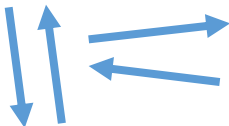
Kullanıcı girişlerini şekilsel olarak göstermek için paralelkenar kullanılır. Örneğin; Kullanıcıdan istenen ad, soyad, yaş vb.)

ÇIKTI

Yukarıda kullanıcıdan alınan bilgileri ekrana yazdırılması gerektiğinde yandaki şekil kullanılır.

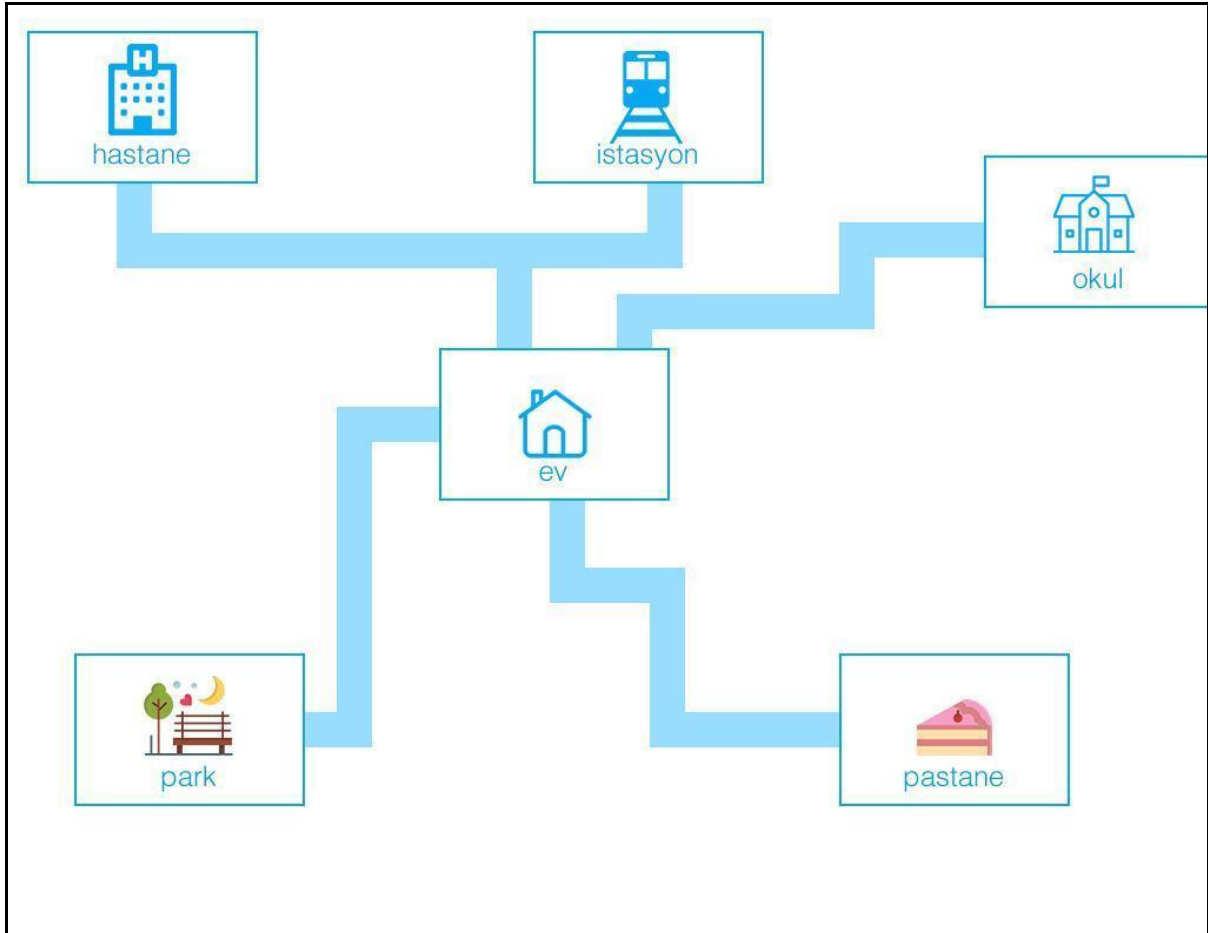
Karar
Yapıları

Algoritma akışında işlemler tek yönlü iken karar yapılarında iki farklı ok çıkabilir ve bazı işlemlerin tekrarlanmasını sağlayabilir. Bu şekilde bir



Yandaki oklar algoritmanın akış yönünü bize belirtir.

EK 3. Nereye Gidelim Haritası ve Görevler



Oda 1 Görev: Evinden çıkıp okula gitmek isteyen Arda'nın, izlemesi gereken yolu tarif eden bir algoritma oluşturunuz.

Oda 2 Görev: Evinden çıkıp hastaneye gitmek isteyen Karaca'nın, izlemesi gereken yolu tarif eden bir algoritma oluşturunuz.

Oda 3 Görev: Evinden çıkıp parka gitmek isteyen Defne'nin izlemesi gereken yolu tarif eden bir algoritma oluşturunuz.

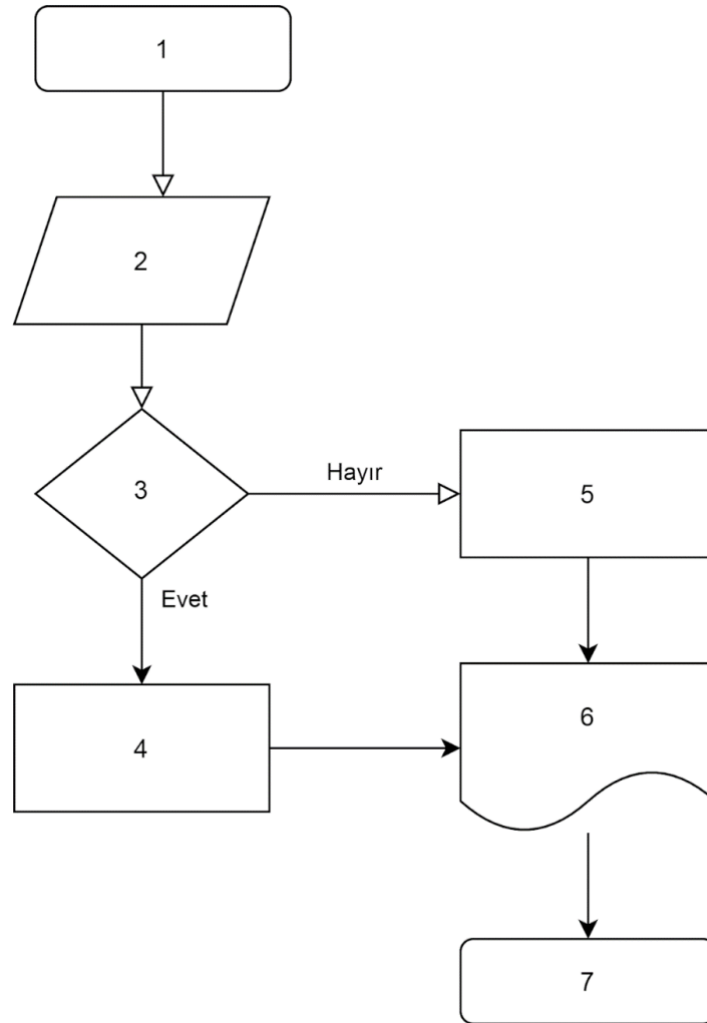
Oda 4 Görev: Evinden çıkıp pastaneye gitmek isteyen Güney'in, izlemesi gereken yolu tarif eden bir algoritma oluşturunuz.

Oda 5 Görev: Hastaneden istasyona gitmek isteyen Karaca'nın izlemesi gereken yolu tarif eden bir algoritma oluşturunuz.

EK 4. Çıkartma Algoritması Çalışma Kâğıdı

Görev: Aşağıdaki problemin çözümüne yönelik sözde kodu verilen çıkartma algoritmasını doğru sıralayarak akış diyagramındaki numaralarla eşleştiriniz.

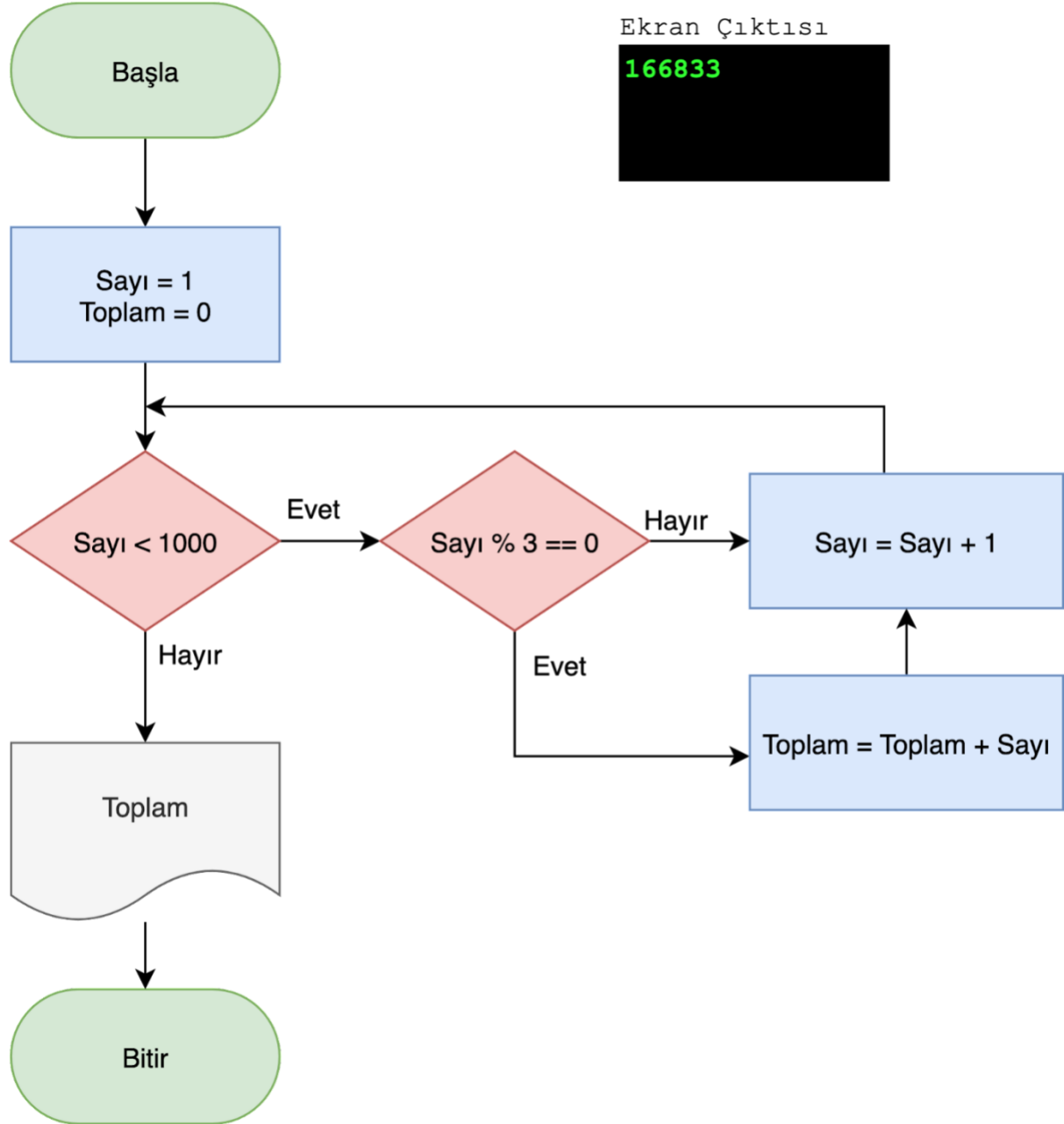
Problem: Klavyeden girilen iki sayı arasında küçük olanı, büyük olandan çıkartma algoritması.



Sözde Kod	Numarası	Sözde Kod	Numarası
Başla		Gir, Sayi1, Sayi2	
Yaz, Sonuc		Eğer Sayi1 > Sayi2	
Değilse, Sonuc = Sayi2 - Sayi1		Bitir	
Sonuc=Sayi1 - Sayi2			

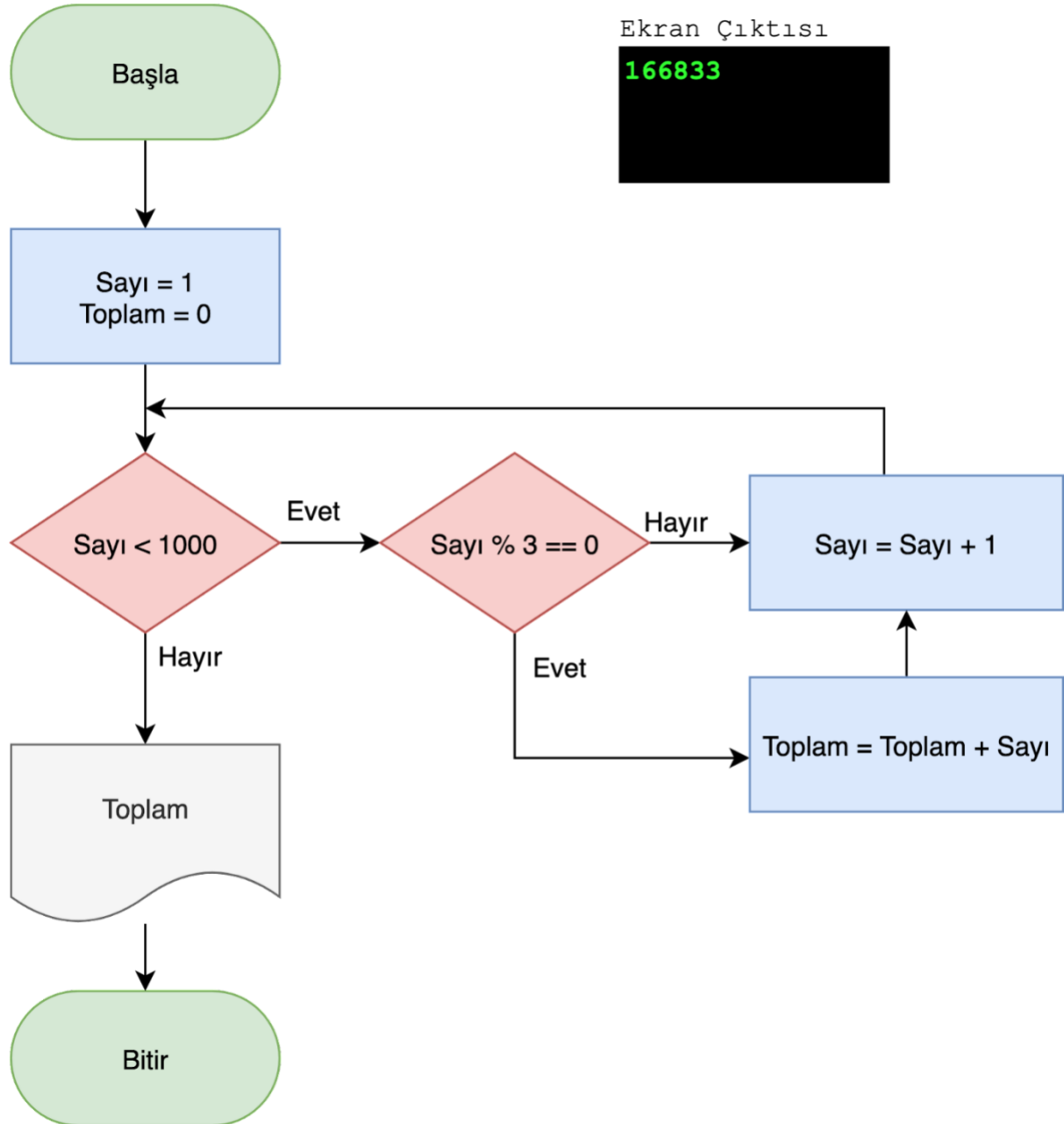
EK 5. Grup Görevleri**Grup 1**

Aşağıda verilen akış şemasında $Sayı = Sayı + 1$ basamağı çıkarılırsa, algoritmanın temel problemi nasıl değişir?



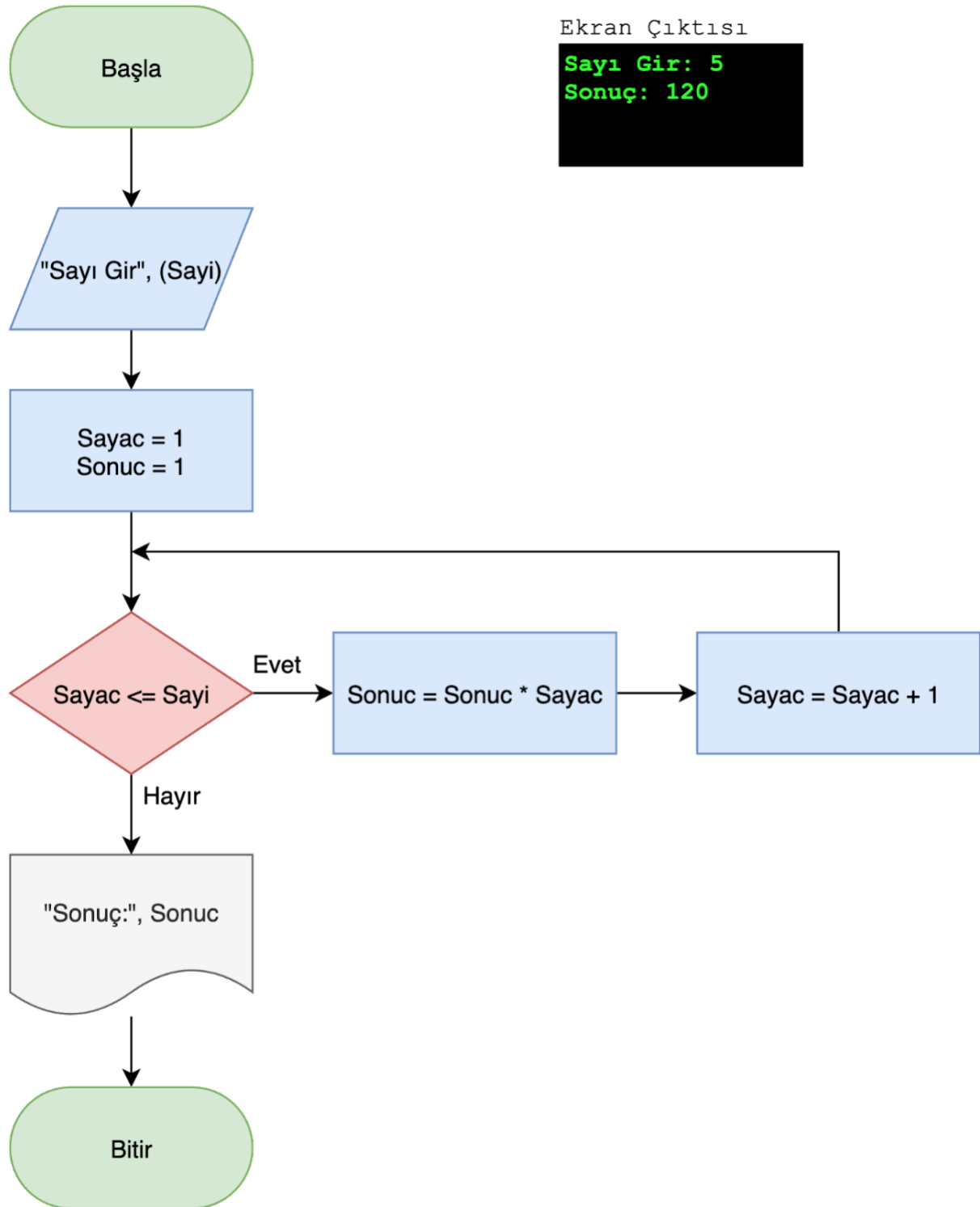
Grup 2

Aşağıda verilen akış şemasında Sayı %3 == 0 olan basamak Sayı %9 == 0 ile değiştirilirse, algoritmanın temel problemi nasıl değişir?



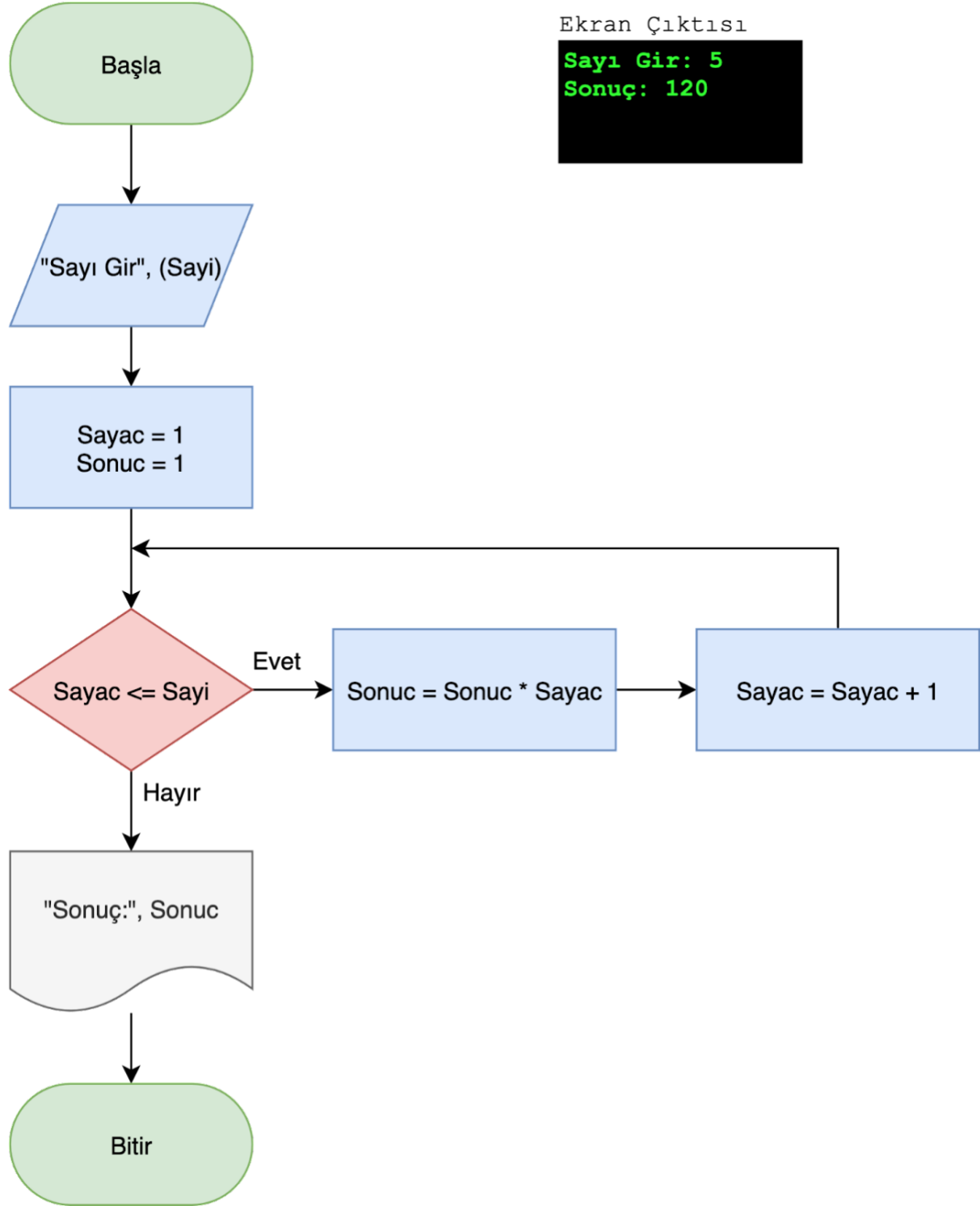
Grup 3

Aşağıda verilen akış şemasına ait programda $\text{Sayac} \leq \text{Sayi}$ basamağı çıkarılırsa program çıktısı nasıl değişir?



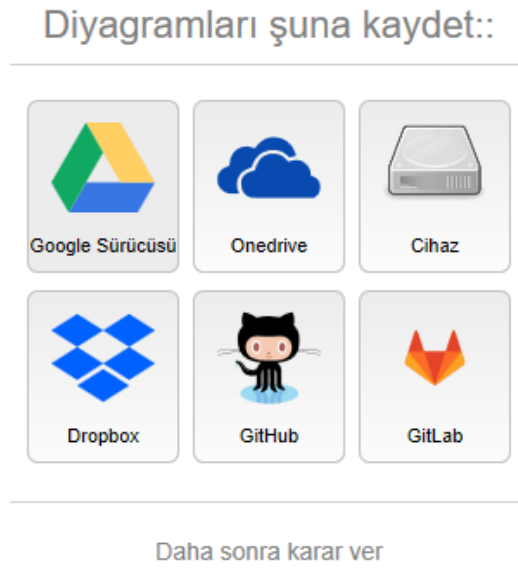
Grup 4

Aşağıda verilen akış şemasına ait programda girilen sayı 4 olsaydı, algoritmanın ekran çıktısı nasıl değişirdi?



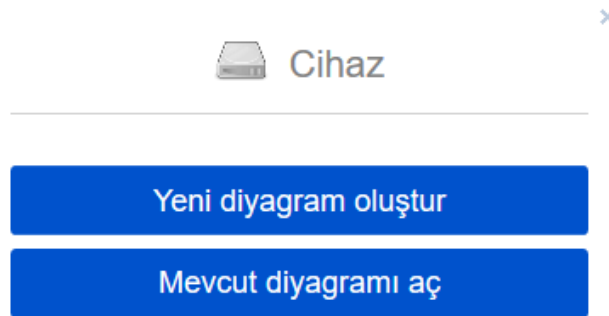
EK 6. App.diagrams.net Program Kullanım Kılavuzu

Akış diyagramını bilgisayar ortamında çizmek için app.diagrams.net isimli websitesi kullanacağız. Ücretsiz olan bu uygulama sayesinde çizdiğimiz şekilleri kaydedebilir, istediğimiz kişilerle paylaşabiliriz. App.diagrams.net isimli uygulamaya ilk giriş yaptığımızda aşağıdaki görüntü karşımıza gelecektir. Çizdiğimiz diyagramı nereye kaydetmek istediğimizi belirtiyoruz. Cihaz seçeneğini tıklayarak, şimdilik kendi bilgisayarımızda kaydedelim.



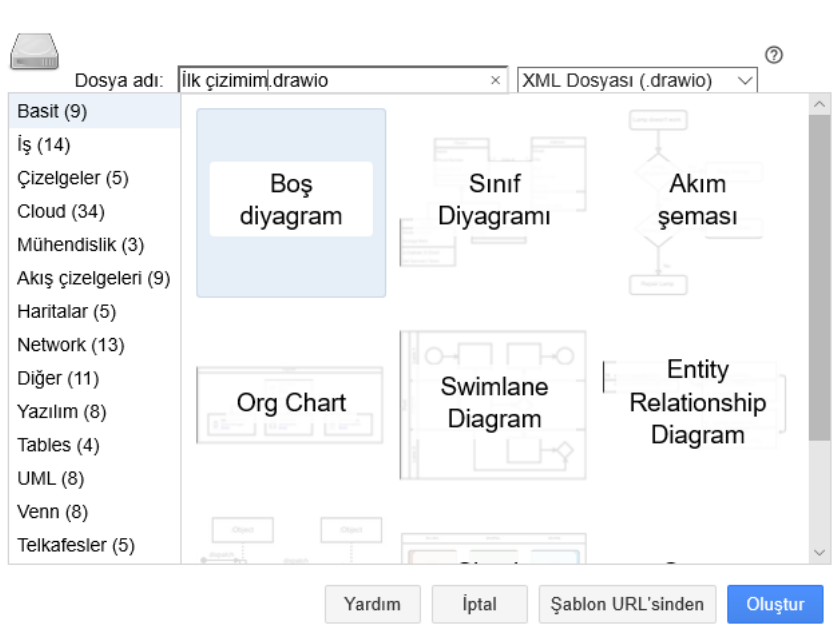
Resim 13. App.diagrams.net giriş ekranı

Ardından, yeni diyagram oluştur butonuna tıklayarak boş bir sayfa oluşturalım.



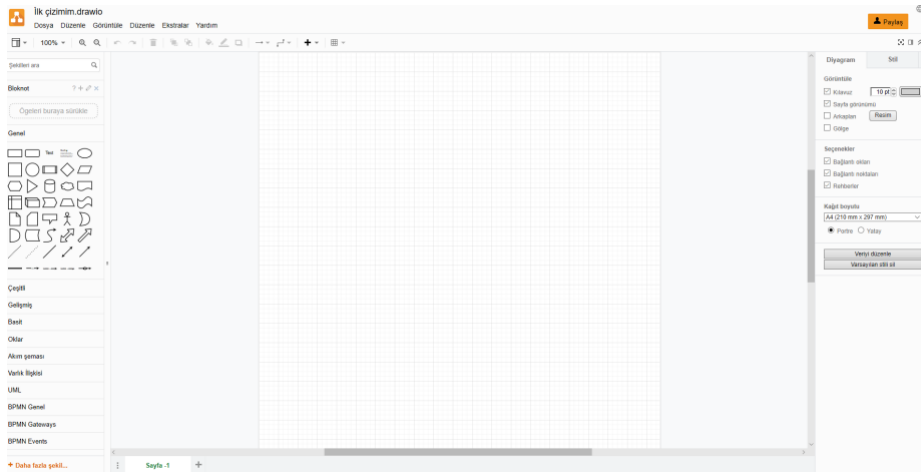
Resim 14. App.diagrams.net yeni diyagram oluşturma ekranı

App.diagrams.net ile basitten profesyonel seviyeye kadar birçok seviyede çizim yapılabilir. Biz şimdilik, boş diyagram seçeneğini seçerek bir isim verebiliriz.



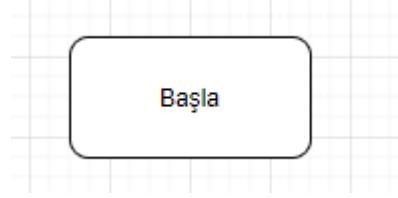
Resim 15. App.diagrams.net yeni diyagram tipi seçme ekranı

Karşımıza aşağıdaki çizim ekranı gelecektir.



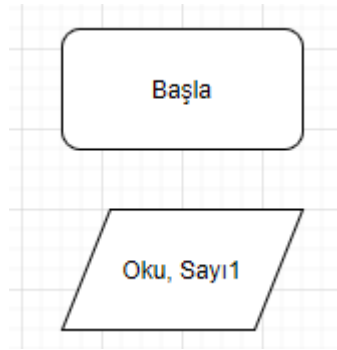
Resim 16. App.diagrams.net çizim ekranı

Bu aşamada **Genel** isimli araç kutusunda öğrendiğimiz bloklar mevcuttur. Ekleme istediğimiz bloğu çizim alanına sürükleyip bırak yapıyoruz. Ardından blok üzerine çift tıklayarak içerisine yazı ekliyoruz.



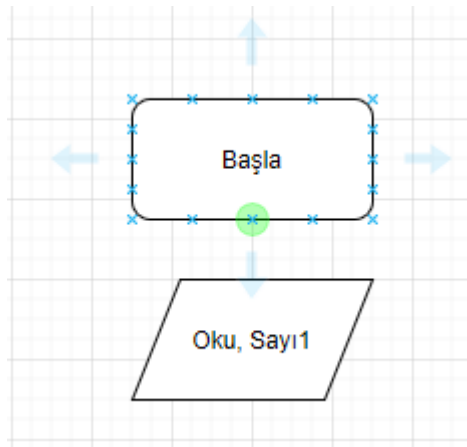
Resim 17. Genel bloğu

Ardından bir tane de paralel kenar ekleyelim.



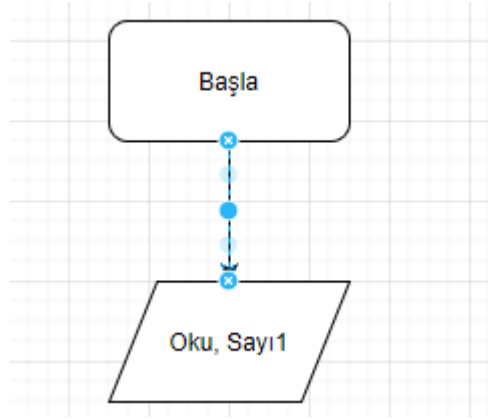
Resim 18. Paralel kenar bloğu

İki bloğu birbirine bağlamak için ilk bloğun üzerine fareyi getiriyoruz. Fare üzerine geldiğinde, çizgi çizebileceğimiz noktaları görüyoruz.



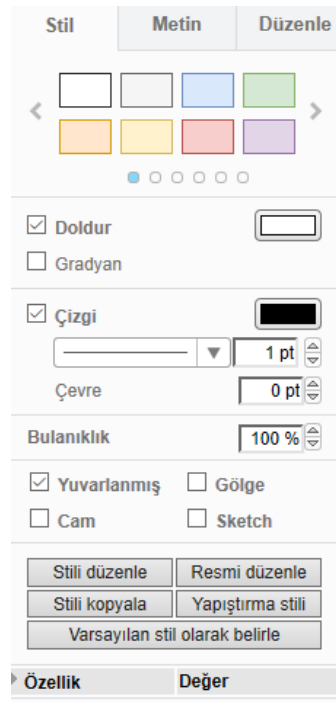
Resim 19. Blokları bağlama öncesi

Ardından yine fare yardımıyla, hangi sonraki blok ile bağlantıyı tamamlıyoruz.



Resim 20. Blokları bağlama sonrası

Eğer bloğun renklerini değiştirmek istersek sağ paneli kullanabiliriz.



Resim 21. Blok özellikleri ekranı

EK 7. Hafta 2 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 3. Algoritmada Değişkenler ve Değerleri

Kazanımlar

- K1. Verilen algoritmada temel kavramları analiz eder.
- K2. İhtiyaç duyulan algoritmayı tasarlar.
- K3. Bir problemin çözümünde değişkenleri kullanır.
- K4. Verilen algoritmanın akış diyagramını sözde koda dönüştürür.
- K5. Verilen algoritmanın akış diyagramındaki değişkenleri ayırt eder.

Haftanın Amacı ve Materyaller

Bu haftanın amacı bir problemin çözümü için tasarlanan akış diyagramlarında kullanılan değişkenler ve değişkenlerin değerlerini keşfetmektir.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız](#).

Önerilen Ders Akışı (50 + 50 + 50 + 50 dk.)

GİRİŞ (10 dk.)

GELİŞME

Algoritma Terimlerini Keşfedelim (40 dk.)

Ders Arası (10 dk.)

Otomatik Park Edelim (20 dk.)

Değişkenler Değerlidir (30 dk.)

Ders Arası (10 dk.)

Algoritmayı Adım Adım Çalıştıralım (40 dk.)

EK 1. Kod Çalıştırma Tablosu

SONUÇ (10 dk.)

EK 2. Hafta 3 Süreli Ödev (50 dk.)

EK 3 Ders Dışı Aktivite Sunusu

EK 4 Derleyici Kullanım Kılavuzu

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 3 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 3 sunumunu ekran paylaşımına açar. Eğitmen geçen hafta işledikleri konu hakkında özet geçerek *ön bilgileri hatırlatır* ve bu haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

B. Gelişme

Algoritma Terimlerini Keşfedelim

Süre: 40 dk.

Kazanımlar: K3. Değişken kavramının özelliklerini bilir.

Materyaller: Hafta 3 Sunum

Hazırlık: Hafta 3 Sunumunda algoritma terimlerini keşfet slayt açılır.

Uygulama: Eğitmen bu etkinlikte sunum üzerinden ilerleyecektir. Öğrencilere sunumda EK 1 üzerinden algoritma terimleri hakkında bilgi verilecektir. Bu aşamada çiftleri toplama algoritması üzerinden terimler örnek görevler üzerinden açıklanır. Bazı terimlerde öğrencilerden rastgele seçimler yapılarak, performanslarını yansıtmaları için onlara söz hakkı verilir. Bu terimler ve çiftleri toplama algoritması ile ilişkisi aşağıdaki konu içeriğine uygun olacak şekilde sunulur.

Konu İçeriği:

1) Tanımlayıcı: Algoritmadaki değişkenleri, sabitleri, kayıt alanlarını ve özel bilgi tiplerini adlandırmak için programcı tarafından gerçekleştirilen işlemlerdir. Tanımlayıcıların yerini tuttukları ifadelere çağrışım yapacak şekilde adlandırılmaları algoritmanın anlaşılması açısından önemlidir. Tanımlayıcı isimlerinde İngiliz alfabesindeki “A-Z” ve “a-z” arası harfler, “0-9” arası

rakamlar, sembollerden alt çizgi “_” kullanılabilir. Bununla birlikte tanımlayıcı ismi, rakamla başlayamaz veya sadece rakamlardan oluşamaz.

Soru: Çiftleri toplama algoritmasında kullanılan tanımlayıcı kuralları nelerdir?

2) Değişken: Algoritmanın her çalıştırılmasında, girdiğimiz değerleri alan veya programın çalışmasıyla bazı değerlerin atandığı bellek alanlarıdır. Değişken isimlendirilmeleri, tanımlayıcı kurallarına uygun biçimde yapılmalıdır. Örneğin bir ismin aktarıldığı değişken “ad” olarak, bir isim ve soy ismin aktarıldığı değişken “adSoyad”, ev telefon numarasını aktarıldığı değişken “evTel”, ev adresinin aktarıldığı değişken “evAdres” ve iş adresinin aktarıldığı değişken “isAdres” olarak tanımlanabilir. Burada özellikle İngiliz alfabesini kullandığımıza dikkat edelim.

Görev: Çiftleri toplama algoritmasındaki değişkenleri bulup, isimlerini değiştirmeyi deneyin.

3) Sabit: Uygulamanın çalıştığı süre boyunca, içeriği sabit olan değer ve ifadelerin saklanması için kullanılır. Algoritma boyunca kolay takip edilebilmesi için kullanılmaktadır. Değişkenler gibi isimlendirme kurallarına uygun olarak oluşturulmalıdırlar.

Görev: Çiftleri toplama algoritmasındaki sabitleri bulun.

4) Atama/Aktarma: Bir değişkene değer atamak için gerçekleştirilen işlemdir. Algoritma adımlarında sağdaki değeri soldaki değişkene atamak için kullanılan bir işlemdir.

Görev: Çiftleri toplama algoritmasındaki değişken atama ifadelerini bulun.

5) Sayaç: Algoritma tasarımlarında bazı işlemlerin belirli sayıda yaptırılması ve bu süreçte oluşan değerlerin sayılması gerekebilir. Bu amaçla kullanılan sayma işlemlerine sayaç denir.

Görev: Çiftleri toplama algoritmasına yeni bir sayaç eklemek için problemi değiştirin.

6) Döngü: Algoritma tasarımlarında bir veya birden fazla işlem satırını, bir koşula bağlı olarak, belirli sayıda veya bir koşul sağlandığı sürece tekrarlayarak çalıştıran kalıplardır. Örneğin 1 ile 100 arasındaki çift sayıların toplamını hesaplayan programda $T=2+4+6 \dots +100$ hesabını teker teker yapmak yerine 1 ile 100 arasında ikişer artan bir döngü kullanmak ve döngü değişkenini toplam hesabında kullanmak daha uygun olacaktır.

Görev: Çiftleri toplama algoritmasındaki döngüyü bulup, önemini tartışın.

7) Ardışık Toplama/Çarpma: Algoritmelerde var olan değere yeni bir değerin eklenmesi ya da var olan değerin yeni bir değerlerle çarpılarak oluşan bu yeni değerin kullanılması işlemidir.

Görev: Çiftleri toplama algoritmasındaki ardışık toplama ifadesi yerine, ardışık çıkarma yapıldığında oluşan yeni problemi tartışın.

Otomatik Park Edelim

Süre: 20 dk.

Kazanımlar: K2. İhtiyaç duyulan algoritmanın tasarımını sağlar.

Materyaller: Hafta 3 Sunum

Hazırlık: Hafta 3 Sunumunda Otomatik Park Etme algoritması başlıklı slayt açılır.

Uygulama: Eğitimci sunum üzerinde otomatik park etme algoritmasını açarak uyarıcı materyali sunar. Öğrencilere bu algoritma üzerinde aşağıdaki problemi sorar ve algoritmayı tekrar düşünmelerini ister.

Algoritmayı çalıştırdığınızda nasıl bir sonuçla karşılaşırız?

Eğitmen öğrencilerin yanıtlarını sohbet ortamından yazarak performanslarını göstermelerini ister. Sohbet ortamında gelen yanıtlara geri bildirimlerde bulunan eğitimci sorunun yanıtını beyin fırtınası yoluyla açıklayarak, algoritmalarda 'Sonsuz döngü' kavramına aşağıdaki gibi dikkat çeker.

*Akıllı araç ilk sokağa geldiğinde yer olmadığını (sensörler yardımıyla) algılayacak ve sonraki sokağa kadar ilerleyecektir. İkinci sokakta P2 ve P3 alanlarının boş olması durumunda P2 alanına aracını park edecektir. Eğer otoparkta boş yer yok ise, 2. ve 3. adımlarda **sonsuz döngü** dediğimiz istenmeyen durumla karşılaşılacaktır.*

Eğitmen daha sonra ikinci bir soru yöneltir ve tekrar sohbetten yanıtları alır.

Peki bu algoritmanın daha iyi çalışması (sonsuz döngüden kurtulması) için basamaklarda nasıl bir değişiklik yapılmalıdır?

Öğrencilerden gelen yanıtlara geri bildirimlerde bulunarak sonsuz döngüden kurtulma konusuna dikkat çeker. Konuyu aşağıdaki gibi özetler.

Bu durumu da göz önüne alarak, otoparkın sonuna gelindiğinde akışın sonlanmasını istiyorsak aşağıdaki şekilde yazabiliriz.

Algoritmayı Sonsuz Döngüden Kurtarma:

1. Başla
2. Sonraki sokağa kadar ilerle.
3. Eğer otoparkın sonuna geldiyse 7. adıma git.
4. Eğer sokakta yer yoksa 2. adıma git.
5. Sokağa gir.
6. İlk uygun yere park et.
7. Bitir.

Eğitmene Öneriler: Eğitimci öğrencilere algorithmadaki problemi keşfetmeleri için ipuçları kullanabilir. Örneğin; 'Basamakları tek tek çalıştırmayı dene'; 'İkinci basamağı tekrar çalıştırmayı düşünebilirsin'; 'son basamağın nasıl çalıştığına dikkat et' vb. gibi ifadelerle problemin kaynağını doğrudan öğrenciye aktarmaktan kaçınılmalıdır.

Değişkenler Değerlidir

Süre: 30 dk.

Kazanımlar: K3. Değişken kavramının özelliklerini bilir.

Materyaller: Hafta 3 Sunumu

Hazırlık: Hafta 3 Sunumunda Değişkenler Değerlidir başlıklı slayt açılır.

Uygulama: Eğitimci öğrencilere aşağıdaki örnek olayı verir.

Örnek Olay: Arkadaşınız aklından iki sayı tutmuştur ve sizden bu iki sayıdan büyük olanı bulmanızı istiyor. Bunun için bir algoritma yazmak istiyorsunuz.

Eğitmen ilgili algoritmanın değişken, değişken değerleri ve ekran çıktısını içeren aşağıdaki talimatı verir.

Talimat 1: Örnek olaya ilişkin algoritma için değişkenleri ve değerlerini düşünün. Hangi değişkenler algoritmada yer almalıdır? Değişken isimleri nasıl olmalıdır?

Sohbet üzerinden öğrencilerden olası değişkenlerin isimlerini yazmaları istenir. Bu şekilde değişken kavramı ile ilgili değişken ismi yazma üzerine öğrenci performansları alınır. Eğitimci gelen yanıtlara bağlı olarak aşağıdaki gibi değişkenler ve değerlerine dikkat çekerek öğrencilere geri bildirim verir.

Değişken zamanla değeri değişen veri saklayıcıdır. Örneğin yaş bilgisi değişkendir. Yaşınızın kaç olduğu ise değerdir. Bilgisayar ortamında değişkenler hafızadan yer ayırmak için kullanılır. Daha sonra içerisine değerler yazarak kullanılır. İçerisindeki değerleri program çalıştığı süre boyunca unutmazlar ve siz değiştirmedığınız sürece değişmezler. Algoritma ya da program yazarken genellikle daha kısa ve anlamlı isimler veririz. Değişken ve değerlere günlük yaşamdan örnekleri aşağıdaki tabloda verilmektedir.

Tablo 10. Değişken örnekleri

Değişken	Değişken İsmi	Değer
Adınız	Ad	"Ahmet", "Mehmet", "Ali", "Ayşe"
Yaşınız	Yas	"12", "13", "25"
Boyunuz	Boy	"150 cm", "140 cm"
Telefon numaranız	Tel	"05555555555"
Sınıfınız	Sinif	"4", "5", "6", "7"

Eğitmen bir sonraki slayta ilerleyerek değişkenlerin sözde kodlarına ilişkin ekran çıktısını düşünmelerini ister. Eğitimci öğrencilere ikinci bir talimat verir.

Talimat 2: Örnek olaydaki algoritmanın sözde kodlarına ilişkin ekran çıktısını düşünün. Değişken değerlerini değiştirip, ekran çıktısını birlikte tahmin etmeye çalışalım.

Öğrencilerden sohbette bir değişken değeri istenir ve ilk yazılan değer kod üzerinde eğitmen eşliğinde test edilir. Ekran çıktısının nasıl değiştiğine dikkat çekilir. Bu noktada eğitmen yeni değişken değerini çalıştırırken sözde koda ait her bir adımın açıklaması için aşağıdaki tabloyu kullanılır.

Tablo 11. Sözde kod örneği

Adım	Açıklama
Başla	Programın Başlangıcı
Oku, (Sayi1,Sayi2)	Kullanıcıdan değerlerin alınması. Burada biz klavyeden alındığı düşünüyoruz. Eğer bir mobil cihaz üzerinde ya da web sitesi üzerinde çalışacaksa farklı teknikler kullanılabilir.
Eğer Sayi1>=Sayi2	Sayi1, Sayi2 'den büyük eşit olup olmadığı kontrol ediliyor. Eğer bu koşul doğru ise bir alt satıra devam edeceğiz. Değilse aksi takdirde yazan satıra gideceğiz. Eğer aksi takdirde satırı yok ise bir alt adımdan devam edeceğiz.
3.1) Yaz, Sayi1	Ekрана Sayi1 'in değerini yaz.
Aksi takdirde	Sayi1, Sayi2 'den büyük değilse
4.1) Yaz, Sayi2	Ekрана Sayi2 'nin değerini yaz
Bitir.	Programın Sonu

Algoritmayı Adım Adım Çalıştırma

Süre: 40 dk.

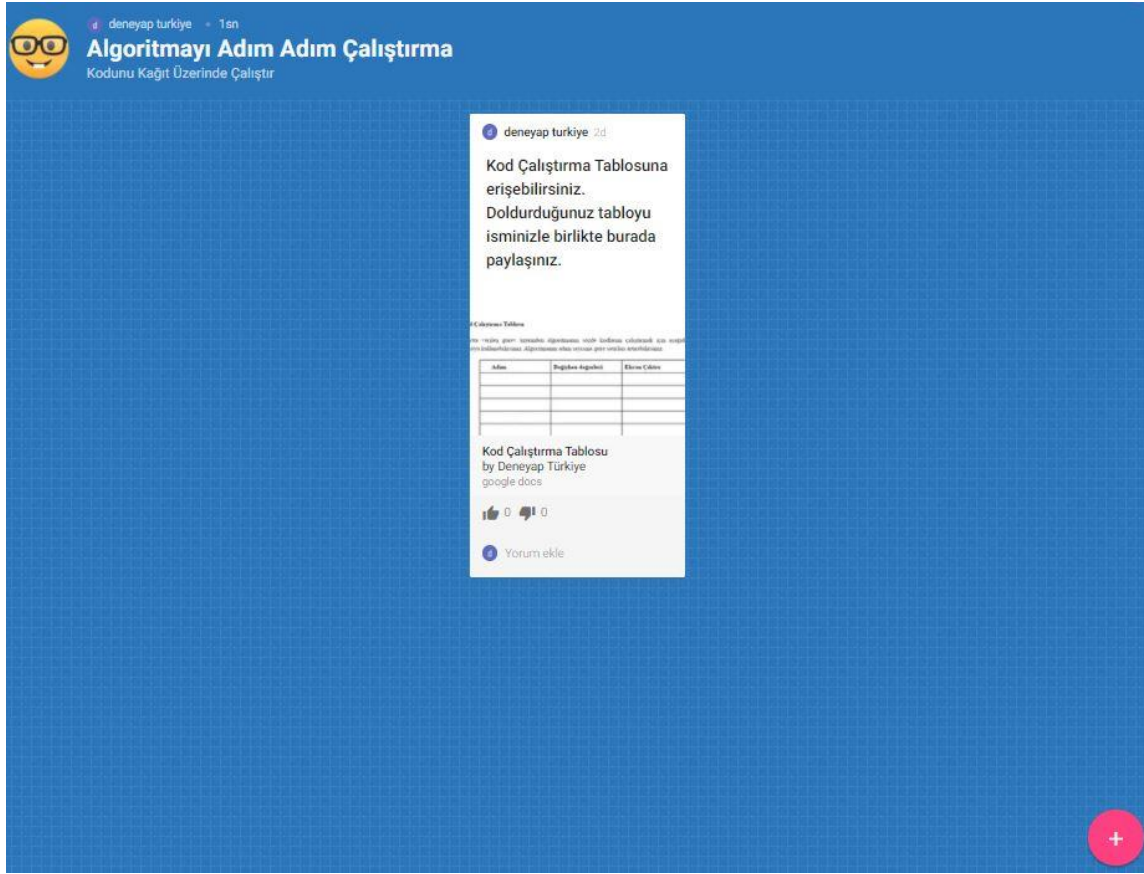
Kazanımlar: K4. Akış diyagramlarını sözde koda dönüştürür.

K5. Akış diyagramında değişkenleri ayırt eder.

Materyaller: Dijital Tartışma Panosu: Bireysel Çalışma

EK 1. Kod Çalıştırma Tablosunu Doldurma

Hazırlık: Hafta 3 sunumunda etkinliğin slaytı açılır. Bireysel öğrenci ürünlerinin gönderilmesi için duvar temalı padlet ortamında bir dijital tartışma panosu oluşturulur. Link ders sırasında öğrencilere paylaşılan notlar ya da sohbet kısmını kullanır. Örnek bir tasarım Resim 22'de gösterilmiştir. EK 1 materyali dijital tartışma panosuna yüklenecektir.



Resim 22. Algoritmayı Adım Adım Çalıştırma Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci slaytta ilerleyerek öğrencilere akış şeması boş olan bir algoritma görevi gösterir. Eğitimci öğrencilere bu akış şemasındaki sözde kodları doğru yerlere yerleştirip, kodları adım adım çalıştırmayı deneyeceklerini belirtir. Bunun için ilk olarak eğitimci anket kullanır. Anket ya da sohbet aracılığıyla öğrencilere akış şemasındaki birinci şekle gelecek sözde kodu soracaktır. Benzer şekilde eğitimci anket ya da sohbet aracılığıyla tüm şekilleri öğrencilerle birlikte doldurmaktadır. Bu şekilde öğrencilerin performanslarını yansıtılmaları beklenir. Daha sonra slaytta ilerlenir ve akış şemasının doldurulmuş hali gösterilir. Bu aşamada eğitimci öğrencilere yeni bir görev verir. Görev öncesi öğrencilerden ilk olarak EK 1'i bilgisayarlarına indirmeleri istenir. Eğitimci ders sırasında materyalinin bağlantı linkini ve materyali doldurduktan sonra gönderi yapacakları dijital tartışma panosunun linkini paylaşılan notlar kısmından iletir. EK 1'i indiren öğrenciler slaytta eğitimcinin gösterdiği akış şemasını adım adım kod tablosunda çalıştırmak için kullanır. Öğrenciler bu işlemi bireysel olarak kendi bilgisayarlarında dolduracaktır. Bunun için öğrencilere 6-7 dk.lık bir müzik açılır. Müzik sonunda doldurdıkları materyali herkesin görebilmesi için tartışma panosuna yüklemeleri istenir.

Eğitimci öğrencilere 2 dk.'lık bir müzik daha açar. Öğrencilerden bu sefer padlete yüklenen dosyaları incelemeleri ve en doğru olduğunu düşündükleri gönderileri beğenmelerini ister. Bu şekilde öğrenciler diğerlerinin gönderilerini değerlendirerek, performanslarını sergileyecektir. Eğitimci müzik sonunda en çok beğeni alan kişinin gönderisi üzerinden konuyu özetler ve varsa hatalarla ilgili öğrencilere geri bildirimde bulunur.

Konu İçeriği:

Görev: Bir okuldaki 10 öğrencinin sınav notunun ortalamasını hesaplamak istiyorsunuz. Bunun için akış şemasındaki boş yerleri doldurarak algoritmayı tasarlayınız. Bu akış şemasındaki sözde kodları oluşturup adım adım kodları çalıştırmak için Kod Çalıştırma tablosunu oluşturun.

Hazırlanacak kod çalıştırma tablosu aşağıdaki gibi olmalıdır.

Tablo 12. Kod çalıştırma tablosu

Adım	Değişken değerleri	Ekran Çıktısı
Başla		
Sayac=0, Toplam=0	Sayac=1 Toplam=0	
Eğer $0 < 10$	Doğru, Evet	
Oku, Sayi	Sayi=1	
Toplam = $0 + 1$	Toplam = 1	
Sayac = $0 + 1$	Sayac =1	
Eğer $1 < 10$	Doğru, Evet	
Oku, Sayi	Sayi=2	
Toplam = $1 + 2$	Toplam = 3	
Sayac = $1 + 1$	Sayac =2	
Eğer $2 < 10$	Doğru, Evet	
Oku, Sayi	Sayi=3	
Toplam = $3 + 3$	Toplam = 6	
Sayac = $2 + 1$	Sayac =3	
Eğer $3 < 10$	Doğru, Evet	
Oku, Sayi	Sayi=4	
Toplam = $6 + 4$	Toplam = 10	
Sayac = $0 + 1$	Sayac =4	
... Girilen Sayılar: 1,2,3,4,5,6,7,8,9,10		
Eğer $9 < 100$	Doğru, Evet	
Oku, Sayi	10	
Toplam = $45 + 10$	Toplam = 55	
Sayac = $9 + 1$	Sayac =10	
Eğer $10 < 10$	Yanlış, Hayır	
Yaz, 55 / 10		5.5
Bitir	Sayac = 10 Toplam=55	

Eğitmene Öneriler: Öğrencilerin hazırladığı tablolar üzerinden beyin fırtınası yoluyla eğitmen adımları aşağıdaki gibi özetler.

Tablo 13. Sözde kod örneği

Adım	Açıklama
Başla	Programın Başlangıcı
Sayac=0, Toplam=0	Burada bize iki adet değişken gerekli. Kaç adet sayı girildiğini Sayac isimli değişkende tutuyoruz. İsteddiğimiz sayıların toplamını ise Toplam isimli değişkende tutuyoruz.
Sayac<10 Olduğu Sürece	Buradaki koşulumuz 10 değerine ulaşmış ulaşmadığımız. Sayac değişkenimiz 10'a ulaşana kadar devam edeceğiz.
3.1) Oku, Sayı	Klavyeden sayı al. Sayı isimli değişkene at.
3.2) Toplam=Toplam+Sayı	Toplam değişkenine bu sayıyı ekle.
3.3) Sayac= Sayac+1	Elimizdeki Sayac değerini bir arttırıyoruz.
Yaz, Toplam/10	Ekrana Toplam/10 değerini yazdırıyoruz.
Bitir.	Programın sonu

C. Sonuç

Süre: 10 dk.

Materyal: EK 2 Hafta 3 Süreli Ödev

EK 3 Ders Dışı Aktivite Sunusu

EK 4 Derleyici Kullanım Kılavuzu

Hazırlık: EK 2 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek haftanın konusu hakkında bilgi verilir.

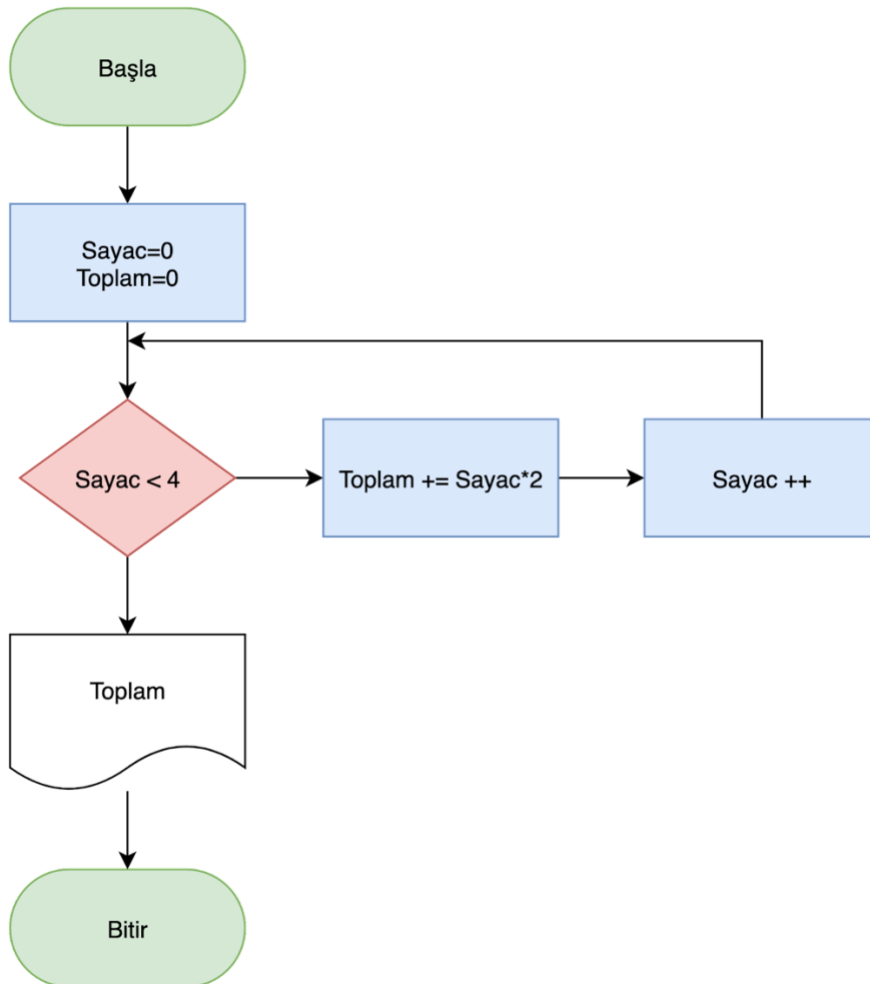
Ders dışı uygulama: Bu uygulama asenkron olarak ÖYS üzerinden yürütülecek iki aşamadan oluşmaktadır. İlk aşamada öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. EK 2 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitmen

görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Ders dışı uygulamanın ikinci aşamasında ise, öğrenciler bireysel olarak C++ programlamanın tarihi, derleyici ihtiyacı ve seçimi ile ilgili bilgi edinebilecekleri EK 3 sunusunu inceleyecektir. Bu sunumun sonunda yer alan EK 4 derleyici kurulum kılavuzunu takip ederek, bilgisayarlarına kurulum gerçekleştirecektir. Bunun için üçüncü hafta dersine gelmeden önce öğrencilerin derleyiciyi bilgisayarlarında kurmuş olmaları beklenmektedir.

Süreli Ödev Yanıtlar: Görevler, öğrenciye verilecek beceri rozetleri ile isimlendirilmiştir. Her bir rozetin altında ilgili görevin yanıtları verilmektedir. Eğitimci bu yanıtları uygulama süresinin sonunda GitHub ya da ÖYS üzerinden öğrencilerle paylaşır.

Kodlayıcı: Akış diyagramını veren arkadaşın bu diyagramın sözde kod halini öğretmenine sunmak istiyor. Arkadaşına yardım etmek için sözde kodları oluşturur musun?



Cevap:

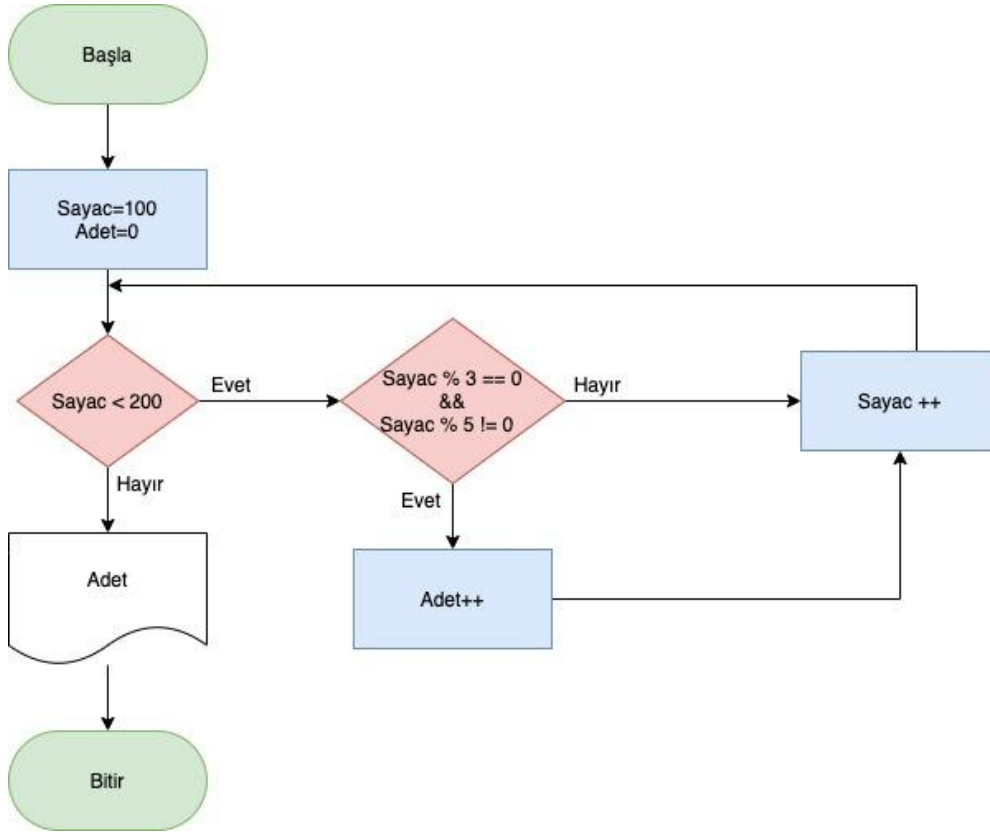
- Başla
- Sayac=0, Toplam=0
- Sayac < 4 Olduğu Sürece
 - a) Toplam = Toplam + Sayac * 2
 - b) Sayac ++
- Yaz, Toplam
- Bitir.

Analizci: Kodlayıcı rozetine ait görevde yer alan akış diyagramını kâğıt üzerinde adım adım çalıştıran tabloyu hazırlayıp, her bir değişkenin ekran çıktısını tahmin edebilir misin?

Tablo 14. Sözde kod çalıştırma tablosu

Değişken	Değer	Ekran Çıktısı
Sayac=0, Toplam=0	Sayac=0, Toplam=0	
Eğer Sayac < 4	Doğru	
Toplam = Toplam + Sayac * 2	Toplam = 0 + 0*2 = 0	
Sayac++	Sayac= 1	
Eğer Sayac < 4	Doğru	
Toplam = Toplam + Sayac * 2	Toplam = 0 + 1*2 =2	
Sayac++	Sayac= 2	
Eğer Sayac < 4	Doğru	
Toplam = Toplam + Sayac * 2	Toplam = 2 + 2*2 =6	
Sayac++	Sayac= 3	
Eğer Sayac < 4	Doğru	
Toplam = Toplam + Sayac * 2	Toplam = 6 + 3*2 =12	
Sayac++	Sayac= 4	
Eğer Sayac < 4	Yanlış	
Yaz, Toplam		12
Bitir		

Tasarlayıcı: İki arkadaş 100 ile 200 arasında 3'e bölünen ancak 5 ile bölünmeyen kaç sayı olduğunu merak edip bir yarışa girmiştir. Biri diğerini geçmek için daha hızlı bir yöntem olan bir algoritma oluşturacaktır. Buna göre hazırlayacağı algoritmanın sözde kod ve akış diyagramı nasıl olmalıdır? Onun yerine bunları sen oluşturabilir misin?



Hafta 3. Ders Materyalleri

EK 1. Kod Çalıştırma Tablosu

Algoritmanın sözde kodlarını çalıştırmak için aşağıdaki tabloyu kullanabilirsiniz. Algoritmanın adım sayısına göre satırları artırabilirsiniz.

Adım	Değişken değerleri	Ekran Çıktısı

EK 2. Hafta 3 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Diğer Ekler: Ders Dışı Aktivite Materyalleri

*Aşağıdaki eklere erişmek için materyaller klasörüne gidiniz.

EK 3. Ders Dışı Aktivite Sunumu

EK 4. Derleyici Kurulum Kılavuzu

Hafta 4. C++ Dilinde Değişken ve Veri Tipleri

Kazanımlar

- K1. C++ programlama dilinde değişken tanımlamayı bilir.
- K2. C++ programlama dilinde sabitleri tanımlamayı bilir.
- K3. C++ programlama dilinde kullanılan veri tiplerini bilir.
- K4. C++ temel giriş/çıkış akışlarını bilir.
- K5. C++ işleçlerinin kullanımını bilir.

Haftanın Amacı ve Materyaller

Bu haftanın amacı, C++ değişken tanımlama adımlarının veri tipleri ile birlikte öğrenilerek sabit tanımlamanın nasıl gerçekleştirildiğini öğrenmektir. Ayrıca C++ temel giriş/çıkış akışlarının nasıl kullanıldığını, C++ işleçlerin örnekler üzerinde nasıl çalıştığını görmelerini sağlamaktadır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız](#).

Önerilen Ders Akışı (50 + 50 + 50 + 50 dk.)

GİRİŞ (10 dk.)

GELİŞME

Değişkenlere İsim Verelim (40 dk.)

EK 1. Kural Kimde Saklı

EK 2. Değişken İsimleri: Destekleyici Bilgiler

Ders Arası (10 dk.)

Sabitleri Ayıralım (10 dk.)

Örnekleri Çoğaltalım (30 dk.)

Çıktıları Karşılaştıralım (10 dk.)

Ders Arası (10 dk.)

Operatörlerle Yüzleşelim (40 dk.)

EK 3. Operatörler: Destekleyici Bilgiler

EK 4. Görev Kodları

SONUÇ (10 dk.)

EK 5. Hafta 4 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 4 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 4 sunumunu ekran paylaşımına açar. Eğitmen üçüncü hafta işledikleri konu hakkında özet geçer ve öğrencilerin kazandıkları rozetleri sorar. Asenkron olarak takip ettikleri derleyici kurulumunun öğrenciler tarafından tamamlanıp tamamlanmadığı sözlü olarak kontrol edilir. Rozet görevleri ve derleyici kurulumu üzerinden *ön bilgileri hatırlatır* ve dördüncü haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

Eğitmene Öneriler: Kurulumu tamamlamamış, kurulum sırasında sorunla karşılaşmış öğrencilere eğitmen söz hakkı verebilir. Benzer hatalar ile karşılaşmış ve sorunu çözmüş öğrencilere de söz hakkı tanınır. Aksi durumda eğitmen hatanın nasıl giderileceği konusunda geri bildirimde bulunur. Bu şekilde diğer öğrencilerin de benzer sorunlara karşı çözüm üretmeleri sağlanmış olur. Zamana bağlı olarak eğitmen öğrencilere sağlayacağı bu desteği ders dışında sorunla karşılaşan öğrencilere yardımcı olarak da tamamlayabilir.

B. Gelişme

Değişkenlere İsim Verelim

Süre: 40 dk.

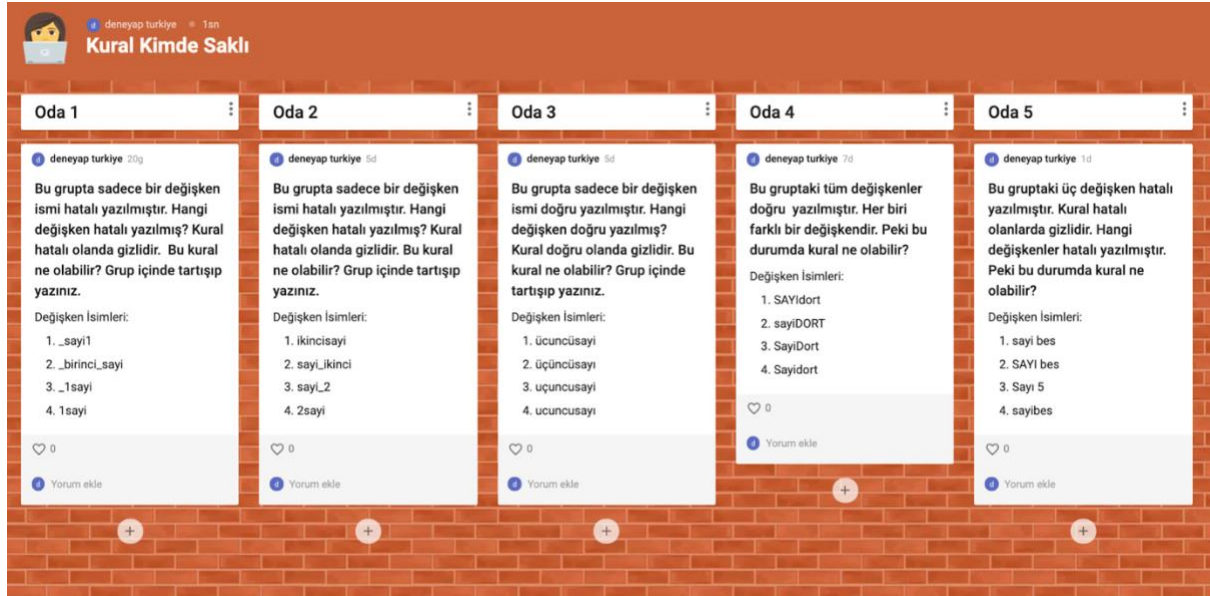
Kazanımlar: K1. C++ programlama dilinde değişken tanımlamayı bilir.

Materyaller: Dijital tartışma panosu: Grup Çalışması

EK 1. Kural Kimde Saklı

EK 2. Destekleyici Bilgiler

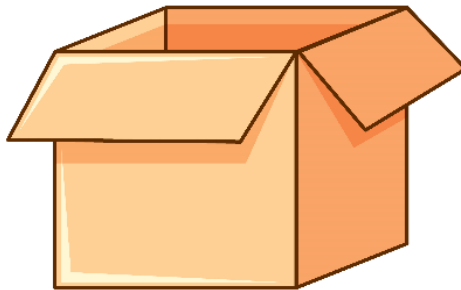
Hazırlık: Eğitimci derse girmeden önce grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Link ders sırasında öğrencilere paylaşılan notlar kısmından iletilecektir. Eğitimci padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. Örnek bir tasarım Resim 23'te gösterilmiştir. EK 1 materyali dijital tartışma panosuna ayrı grup görevleri halinde yüklenecektir. EK 2 ise destekleyici materyal olarak panoda bağlantı şeklinde verilmelidir.



Resim 23. Kural Kimde Saklı Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci Hafta 4 sunusunda “Değişkenlere İsim Verelim” başlıklı slaytına devam eder. Eğitimci slayt üzerinden öğrencilere aşağıdaki hatırlatmayı yapar.

“C++ programlarında veri değerlerinin bilgisayarın belleğinde saklanabildiği saklayıcı yapılara (aşağıdaki kutu gibi düşünebiliriz) değişken adı verilir. Saklanan değere değişkenin adı kullanılarak ulaşılabilir.”



Bir değişkenin bir programda kullanılabilmesi için önce tanımlanması gerekir. Bir değişken tanımlandığında, tür belirtilir ve uygun miktarda bellek ayrılır. Bu bellek alanı, değişkenin adı referans alınarak ele alınmaktadır. Aynı veri türünden birden fazla değişken adı oluşturulmak istenirse bunlar virgülle ayrılacak şekilde aşağıdaki gibi yazılabilir:

veri-tipi **degisken_adi**;
veri-tipi **deg_adi1, deg_adi2, deg_adi3**;

Daha sonra, eğitmen iki farklı grup etkinliği yapacaklarını ve ilk etkinlikte değişkenlere isim verme kuralları üzerinde duracaklarını belirtir. Öğrenciler dörderli gruplar halinde 10 dk. süre ile çalışma odalarına dağıtılır. Gruplara dağılmadan önce paylaşılan notlar kısmından etkinliğin padlet linki paylaşılır. Etkinlikte her grup bir tane değişken belirleme kuralına erişmelidir. Bu kurala erişmek için öğrenciler grup görevlerinde verilen değişken isimlerinin ortak noktalarını keşfetmeye çalışır. Gruplar beş dakika grup içinde tartışma yapar ve öngördükleri kuralı grup sütununda yazar. Bu şekilde öğrenci performanslarının sergilenmesi beklenir. Öğrenciler ana odaya geri döndüklerinde, eğitmen dijital panoyu ekran paylaşımı yaparak öğrenci ürünleri üzerinden kuralları özetler. Aynı zamanda eğitmen bu şekilde geri bildirimde bulunmaktadır.

Derste erişilmesi gereken temel kurallar aşağıdaki gibi olmalıdır.

1. İsim alt çizgi ile başlayabilir ancak sayı ile başlayamaz. Diğer her karakter bir harf, alt çizgi veya sayı olabilir.
2. İsim bir harf ile başlayabilir ancak sayı ile başlayamaz. Diğer her karakter bir harf, alt çizgi veya sayı olabilir.
3. Değişken adları C++'da büyük/küçük harfe duyarlıdır; bu nedenle "numara", "Numara" ve "NUMARA" gibi değişkenler üç ayrı değişken olarak ele alınır.
4. İsim içerisinde Türkçe karakter kullanılmaz.
5. İsim yazarken boşluk bırakılmaz.

Eğitmen tüm kurallar tamamlanınca değişken ismi olmayan yasaklı kelimelerin olduğundan bahseder. Bunun için "Değişken İsmi Olamazlar" slaytı ile devam edilir. Eğitmen ayrıca öğrencilere aşağıdaki ipucunu vererek önerilerde bulunur.

İPUCU:

C++ programlamada, değişkenlerin adları için küçük harfler kullanmak standart bir uygulamadır. Ayrıca programlarınızın okunabilirliğini artırmak için, ogrenci_yas, sinav_notu vb. gibi daha uzun ve daha açıklayıcı adlar seçebilirsiniz.

Değişken belirleme isimlerinin keşfedilmesinden sonra, eğitmen aynı dijital tartışma panosu üzerinden ikinci bir grup çalışması yapacaklarını belirtir. Öğrenciler tekrar rastgele gruplara dağıtılır. Şimdiki görevleri ise, atandıkları odaya ait belirlenen kurala uygun en az üç adet değişken ismi tanımlamaktır. Öğrenciler bu şekilde bir önceki gruplarından ayrılarak yeni gruplar içinde belirlenen kuralları uygulamaktadır. Bu şekilde öğrenciler performanslarını yansıtmaktadır. Bu görev için 10 dk. süre tanınır. Süre sonunda eğitmen padlet ortamını ekran paylaşımına açarak, öğrencilerin gönderileri üzerinden kuralların kontrolünü sağlar. Varsa öğrenci hataları düzeltilir ve konu pekiştirilerek yeni durumlara transferi sağlanmış olur.

Eğitmene Öneriler: Eğitmen etkinlikte bilmeseler de bir tahminde bulunmaları için öğrencileri teşvik eder. Ders sonunda eğitmen padlet etkinliğinin pdf çıktısını alarak ÖYS üzerinde ilgili hafta altına ekler ve bu şekilde öğrenci ürünlerini ders sonrası erişime açar. Eğitmen öğrenci gönderilerini çalışma gruplarının görev süresi içerisinde incelemelidir. Bu şekilde öğrenciler geri döndüğünde gönderileri zaten incelemiş olarak hazırda bulunur ve zamandan kazanılmış olur.

Sabitleri Ayırılım

Süre: 10 dk

Kazanımlar: K2. C++ programlama dilinde sabitleri tanımlamayı bilir.

Materyaller: Hafta 4 Sunum

Hazırlık: Eğitimci sunumun “Sabitleri Ayırılım” slaytında bekler.

Uygulama: Eğitimci sunum üzerinden sabitleri ayırılım etkinliğine devam eder. Burada değişken ismi tanımlamalarının ardından sabitler için aşağıdaki kısa bilgi ile konuya giriş yapar.

“Bir programın yürütülmesi sırasında içeriği hiç değişmeyecek olan veriler, bir değişken yerine sabit bir tanımlama ile saklanmalıdır.”

Eğitmen öğrencilerin dikkatini slayttaki örnek koda çevirmelerini ister ve aşağıdaki talimatı verir:

Bu kodda yer alan birtakım sabitler var. Bunları bulup, sohbetten tahminlerinizi yazınız. Bunu yaparken üç temel kurala dikkat ediniz:

“const” anahtar sözcüğünü kullanarak tanımlarız.

Sabit adları, değişken adlarından ayırt etmek için büyük harfle yazılır.

Sabitler her zaman tanımlama sırasında değerlerini almalıdır.

Eğitmen öğrencilerin tahminleri için 3 dk.’lık bir sözsüz müzik açar. Müzik sonunda etkinlik sona erer. Etkinlik sonunda eğitimci sohbetteki tahminlerden yararlanır ve öğrencilere ismiyle hitap ederek geri bildirimde bulunur. “Ayşe ve diğer tüm arkadaşların da belirttiği gibi....” şeklinde öğrencilere referans vererek özetleme yapan eğitimci, konuyu aşağıdaki bilgilerden yararlanarak açıklar:

Bir programın yürütülmesi sırasında içeriği hiç değişmeyecek olan veriler, bir değişken yerine sabit bir tanımlama ile saklanmalıdır. Bu, derleyicinin kodu hatalar açısından kontrol etmesini sağlar. Yazdığımız program, sabitte depolanan değeri değiştirmeye çalışırsa, derleyici bir hata bildirir ve derleme başarısız olur.

Aşağıda verilen bilgileri sabit olarak değerlendirebilirsiniz.

Pi sayısı

Evinizdeki oda sayısı

Eviniz ile okulunuz arasındaki uzaklık

Kimlik numaranız

Öğrenci numaranız

Doğum tarihiniz

`const veriTipi SABITADI = deger;`

`const double PI = 3.14159265;`

Sabitleri kullanmak çok fazla avantaj sağlar. Örneğin matematikteki Pi sayısını her kullanmamız gerektiğinde, program boyunca 3.14 gibi bir sayı yazdığımızı varsayalım. Ardından uygulamanızın daha yüksek bir hassasiyet gerektirdiğini fark edersek; her 3.14 sayısını, 3.14159265 gibi daha yüksek hassasiyet değeri ile değiştirmemiz gerekir. Bu değişikliği uygun bir şekilde hatasız olarak gerçekleştirmek de zaman alacaktır. Bunun yerine, Pi sayısını 3.14 değeri ile sabit olarak tanımlamış olsaydık, daha sonra hassasiyeti artırmamız gerektiğinde sadece en baştaki tanımlamadaki değeri değiştirmemiz yeterli olacaktır.

Örnekleri Çoğaltalım

Süre: 30 dk.

Kazanımlar: K3. C++ programlama dilinde kullanılan veri tiplerini bilir.

Materyaller: Hafta 4 Sunum

Hazırlık: Eğitimci Hafta 4 sunusunda “Örnekleri Çoğaltalım” başlıklı slayta devam eder.

Uygulama: Eğitimci slayt üzerinden öğrencilere veri tipleri hakkında açıklamalarda bulunur ve bir örnek verir. Bu şekilde uyarıcı materyal ile yeni bilgi sunulur. Eğitimci bir veri tipine ait kendi örneğini açıkladıktan sonra öğrencilere 1 dk. süre tanıyarak öğrencilerin örneği artırmalarını ister. Bunun için sohbet ortamından “Şimdi sen de kendi örneğini sohbetten bana yaz” ifadesiyle öğrenciler yönlendirilir. Bu şekilde eğitimci öğrencilerden performanslarını göstermelerini bekler ve öğrenci yanıtları üzerinden geri bildirimde bulunarak konuyu özetler. Bu etkinlikteki bir diğer görev ise, değişken tanımlama kodlarını örnek olarak gösterirken kullanılan çıktıları tahmin etmedir. Eğitimci bunun için örneğin bool değişken tipinde örnekler üzerine konuştuktan sonra, bool değişken tanımlama için örnek bir kodun olduğu slayta ilerler. Uyarıcı yeni bilginin sunulduğu bu aşamada öğrencilere anket aracılığıyla kodun çıktısını tahmin etmeleri istenir. Anket soruları öğrencilere sunu üzerinden iletilir. Eğitimci anket sorularını 30 sn. süre ile açık tutar. Maksimum 1 dk. süre içinde kapatır. Eğitimci gelen anket yanıtları ile öğrenci performanslarını topladıktan sonra, değişken tanımlama veri tipindeki anahtar noktalara değinerek konuyu özetler.

Eğitmene Öneriler: Eğitimci anket ile yanıt beklerken tüm öğrencilerin yanıt vermesini teşvik etmelidir. Sohbet özelliği kullanılırken 1 dk.’lık kısa ve sözsüz müzikler açılabilir. Bu süre sonunda sohbet paylaşımları sona erdirilir ve gelen yanıtlar üzerinden eğitimci konuyu açıklar.

Çıktıları Karşılaştıralım

Süre: 10 dk.

Kazanımlar: K4. C++ temel giriş/çıkış akışlarını bilir.

Materyaller: Hafta 4 Sunum

Hazırlık: Eğitimci Hafta 4 sunusunda “Çıktıları Karşılaştıralım: Cin-Cout Arasındaki Fark” başlıklı slayta devam eder.

Uygulama: Bu slaytta eğitimci kısaca cin ve cout arasındaki çıktı farkları ve temel giriş çıkış işlemlerini aktarmalıdır. Slaytta iki farklı kod gösterilir ve öğrencilere sohbet üzerinden bu iki kod

çiktısının ne olacağını tahmin etmeleri istenir. 1 dk. sohbet üzerinden öğrenciler performanslarını gösterir. Bu süre içerisinde eğitmen sözsüz kısa bir müzik başlatır. Sonunda eğitmen sohbet üzerinden gelen yanıtlardan da yararlanarak öğrencilerin çıktılar arasındaki farkı keşfetmelerini sağlar. Bunun için beyin fırtınası yöntemi kullanılır. Eğitmen cin ve cout komutlarının görevlerini özetleyerek açıklar.

Eğitime Öneriler: Eğitmen konuyu özetlerken aşağıdaki bilgilerden yararlanır.

C++ programlama, giriş ve çıkış işlemlerini gerçekleştirmek için birçok farklı kütüphane sunmaktadır. C++ 'da giriş ve çıkış, bayt serisi şeklinde veya daha yaygın olarak akış olarak bilinir. Giriş akışında, bayt akış yönü aygıttan (klavye, disk sürücü, ağ bağlantısı, vb.) ana belleğe doğrudur. Çıkış akışında ise bayt akış yönü tersine ana bellekten aygıtı (ekran, yazıcı, disk sürücü, vb.) doğrudur. Girdi/çıkış işlemleri için C++ 'da bulunan başlık dosyaları şunlardır:

iostream: Standart giriş/çıkış akışını temsil eder. Bu başlık dosyası cin, cout, cerr vb. gibi nesnelerin tanımlarını içerir.

io manip: Giriş/çıkış manipulatorleri anlamına gelir. Bu dosyalarda bildirilen yöntemler akışları işlemek için kullanılır. Bu dosya setw, setprecision vb. tanımları içerir.

fstream: Bu başlık dosyası temel olarak dosya akışını açıklar. Bu başlık dosyası, bir dosyadan girdi olarak okunan verileri veya dosyaya çıktı olarak yazılan verileri işlemek için kullanılır.

C++'da bulunan iki anahtar kelime cin ve cout çıktıları yazdırmak ve girdi almak için kullanılmaktadır. Bu iki ifade girdi ve çıktı almak için en temel yöntemlerdir. C++'da cin ve cout kullanmak için programda iostream başlık dosyasını eklemek gerekmektedir.

Operatörlerle Yüzleşelim

Süre: 30 dk.

Kazanımlar: K8. Bit işlemleri ile veriler üzerinde hesaplamalar yapar.

Materyaller: Dijital tartışma panosu: Bireysel Çalışma

EK 3. Destekleyici Bilgiler

EK 4. Görev Kodları

Hazırlık: Öğrencilerin bilgisayarında Code::Blocks yüklü ve aktif çalışır durumda olmalıdır. Eğitmen derse girmeden önce bireysel çalışma için raf temalı bir dijital panoyu padlet ortamında oluşturur. Link ders sırasında öğrencilere paylaşılan notlar kısmından iletilecektir. Eğitmen padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. Örnek bir tasarım Resim 24'te gösterilmiştir. EK 3, dijital tartışma panosunda bağlantı olarak verilirken, EK 4 materyali dört grup halinde ayrı ayrı dijital tartışma panosuna yüklenecektir.

Operatörlerle Yüzleşelim

Oda 1
deneyap turkiye 2d
Görev Kodu 1
Kod çıktısına ilişkin tahminlerinizi sütun altındaki artı işaretine tıklayarak yazınız.

Oda 2
deneyap turkiye 2d
Görev Kodu 2
Kod çıktısına ilişkin tahminlerinizi sütun altındaki artı işaretine tıklayarak yazınız.

Oda 3
deneyap turkiye 3d
Görev Kodu 3
Kod çıktısına ilişkin tahminlerinizi sütun altındaki artı işaretine tıklayarak yazınız.

Oda 4
deneyap turkiye 1d
Görev Kodu 4
Kod çıktısına ilişkin tahminlerinizi sütun altındaki artı işaretine tıklayarak yazınız.

Destekleyici Bilgiler

Aritmetik Operatörler

Operatör	Açıklama	Kullanım
+	Eklemeye	a + b
-	Çıkartmaya	a - b
*	Çarpmaya	a * b
/	Bölme	a / b
%	Bölme kalanı	a % b
++	Birbirine eklenme	a++
--	Birbirinden çıkartma	a--

Karşılaştırma Operatörleri

İşare	Açıklama	Kullanım
==	Eşitlik	a == b
!=	Eşitlik değil	a != b
>	Büyük	a > b
<	Küçük	a < b
>=	Büyük eşitlik	a >= b
<=	Küçük eşitlik	a <= b

Resim 24. Operatörlerle Yüzleşelim Örnek Dijital Tartışma Panosu

Uygulama: Öğitmen bu etkinliği öğrencilerin bireysel olarak çalışacağı dijital tartışma panosu üzerinde tamamlar. Bu nedenle öğretmen ekran paylaşımı olarak oluşturduğu dijital panoyu gösterir. Panoda operatörler ile ilgili dört farklı kod görevi vardır. Her öğrenci kendi bilgisayarında bu kodları derleyici üzerinde yazarak çalışacaktır. Bu görevde öğrencilerden ilk olarak kodları 5 dk. süre içinde incelemeleri ve istedikleri görevlerin altına kodun çıktısına ilişkin tahminlerini yazmaları istenir. Bu süre içerisinde öğrencilere sözsüz bir müzik açılır. Müzik sonunda öğretmen ikinci bir talimat verir. Bu kodlara ilişkin tahminlerinin doğru olup olmadığını anlamak için öğrencilerden kodları kendi bilgisayarlarında yazıp kontrol etmeleri istenir. Her öğrenci dört kodu bilgisayarında 10 dk. süre içinde yazıp kontrol etmeye çalışır. Öğrenciler kodu yazıp kontrol ettikleri ekranın görüntüsünü alır. Daha sonra dijital panoda ilgili görevin altında tahminini yazdığı iletiyi düzenle diyerek ekran görüntüsünü panoda paylaşır. Böylece öğrencilerin kod görevleri altında bir tahmini ve tahminini teyit ettiği kod satırları bulunacaktır. Görevler tamamlanırken öğrencilerin takıldıkları yerlerde destekleyici bilgilerden yararlanmaları istenir. Öğitmen çalışma sırasında 10 dk.'lık bir odaklanma müziği açar. Süre sonunda müzik de sona ermelidir.

Öğrenciler bu etkinlikte belirtilen süreler içinde istediği kadar göreve yanıt verebilmeli ve istediği kod görevinden başlayabilmelidir. Ancak tüm görevleri incelemelerini teşvik etmek amacıyla öğretmen en çok kod görevine katkı gösteren arkadaşların ödüllendirileceğini belirtir. Etkinlik sonunda öğretmen dört göreve ilişkin kod çıktılarına öğrenci gönderilerine de değinerek paylaşır. En çok kod görevine doğru tahminde bulunup, tahminini ekran görüntüsü ile teyit eden öğrenciler ödüllendirilir.

Eğitime Öneriler: Öğrenciler dijital panoya gönderi iletirken isimlerini de yazmalıdır. Bu şekilde ödül sisteminde yer alacak öğrenciler belirlenebilir. Öğitmen ödülün ne olacağına öğrencilerle birlikte karar vermelidir ya da basitçe bu öğrencileri alkışlama yöntemi kullanılabilir. Öğrenciler destekleyici bilgileri açıp incelemeyen eğitime doğrudan soru yöneltebilir. Bu durumlarda

onların materyal üzerinde çalışmalarını teşvik eden “*Bu konudaki soruna destekleyici bilgiler cevap verecektir. Lütfen materyali sistemden açıp inceleyin*” şeklinde geri bildirimler verilmelidir.

C. Sonuç

Süre: 10 dk.

Materyal: EK 5 Hafta 4 Süreli Ödev

Hazırlık: EK 5 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.’sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek haftanın konusu hakkında bilgi verilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS’den yapacaktır. EK 5 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitimci görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar: Görevler, öğrenciye verilecek beceri rozetleri ile isimlendirilmiştir. Her bir rozetin altında ilgili görevin yanıtları verilmektedir. Eğitimci bu yanıtları uygulama süresinin sonunda GitHub ya da ÖYS üzerinden öğrencilerle paylaşır.

Kodlayıcı: Kodlayıcı rozetine ait iki görev bulunmaktadır. Aşağıdaki görevlerden birinin yapılması kodlayıcı rozetinin kazanılması için yeterlidir.

1. Kullanıcıdan alacağınız yarıçap bilgisi ve tanımlayacağınız sabit $\pi = 3.14$ değerini kullanarak dairenin alanı ve çevresini hesaplayıp ekrana yazdıran programı kodlayınız.

```
#include <iostream>
using namespace std;
int main()
{
    int yari_cap;
    const float PI_DEGERI = 3.14;
    float alan, cevre;
```

```

    cout << "Dairenin yarıcap uzunlugunu giriniz: ";
    cin >> yari_cap;
    alan = PI_DEGERI * yari_cap * yari_cap;
    cout << "Dairenin alani: " << alan;
    cevre = 2 * PI_DEGERI * yari_cap;
    cout << "\nDairenin cevresi: " << cevre;
    return(0);
}

```

2. 2 kişinin yaşları toplamını hesaplayıp, ekrana yazdıran programın kodunu yazınız.

```

#include <iostream>
using namespace std;
int main()
{
    int yas1, yas2, yas3;
    cout << "1. kisinin yasini giriniz:";
    cin >> yas1;
    cout << "2. kisinin yasini giriniz:";
    cin >> yas2;
    cout << "\nYaslarinizin toplami: " << yas1+yas2;
    return 0;
}

```

Denetleyici: Verilen değişken tanımlamalarından ve ilk değer atamalarından hangileri uygun olmayan değişken tanımlı ve değer atamasıdır?

int __xyz5;	int _xyz5;	bool xyz = -1
short xyz = 34452;	int xyz=5.2;	int xyz = '*';
int xyz = 34452;	char xyz = '\192';	float xyz = 12345.12345;

Tasarlayıcı: Verilen C++ kodunun çıktısı tahmin edin.

Kod

```
#include <iostream>
using namespace std;
int main() {
    const int x;
    x = 10;
    cout << x<<;
}
```

Çıktı

Verilen kodda sabit olarak tanımlanan x değişkenine tanımlama satırının dışında değer atanmaya çalışılmaktadır. Daha önce belirtildiği gibi sabitler tanımlanırken ilk değerlerini alır ve daha sonra içerikleri değiştirilemez.

Analizci: 2 kişinin yaşları toplamını hesaplayan, program kodu verilmiştir. Kod üzerinde eksik olan satırları tamamlayınız.

Eksik Kodlar

```
#include <iostream>
using namespace std;
int main() {
    int yas1, yas2;
    cout << "1. kişinin yasini giriniz:";
    cin >> yas2;
    return 0;
}
```

Tamamlanmış Kodlar

```
#include <iostream>
using namespace std;
int main() {
    int yas1, yas2;
    cout << "1. kişinin yasini giriniz:";
    cin >> yas1;
    cout << "2. kişinin yasini giriniz:";
    cin >> yas2;
    cout << "\nYaşlarınızın toplamı: " << yas1+yas2;
    return 0;
}
```

Hafta 4. Ders Materyalleri

EK 1. Kural Kimde Saklı

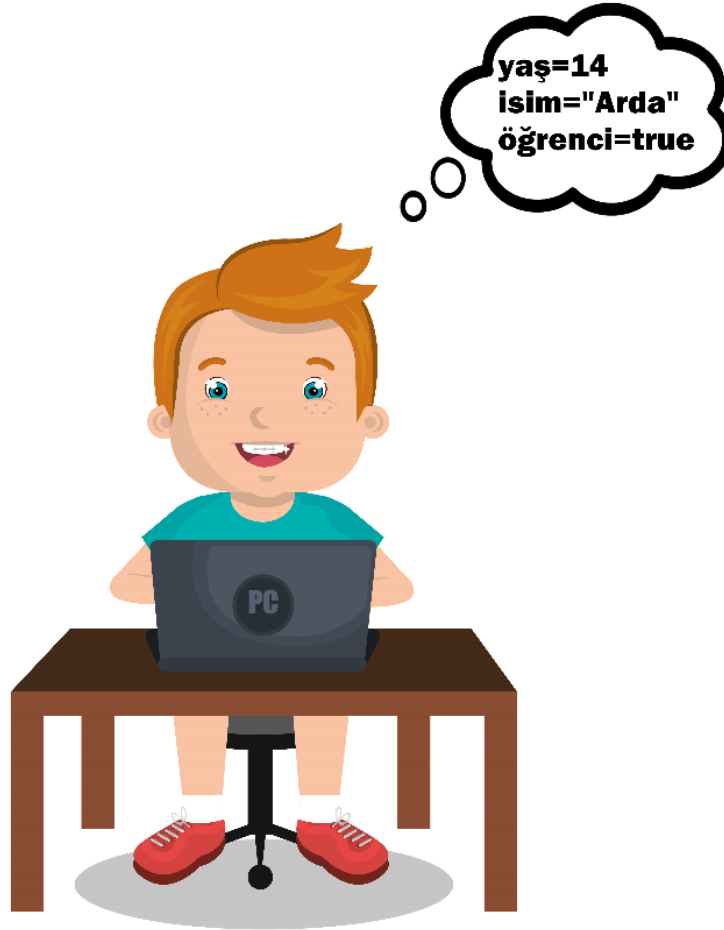
Aşağıdaki materyal dijital tartışma panosunda her bir oda için ayrı ayrı eklenmelidir. Odaların erişmesi gereken kurallar ayrıca materyalde bulunsa da bunlar panoda öğrencilerin uygulaması bittikten sonra eğitmen tarafından yazılmalıdır.

Oda 1: Görev				
Bu grupta sadece bir değişken ismi hatalı yazılmıştır. Hangi değişken hatalı yazılmış? Kural hatalı olanda gizlidir. Bu kural ne olabilir? Grup içinde tartışıp yazınız.	_sayi2	_ikinci_sayi	_2sayi	2sayi
<i>Kural 1: Değişken ismi alt çizgi ile başlayabilir. Ancak numara ile değil.</i>				
Oda 2: Görev				
Bu grupta sadece bir değişken ismi hatalı yazılmıştır. Hangi değişken hatalı yazılmış? Kural hatalı olanda gizlidir. Bu kural ne olabilir? Grup içinde tartışıp yazınız.	beşincisayi	sayi_beşinci	sayi_5	5sayi
<i>Kural 2: Değişken ismi bir harf ile başlayabilir. Ancak numara ile değil.</i>				

Oda 3: Görev				
Bu grupta sadece bir değişken ismi doğru yazılmıştır. Hangi değişken doğru yazılmış? Kural doğru olanda gizlidir. Bu kural ne olabilir? Grup içinde tartışıp yazınız.	ücuncüsayı	üçüncüsayı	uçuncusayı	ucuncusayı
<i>Kural 3: Değişken isminde Türkçe karakter kullanılmaz.</i>				
Oda 4: Görev				
Bu gruptaki tüm değişkenler doğru yazılmıştır. Her biri farklı bir değişkendir. Peki bu durumda kural ne olabilir?	sayiDORT	SAYIdort	SayiDort	Sayidort
<i>Kural 4: Değişken adları C++'da büyük/küçük harfe duyarlıdır; bu nedenle "numara", "Numara" ve "NUMARA" gibi değişkenler üç ayrı değişken olarak ele alınır.</i>				
Oda 5: Görev				
Bu gruptaki üç değişken hatalı yazılmıştır. Kural hatalı olanlarda gizlidir. Hangi değişkenler hatalı yazılmıştır. Peki bu durumda kural ne olabilir?	SAYI iki	sayi iki	Sayı 2	sayiki
<i>Kural 5: Değişken ismi yazılırken boşluk kullanılmaz.</i>				

EK 2. Değişken İsimleri: Destekleyici Bilgiler

C++ programlarında veri değerlerinin bilgisayarın belleğinde saklanabildiği saklayıcı yapılara değişken adı verilir. Saklanan değere değişkenin adı kullanılarak ulaşılabilir. Değişken dediğimiz kavram bir öğrencinin yaşı, ismi veya öğrencilik durumu olabilir.



Değişken kavramını daha iyi anlamak için şu örnekleri kullanabiliriz.

Bir bardak ve bir şişe süt olduğunu varsayalım. Bardağa biraz süt koyalım ve bardakta ne kadar süt olduğuna bakalım. Bardağa biraz daha süt koyalım ve tekrar bardakta ne kadar süt olduğuna bakalım. Sütün miktarı bardak dolana kadar az az değişerek herhangi bir değer olabilir. İşte bu nedenle süt miktarına “değişken” diyoruz. Yani değişebilen bir miktar söz konusudur.



Kilomuz ve yaşıımız bir deęiřkendir.



Kumbaradaki para miktarımız bir deęiřkendir.



Günlük yürüme miktarımız bir deęiřkendir.



Genel olarak farklı kişilerin adları ve soyadları değişkendir (tabi ki aynı olması da mümkündür).



Bir değişkenin bir programda kullanılabilmesi için önce tanımlanması gerekir. Bir değişken tanımladığınızda, tür belirtilir ve uygun miktarda bellek ayrılır. Bu bellek alanı, değişkenin adı referans alınarak ele alınmaktadır.

veri-tipi **degisken_adi**;

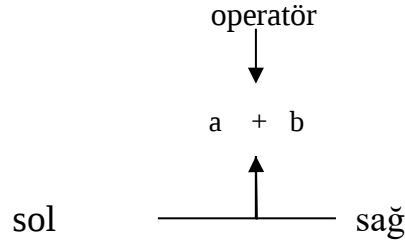
Sayılar, karakterler ve hatta tam kayıtlar gibi veriler, bir program tarafından işlenmelerini sağlamak için değişkenlerde saklanır. Değişkenler, özellikle bir sınıfa aitlerse, nesne olarak da adlandırılır. Aynı veri türünden birden fazla değişken adı oluşturulmak istenirse bunlar virgülle ayrılacak şekilde aşağıdaki gibi yazılabilir:

veri-tipi **deg_adi1, deg_adi2, deg_adi3**;

EK 3. Operatörler: Destekleyici Bilgiler

Aritmetik Operatörler

Matematiksel hesaplamalar bilgisayar programlarında yaygın olarak kullanılmaktadır. Programlama dilinde bir operatör, bir değer veya değişken üzerinde çalışan bir semboldür. Aşağıda görüldüğü üzere a değişkeni sol işlenen, b değişkeni sağ işlenen ve + sembolü de operatörü oluşturmaktadır.



Aşağıdaki tabloda, bilgisayar programlamasında kullanılan önemli aritmetik operatörler listelenmiştir.

Tablo 15. Önemli aritmetik operatörler

Operatör	Açıklama	Kullanım
+	İki işleneni birbirine ekler.	$x + y$
-	İkinci işleneni birinciden çıkarır.	$x - y$
*	İki işleneni çarpar.	$x * y$
/	İkinci işlenenle birinciye böler.	x / y
%	Tamsayı bölümünün kalanını verir.	$x \% y$
++	Sayıyı bir arttırma	$x++$ veya $++x$
--	Sayıyı bir azaltma	$x--$ veya $--x$

Tekli operatör olan ++ veya -- işleçleri değişken değerlerini arttırmak için kullanılır. “x++” ifadesinde x değişkeni var olan değeri ile işleme girer ve işlem tamamlandıktan sonra x değişkeninin değeri bir arttırılır. “++x” ifadesinde ise önce değişkenin değeri bir arttırılır ve daha sonra işlem yapılır. Aynı şekilde -- operatörü içinde benzer kullanım geçerlidir. Aşağıdaki örneği inceleyiniz.

```
a = 5
```

```
b = ++a    // a değeri 6 olur, b değeri 6 olur
```

```
c = 2
```

```
d = c++;    // c değeri 3 olur, d değeri 2 olur
```

```
a = 5
```

```
b = --a    // a değeri 4 olur, b değeri 4 olur
```

```
c = 2
```

```
d = c--;    // c değeri 1 olur, d değeri 2 olur
```

UYARI:

Bir ön-ek operatörünün değişken değerini hemen değiştirdiğini, bir son-ek operatörünün değişken değerini daha sonra değiştirdiğini unutmayalım.

Karşılaştırma Operatörleri

Bu işleçlerin amacı iki değişken veya değişken grubunu belirtilen şarta göre karşılaştırmaktır. Bu karşılaştırmalar aynı türdeki değişkenler arasında olmalıdır. Bu işleçleri özellikle ilerleyen haftalarda göreceğimiz koşul yapıları ve döngülerde kullanılır. Yapılan karşılaştırmalar doğruysa "1" yanlışsa "0" sonucunu döndürür.

Tablo 16. Karşılaştırma operatörleri

İşleç	Açıklama	Kullanım
<	Küçüktür	$x < y$
>	Büyüktür	$x > y$
<=	Küçük eşittir	$x \leq y$
>=	Büyük eşittir	$x \geq y$
==	Eşittir	$x == y$
!=	Eşit değildir	$x != y$

UYARI:

Programlamada en çok yapılan hatalardan biri iki ifadeyi karşılaştırmak için atama operatörünün (=) kullanılmasıdır. Böyle bir atama yaptığınızda soldaki değer bir değişkense derleyici bir hata mesajı oluşturmaz. Bu hata, programlamaya yeni başlayanların en çok dikkat etmesi gereken bir hatadır.

Atama Operatörleri

Atama işleçleri, bir değişkene değer atamak için kullanılır. Atama operatörünün sol taraftaki işleneni değişkendir ve atama operatörünün sağ taraftaki işleneni bir değerdir. Farklı tür atama operatörleri aşağıda tabloda verilmektedir.

Tablo 17. Atama operatörleri

İşleç	Açıklama	Kullanım
=	Sağdaki değeri soldaki değişkene atamak için kullanılır.	$x = y$ $z=10$
+=	Bu işleç, '+' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değere ekler ve ardından sonucu soldaki değişkene atar.	$x+=y$ $(x=x+y)$
-=	Bu işleç, '-' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değerden çıkarır ve ardından sonucu soldaki değişkene atar.	$x-=y$ $(x=x-y)$
=	Bu işleç '' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değerle çarpar ve ardından sonucu soldaki değişkene atar.	$x*=y$ $(x=x*y)$
/=	Bu işleç '/' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değere tam böler ve ardından sonucu soldaki değişkene atar.	$x/=y$ $(x=x/y)$
%=	Bu işleç '%' ve '=' operatörlerinin birleşimidir. Önce soldaki değişkenin geçerli değerini sağdaki değere göre bölümünden kalanı alır ve ardından sonucu soldaki değişkene atar.	$x\%=y$ $(x=x\%y)$

Mantıksal Operatörler

Mantıksal operatörler programlama dillerinde çok önemli bir yere sahiptir ve belirli koşullara göre karar vermemize yardımcı olurlar. İki koşulun sonucunu birleştirmek istediğimizde mantıksal VE ve VEYA istediğimiz sonucu üretmemize yardımcı olur. Bütün şartların sağlanması için koşullar arasına VE, herhangi birinin sağlatılması isteniyorsa koşullar arasına VEYA ve koşulu sağlamayanlar isteniyorsa DEĞİL işleci kullanılmalıdır.

Tablo 18. Mantıksal operatörler

İşleç	Açıklama	Kullanım
&&	Mantıksal VE	x && y
	Mantıksal VEYA	x y
!	Mantıksal DEĞİL	!x

Mantıksal işleçlerin sonucu şu şekilde belirlenmektedir. Eğer x&&y kullanılıyor ise her iki değişkenin değeri "1" olursa sonuç "1" olur, aksi hâlde sonuç "0" olur. Eğer x||y kullanılıyor ise her iki değişkenin değeri "0" olursa sonuç "0" olur, aksi hâlde sonuç "1" olur. Eğer !x kullanılıyor ise değişkenin değeri "1" ise sonuç "0", "0" ise sonuç "1" olacaktır. A ve B değişkenleri için oluşturulan aşağıdaki tabloyu inceleyebilirsiniz.

Tablo 19. Mantıksal operatörlerin kullanımı

A	B	A && B	A B	! A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

UYARI:

x veya x+1 gibi sayısal bir değer, değeri 0 ise "false", 0 dışında bir değer ise "true" olarak yorumlanır.

Operatör Önceliği

C++ programlarınızı oluştururken aritmetik operatörlerin önceliğine dikkat etmeniz gerekmektedir. Operatör önceliği değerlendirme sırasını, yani operatörlerin ve işlenenlerin nasıl gruplandığını belirler. Aşağıdaki tablodaki öncelik sırasını inceleyiniz.

Tablo 20. Operatör öncelikleri

Öncelik	Operatör	İşlem Yönü
1	()	Soldan sağa
2	~ ++ --	Sağdan sola
3	* / %	Soldan sağa
4	+ -	Soldan sağa
5	<< >>	Soldan sağa
6	< <= > >=	Soldan sağa
7	== !=	Soldan sağa
8	&	Soldan sağa
9	^	Soldan sağa
10		Soldan sağa
11	&&	Soldan sağa
12		Soldan sağa
13	= += -= *= /= %=	Sağdan sola

EK 4. Görev Kodları

```
#include <iostream>
using namespace std;
int main()
{
    int a = 14, b = 3;
    cout << "Ekleme: " << (a + b) << endl;
    cout << "Cikarma: " << (a - b) << endl;
    cout << "Carpma: " << (a * b) << endl;
    cout << "Bolme: " << (a / b) << endl;
    cout << "Mod alma: " << (a % b) << endl;
    cout << "Arttirma: " << a++ << endl;
    cout << "Arttirma: " << ++a << endl;
    cout << "Azaltma: " << b-- << endl;
    cout << "Azaltma: " << --b << endl;
    return 0;
}
```

Kodun Çıktısı (Etkinlik sonunda öğrencilerle paylaşılır.)

```
Ekleme: 17
Cikarma: 11
Carpma: 42
Bolme: 4
Mod alma: 2
Arttirma: 14
Arttirma: 16
Azaltma: 3
Azaltma: 1
```

```
#include <iostream>
using namespace std;
int main()
{
    int x = 8, y = 7;
    cout << (x < y) << endl;
    cout << (x > y) << endl;
    cout << ((x-1) <= y) << endl;
    cout << (x >= (y+2)) << endl;
    cout << (x == y) << endl;
    cout << (x != y) << endl;
    return 0;
}
```

Kodun Çıktısı (Etkinlik sonunda öğrencilerle paylaşılır.)

```
0
1
1
0
0
1
```

```
#include <iostream>
using namespace std;
int main()
{
    int x = 7, y = 1;
    cout << ((x <= y) || (y>1)) << endl;
    cout << ((x > 6) && (y==1)) << endl;
    cout << (x && !y) << endl;
    cout << (!(x-1) || (y+1 > 1)) << endl;
    return 0;
}
```

Kodun Çıktısı (Etkinlik sonunda öğrencilerle paylaşılır.)

```
0
1
0
1
```

```
#include <iostream>
using namespace std;
int main()
{
    float x, y, z;
    x = 4.3;
    y = x;
    z = x + 2.5 + (y * 3.0);

    cout << "x: " << x << endl;
    cout << "y: " << y << endl;
    cout << "z: " << z << endl << endl;

    x = y = 2.1;
    cout << "x: " << x << endl;
    cout << "y: " << y << endl << endl;

    x+= 3.8;
    y+= y * 3.0;
    cout << "x: " << x << endl;
    cout << "y: " << y << endl << endl;

    x/= 2.0;
    y*= 4.0;
    cout << "x: " << x << endl;
    cout << "y: " << y << endl << endl;

    return 0;
}
```

Kodun Çıktısı (Etkinlik sonunda öğrencilerle paylaşılır.)

```
x: 4.3  
y: 4.3  
z: 19.7
```

```
x: 2.1  
y: 2.1
```

```
x: 5.9  
y: 8.4
```

```
x: 2.95  
y: 33.6
```

EK 5. Hafta 4 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 5. Karar Mantık Yapıları

Kazanımlar

- K1. C++ programında karar yapılarını tanımlayarak kullanabilir.
- K2. C++ programında tekli karar yapısını tasarlayıp kodlayabilir.
- K3. C++ programında çoklu karar yapısını tasarlayıp kodlayabilir.
- K4. C++ programında iç içe karar yapısını oluşturarak kodlayabilir.

Haftanın Amacı ve Materyaller

Bu haftanın amacı öğrencilerin programlamada karar yapılarını kullanarak programın akışını kontrol edebilmelerini sağlamaktır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız](#).

Önerilen Ders Akışı (50 + 80 + 30 + 50 dk.)

GİRİŞ (10 dk.)

Biliyor muydunuz?

GELİŞME

Karar Yapılarını Tanıyalım (40 dk.)

EK 1. Karar Yapılarını Tanıyalım Görev Kâğıtları

Ders Arası (10 dk.)

Görevleri Kodlayalım (35 dk. + 10 dk.(ara)+ 35 dk.) (Toplam 80 dk.)

EK 1. Karar Yapılarını Tanıyalım Görev Kâğıtları

Ders Arası (10 dk.)

İç İçe Koşula Farklı Bir Bakış (Switch) (20 dk.)

EK 2. SWITCH Kullanımı

SONUÇ (10 dk.)

EK 3. Hafta 4 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Dikkat Çekme ve Hedeften Haberdar Etme: Biliyor muydunuz?

Süre: 10 dk.

Materyal: Hafta 5 Sunum

Hazırlık: Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 5 Sunum).

Uygulama: Eğitmen öğrencilere sunum üzerinden aşağıda verilen soruyu sözlü olarak sorarak düşüncelerini ister ve anketi başlatır.

Anket Sorusu: Sizce Apple’ın kurucusu Steve Jobs, Facebook’un kurucusu Mark Zuckerberg, Microsoft Kurucusu Bill Gates, Google’ın kurucusu Larry Page, Tesla’nın kurucusu Elon Musk şirketlerini kurarken sizce hangi yaş aralıklarındaydı?

Öğrencilerden ankete yanıtları alındıktan sonra, eğitmen sunumun “*Biliyor muydunuz?*” başlıklı slaydına gelerek doğru cevabı öğrencilere gösterir.

Bu etkinlikteki amaç öğrencilere günümüzde kurulan yazılım şirketlerinin kurucularının yaşının öğrencilere gösterilerek, onların da gerekli algoritmik düşünme ve yazılım bilgilerini kullanarak günümüz dünyasını değiştirebilecek bir girişim yapabileceklerini hissettirmektir.

Öğrencilerin dikkat çekme kısmı yukarıdaki gibi tamamlandıktan sonra, eğitmen geçen hafta işledikleri konu hakkında özet geçerek ön bilgileri hatırlatır ve bu haftanın amacını paylaşarak öğrencileri hedeften haberdar eder.

B. Gelişme

Karar Yapılarını Tanıyalım

Süre: 40 dk.

Kazanımlar: K1. C++ programında karar yapılarını tanımlar.

Materyaller: Hafta 5 Sunum

EK 1. Karar Yapılarını Tanıyalım Görev Kâğıtları

Hazırlık: Hafta 5 Sunumunda “Algoritma Çıktıları Ne Olur” adlı slaytlar sırası geldikçe açılır.

Uygulama: Eğitimci bu ders için hazırlanan beş görevi sunum üzerinden ikiye dakika arayla öğrencilerine gösterir. Her görev için eğitimci 1 ile 100 arasında rastgele bir sayıyı öğrencilerine söyler. Eğitimci söylediği sayı ve verilen algoritmalara göre nasıl bir çıktı olabileceğini öğrencilere sorar ve cevapları sohbet kısmından ister. Eğitimci bulunan çıktı sonuçlarının doğru veya yanlışlığı hakkında öğrencilere geri bildirim vererek dersteki her bir öğrencilerin doğru sonuca ulaşması hedeflenir.

Not: Her slayt değişiminde eğitimci karar yapılarına uygun sayılar belirleyerek ve her değişimde bu sayıları değiştirerek, öğrencilerin o sayıya yönelik programda hangi çıktıyı vereceği öğrenciler tarafından bulunması istenir. Ayrıca, öğrenciler etkinliği bitirdikten sonra her bir görev kâğıdının günlük yaşamda hangi problemi çözmek için yazıldığının öğrenciler tarafından tahmin edilmesi istenir ve bununla ilgili eğitimci eşliğinde tartışma gerçekleştirilir.

Eğitime Öneriler: Yukarıdaki etkinliğin amacı; öğrencinin program akışının bir noktasında verilen karara göre yapılacak işlemleri göstermek ve verilen karara bağlı olarak ekran çıktısının nasıl değişebileceğini öğrencilerin görmesini sağlamaktır.

Yukarıdaki etkinlik bittikten sonra eğitimcinin karar yapılarının C++ dilinde nasıl kodlandığına yönelik aşağıda gibi kısa bir bilgilendirme yaparak sunu üzerinden özetleme yapması beklenir.

Programlama dillerinin olmazsa olmazı karar ifadeleridir. Karar ifadeleri program akışına yön vermek için kullanılır. Belirli bir şartlara (sıcaklık değeri, klavyede bir tuşa basılıp basılmadığı, bir değer girilip girilmediği) göre yapılması gereken işlemleri karar ifadeleri ile gerçekleştiririz. C++ programlama dilinde bu amaç için kullanılan “if”, “if-else” ve “switch” ifadeleridir.

Eğer koşula göre yapılacak bir veya daha çok işlem varsa ilgili blok “{” “}” arasına yazılır. Tek satırlık işlemler için “{” “}” karakterlerine gerek yoktur.

if(koşul)

{

koşul doğru ise yapılacak işlemler

}

yerine aşağıdaki şekilde yazılabilir.

if(koşul)

koşul doğru ise yapılacak işlem

Bazı durumlarda koşul gerçekleşmediğinde de işlem yapmak isteriz. Koşulun gerçekleşmemesi durumunu işleme almak için “aksi takdirde” anlamına gelen “else” komutu yazılır.

if(koşul)

{

koşul doğru ise yapılacak işlemler

}

else

{

koşul yanlış ise yapılacak işlemler

}

Bazı durumlarda koşullarımız birden fazla olabilir. Bu durumlarda her bir koşul için “else if” kullanılmaktadır.

if(kosul1)

 kosul1 için yapılacak işlem

else if(kosul2)

 kosul2 için yapılacak işlem

else if(kosul3)

 kosul3 için yapılacak işlem

else

 diğer tüm durumlar için yapılacak işlem

Görevleri Kodlayalım

Süre: 50 dk.

Kazanımlar: K1: C++ programında tekli karar yapılarını bilir.

K2: C++ programında çoklu karar yapılarını bilir.

K3: C++ programında iç içe karar yapısını bilir.

Materyaller: Hafta 5 Sunum

EK 1. Karar Yapılarını Tanıyalım Görev Kâğıtları

Hazırlık: Eğitmenden bir önceki etkinlikteki öğrenciler tarafından çıktısı bulunmaya çalışılan algoritmaların nasıl koda çevrildiğine yönelik hazırlanan slaytları sırası geldikçe açması beklenir. Ayrıca kodlama yapılacağı için Code::Blocks programı açılır.

Uygulama: Bu bölüm 5 görevden oluşmaktadır. Her bir görev için eğitmenin o görev esnasında neler yapacağı aşağıda sırasıyla verilmektedir.

Görev 1: Klavyeden girilen bir sayının 3'e bölünüp bölünmeyeceğini ekrana yazdıran kodu yazınız.

Yukarıdaki görev için "Görev 1" başlıklı slayt açılır. Daha sonra eğitmen Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğitmen her satır kodun ne anlama geldiğini ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 5 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eğitmen öğrencilerin kodları kendi bilgisayarlarında yazdıklarından emin olduktan ve ihtiyacı olan öğrencilere gerekli geri bildirimleri verdikten sonra bir sonraki görev ve slayda geçer.

Görev 2: Klavyeden girilen sıcaklık değerine göre, suyun bulunacağı hali ekrana yazdıran programı nasıl yazardınız? Eksik kodları dolduralım...

Not: sıcaklık<0 ise katı

Sıcaklık 0 ile 100 arasında ise sıvı

Sıcaklık 0 ile 10000 arasında ise gaz

Sıcaklık 10000'den büyük ise plazma.

Yukarıdaki görev için "Görev 2" başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğunu öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.'lık süre verildiği

belirtilerek yukarıdaki belirtilen görevdeki eksik kodları tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapması istenir. Eğitimci bitirip el kaldıran öğrencilerden rastgele birini seçerek ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitimci kodun nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer.

Görev 3: Klavyeden girilen bir öğrencinin yıl sonu not ortalaması değerine göre, öğrencinin bir üst sınıfa geçip geçmeyeceğini, eğer geçiyorsa belge alıp almayacağını, eğer belge alıyorsa takdir belgesi mi teşekkür belgesi mi alacağını gösteren programı yazınız? Eksik kodları dolduralım...

Not: ortalama ≥ 50 ise öğrenci geçer

70 $\leq$ ortalama < 85 ise öğrenci teşekkür belgesi alır.

ortalama ≥ 85 ise öğrenci takdir belgesi alır.

Yukarıdaki görev için “Görev 3” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğunu öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodların tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapması istenir. Eğitimci bitirip el kaldıran öğrencilerden rastgele birini seçer ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitimci kodun nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer.

Görev 4: Bir sinema salonu yaşı 18’den küçük olanlar için 15 TL, 65 yaşından büyükler için 10 TL alırken, diğer tüm yaş gruplarından 20 TL almaktadır. Bu bilgiler doğrultusunda girilen yaşa göre kişinin ödeyeceği miktarı söyleyen programı nasıl kodladınız? Eksik kodları dolduralım.

Yukarıdaki görev için “Görev 4” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğunu öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodların tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapması istenir. Eğitimci bitirip el kaldıran öğrencilerden rastgele birini seçer ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitimci kodun nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer.

Görev 5: Bir işletmede çalışanların aylık maaşı çalışma saatlerine göre ödenmektedir. Eğer kişinin çalışma saati 100’ün altında ise 1 kat, 100-250 arası 2 kat, 250 saatten fazla çalışırsa 3 kat ücret

ödenmektedir. Saatlik ücret: 5 lira ise, girilen saat bilgisine göre maaşı hesaplayan programı yazınız.

Yukarıdaki görevi öğrencilerin kendi bilgisayarlarında 10 dk. içerisinde kodlaması beklenir. Bu esnada öğretmenler öğrencilerin sıkıştığı noktalarda ipuçları vererek onlara destek olabilir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapılması istenir. Öğitmen bitirip el kaldıran öğrencilerden rastgele birini seçer ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da öğretmen kodun nasıl yazıldığını kendisi de yazarak özetler, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister.

Eğitmene Öneriler: Yukarıdaki etkinlikteki amaç öğrencilerin kod yazarken, karar yapılarını nerede ve ne zaman kullanılacağına yönelik farkındalığını artırmaktır. Yukarıdaki görevlerin tamamını öğrencilerin kendi bilgisayarlarında kodlaması beklenmektedir. Bunun için öğretmenlerin etkinlikler esnasında sürekli farklı öğrencilerinden isteyeceği ekran paylaşımı onların her etkinliğe hazır girmesi için motivasyonunun artmasını sağlayacaktır.

İç İçe Koşula Farklı Bir Bakış (Switch)

Süre: 20 dk.

Kazanımlar: K4. C++ programında iç içe karar yapısını bilir.

Materyaller: Hafta 5 Sunum

EK2. SWITCH Kullanımı

Hazırlık: Öğitmen “İç İçe Koşula Farklı Bir Bakış” adlı slayda gelir. Ayrıca kodlama yapılacağı için Code::Blocks programı açılır.

Uygulama: Öğitmen derse aşağıdaki soruyu sorarak, öğrencilerden yanıtlarını chat kısmına yazmasını ister.

Soru: Koşul yapılarını kodlarken sadece if/ if-else komutları mı kullanılıyor acaba?

Cevapları aldıktan sonra öğretmen soruyu aşağıdaki gibi özetler.

“Birden çok koşul olduğu durumlarda if/if-else yapılarını kullanabileceğimiz gibi “switch” bloğunu da kullanabiliriz.”

Basit bir hesap makinesi yapımı için yazılabilecek iki farklı kod bloğu öğrencilere sunum üzerinden gösterilerek, if yapısı ile switch yapısı konusunda kod yazımında ne gibi farklılıkların olduğuna yönelik sınıf içi tartışma ortamı oluşturulur.

If/ If Else Kullanımı	Switch/Case Kullanımı
<pre> #include <iostream> using namespace std; int main() { char islem; cin >> islem; if(islem == '+') cout << "Toplama islemi"; else if(islem == '-') cout << "Cikarma islemi"; else if(islem == '*') cout << "Carpma islemi"; else if(islem == '/') cout << "Bolme islemi"; else cout << "Hatali giris."; } </pre>	<pre> #include <iostream> using namespace std; int main() { char islem; cin >> islem; switch(islem) { case '+': cout << "Toplama islemi"; break; case '-': cout << "Cikarma islemi"; break; case '*': cout << "Carpma islemi"; break; case '/': cout << "Bolme islemi"; break; default: cout << "Hatali giris."; } return 0; } </pre>

Daha sonra eğitmen tarafından aşağıdaki gibi konu öğrencilerinde katılımı ile özetlenir.

Bir değişkenin değerini birden çok “if” koşulu ile kontrol etmek yerine daha “switch” ve “case” bloklarını kullanabiliriz. Her koşul durumunu “case” komutları ile belirterek daha kolay okunabilir bir kod parçası oluşturabiliriz. Burada kullanılacak değerler sabit olacaktır ve kullanılacak olan

değişkenin kullanacağımız durumlarını önceden biliyor olmamız gerekiyor. Örneğin int tipindeki Not değişkeni için sabit 1, 2, 3, 4, 5 değerleri için switch-case söz dizimi şu şekildedir;

```
switch(Not)
{
    case 1:
        Yapılacak işlemler
        break;
    case 2:
        Yapılacak işlemler
        break;
    ...
    default:
        Yapılacak işlemler
        break;
}
```

break: Bu komut, koşul bloğundan çıkmamızı sağlar.

default: Verilen koşulların gerçekleşmemesi halinde çalışır.

Switch kullanımı yukarıdaki gibi özetlendikten sonra eğitimci eşliğinde öğrencilerin, basit hesap makinesi yapımını switch bloğunu kullanarak kendi bilgisayarlarında kodlaması istenir. Öğrenciler bu kodu yazarken eğitmeni takip eder.

C. Sonuç

Süre: 10 dk.

Materyal: EK 3 Hafta 5 Süreli Ödev

Hazırlık: EK 3 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta "Döngüler" konusunun işleneceği bildirilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. EK 3 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitmen görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar

Analizci:

```
#include <iostream>
using namespace std;
int main()
{
    int sayi = 13;
    if(sayi % 2 == 0)
        cout << sayi /2;
    else
        cout << (sayi-1) /2;
}
```

Yukarıdaki kodu bilgisayarına yazan bir programcı sizce ne programlamak istemiştir?

Cevap: Verilen sayının yarısını bulmaya çalışmaktadır. Sayı tek olsa dahi orta noktası bulunmaktadır.

Denetleyici:

```
#include <iostream>
using namespace std;
int main()
{
    int sayi = 13;
    if(sayi % 2 == 0)
        cout << sayi /2;
    else
        cout << (sayi-1) /2;
}
```

Yukarıdaki kodu bilgisayarına yazan bir programcı nasıl bir ekran çıktısı ile karşılaşacaktır. Çıktıyı tahmin ettikten sonra, lütfen kodları bilgisayarınıza yazarak tahmininizi test ediniz.

Cevap: 6

Kodlayıcı: Ali adlı bir öğrenci istediği programı yazmaya başlamadan önce yandaki akış şemasını oluşturmuştur. Fakat akış şemasını koda dönüştürme konusunda yardıma ihtiyacı vardır. Bunun için Ali'ye yardım ediniz.

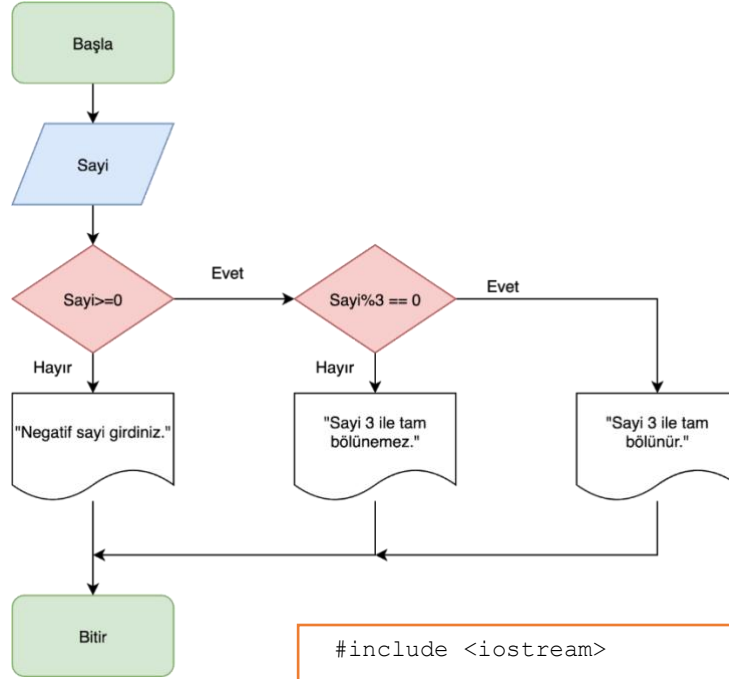
Cevap:

```
#include <iostream>
using namespace std;
int main()
{
    int sayi;
    cin >> sayi;
    if(sayi>10)
        sayi = sayi -10;
    else
        sayi = sayi +10;
    cout << sayi;
}
```

Hafta 5. Ders Materyalleri

EK1. Karar Yapılarını Tanıyalım Görev Kâğıtları (Toplam 5 görev)

1. Görev



1. Grup Ekran Çıktısı:

.....

2. Grup Ekran Çıktısı:

.....

3. Grup Ekran Çıktısı:

.....

4. Grup Ekran Çıktısı:

.....

5. Grup Ekran Çıktısı:

.....

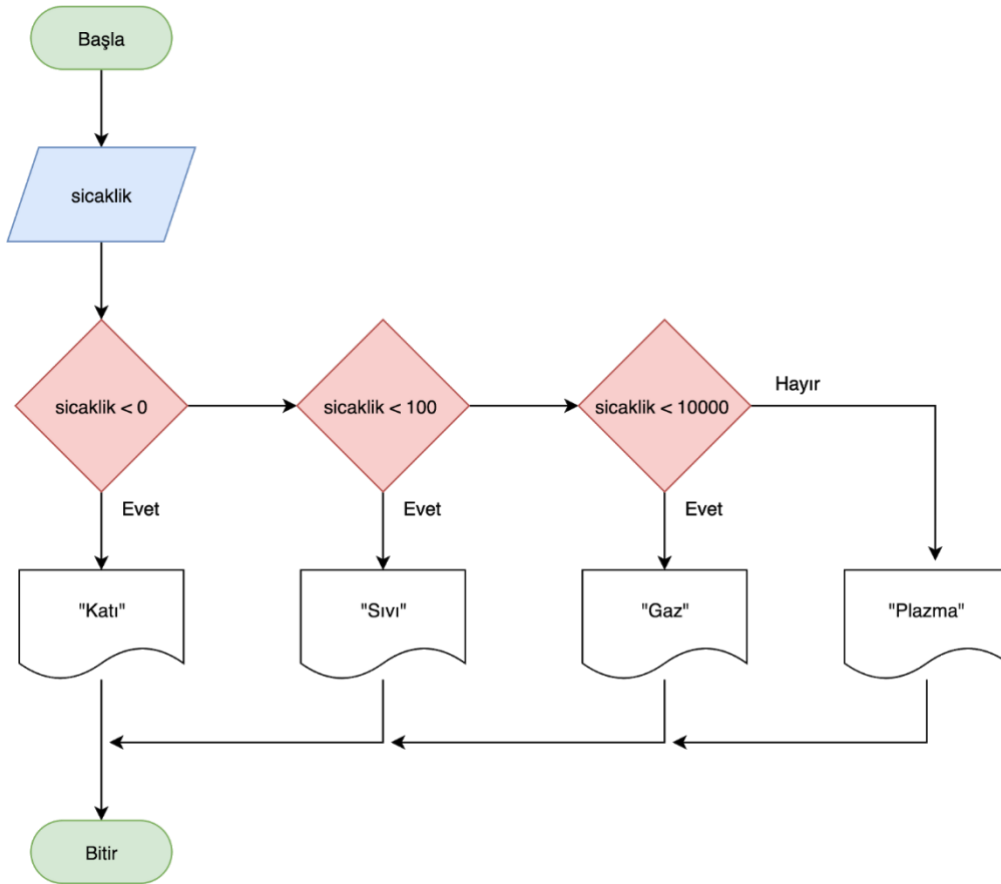
```

#include <iostream>
using namespace std;
int main()
{
    int sayi;
    cin >> sayi;

    if(sayi >= 0)
    {
        if(sayi%3 == 0)
            cout << "Sayı 3 ile tam bolunur.";
        else
            cout << "Sayı 3 ile tam bolunemez.";
    }
    else
    {
        cout << "Negatif sayı girdiniz.";
    }

    return 0;
}
  
```

2. Görev



1. Grup Ekran Çıktısı:

.....

2. Grup Ekran Çıktısı:

.....

3. Grup Ekran Çıktısı:

.....

4. Grup Ekran Çıktısı:

.....

5. Grup Ekran Çıktısı:

.....

```

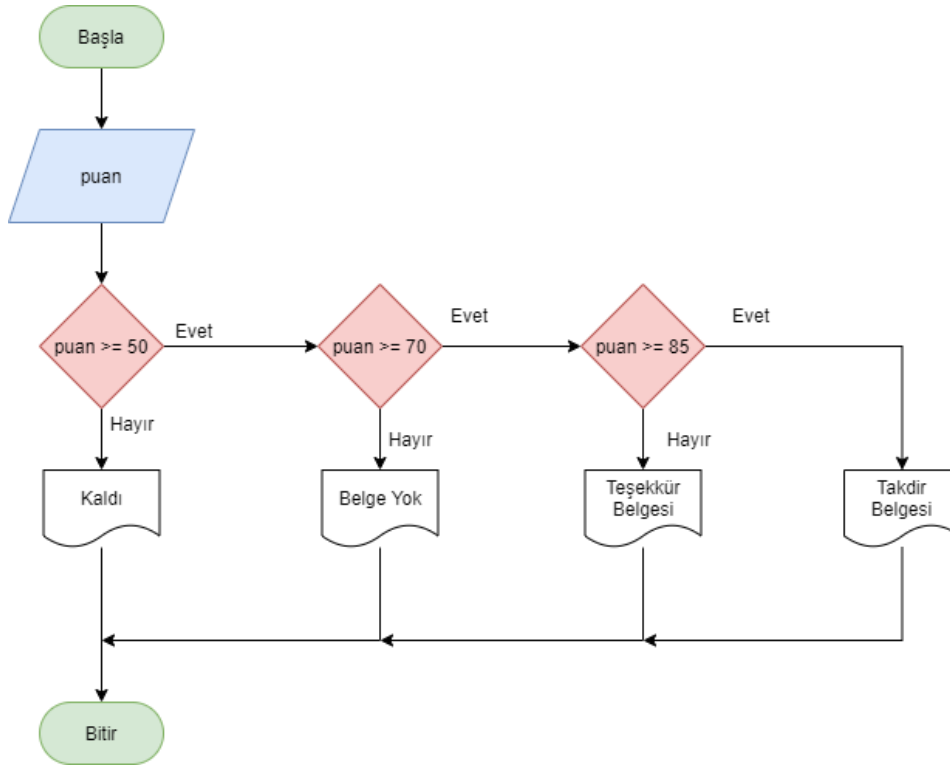
#include <iostream>
using namespace std;

int main()
{
    int sicaklik;
    cin >> sicaklik;

    if(sicaklik < 0)
        cout << "Kati";
    else if(sicaklik < 100)
        cout << "Sivi";
    else if(sicaklik < 10000)
        cout << "Gaz";
    else
        cout << "Plazma";

    return 0;
}
  
```

3. Görev



1. Grup Ekran Çıktısı:

.....

2. Grup Ekran Çıktısı:

.....

3. Grup Ekran Çıktısı:

.....

4. Grup Ekran Çıktısı:

.....

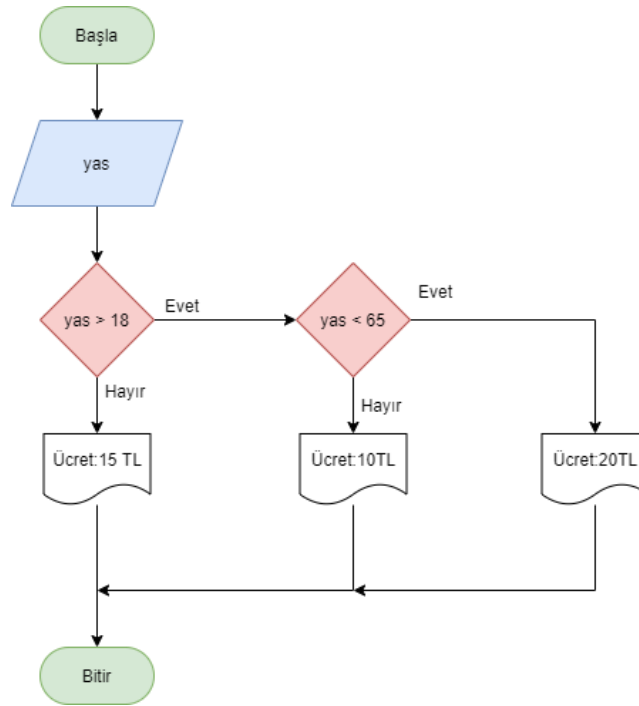
5. Grup Ekran Çıktısı:

.....

```

#include <iostream>
using namespace std;
int main()
{
    int puan;
    cout << "notu giriniz:";
    cin >> puan;
    if(puan<50)
        cout << "kaldi";
    else if(puan<70)
        cout << "belge yok";
    else if(puan<85)
        cout << "tesekkür belgesi";
    else if(puan<=100)
        cout << "takdir belgesi";
    else
        cout <<"Hatali giris";
    return 0;
}
  
```

4. Görev



1. Grup Ekran Çıktısı:

.....

2. Grup Ekran Çıktısı:

.....

3. Grup Ekran Çıktısı:

.....

4. Grup Ekran Çıktısı:

.....

5. Grup Ekran Çıktısı:

.....

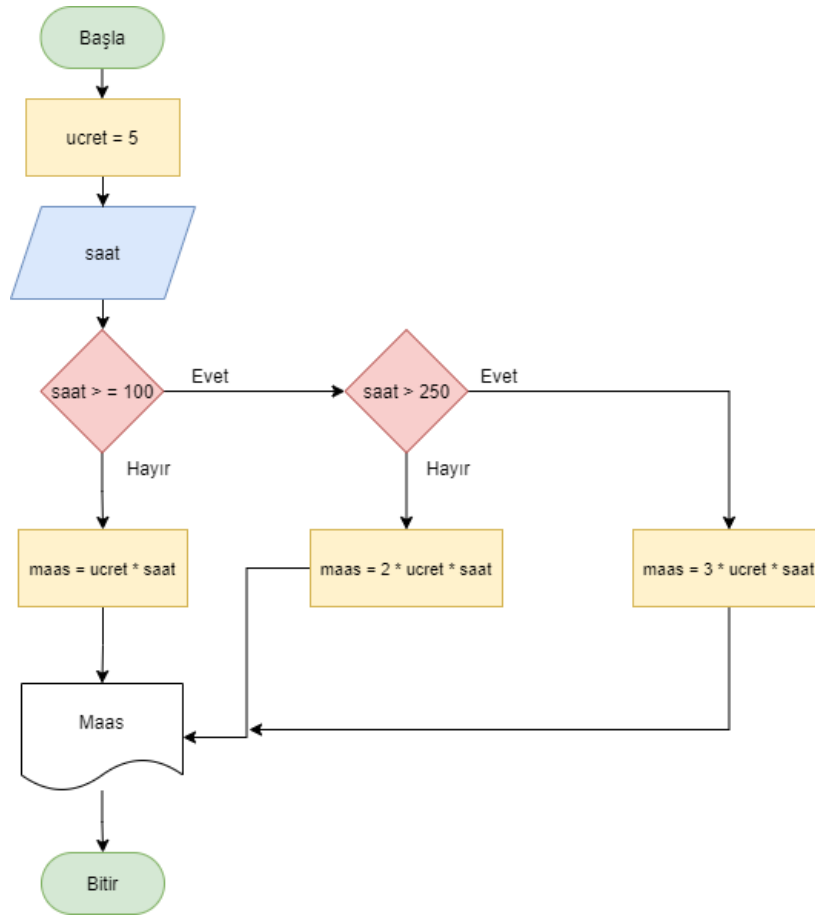
```

#include <iostream>
using namespace std;
int main()
{
    int yas;
    cout << "yasinizi giriniz:";
    cin >> yas;

    if(yas<18)
        cout << "Ucret: 15 TL ";
    else if(yas<65)
        cout << "Ucret: 20 TL ";
    else
        cout << "Ucret: 10 TL";
    return 0;
}

```

5. Görev



1. Grup Ekran Çıktısı:

.....

2. Grup Ekran Çıktısı:

.....

3. Grup Ekran Çıktısı:

.....

4. Grup Ekran Çıktısı:

.....

5. Grup Ekran Çıktısı:

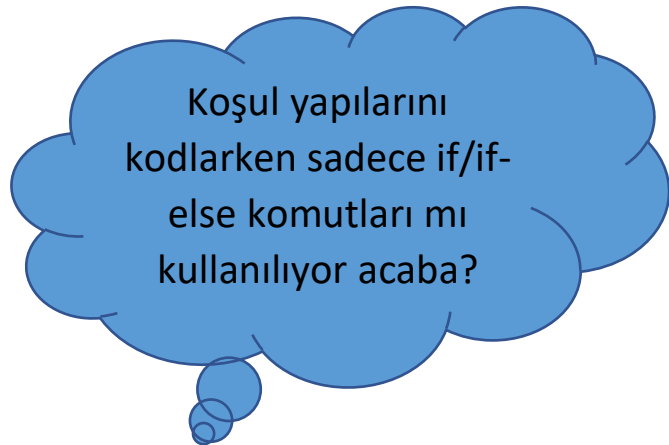
.....

```

#include <iostream>
using namespace std;
int main()
{
    int saat,maas;
    cout << "kac saat calisti:";
    cin >> saat;
    if(saat<100)
        maas = saat * 5;
    else if(saat<250)
        maas = saat * 5 * 2;
    else
        maas = saat * 5 * 3;
    cout << "Maasiniz: "<< maas;
    return 0;
}

```

EK 2. SWITCH Yapısı Kullanımı



Cevap: Birden çok koşul olduğu durumlarda “switch” bloğunu kullanabiliriz. Gelin if ile switch kullanımını hesap makinesi yapımı için gereken kodlamayı yaparak kıyaslayalım.

If/ If Else Kullanımı

```
#include <iostream>
using namespace std;
int main()
{
    char islem;
    cin >> islem;
    if(islem == '+')
        cout << "Toplama islemi";
    else if(islem == '-')
        cout << "Cikarma islemi";
    else if(islem == '*')
        cout << "Carpma islemi";
    else if(islem == '/')
        cout << "Bolme islemi";
    else
        cout << "Hatali giris.";
}
```

Switch/Case Kullanımı

```
#include <iostream>
using namespace std;
int main()
{
    char islem;
    cin >> islem;
    switch(islem){
        case '+':
            cout << "Toplama islemi";
            break;
        case '-':
            cout << "Cikarma islemi";
            break;
        case '*':
            cout << "Carpma islemi";
            break;
        case '/':
            cout << "Bolme islemi";
            break;
        default:
            cout << "Hatali giris.";
    }
}
```

EK 3. Hafta 5 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 6. Döngü Yapıları

Kazanımlar

K1. C++ programlama dilinde döngü mantığını kavrar.

K2. C++ programlama dilinde döngü oluşturabilir.

Haftanın Amacı ve Materyaller

Bu haftanın amacı öğrencilerin programlamada döngü yapılarını kullanarak programın akışını kontrol edebilmelerini sağlamaktır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız.](#)

Önerilen Ders Akışı (30 + 50 + 60+ 50 dk.)

GİRİŞ (10 dk.)

GELİŞME

Döngüleri Tanıyalım (20 dk.)

EK 1. Döngüleri Tanıyalım

Ders Arası (5 dk.)

Döngüleri Ne İçin Kullanıldığını Keşfediyorum (50 dk.)

EK 2. Koddaki Döngülerin Ne İçin Yazıldığını Keşfediyorum.

EK 3. Döngü Çeşitleri Afişi

Ders Arası (10 dk.)

Döngü Görevlerini Kodlayalım (60 dk.)

EK 4. Döngü Görevlerini Bilgisayarlarımızda Kodlayalım

SONUÇ (10 dk.)

EK 5. Hafta 6 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 6 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 6 sunumunu ekran paylaşımına açar. Eğitmen bir önceki hafta işledikleri konu olan karar yapıları hakkında özet geçer ve öğrencilerin kazandıkları rozetleri sorar. Rozet görevleri ve karar yapıları üzerinden *ön bilgileri hatırlatır* ve altıncı haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

Eğitmene Öneriler: Verilen süreli ödevlerde sorunla karşılaşmış öğrencilere eğitmen söz hakkı verebilir. Benzer hatalar ile karşılaşmış ve sorunu çözmüş öğrencilere de söz hakkı tanınır. Öğrencilerden dönüt alınmazsa eğitmen hatanın nasıl giderileceği konusunda geri bildirimde bulunur. Bu şekilde diğer öğrencilerin de benzer sorunlara karşı çözüm üretmeleri sağlanmış olur. Zamana bağlı olarak eğitmen öğrencilere sağlayacağı bu desteği ders dışında sorunla karşılaşan öğrencilere yardımcı olarak da tamamlayabilir.

B. Gelişme

Döngüleri Tanıyalım

Süre: 20 dk.

Kazanımlar: K1. C++ programında döngü mantığını kavrar.

Materyaller: Hafta 6 Sunum

EK1. Döngüleri Tanıyalım

Hazırlık: Hafta 6 Sunumunda “Hangi Blok Kodları Kullanırdınız” adlı iki slayt etkinlik sırası geldikçe açılır.

Uygulama: “Hangi Blok Kodları Kullandınız 1” adlı slayt açılarak; arının tüm çiçeklerdeki nektarı alması için ilerle ve nektarı al bloklarını kaçar kez kullanması gerektiği öğrencilere anket yoluyla 1 dk. zaman verilerek sorulur. Yanıtlar geldikten sonra doğru cevap eğitmen tarafından sözlü bir şekilde söylenir. Daha sonra “Hangi Blok Kodları Kullandınız 2” slaydı açılarak ekrandaki kodları kullanarak, arının tüm çiçeklerdeki nektarı alması için ilerle ve nektarı al bloklarını kaçar kez kullanırdınız sorusu yine anket yoluyla 1 dk. zaman verilerek sorulur. Etkinliklerin cevabı aşağıda verilmiştir.



Çözüm 1



Çözüm 2

Eğitime Öneriler: Öğrencilerin her iki anket için verdiği yanıtların karşılaştırması istenir ve öğrencilere şu soru sorulur? Görev 1 de 1000 tane çiçek olsaydı ve arıya tüm nektarı aldıracak olsaydı 1000 kez mi ilerle ve nektarı al kodunu kullanırsınız? Öğrencilerin bu soru üzerinde tartışması sağlanarak döngünün neden kullanıldığı hakkında fikir sahibi olması sağlanır. Daha sonra öğretmen aşağıdaki bilgiler doğrultusunda döngüler konusunu sunuda da belirtildiği üzere özetler.

Döngüler, birden çok kez tekrar eden görevleri gerçekleştirmek için kullanılmaktadır. Tekrar sayısı önceden bilinebildiği gibi, programın çalışma sırasında da belirlenebilir. Tekrar sayısı kullanıcı girişine de bağlı olabilir. Örneğin ekrana 10 kez uyarı göndermek için veya kullanıcı şifresini doğru girene kadar uyarı göndermek için döngüler kullanılabilir.

Döngü kullanmamak bir kaba 12 bardak süt dökmek için 12 tane ayrı bardak kullanmaya benzer. Yapılabilir ama mantıklı değildir.

Döngüleri Ne İçin Kullanıldığını Keşfetme

Süre: 50 dk.

Kazanımlar: K2. C++ programında döngü oluşturabilir.

Materyaller: Hafta 6 Sunum

EK2. Koddaki Döngülerin Ne İçin Yazıldığını Keşfediyorum.

EK3. Döngü Çeşitleri Afişi

Hazırlık: Eğitimci sunum üzerinden döngü çeşitleri slaydını veya ekte bulunan afişin yüksek çözünürlüklü halini açar. Ayrıca kodlama yapılacağı için Code::Blocks programı açılır.

Uygulama: Eğitimci döngüler için hazırlanan afiş öğrencilerle de ÖYS üzerinden paylaşarak döngüler hakkında aşağıdaki gibi bir özetleme yapar.

Döngüleri C++ programlama dilinde “for” “while” ve “do-while” komutları ile gerçekleştirebiliriz. Bir döngüyü istediğimiz komut ile yapabiliriz. Üç tane olmasının nedeni ise bazı işlemleri bazı döngüler ile yapmak daha kolay olmaktadır. Döngüler için oluşturulan afiş öğrencilerle paylaşılır.

Özetleme yaptıktan sonra eğitimci sunum üzerinde bulunan 10 farklı görevin ne amaçla yazıldığını öğrencilere teker teker sorar ve öğrencilerin yanıtların chat kısmından yazmasını ister. Her bir görev sonrası öğrencilerin verdiği yanıtların doğruluğunu kontrol etmesi için kodu kendi bilgisayarlarında yazması istenir.

Eğitmene Öneriler: Yukarıdaki kodlamalar ve verilen görevlerin amacını bulma bittikten sonra döngüler sunumda da verildiği üzere aşağıdaki gibi özetlenir.

1. Bir döngüde olması gerekenler şunlardır;
2. Kontrol değişkeni: Döngü başlangıç ve bitişi hangi değişken üzerinden kontrol edilecek.
3. Değişken başlangıç değeri
4. Bitiş koşulu
5. Değişken güncellemesi: Her tekrar sonrasında kontrol değişkeni nasıl etkilenecek?
6. Döngü gövdesi: Döngü içerisinde gerçekleştirilecek işlemler.

Döngü Görevlerini Kodlayalım

Süre: 60 dk.

Kazanımlar: K2. C++ programında döngü oluşturabilir.

K3. C++ programında iç içe döngü yapısını bilir.

Materyaller: Hafta 6 Sunum

EK4. Döngü Görevlerini Bilgisayarlarımızda Kodlayalım

Hazırlık: Eğitimciden bu bölüm için döngüler ile ilgili çözülmesi gereken beş görevin bulunduğu slaytları sırası geldikçe açması beklenir. Ayrıca kodlama yapılacağı için Code::Blocks programı açılır.

Eğitmene Öneriler: Eğitimci görevleri öğrencilere sunu üzerinden göstermeden önce C++ programlama dilinde nasıl rastgele sayı üretildiği hakkında Hafta 5 sunumunda da verildiği üzere aşağıdaki gibi kısa bir bilgi verir. Rastgele sayı üretimi bu bölümde verilen bazı görevlerin çözümünde kullanılacağı için öğrenciler için ön bilgi niteliğindedir.

“C++ programlama dilinde rastgele sayı üretmek için rand() hazır fonksiyonu kullanılır. Bu fonksiyon 0 ile üst sınır (en az 32767 en çok RAND_MAX) arasında rastgele sayı üretir. Üst sınırı

sınırlandırmak için mod (%) operatörü kullanılır. 0-100 (100 hariç) arasında rastgele sayı üretmek istersek `rand()%100` şeklinde kullanırız. alt sınırı arttırmak/azaltmak istersek toplama işlemini kullanırız. Örneğin 10 ile 100 arasında rastgele sayı üretmek için `10 + (rand() % 90)` şeklinde kullanırız. `srand()` fonksiyonu `rand()` fonksiyonu için hazırlayıcı fonksiyondur. Rastgele sayı üretici için kullanılacak başlangıç değerini ayarlar. Eğer `srand()` kullanılmaz ise program her çalıştırıldığında aynı rastgele değerler elde edilir. “`srand(time(0));`” satırı main bloğunun başına eklenerek kullanılabilir.”

Uygulama: Bu bölüm 5 görevden oluşmaktadır. Her bir görev için eğitmenin o görev esnasında neler yapacağı aşağıda sırasıyla verilmektedir.

Görev 1: Merve adlı öğrenci 1’den 20’ye kadar olan tek sayıları bulan bir program yazıp ekranda göstermek istemektedir. Merve bunun için sizce nasıl bir kod yazmalı?

Yukarıdaki görev için “Görev 1” başlıklı slayt açılır. Daha sonra eğitmen Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğitmen her satır kodun ne anlama geldiğini ve ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 5 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eğitmen öğrencilerin kodları kendi bilgisayarlarında yazdıklarından emin olduktan ve ihtiyacı olan öğrencilere gerekli geri bildirimleri verdikten sonra bir sonraki görev ve slayda geçer. Görev 1’in cevabı aşağıdaki gibidir. Eğitmenin aşağıdaki gibi 3 farklı döngü yapısını kullanarak öğrencilere kodlamayı göstermesi beklenmektedir.

Cevap 1:

For

```
int i;
for(i=1;i<20;i++)
    if(i%2==1)
        cout << i << endl;
```

While

```
int i=1;
while(i<20)
{
    if(i%2==1)
        cout << i << endl;
```

```

        i++;
    }

```

Do-While

```

int i=1;

do
{
    if(i%2==1)
        cout << i << endl;

    i++;
} while(i<20);

```

Görev 2: Ali kardeşi Buğra'nın 1'den 30'a kadar 3'erli olarak sayı saymasını istemektedir. Buradan hareketle kardeşinin doğru sayıp saymadığını kontrol etmesi için bilgisayarda 1'den 30'a kadar küçükten büyüğe olacak şekilde 3'e bölünebilecek sayıları ekranda göstermek istiyor. Bunun için nasıl bir kod yazmalı?

Yukarıdaki görev için "Görev 2" başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğu bildirilir ve öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.'lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodların tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapması istenir. Eğitimci bitirip el kaldıran öğrencilerden rastgele birini seçerek öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitimci kodun nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 2'in cevabı aşağıdaki gibidir.

Cevap 2:

```

#include <iostream>

using namespace std;

int main()
{
    for(int i=0; i < 30; i++)
    {

```

```

if(i%3==0)

cout << i << endl;

}

return 0;

```

Görev 3: Rafet öğretmen sınıfında bulunan 10 öğrencinin matematik dersinde aldığı notları klavyeden teker teker girerek sınıfın matematik dersi not ortalamasını bulan bir program yazmak istiyor. Bunun için nasıl bir kod yazmalıdır?

Yukarıdaki görev için “Görev 3” başlıklı slayt açılır. Daha sonra eğitimci Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğitimci her satır kodun ne anlama geldiğini ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 10 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eğitimci öğrencilerin kodları kendi bilgisayarlarında yazdıklarından emin olduktan ve ihtiyacı olan öğrencilere gerekli geri bildirimleri verdikten sonra bir sonraki görev ve slayda geçer. Görev 3’ün cevabı aşağıdaki gibidir.

Cevap 3:

```

#include <iostream>

using namespace std;

int main()
{
    int toplam = 0;
    for(int i=0; i < 10; i++)
    {
        int puan;
        cout << i+1 << ". öğrenci puanı:";
        cin >> puan;
        toplam += puan;
    }
    cout << "ortalama: " << toplam /10 ;
}

```

Görev 4: Defne öğretmen aldığı 10 tane kitabı sınıfındaki öğrencilere çekiliş yoluyla dağıtmak istemektedir. Sınıftaki öğrencilerin okul numaraları 50 ile 100 arasındadır (100 hariç). Defne öğretmen bunun için 50 ile 100 arasında 10 tane rastgele bir sayı üreten program yazarak kitapları okul numarası rastgele çıkan öğrencilere verecektir. Bunun için nasıl bir kod yazmalıdır?

Yukarıdaki görev için “Görev 4” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğu bildirilir ve öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodların tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapması istenir. Öğitmen bitirip el kaldıran öğrencilerden rastgele birini seçer ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitmen kodun nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 4’ün cevabı aşağıdaki gibidir.

Cevap 4:

```
include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    srand(time(0));
    for(int i=0; i<10;i++)
        cout << 50 + rand() % 50 <<endl;
}
```

Görev 5: Ahmet okul kütüphanesindeki raflara herkesin kolayca kitapları bulabilmesi için sayı etiketleri yapıştırmak istiyor. Kütüphanede 30 tane raf olduğu düşünülürse Ahmet’in 1 den 30’a kadar sayıları sıralayıp ekranda göstermesi gerekmektedir. Buradan hareketle Ahmet’in nasıl bir kod yazması gereklidir, bilgisayarımızda kodlayalım.

Yukarıdaki görev için “Görev 5” başlıklı slayt açılır. Daha sonra eğitmen Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğitmen her satır kodun ne anlama geldiğini ve ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 7 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eğitmen öğrencilerin kodları kendi bilgisayarlarında yazdıklarından emin olduktan ve ihtiyacı olan öğrencilere gerekli geri bildirimleri verdikten sonra bir sonraki görev ve slayda geçer. Görev 5’in cevabı aşağıdaki gibidir.

Cevap 5:

```

int sayi = 1;

while(sayi<30)    {

    cout << sayi <<endl;

    sayi ++;

}

```

Eğitime Öneriler: Yukarıdaki etkinlikteki amaç öğrencilerin kod yazarken, döngü yapılarını nerede ve ne zaman kullanılacağına yönelik farkındalığını artırmaktır. Yukarıdaki görevlerin tamamını öğrencilerin kendi bilgisayarlarında kodlanması beklenmektedir. Bunun için öğretmenlerin etkinlikler esnasında sürekli farklı öğrencilerinden isteyeceği ekran paylaşımı onların her etkinliğe hazır girmesi için motivasyonunun artmasını sağlayacaktır.

C. Sonuç

Süre: 10 dk.

Materyal: EK 5 Hafta 6 Süreli Ödev

Hazırlık: EK 5 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta "Diziler ve Katarlar" konusunun işleneceği bildirilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. EK 5 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler ileildikten sonra eğitim görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödevler Yanıtlar: Görevler, öğrenciye verilecek beceri rozetleri ile isimlendirilmiştir. Her bir rozetin altında ilgili görevin yanıtları verilmektedir. Eğitim bu yanıtları uygulama süresinin sonunda GitHub üzerinden öğrencilerle paylaşır.

1) **Analizci:** Aşağıdaki kod ekrana kaç kez merhaba yazar. (Cevap 0)

```
for(int i=0;i>10;i++)
{
    cout << "merhaba" << endl;
}
```

2) **Kodlayıcı:** Ahmet bilgisayara rastgele 2 sayı ürettirip, üretilen sayıdaki büyük olanı ekrana yazdırmak istiyor. Ahmet'in nasıl bir kod yazması gerekmektedir?

```
#include <iostream>
```

```
#include <cstdlib>
```

```
#include <ctime>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int sayi1, sayi2;
```

```
    srand(time(0));
```

```
    sayi1 = rand();
```

```
    sayi2 = rand();
```

```
    if(sayi1>sayi2)
```

```
        cout << sayi1;
```

```
    else
```

```
        cout << sayi2;
```

```
}
```

- 3) **Kodlayıcı:** Bir sınıftaki öğrencilerin numarası 5 ile 25 arasında değişmektedir. Sınıfa giren matematik öğretmeni Sercan, 3 ile tam bölünebilen öğrenci numaralarını bulduran bir program yazmak istediğini belirtmiştir. Siz matematik öğretmeninize kodları nasıl yazarak yardımcı olurdunuz?

```
for(int i=5;i<25;i++)  
    if (i%3 == 0)  
        cout << i << endl;
```

Hafta 6. Ders Materyalleri

EK 1. Döngüleri Tanıyalım Görev 1



ilerle ▼

sağa dön ↻ ▼

sola dön ↺ ▼

nektarı al

* https://studio.code.org/s/express-2020/lessons/22/levels/1?section_id=2984849

Yukarıdaki arının tüm çiçeklerdeki nektarı almasını isteseydiniz **blok** kodlardan hangisini kaç kere kullanırdınız?

EK1. Döngüleri Tanıyalım Görev 2



ilerle ▼

bu işlemleri 5 kez tekrarla
yap

sağa dön ↻ ▼

* https://studio.code.org/s/express-2020/lessons/22/levels/1?section_id=2984849

Yukarıdaki arının tüm çiçeklerdeki nektarı almasını isteseydiniz **blok** kodlardan hangisini kaç kere kullanırdınız?

EK 2. Koddaki Döngülerin Ne İçin Yazıldığını Keşfediyorum.**Kod****Problem**

Sol kısımda bulunan kodlar hangi temel problem ya da amaç için yazılmıştır? Bu sütunda belirtiniz.

```

for(int sayi=0;sayi<10;sayi++)
for(int sayi=10;sayi>=0;sayi--)
for(int sayi=10;sayi>=0;sayi-=2)
for(int i=15;i>=0;i-=3)

#include <iostream>
using namespace std;
int main()
{
    for(int i=0;i<10;i++)
        cout << "DENEYAP" << endl;
    return 0;
}

#include <iostream>
using namespace std;
int main()
{
    for(int i=1; i<10; i++)
    {
        cout << i << endl;
    }
    return 0;
}

int sayi = 1;
while(sayi<100)
{
    cout << sayi << endl;
    sayi ++;
}

int i;
for(i=1;i<20;i++)
    if(i%2==1)
        cout << i << endl;

```

```
int i=1;
while(i<20)
{
    if(i%2==1)
        cout << i << endl;
    i++;
}
```

```
int i=1;
do
{
    if(i%2==1)
        cout << i << endl;
    i++;
}while(i<20);
```

EK 3. Döngü Çeşitleri Afişi (Yüksek çözünürlüklü hali için materyaller klasörüne gidiniz.)

#C++Ogreniyorum

Döngüler Nasıl Kullanılır?

Source: DeneYap İçerik Geliştirme

FOR Döngüsü

For döngüsünde değişkene ilk değer atanır.
Her bir adımdaki artış değeri değişkene eklenir.
Koşul sağlandığı sürece çalışmaya devam eder.
Döngünün temel söz dizimi aşağıdaki gibidir;

for(değişken=ilk deger; koşul; her adımdaki değişim)

While Döngüsü:

1. While döngüsü durdurma kriteri sağlanana kadar çalışmaya devam eder.

Kullanımı:

```
while (koşul)
{
    koşul doğru olduğu sürece yapılacak işlemler.
}
```

WHILE

2. while döngüsü içerisinde kontrol değişkeninin değeri güncellenmelidir. Aksi takdirde sonsuz döngüye girebilir ve program hiç sonlanmaz!

do-while döngüsü:

1. do while döngüsünün, while döngüsünden farkı önce işlem yapılır, sonra koşul kontrol edilir.

Kullanımı:

```
do
{
    işlemler
}while(koşul);
```

DO WHILE

Birden fazla kontrol değişkeni kullanmamız gerekir ise, döngüleri iç içe kullanırız.

Resim 25. Döngü çeşitleri afişi

EK4. Döngü Görevlerini Bilgisayarlarımızda Kodlayalım

1

Merve adlı öğrenci 1'den 20'ye kadar olan tek sayıları bulan bir program yazıp ekranda göstermek istemektedir. Merve bunun için sizce nasıl bir kod yazmalı?

2

Ali kardeşi Buğra'nın 1'den 30'a kadar 3'erli olarak sayı saymasını istemektedir. Buradan hareketle kardeşinin doğru sayıp saymadığını kontrol etmesi için bilgisayarda 1'den 30'a kadar küçükten büyüğe olacak şekilde 3'e bölünebilecek sayıları ekranda göstermek istiyor. Bunun için nasıl bir kod yazmalı?

3

Caner öğretmen sınıfında bulunan 5 öğrencinin matematik dersinde aldığı notları klavyeden girerek bulunmasını isteyen bir program yapmak istiyor. Bunun için nasıl bir kod yazmalı.

4

Defne öğretmen aldığı 10 tane kitabı sınıfındaki öğrencilere çekiliş yoluyla dağıtmak istemektedir. Sınıftaki öğrencilerin okul numaraları 50 ile 100 arasındadır. Defne öğretmen bunun için 50 ile 100 arasında 10 tane rastgele sayı üreten bir program yazarak çıkan numaraya sahip öğrencilere kitabı vermeyi planlamaktadır. Bunun için nasıl bir kod yazmalıdır.

5

Ahmet okul kütüphanesindeki raflara herkesin kolayca kitapları bulabilmesi için sayı etiketleri yapıştırmak istiyor. Kütüphanede 30 tane raf olduğu düşünülürse Ahmet'in 1 den 30'a kadar sayıları sıralayıp ekranda göstermesi gerekmektedir. Buradan hareketle Ahmet'in nasıl bir kod yazması gereklidir, bilgisayarımızda kodlayalım.

EK 5. Hafta 6 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 7. Diziler ve Katarlar

Kazanımlar

- K1. C++ programlamada dizi kavramını açıklar.
- K2. C++ programlamada tek boyutlu ve çok boyutlu dizileri kullanır.
- K3. C++ programlamada dizilere farklı türde değerler atar.
- K4. Dizileri döngü içinde kullanır.
- K5. Karşılaşılan problemlere dizileri kullanarak çözüm üretir.
- K6. Diziler ve katarlar arasındaki farkı ayırt eder.

Haftanın Amacı ve Materyaller

Bu haftanın amacı, tek boyutlu ve çok boyutlu dizi tanımlama ve diziler üzerinde farklı işlemlerin gerçekleştirilmesi ile örnek çözümlerin öğretilmesi amaçlanmaktadır. Katarların tanımlanması ve kullanımını uygulayarak dizilerden farkını öğrenir.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız](#).

Önerilen Ders Akışı (50 + 50 + 50 + 50 dk.)

GİRİŞ (5 dk.)

GELİŞME

Dizileri Tanıyalım (25 dk.)

EK 1. Örnek Dizi

Dizilere Değer Verelim (20 dk.)

Ders Arası (10 dk.)

Döngülerle Diziler (20 dk.)

EK 2. Grup Görevleri

Dizilerle Kodlayalım (30 dk.)

EK 3. Kod Satırlarını Tamamla

Ders Arası (10 dk.)

Kodlama Ekibi (30 dk.)

Farkı Keşfedelim (10 dk.)

SONUÇ (10 dk.)

EK 4. Hafta 7 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 7 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken, *dikkat çekmek için* eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 7 sunumunu ekran paylaşımına açar. Eğitmen bir önceki hafta işledikleri konu olan döngüler hakkında özet geçer ve öğrencilerin kazandıkları rozetleri sorar. Rozet görevleri ve döngüler üzerinden *ön bilgileri hatırlatır* ve yedinci haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

B. Gelişme

Dizileri Tanıyalım

Süre: 25 dk.

Kazanımlar: K1. C++ programlamada dizi kavramını açıklar.

K2. C++ programlamada tek boyutlu ve çok boyutlu dizileri kullanır.

Materyaller: Hafta 7 Sunum

Çeşitli madeni paralar, kâğıt ve kalem

Dijital Tartışma Panosu: Bireysel Çalışma

EK 1. Örnek Dizi

Hazırlık: Eğitmen dersin başında öğrencilerden yanlarında çeşitli büyüklüklerde madeni para (bir liradan 6 tane, 50 kuruştan 8 tane gibi...) bulundurmalarını ister. Bu mümkün değilse kalem ve kâğıt kullanabileceklerini söyler. Eğitmen derse girmeden önce bireysel çalışma için duvar temalı bir dijital panoyu padlet ortamında oluşturur. Oluşturulan linki, ders sırasında öğrencilere paylaşılan notlar kısmından iletacaktır.

Uygulama: Bu etkinlikte öğretmen dizi ve indis kavramları üzerine sunumdan ilerleyerek yeni bilgiyi verir. Tek boyutlu ve çift boyutlu diziler hakkında örnekler üzerinden açıklamalar yapılır. Bazı slaytlarda sohbet ya da anket kullanılarak konu ile ilgili yöneltilen soru hakkında öğrenci tahminleri alınır. Bu şekilde öğretmen yeni bilginin sunumunu yaparken, öğrencilerin de performanslarını yansıtmaları beklenir. Daha sonra öğretmen öğrenci yanıtları üzerinden geri bildirimlerde bulunur. Bu aşamada maksimum 10 dk. geçirilir.

Öğrenciler bu etkinlikte öğrendiklerinin kalıcılığı ve transferi için bir uygulama yapmaktadır. Öğitmen sunumda “Dizi Talimatları” başlıklı slaytta bekler ve uygulamayı öğrencilere açıklar. Bu slaytta öğrencilerden yanlarında bulunan çeşitli büyüklüklerdeki madeni paralar ile tek ve çok boyutlu diziler oluşturmaları istenmektedir. Tüm talimatlar bireysel olarak beyaz bir kâğıt üzerinde oluşturulmalıdır. Oluşturulacak diziler için öğretmen sunumda aşağıdaki talimatlarla verilmektedir:

Elinizdeki madeni paraları ya da kâğıt ve kalem kullanarak,

1. *Tek boyutlu bir dizi oluşturun. Dizin boyutunu ve dizi indislerini dizi elemanlarının altına yazın.*
2. *Birinci talimatta oluşturduğunuz tek boyutlu dizinin hemen altına, aynı diziden iki tane daha ekleyin. Bu şekilde elde ettiğiniz çok boyutlu dizinin boyutunu ve dizi elemanlarının indis numaralarını her bir elemanın altına yazın.*

Bu talimatlarda dizi boyutları değişmekte ve öğrencinin dizinin boyut bilgisini ve indis numaralarını da kâğıtlarına yazmaları gerekmektedir. Aynı zamanda öğretmen öğrencilere paylaşılan notlar kısmında dijital tartışma panosunun linkini iletir. Görev tamamlanınca öğrenciler hazırladıkları dizilerin fotoğraflarını çekerek panoya göndermelidir. Bu etkinlikte madeni para bulamayan öğrenciler bir kâğıda farklı büyüklükte daireler çizerek dizileri oluşturabilirler. Talimatlar açıklandıktan sonra öğretmen 5 dk.’lık sözsüz bir odaklanma müziği başlatır. Müzik sona erdiğinde etkinlik de sona erecektir. Öğitmen etkinlik boyunca sunu aracılığıyla dizi talimatlarını ekranda tutarken, etkinlik sonunda dijital tartışma panosunu ekran paylaşımı yapar. Talimatlara göre hazırlanmış bir örnek EK 1’de verilmiştir. Panoda tüm öğrenci gönderileri yapıldıktan sonra öğretmen de kendi örneğini EK 1’in ekran görüntüsünü alarak panoya iletir. Bu şekilde öğrenci gönderileri üzerinden özellikle hatalı dizilen örnekler bulunur ve konu tekrar edilir. Çok fazla hatalı gönderilerde öğretmen konuyu özellikle kendi ilettiği EK 1 örneği üzerinden açıklar ve yanlış öğrenmeleri düzeltir.

Eğitmene Öneriler: Etkinlik sırasında müzik sona erdiğinde bireysel olarak oluşturulan dizilerin fotoğrafları öğrenciler tarafından dijital panoya gönderilmiş olmalıdır. Öğitmen bu nedenle müzik bitimine yakın hatırlatmada bulunmalıdır.

Dizilere Değer Verelim

Süre: 20 dk.

Kazanımlar: K2. C++ programlama dilinde tek boyutlu ve çok boyutlu dizileri kullanır.

K3. C++ programlama dilinde dizilere değer atar.

Materyaller: Hafta 7 Sunum

Hazırlık: Eğitimci Hafta 7 sunumunda “Dizilere Değer Verelim” başlıklı slaytları kullanır.

Uygulama: Bu etkinlikte eğitimci sunumda ilerleyerek tek boyutlu ve çok boyutlu dizilere değer atama ile ilgili yeni bilgiyi verir. Tek boyutlu ve çift boyutlu diziler hakkında örnekler üzerinden açıklamalar yapılır. Bazı slaytlarda anket kullanılarak, konu ile ilgili yöneltilen soru hakkında öğrenci tahminleri alınır. Bu şekilde eğitimci yeni bilginin sunumunu yaparken, öğrencilerin de performanslarını yansıtmaları beklenir. Daha sonra eğitimci öğrenci yanıtları üzerinden geri bildirimlerde bulunur.

Döngülerle Diziler

Süre: 20 dk.

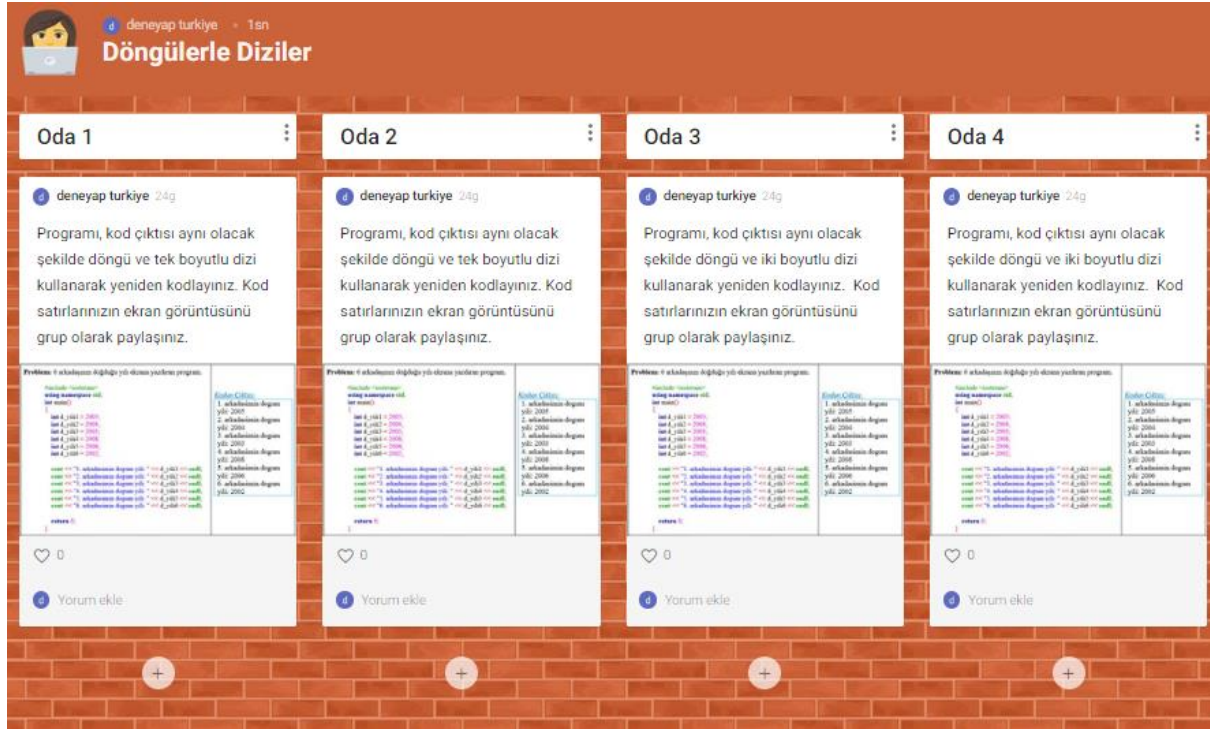
Kazanımlar: K4. Dizileri döngü içinde kullanır.

Materyaller: Hafta 7 Sunum

Dijital Tartışma Panosu: Grup Görevleri

EK 2: Grup Görevleri

Hazırlık: Eğitimci Hafta 7 sunumunda “Döngülerle Diziler” başlıklı slaytları kullanır. Eğitimci derse girmeden önce dört farklı grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletilecektir. Eğitimci padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. EK 2’deki materyalde yer alan iki farklı görev oda 1 ve oda 2 birinci görev; oda 3 ve oda 4 ikinci görev olacak şekilde gruplara yüklenecektir. Örnek bir tasarım Resim 26’da gösterilmiştir.



Resim 26. Döngülerle Diziler Örnek Padlet Panosu

Uygulama: Bu etkinlikte öğretmen ilk olarak sunumu kullanarak aşağıdaki açıklamayı yapar.

Dizilere ilk değer atamanın diğer bir yolu da döngüleri kullanmaktır. Bunu gerçekleştirmek için, önce diziyi normalde yaptığımız gibi tanımlar ve daha sonra oluşturacağımız döngü içerisinde istediğimiz değerleri atarız.

Açıklamanın ardından bir örnek kod gösterilir. Kod çıktısı öğrencilerle birlikte incelenir. Ardından öğretmen, “Döngülerle Diziler: Sıra Sizde” etkinliğine gelir. Bu etkinlikte öğrencilere döngü kullanılmadan oluşturulan dizilerle ilgili bir program vardır. Öğrenciler, bu programı döngü ve diziler kullanarak gruplar halinde yeniden kodlayacaktır. EK 2’de yer alan iki görev için dört grup oluşturulur. Burada iki farklı grubun aynı görev üzerinde çalışması sağlanarak, birbirini doğrulayan görevler kullanılması ve böylece akran öğrenmesinin desteklenmesi hedeflenmektedir. Öğretmen öğrencilere padlet linkini paylaşılan notlar kısmından iletir. Öğrenciler burada dört ayrı grup sütunu ve her sütun altında öğretmen tarafından eklenmiş görev kartını görecektir. Daha sonra öğretmen öğrenciler arasından dört kişiyi moderatör yapar ve diğerlerini dört moderatörün bulunduğu çalışma odasına rastgele göndereceğini belirtir. Moderatörler görevle ilgili grup içinde tartışılan kodları derleyici üzerinde yazma ve grup adına padlete yükleme görevini üstlenir. Aynı zamanda kodları ortak yazabilmek için moderatör ekranını grup ile paylaşabilir. Örneğin birinci odadaki moderatör grupça yazılan kod satırlarının ekran görüntüsünü padlet ortamındaki birinci görev sütununda paylaşır ya da doğrudan kod satırlarını yazar. Bu şekilde her grup kendi içerisinde 10 dk. görev üzerinde çalışır.

Süre bitiminde ana ekrana dönen öğrencilerin padlet ortamı öğretmen tarafından ekranda paylaşılmış haldedir. Öğretmen oda 1 ve oda 2’dekilerin aynı görev üzerinde çalıştıklarını belirterek

iki grup sonuçlarını karşılaştırır. Grup kodları arasındaki farklılıklar ya da benzerlikler üzerinden görevlerin doğru yanıtları öğrencilerle birlikte keşfedilir. Benzer şekilde oda 3 ve oda 4 görevlerine de eğitmen geri bildirimde bulunur. Böylece döngü ve diziler kullanarak aynı problem için birden fazla kodlama yapılabileceği konusunda öğrencilere geri bildirimde bulunulur.

Eğitime Öneriler: Eğitmen doğru yanıtları tartışma panosunda paylaşabilir. Ayrıca eğitmen tartışma panosunun pdf halini alarak, ders sonrası öğrencilere Moodle üzerinden iletebilir. Görev yanıtları aşağıdaki gibidir:

Görev 1 Yanıtı	Görev 2 Yanıtı
<pre>#include <iostream> using namespace std; int main() { int yaslar[5] = {15, 14, 17, 12, 16}; int i; for(i=0; i<5; i++){ cout << i+1 << ". arkadasimin yasi: " << yaslar[i] << endl; } return 0; }</pre>	<pre>#include <iostream> using namespace std; int main(){ int d_yili[2][3] = {{2005, 2004, 2003}, {2008, 2006, 2002}}; int k = 1; for(int i=0; i<2;i++){ for(int j=0; j<3; j++,k++){ cout << k << ". arkadasimin dogum yili: " << d_yili[i][j] << endl; } } return 0; }</pre>

Etkinlik: Dizilerle Kodlayalım

Süre: 30 dk.

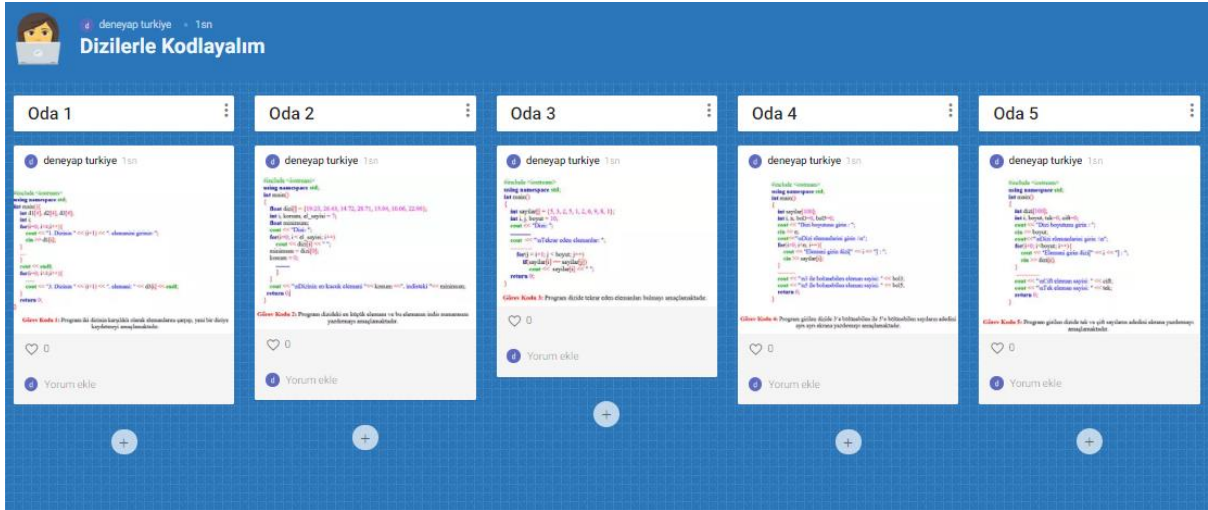
Kazanımlar: K5. Diziler üzerinde istenen işlemleri gerçekleştirir.

Materyaller: Hafta 7 Sunum

Dijital Tartışma Panosu: Grup Görevleri

EK 3: Kod Satırlarını Tamamla

Hazırlık: Eğitmen Hafta 7 sunumunda “Dizilerle Kodlayalım” başlıklı slaytı kullanır. Eğitmen derse girmeden önce beş farklı grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletecektir. Eğitmen padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. EK 3’teki görevler grup sütunlarına yüklenecektir. Örnek bir tasarım Resim 27’de gösterilmiştir.



Resim 27. Dizilerle Kodlayalım Örnek Padlet Panosu

Uygulama: Öğitmen bu etkinlikte bir öncekine benzer şekilde öğrencilere grup içinde çalışacakları kodlama görevleri verir. Öğrencilere etkinlik hakkında sunumun “Döngülerle Diziler: Grup Çalışması” başlıklı slaytı üzerinden aşağıdaki açıklama yapılır.

Görevlerin her biri gruplara göre değişen kod satırları içermektedir. Ancak bu kod satırlarında programın düzgün çalışmasını sağlayan dizi tanımlama satırları eksiktir. Buna göre eksik kodları tamamlayarak, programın doğru çalışmasını sağlayınız. Koda ilişkin ekran çıktısını grup olarak dijital panoya yükleyiniz.

EK 3’te beş farklı görev için dörder kişilik gruplar oluşturulur. Öğitmen öğrencilere padlet linkini paylaşılan notlar kısmından iletir. Öğrenciler burada beş ayrı grup sütunu ve her sütun altında eğitimci tarafından eklenmiş görev kartını görecektir. Öğitmen her gruptan bir öğrenciyi moderatör olarak tanımlar ve diğerlerini de rastgele gruplara dağıtır. Moderatörler görevle ilgili grup içinde tartışılan kodları derleyici üzerinde tamamlama ve grup adına padlete yükleme görevini üstlenir. Aynı zamanda kodları ortak yazabilmek için moderatör ekranını grup ile paylaşabilir. Örneğin birinci odadaki moderatör grupça yazılan kod satırlarının ekran görüntüsünü padlet ortamındaki birinci görev sütununda paylaşır ya da doğrudan kod satırlarını yazar. Bu şekilde her grup kendi içerisinde 10 dk. görev üzerinde çalışır. Süre bitiminde ana ekrana dönen öğrencilerin padlet ortamı eğitimci tarafından ekranda paylaşılmış haldedir. Öğitmen grup görevlerinin doğru yanıtlarını da her grup altına yükler. 3 dk.lık bir sözsüz müzik başlatır. Müzik süresince öğrencilerden kendi yanıtlarını eğitimcinin ki ile karşılaştırmaları ve diğer gruplara ait görevlerin de incelenmesini ister. Müzik sonunda eğitimci hatalı olan noktalar ya da öğrencilerin aklına takılan kısımlarla ilgili sorular üzerinden geri bildirimde bulunarak konuyu özetler.

Eğitime Öneriler: Eğitimci tartışma panosunun pdf halini alarak, ders sonrası öğrencilerle Moodle üzerinden paylaşabilir. Bu şekilde öğrencilerden grup çalışmasına dahil olmadıkları kod görevlerini sanal ders sonrası bireysel olarak da derleyici üzerinde çalışmaları istenebilir. Verilen grup görevlerinin doğru yanıtları aşağıdadır:

```
#include <iostream>
using namespace std;
int main(){
    int dizi1[5], dizi2[5], dizi3[5];
    int i;
    for(i=0; i<5;i++){
        cout << "1. Dizinin " << (i+1) << ".
elemanini giriniz: ";
        cin >> dizi1[i];
    }
    cout << endl;
    for(i=0; i<5;i++){
        cout << "2. Dizinin " << (i+1) << ".
elemanini giriniz: ";
        cin >> dizi2[i];
    }
    cout << endl;
    for(i=0; i<5;i++){
        dizi3[i] = dizi1[i] + dizi2[i];
        cout << "3. Dizinin " << (i+1) << ".
elemani: " << dizi3[i] << endl;
    }
    return 0;
}
```

Görev Kodu 1

Yandaki program iki dizinin karşılıklı olarak elemanlarını toplayıp, yeni diziye kaydetmeyi amaçlamaktadır.

```
#include <iostream>
using namespace std;
int main()
{
    int sayilar[] = {19, 11, 21, 13, 15};
    int i, x, y, n = 5;
    cout << "Dizi: ";
    for(i=0; i < n; i++)
        cout << sayilar[i] << " ";
    x = sayilar[0];
    y = 0;
    for(i=1; i < n; i++){
        if(x < sayilar[i]){
            x = sayilar[i];
            y = i;
        }
    }
    cout << x << " ve " << y;
    return 0;
}
```

Görev Kodu 2

Yandaki program dizideki en büyük elemanı ve bu elemanın indis numarasını yazdırmayı amaçlamaktadır.

```

#include <iostream>
using namespace std;
int main()
{
    int dizi[] = {12, 67, 78, 45, 78, 78, 32, 16,
16, 57};
    int i, j, n = 10;
    cout << "Dizi: ";
    for(i = 0; i < n; i++)
        cout << dizi[i] << " ";

    cout << "\nElemanlar: ";
    for(i = 0; i < n-1; i++)
        if(dizi[i] == dizi[i+1])
            cout << dizi[i] << " ";
    return 0;
}

```

Görev Kodu 3

Yandaki program dizide ardışık olarak tekrar eden elemanları bulmayı amaçlamaktadır.

```

#include <iostream>
using namespace std;
int main()
{
    int dizi[100];
    int i, boyut, tek=0, cift=0;
    cout << "Eleman sayisini girin: ";
    cin >> boyut;
    cout<<"\nDizi elemanlarini girin:\n";
    for(i=0; i<boyut; i++){
        cout << "Elemani girin dizi[" << i << "]:
";
        cin >> dizi[i];
    }
    for(i=0; i<boyut; i++){
        if(dizi[i]%2==0)
            cift++;
        else
            tek++;
    }
    cout << "\nCift eleman sayisi: " << cift;
    cout << "\nTek eleman sayisi: " << tek;
    return 0;
}

```

Görev Kodu 4

Yandaki program girilen dizide tek ve çift sayı adedini ekrana yazdırmayı amaçlamaktadır.

```

#include <iostream>
using namespace std;
int main()
{
    int sayilar[100];
    int i, n, bol5=0;
    cout << "Eleman sayisini girin: ";
    cin >> n;
    cout<<"\nDizi elemanlarini girin:\n";
    for(i=0; i<n; i++){
        cout << "Elemani girin dizi[" << i << "]: ";
        cin >> sayilar[i];
    }
    for(i=0; i<n; i++){
        if(sayilar[i]%5==0)
            bol5++;
    }
    cout << "\n5 ile bolunebilen eleman sayisi: "
    << bol5;
    return 0;
}

```

Görev Kodu 5

Girilen dizide 5'e bölünebilen sayı adedini ekrana yazdırmayı amaçlamaktadır.

Kodlama Ekibi

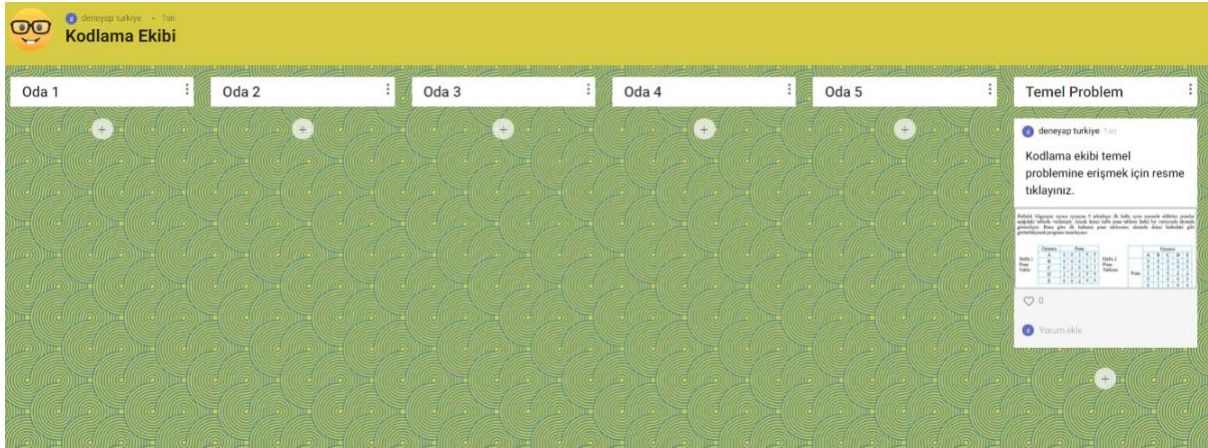
Süre: 30 dk.

Kazanımlar: K5. Diziler üzerinde istenen işlemleri gerçekleştirir.

Materyaller: Hafta 7 Sunum

Dijital Tartışma Panosu: Grup Görevleri

Hazırlık: Eğitimci Hafta 7 sunumunda "Kodlama Ekibi" başlıklı slaytı kullanır. Eğitimci derse girmeden önce beş farklı grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletecektir. Eğitimci padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. Örnek bir tasarım Resim 28'de gösterilmiştir.



Resim 28. Kodlama Ekibi: Grup Çalışması Örnek Padlet Tablosu

Uygulama: Eğitimci öğrencilere bu etkinlikte bir problem üzerinde ilk olarak bireysel çalışacaklarını daha sonra ise bireysel çalışmalarını grup olarak tamamlayacaklarını belirtir. Temel olarak herkes tek bir problemin çözümü için çalışırken, bireysel çalışmalar grup çalışmasının tamamlayıcısı olmaktadır. Başka bir ifadeyle bireysel görevler birleştirildiğinde bir problemin çözümüne ilişkin kodlar ortaya çıkmaktadır. Bunun için eğitimci sunumun “Kodlama Ekibi: Bireysel Çalışma” başlıklı slaytı üzerinden aşağıda verilen problemi öğrencilerle paylaşır.

Haftalık oyun oynayan üç arkadaşın oyun sonunda aldıkları puanların ilk iki haftası aşağıda verilmiştir. Üçüncü hafta ise ilk iki haftanın puan toplamından oluşmalıdır. Buna göre bu üç arkadaşın üçüncü hafta puanlarını ekrana yazdıran programı tasarlayınız.

Hafta 1:	Oyuncu				Puan			
	Ayşe				1	0	1	
	Burcu				0	0	0	
	Can				0	2	0	

Hafta 2:	Oyuncu				Puan			
	Ayşe				1	1	1	
	Burcu				2	0	0	
	Can				0	2	0	

Hafta 3								
Oyuncu				Puan				
Ayşe				?	?	?		
Burcu				?	?	?		
Can				?	?	?		

Eğitmen öğrenci isimlerini alfabetik sıralayarak, sınıfı ikiye ayırır ve grupların görevlerini açıklar. Birinci grup, problemdeki ilk hafta skorlarını sırayla ekrana yazdıran kod satırları üzerinde bireysel olarak çalışırken ikinci gruptakiler ikinci hafta skorlarının ekrana yazdırılması üzerine çalışır. Her iki gruptaki öğrenciler ilk etapta probleme ilişkin görevlerini tamamlamak üzere 5 dk. kadar bireysel çalışma gerçekleştirir. Bu süre içinde eğitmen 5 dk.'lık sözsüz bir odaklanma müziği başlatır. Müzik devam ederken eğitmen alfabetik sıralamaya göre ayrılan birinci gruptaki öğrencilerin kullanıcı özelliklerini moderatör olarak tanımlar. Moderatör tanımlamanın temel amacı, sınıf içinde grup ayrımının belirlenmesi ve ikinci etapta çalışma odalarına dağılırken homojen grupların oluşturulması içindir. Müzik sonunda eğitmen herhangi bir geri bildirimde bulunmadan ikinci etaba geçeceklerini belirtir. Eğitmen ikinci etap için grupları oluşturmadan önce sunumda “Kodlama Ekibi: Grup Çalışması” başlıklı slayta ilerleyerek öğrencilere açıklama yapılır.

İkinci etapta iki gruptaki öğrenciler birleştirilerek dörderli kodlama ekiplerini oluşturur. Ekipler oluşturulurken eğitmen beş çalışma odasına ilk olarak moderatör olmayan grup 2 öğrencilerini rastgele dağıtır. Sistem moderatör olan grup 1'i bu işlemde otomatik olarak atamaz. Bunun için eğitmen beş odaya moderatör öğrencilerden ikişer kişiyi elle dağıtmalıdır. İkinci etapta bireysel çalışmaların birbirlerine aktarılması önemlidir. Çünkü öğrenciler grup halinde temel problemi çözmek için bireysel yazdıkları kod satırlarını bir araya getirecektir. Bu nedenle kodlama ekibinde grup 1'den iki, grup 2'den iki öğrencinin bulunması önemlidir. Böylece ekran öğrenimi desteklenmelidir. Dört kişiden oluşan çalışma gruplarına 10 dk. süre verilir. Odalarda tamamlanan kodların ekran çıktıları gruplar tarafından padlet linkine gönderilir. Oluşturulan grupların görevi sunum üzerinden öğrencilere aşağıdaki şekilde aktarılır.

Grup Görevi: İlk hafta ve ikinci hafta skorlarını ekrana yazdıran kod satırlarını birleştirin. Bu iki skorların toplamını ekrana yazdıran yeni kod satırlarını oluşturun. Şimdi elinizde temel problemi çözecek kod satırı parçaları vardır. Bunları bir araya getirerek temel problemin çıktısını oluşturacak programı çalıştırın.

Süre sonunda ana odaya dönen grupların gönderileri için eğitmen padlet sayfasını ekran paylaşımı yapmıştır. Eğitmen öğrencilerin diğer gruplar tarafından iletilen ekran çıktılarında benzer ve farklı noktalara dikkat çeker ve ayrı bir sütunda doğru kod satırlarını padlete yükler. Bu şekilde öğrenci gönderileri ile problemin çözümü özetlenir.

Eğitmene Öneriler: Eğitmen problemi padlet ortamına yükleyebilir. Öğrenciler sadece ikinci etapta çalışma odalarına gönderilir. Bireysel çalışmalar sırasında odalar oluşturulmayacaktır. Dört kişilik grup kodlama ekibini oluşturduğunda, turuncu çerçeve içindeki kod satırlarına grup 1 (moderatörler), yeşil çerçeve içindeki kod satırlarına ise grup 2 öğrencilerinin bireysel olarak erişmesi beklenir. Kodlama ekibi ise turuncu ve sarı kodları birleştiren diğer kod satırları ile birlikte yeşil kod satırlarını tamamlamalıdır. Temel problemin kodlarını aşağıda bulabilirsiniz.

```

1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int i, j, satir=3, sutun=3, matris1[3][3], matris2[3][3],
        sonuc[3][3];
6.     for(i = 0; i < satir; i++) {
7.         for(j = 0; j < sutun; j++) {
8.             cout << "1. Matrisin [" <<i<< ", " <<j<<"]. elemanini
        giriniz: ";
9.             cin >> matris1[i][j];
10.        }
11.    }
12.    cout << endl;
13.    for(i = 0; i < satir; i++) {
14.        for(j = 0; j < sutun; j++) {
15.            cout << "2. Matrisin [" <<i<< ", " <<j<<"]. elemanini
        giriniz: ";
16.            cin >> matris2[i][j];
17.        }
18.    }
19.    cout<<"\nSonuc Matrisi: " << endl;
20.    for(i = 0; i < satir; i++) {
21.        for(j = 0; j < sutun; j++) {
22.            sonuc[i][j] = matris1[i][j] + matris2[i][j];
23.            cout << sonuc[i][j] << " ";
24.        }
25.        cout << endl;
26.    }
27.    return 0;
28. }
29.

```

Farkı Keşfedelim

Süre: 10 dk.

Kazanımlar: K6. Diziler ve katarlar arasındaki farkı anlar.

Materyaller: Hafta 7 Sunum

Hazırlık: Eğitimci Hafta 7 sunumunda “Farkı Keşfedelim” başlıklı slaytı kullanır.

Uygulama: Bu etkinlikte eğitimci sunum üzerinden yeni bilginin sunumunu yapar. Sunumda “Farkı Keşfedelim” başlıklı slayt üzerinde verilen örnek kod öğrencilerle birlikte incelenir. Eğitimci kod satırlarında yer alan katar ve dizi komutlarına dikkat çeker. Bu komutların çıktı üzerinde nasıl bir farklılık oluşturduğunu düşünmelerini ister. Bununla ilgili fark olduğunu düşünen öğrencilerden kullanıcı durumlarını gülen yüz, olmadığını düşünen öğrencilerden ise kullanıcı durumlarını ağlayan yüz olarak değiştirmeleri istenir. Bu şekilde öğrencilerden performans yansıtılmaları beklenir. Daha sonra eğitimci ağlayan yüze dönen bir öğrenciye söz hakkı tanır ve iki komut arasında fark olmadığını nasıl karar verdiğini sorar. Daha sonra bu öğrenciden gülen yüze dönen

arkadaşlarından birine söz hakkı tanıyıp aynı sorunun tersini sorması istenir. Bu şekilde iki farklı grubun birbirini çürütmesi beklenir. Eğitimci öğrenci yanıtlarına bağlı olarak aradaki farkı açıklar ve konuyu özetler.

Eğitmene Öneriler: Eğitimci aradaki farkı açıklarken aşağıdaki içerikten yararlanabilir.

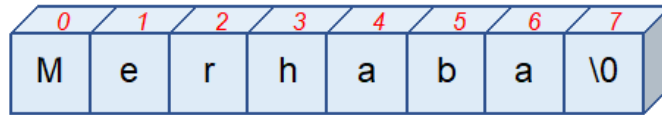
Programlamada metin türünde verilerimizi saklamak için kullanılan özel karakter dizileridir. Katarlar, null ('\0') karakter ile sonlandırılmış tek boyutlu karakter dizileri olarak tanımlanabilir. Aşağıda verilen örnekte, "Merhaba" sözcüğünden oluşan bir katar oluşturulmaktadır. Normalde verilen kelime 7 harften oluşsa da sondaki null karakteri tutmak için de bir karakterlik alan gerektiği için bellekte toplamda 8 karakterlik alan ayrılması gerekmektedir.

char kelime[] = {'M', 'e', 'r', 'h', 'a', 'b', 'a', '\0'};

Dizilerdeki ilk değer atama yöntemlerini hatırlarsanız aşağıdaki gibi bir ilk değer ataması yapabiliriz. Dizi değişkeni, çift tırnak işareti içine alınmış bir karakter dizisi içerir.

char kelime[] = "Merhaba";

Aslında yukarıdaki tanımlamada gördüğünüz üzere null karakteri bir katar sabitinin sonuna yerleştirmesiniz. C++ derleyicisi diziyi oluşturduğunda '\0' değerini dizinin sonuna otomatik olarak ekler. Yukarıdaki tanımlama sonucu bellekte şu şekilde bir yerleşim söz konusu olacaktır.



C. Sonuç

Süre: 10 dk.

Materyal: EK 4 Hafta 7 Süreli Ödev

Hazırlık: EK 4 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta "Nesneler" konusunun işleneceği hakkında bilgi verilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. EK 4 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve

sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitmen görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Tasarlayıcı: Fadime, 10 arkadaşının Facebook beğeni sayılarının toplamını ve beğeni ortalamalarını merak etmektedir. Bu amaçla bir program yazmak ister, ancak yardıma ihtiyacı vardır. Fadime için bu programı sen tasarlar mısın?

```
#include <iostream>
using namespace std;
int main() {
    int sayilar[] = {17, 13, 12, 9, 6, 11, 3, 14, 2, 19};
    int toplam = 0, n = 10;
    float ort;
    cout << "Dizi: " << endl;
    for (int i=0; i < n; i++) {
        cout << sayilar [i] << " ";
        toplam += sayilar[i];
    }
    cout << "\nDizinin toplami: " << toplam << endl;
    ort = (float)toplam / n;
    cout << "Dizinin ortalamasi: " << ort << endl;
    return 0;
}
```

Analizci: Arkadaşın (bilgisayar) aklından 1-9 arasında rastgele bir sayı tutar. Sen de tutulan bu sayıyı 3 tahminde bulmaya çalışan bir program yazıyorsun. Kural gereği tutulan sayıyı 3 tahminde bulamazsan oyun sona erer. Eğer 3 tahminden birinde sayıyı bulursan program tutulan sayıyı kaçınıcı tahmin hakkında bulduğunu ekrana yazdırır.

```
#include <iostream>
#include <ctime>
using namespace std;
int main()
{
    int sayi;
    int tahmin = -1;
    int tahmin_sayisi = 0;
    int tahmin_limiti = 3;
    bool outOfGuesses = false;
    srand(time(NULL));

    sayi = rand() % 9 + 1;
    cout << sayi;
    while(tahmin != sayi && tahmin_sayisi < tahmin_limiti){
        cout << "Tahmininizi girin: ";
        cin >> tahmin;
        tahmin_sayisi++;
    }
    if(tahmin == sayi){
        cout << "Tebrikler, " << tahmin_sayisi << ". denemede kazandınız!" << endl;
    } else {
        cout << "Uzgunum, 3 hakkınızda bilemediniz!" << endl;
    }
    return 0;
}
```

Kodlayıcı: Aşağıdaki matris yapısında voleybol oyuncularının numaraları verilmektedir. Koç, oyuncuları iki maç öncesi aşağıdaki gibi aynı sırada görmek istiyor. İki maçta sıralamanın aynı olduğundan emin olmak için bir kod tasarlamayı düşünüyor, ancak yardıma ihtiyacı var. Koç için bu programı sen tasarlar mısın? Program içerisinde tanımlama bölümünde oyuncuların ilk dizilimi aşağıdaki matristeki gibi olmalıdır.

1. Maç Oyuncu Sırası:

1	3	5	7	9
11	13	15	17	19
2	4	6	8	10
12	14	16	18	20

2. Maç Oyuncu Sırası:

1	3	5	7	9
11	13	15	17	19
2	4	6	8	10
12	14	16	18	20

```
#include <iostream>
using namespace std;
int main()
{
    int matris1[4][5] = { {1, 3, 5, 7, 9},
                          {11, 13, 15, 17, 19},
                          {2, 4, 6, 8, 10},
                          {12, 14, 16, 18, 20}};

    int matris2[4][5] = { {1, 3, 5, 7, 9},
                          {11, 13, 15, 17, 19},
                          {2, 4, 6, 8, 10},
                          {12, 14, 16, 18, 20}};

    bool durum = true;
    int i, j;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 5; j++)
            if (matris1[i][j] != matris2[i][j])
                durum = false;

    if (durum)
        cout << "Sıralama aynı";
    else
        cout << "Sıralama farklı";
    return 0;
}
```

Hafta 7. Ders Materyalleri

EK 1. Dizileri Tanıyalım

Talimat 1

4 elemanlı tek boyutlu dizi



0

1

2

3

Talimat 2

4*3'lük
12 elemanlı 3 boyutlu bir dizi



0,0

0,1

0,2

0,3



1,0

1,1

1,2

1,3



2,0

2,1

2,2

2,3

EK 2. Grup Görevleri

Problem: 5 arkadaşın yaş bilgilerini değişkende saklama ve ekrana yazdırma

```
#include <iostream>
using namespace std;
int main()
{
    int yas1 = 15;
    int yas2 = 14;
    int yas3 = 17;
    int yas4 = 12;
    int yas5 = 16;

    cout << "1. arkadasimin yasi: " << yas1 <<
endl;
    cout << "2. arkadasimin yasi: " << yas2 <<
endl;
    cout << "3. arkadasimin yasi: " << yas3 <<
endl;
    cout << "4. arkadasimin yasi: " << yas4 <<
endl;
    cout << "5. arkadasimin yasi: " << yas5 <<
endl;
    return 0;
}
```

Kodun Çıktısı:

```
1. arkadasimin yasi:
15
2. arkadasimin yasi:
14
3. arkadasimin yasi:
17
4. arkadasimin yasi:
12
5. arkadasimin yasi:
16
```

Görev 1: Programı, kod çıktısı aynı olacak şekilde döngü ve tek boyutlu dizi kullanarak yeniden kodlayınız. Kod satırlarının ekran görüntüsünü grup olarak paylaşınız.

Görev 2: Programı, kod çıktısı aynı olacak şekilde döngü ve iki boyutlu dizi kullanarak yeniden kodlayınız. Kod satırlarının ekran görüntüsünü grup olarak paylaşınız.

EK 3. Kod Satırlarını Tamamla

```

#include <iostream>
using namespace std;
int main(){
    int dizi1[5], dizi2[5], dizi3[5];
    int i;
    for(i=0; i<5;i++){
        cout << "1. Dizinin " << (i+1) << ". elemanini giriniz: ";
        cin >> dizi1[i];
    }
    cout << endl;
    for(i=0; i<5;i++){
        cout << "2. Dizinin " << (i+1) << ". elemanini giriniz: ";
        cin >> dizi2[i];
    }
    cout << endl;
    .....
    return 0;
}

```

Görev Kodu 1: Program iki dizinin karşılıklı olarak elemanlarını toplayıp, yeni bir diziye kaydetmeyi amaçlamaktadır.

```

#include <iostream>
using namespace std;
int main()
{
    int sayilar[] = {19, 11, 21, 13, 15};
    int i, maks, yer, n = 5;
    cout << "Dizi: ";
    for(i=0; i < n; i++)
        cout << sayilar[i] << " ";
    maks = sayilar[0];
    yer = 0;
    .....
}
cout << "\nDizinin en buyuk elemani " << yer << ". indisteki " << maks;
return 0;
}

```

Görev Kodu 2: Program dizideki en büyük elemanı ve bu elemanın indis numarasını yazdırmayı amaçlamaktadır.

```
#include <iostream>
using namespace std;
int main()
{
    int dizi[] = {12, 67, 78, 45, 78, 78, 32, 16, 16, 57};
    int i, j, n = 10;
    cout << "Dizi: ";
    for(i = 0; i < n; i++)
        cout << dizi[i] << " ";
    cout << "\nTekrar eden elemanlar: ";
    .....
    return 0;
}
```

Görev Kodu 3: Program dizide tekrar eden elemanları bulmayı amaçlamaktadır.

```
#include <iostream>
using namespace std;
int main()
{
    int dizi[100];
    int i, boyut, tek=0, cift=0;
    cout << "Dizi boyutunu girin: ";
    cin >> boyut;
    cout<<"\nDizi elemanlarini girin:\n";
    for(i=0; i<boyut; i++){
        cout << "Elemani girin dizi[" << i << "]: ";
        cin >> dizi[i];
    }
    .....
    cout << "\nCift eleman sayisi: " << cift;
    cout << "\nTek eleman sayisi: " << tek;
    return 0;
}
```

Görev Kodu 4: Program girilen dizideki tek ve çift sayıların kaç adet olduğunu ayrı ayrı ekrana yazdırmayı amaçlamaktadır.

```
#include <iostream>
```

```

using namespace std;
int main()
{
    int sayilar[100];
    int i, n, bol5=0;
    cout << "Dizi boyutunu girin: ";
    cin >> n;
    cout<<"\nDizi elemanlarini girin:\n";
    for(i=0; i<n; i++){
        cout << "Elemani girin dizi[" << i << "]: ";
        cin >> sayilar[i];
    }
    .....
    cout << "\n5 ile bolunebilen eleman sayisi: " << bol5;
    return 0;
}

```

Görev Kodu 5: Program girilen dizideki 5 ile bölünebilen sayıların kaç tane olduğunu ekrana yazdırmayı amaçlamaktadır.

EK 4. Hafta 7 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 8. Fonksiyonlar

Kazanımlar

- K1. C++ programlama dilinde fonksiyon tanımlayabilir.
- K2. C++ programlama dilinde fonksiyon oluşturmayı bilir.
- K3. C++ programlama dilinde fonksiyon çağırmaı bilir.

Haftanın Amacı ve Materyaller

Bu haftanın amacı öğrencilerin programlamada fonksiyon oluşturmalarını, fonksiyon kullanabilmelerini sağlamaktır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız.](#)

Önerilen Ders Akışı (30 + 50 + 70+ 50 dk.)

GİRİŞ (10 dk.)

GELİŞME

Fonksiyonları Tanıyalım (20 dk.)

EK 1. Fonksiyonların Özelliklerini Keşfediyorum

Ders Arası (10 dk.)

Fonksiyonların Nasıl Kullanıldığını Keşfediyorum (50 dk.)

EK 2. Fonksiyonların Kullanım Afişi

EK 3. C++ Programında Fonksiyon Tanımlama Görevleri

Ders arası (10 dk.)

Fonksiyonları Kullanarak Kodlama Yapalım (40 dk. + 10 dk. (Ara) + 20 dk.)

EK 4. Fonksiyonları Kullanarak Kodlama Yapma

Ders arası (10 dk.)

SONUÇ (10 dk.)

EK 5. Hafta 8 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 8 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 8 sunumunu ekran paylaşımına açar. Eğitmen bir önceki hafta işledikleri konu olan diziler ve katarlar hakkında özet geçer ve öğrencilerin kazandıkları rozetleri sorar. Rozet görevleri ve karar yapıları üzerinden *ön bilgileri hatırlatır* ve sekizinci haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

Eğitmene Öneriler: Verilen süreli ödevlerde sorunla karşılaşmış öğrencilere eğitmen söz hakkı verebilir. Benzer hatalar ile karşılaşmış ve sorunu çözmüş öğrencilere de söz hakkı tanınır. Aksi durumda eğitmen hatanın nasıl giderileceği konusunda geri bildirimde bulunur. Bu şekilde diğer öğrencilerin de benzer sorunlara karşı çözüm üretmeleri sağlanmış olur. Zamana bağlı olarak eğitmen öğrencilere sağlayacağı bu desteği ders dışında sorunla karşılaşan öğrencilere yardımcı olarak da tamamlayabilir.

B. Gelişme

Fonksiyonları Tanıyalım

Süre: 20 dk.

Kazanımlar: K1. C++ programlama dilinde fonksiyon tanımlayabilir.

Materyaller: Hafta 8 Sunum

EK 1. Fonksiyonların Özelliklerini Keşfediyorum

Hazırlık: Hafta 8 Sunumunda “Fonksiyon Görevlerini Keşfetme” adlı slaytlar sırası geldikçe açılır.

Uygulama: Öğitmen bu ders için hazırlanan beş görevi sunum üzerinden ikişer dakika arayla teker teker öğrencilerine gösterir. Öğitmen her bir slayttaki ayrı ayrı görevler için öğrencilere şu şekilde soru sorar: Sırasıyla göstereceğim beş slaytta fonksiyonların ne işe yaradıklarını ve özelliklerini anlatan; günlük yaşamla ilişkisi kurularak keşfedebileceğiniz örnekler var. Şimdi herkesin her bir örnekten yola çıkarak fonksiyonların ne gibi özellikleri olabileceğine yönelik sohbet kısmından bir şeyler yazmasını istiyorum. Her slayttaki örneği anlamak ve yazmak için zamanımız 2 dk. Daha sonra gruplardan gelen fonksiyonlarla ilgili her bir özelliği beyin fırtınası yoluyla sunum üzerinden özetleyeceğiz.

Not: Her slayt değişiminde öğretmen sizce bu görevi yapacak bir fonksiyonun özelliği ne olabilir şeklinde sorar. Bazı öğrencilerden sohbet kısmından gelecek doğru yanıtları öğrencilerle paylaşarak, evet isimli arkadaşımız fonksiyonların program takibini kolaylaştıracağını ifade etti şeklinde doğru yanıtları seçerek diğer öğrenciler içinde ipucu verir. Eğer sohbet kısmında konu ile ilgili olmayan yanıtlar gelmesi durumunda öğretmen bazı fonksiyon özelliklerini ipucu kullanarak öğrencilerin keşfetmesine olanak verir.

Öğitmen etkinlik sonunda fonksiyonlarla ilgili özellikleri öğrencilerle birlikte aşağıdaki gibi özetlemeye çalışır.

Bir işi alt parçalara bölmek, hem işin takibini kolaylaştırır hem de aynı işi iki kez yapılmasını engeller. Bilgisayar programlamada da bu böyledir. Birden fazla kez kullanılacak işler bir çatı altında fonksiyon yazılarak toplanır. Bu sayede;

1. Program takibi kolaylaşır.
2. Hata çözümü kolaylaşır.
3. Tek noktadan değişiklik yapılır.

Fonksiyonlar işleri bölerek daha az satır kod yazmamızı sağlar. Örneğin; Yazdığımız programın 10 farklı noktasında 5 satırlık işlem yaptırmamız gerekiyor. Eğer bunun için fonksiyon kullanmazsak 50 satır kod oluşacaktır. Fonksiyon kullanarak satır sayısı 15'e düşecektir. 5 satır fonksiyon için 10 satır da fonksiyon çağırma için kullanılacaktır.

Eğitime Öneriler:

Yukarıdaki etkinlik bitimiyle beraber öğrencilerden gelen yanıtlar neticesinde fonksiyonların özellikleri aşağıdaki gibi çıkarılmaya çalışılır. Biz yazılımcılar programlarımız da fonksiyonları kullanarak;

1. Daha kolay hatalarımızı bulabiliriz. (Materyaldeki dördüncü örnek.)
2. Daha doğru çözüm üretebiliriz. (Materyaldeki beşinci örnek.)
3. İhtiyaç duyduğumuz anda belli bir görevi yapması için çağırırız. (Materyaldeki birinci örnek.)
4. İhtiyaç duyduğumuz anda belli bir görevi yapması için çağırırız ve kullanılmasını sağlarız (Materyaldeki üçüncü örnek.)
5. Daha az satır kod yazarız ve programın yönetimi kolaylaştırırız. (Materyaldeki ikinci ve beşinci örnek)

6. İstenilen yerlerde kullanılıyorlar ve kod tekrarını önüyorlar. (Materyaldeki ikinci ve beşinci örnek)

Fonksiyonların Nasıl Kullanıldığını Keşfediyorum

Süre: 50 dk.

Kazanımlar: K2. C++ programlama dilinde fonksiyon oluşturmayı bilir.

Materyaller: Hafta 8 Sunum

EK 2. Fonksiyonların Kullanım Afişi

EK 3. C++ Programında Fonksiyon Tanımlama Görevleri

Hazırlık: Eğitimci sunum üzerinden fonksiyonların kullanımı adlı sunumu veya ekte bulunan afişin yüksek çözünürlüklü halini açar. Kodlama yapılacağı için Code::Blocks programı açılır. Ayrıca öğrencilerden yazdıkları kodların ekran görüntüsü isteneceği için eğitimcinin padlet oluşturması ve linkini öğrencilerle paylaşması gerekmektedir.

Uygulama: Eğitimci fonksiyon kullanımı için hazırlanan afiş ekran paylaşarak gösterir ve fonksiyonlar hakkında aşağıdaki gibi bir özetleme yapar.

Eğitime Öneriler: Yukarıdaki etkinlikleri öğrenciler bitirdikten sonra fonksiyonlar aşağıdaki gibi özetlenir.

C++ programlama dilinde fonksiyon yazmak için üç kısım vardır. Bunlardan ilki geriye döndürülecek değişkenin tipi (dönüş tipi, Eğer geriye değer dönmüyor ise **void** olarak belirtilir.), İkinci olarak fonksiyonun ismi, son olarak fonksiyon içerisinde ihtiyaç duyulan bilgilere parametre denir. Fonksiyon işlemlerini tamamladıktan sonra, eğer çağrıldığı yere değer döndürecek ise **return** komutu ile belirtilir.

Instagram'daki her resmin bir numarası vardır bu numarayı kullanarak işlemler yapılır. Bu numarayı bir fonksiyona parametre olarak göndeririz. Bir fotoğrafa yorum göndermek için fonksiyon yazarsak; fonksiyonun ismi: yorum_yap, alacağı değer: yorum metni, geriye de işlemin başarılı olup olmadığı döndürülür.

```
dönüş_tipi fonksiyon_ismi(parametreler)
{
    Yapılacak işlemler
    return dönüş değeri;
}
```

Fonksiyonları amacını belirtecek şekilde isimlendirmeye özen gösterilmelidir. Çünkü başka biri fonksiyonu kullanmak istediğinde amacını kolayca anlayabilmelidir. Örneğin verilen sayıların ortalamasını alan bir fonksiyon yazıyorsak ismini ortalama_al veya ortalamaAl şeklinde belirtebiliriz. Benzer şekilde kullanıcılara mail atacak bir fonksiyon için mail_at veya mailAt şeklinde isimlendirebiliriz.

Yukarıda verilen örneklerden sonra öğrencilerden sunum üzerinden iki göreve yönelik fonksiyonu nasıl tanımlayacakları Code::Blocks üzerinden istenir. Vereceği yanıtların ekran görüntüsünü padlet üzerinden atılması istenir. Eğitimci fonksiyonları doğru tanımlayan bir öğrencinin paylaşımı üzerinden doğru yazımla ilgili bir dönüt verir. Öğrenciler tarafında doğru yanıt verilememesi durumunda eğitimci ekran paylaşımı yaparak verilen görevlerin nasıl fonksiyon olarak tanımlandığını öğrencilerine gösterir.

Eğitime Öneriler: Yukarıdaki verilen görevlerin amacı öğrencilere fonksiyonların nasıl tanımlandığına yöneliktir. Görevlerin tamamen kodlanması değil, fonksiyonların doğru bir şekilde nasıl tanımlandığını öğrencilerin anlaması önemlidir.

Fonksiyonları Kullanarak Kodlama Yapalım

Süre: 60 dk.

Kazanımlar: K1.C++ programlama dilinde fonksiyonu tanımlayabilir.

K2. C++ programlama dilinde fonksiyon oluşturmayı bilir.

K3. C++ programlama dilinde fonksiyon çağırmaı bilir.

Materyaller: Hafta 8 Sunum

EK 4 Fonksiyonları Kullanarak Kodlama Yapma

Hazırlık: Eğitimciden bu bölüm için döngüleri ile ilgili çözülmesi gereken yedi görevin bulunduğu slaytları sırası geldikçe açması beklenir. Ayrıca kodlama yapılacağı için Code::Blocks programı açılır.

Uygulama: Bu bölüm 6 görevden oluşmaktadır. Her bir görev için eğitimcinin o görev esnasında neler yapacağı aşağıda sırasıyla verilmektedir.

Görev 1: Ekranı 10 kez deneyap ardından 2 kez “Merhaba!” yazan ekrana_yaz isimli bir fonksiyon yazalım.

Yukarıdaki görev için “Görev 1” başlıklı slayt açılır. Daha sonra eğitimci Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğitimci her satır kodun ne anlama geldiğini ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 5 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eğitimci öğrencilerin kodları kendi bilgisayarlarında yazdıklarından emin olduktan ve ihtiyacı olan öğrencilere gerekli geri bildirimleri verdikten sonra bir sonraki görev ve slayda geçer. Görev 1’in cevabı aşağıdaki gibidir.

Cevap 1:

```

void ekrana_yaz()
{
    for(int i=0; i<10;i++)
    {
        cout << "Deneyap" <<endl;
    }
    for(int i=0; i<2;i++)
    {
        cout << "Merhaba!" <<endl;
    }
}

```

Fonksiyonumuzdan geriye herhangi bir bilgi dönmeyeceği için “void” yani “boş” olarak belirtiyoruz. Eğer bir geri dönüş tipi belirtirsek (int, double vs.), kesinlikle geriye o türde bir değer döndürmemiz gerekiyor. fonksiyon içerisinde ilk olarak 10 kez “Deneyap” yazdırıyoruz. ardından da 2 kez “Merhaba!” yazdırıyoruz.

Şimdi bunu sadece fonksiyonun ismini yazarak ana programdan çağıralım:

```

int main()
{
    ekrana_yaz();
}

```

Görev 2: Dikdörtgen şeklinde olan büyük bir arazi üçgensel bölgelere ayrılmak istenmektedir. Bunun içinde araziye ne kadar üçgen sığabileceğini bulmak isteyen bir yazılımcı ihtiyaç duyduğu anda çağırabileceği üçgen alanının hesaplamasına yönelik bir fonksiyon yazmak istemektedir. Yazılımcı bu üçgen alan bulma fonksiyonunu nasıl kodlaması gerekmektedir?

Yukarıdaki görev için “Görev 2” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğunu bildirir ve öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodları tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapması istenir. Eğitimci bitirip el kaldıran öğrencilerden rastgele birini seçerek ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitimci kodun nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 2’in cevabı aşağıdaki gibidir.

Cevap 2:

```
void ucgen_alan_hesapla(double taban, double yukseklik)
{
    double alan = (taban*yukseklik) / 2;
    cout << alan << endl;
}
```

Fonksiyonumuz hazır hale geldi. Artık kullanıma hazır, üçgen alanı hesaplayıp ekrana yazdıran bir fonksiyona sahibiz. Programın istediğimiz yerinde çağırıp kullanabiliriz. Tabanı 2 ve yüksekliği 4 olan bir üçgen için alan aşağıdaki gibi hesaplanır. Parametreleri doğrudan sayı olarak girebildiğimiz gibi değişken ismi de girebiliriz.

```
int main()
{
    ucgen_alan_hesapla(2,4);
}
```

Görev 3: Fonksiyona gönderilen sayı 5 ile tam bölünüyorsa ekrana “tam bölünür.” aksi durumda kalanı yazdıran fonksiyonu yazalım.

Yukarıdaki görev için “Görev 3” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğu bildirilir ve öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodların tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapılması istenir. Öğitmen bitirip el kaldıran öğrencilerden rastgele birini seçerek ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitmen kodun nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 3’ün cevabı aşağıdaki gibidir.

Cevap 3:

```
#include <iostream>
using namespace std;
void bes_ile_bolme(int sayi)
{
    if(sayi % 5 == 0)
        cout << "tam bolunur."<<endl;
    else
        cout << sayi%5<<endl;
}
```

```
int main()
{
    bes_ile_bolme(11);
    bes_ile_bolme(15);
    bes_ile_bolme(12);
}
```

Görev 4: Fonksiyona gönderilen iki sayıdan küçük olanı geriye döndüren fonksiyonu yazalım.

Yukarıdaki görev için “Görev 4” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğu bildirilir ve öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodların tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapması istenir. Eğitimci bitirip el kaldıran öğrencilerden rastgele birini seçer ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitimci kodun nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 4’ün cevabı aşağıdaki gibidir.

Cevap 4:

```
#include <iostream>
using namespace std;
int hangisi_buyuk(int sayi1, int sayi2)
{
    if(sayi1>sayi2)
        return sayi1;
    else
        return sayi2;
}
```

```
int main()
{
    int sayi = hangisi_buyuk(10,20);
    cout << sayi <<endl;

    sayi = hangisi_buyuk(22,11);
    cout << sayi <<endl;
}
```

Görev 5: Fonksiyona gönderilen tam sayı tipindeki dizinin en büyük sayısını ekrana yazan fonksiyonu yazalım.

Yukarıdaki görev için “Görev 5” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğunu bildirir ve öğrencilerden bu eksik kodları nasıl

tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk.'lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodları tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapılması istenir. Öğitmen bitirip el kaldıran öğrencilerden rastgele birini seçer ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitmen kodu nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 5'in cevabı aşağıdaki gibidir.

Cevap 5:

```
void en_buyuk(int dizi[5])
{
    int enbuyuk = dizi[0];
    for(int i=1;i<5;i++)
    {
        if(enbuyuk<dizi[i])
            enbuyuk=dizi[i];
    }
    cout << enbuyuk;
}
```

Ana programımız ise aşağıdaki şekilde olacaktır.

```
int main()
{
    int sayilar[] = {5,3,4,5,8};
    en_buyuk(sayilar);
}
```

Görev 6: İki dizi içerisindeki en büyük iki sayının toplamını bulan programı fonksiyon kullanarak yazalım.

Cevap 6:

Yukarıdaki görev için "Görev 6" başlıklı slayt açılır. Daha sonra eğitmen Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğitmen her satır kodun ne anlama geldiğini ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 10 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eğitmen öğrencilerin kodları kendi bilgisayarlarında yazdıklarından emin olduktan ve ihtiyacı olan öğrencilere gerekli geri bildirimleri verdikten sonra bir sonraki görev ve slayda geçer. Görev 6'nın cevabı aşağıdaki gibidir.

Önceki örneği sadece 1 düzenleme ile kullanabiliriz. Ekran yazdırmak yerine en büyük sayıyı ana programa göndermemiz gerekiyor. Böylece sonraki işlemlerde kullanabiliriz.

```

int en_buyuk(int dizi[5])
{
    int enbuyuk = dizi[0];

    for(int i=1;i<5;i++)
    {
        if(enbuyuk<dizi[i])
            enbuyuk=dizi[i];
    }
    return enbuyuk;
}

```

Ana programı da aşağıdaki gibi yazabiliriz. İlk olarak dizilerimizi tanımlıyoruz. Ardından dizi1'i fonksiyona gönderip en büyüğünü buluyoruz. Sonra dizi2'nin en büyük elemanını bulup ekrana yazdırıyoruz. Son olarak yapmak istediğimiz toplama işlemini gerçekleştiriyoruz.

```

int main()
{
    int dizi1[] = {5,3,4,5,8};
    int dizi2[] = {9,3,4,5,8};

    int en_buyuk_1 = en_buyuk(dizi1);
    cout << "1. dizinin en buyugu:" << en_buyuk_1 << endl;
    int en_buyuk_2 = en_buyuk(dizi2);
    cout << "2. dizinin en buyugu:" << en_buyuk_2 << endl;

    cout << en_buyuk_1 << "+" << en_buyuk_2 << "=" << en_buyuk_1 + en_buyuk_2;
}

```

Eğitmene Öneriler: Yukarıdaki etkinlikteki amaç öğrencilerin kod yazarken, fonksiyonları nerede, ne zaman, nasıl kullanılacağına yönelik onların farkındalığını artırmaktır. Yukarıdaki görevlerin tamamını öğrencilerin kendi bilgisayarlarında kodlanması beklenmektedir. Bunun için öğretmenlerin etkinlikler esnasında sürekli farklı öğrencilerinden isteyeceği ekran paylaşımı onların her etkinliğe hazır girmesi için motivasyonunun artmasını sağlayacaktır.

C. Sonuç

Süre: 10 dk.

Materyal: Hafta 8 Süreli Ödev

Hazırlık: EK 7 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta "Nesneler" konusunun işleneceği bildirilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. "Hafta 8 Süreli Ödev" adlı dosya ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitmen görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar: Görevler, öğrenciye verilecek beceri rozetleri ile isimlendirilmiştir. Her bir rozetin altında ilgili görevin yanıtları verilmektedir. Eğitmen bu yanıtları uygulama süresinin sonunda GitHub üzerinden öğrencilerle paylaşır.

Kodlayıcı: Parametre olarak gönderilen iki sayının toplamını geriye döndüren bir fonksiyonu oluşturmak isteseydiniz nasıl bir kod yazardınız?

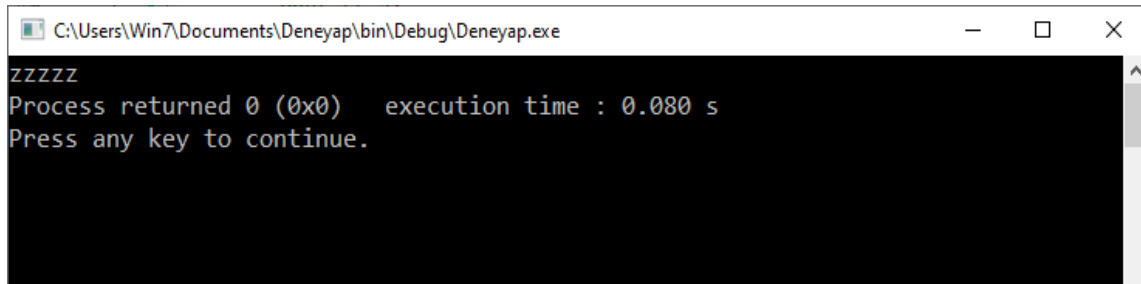
```
#include <iostream>

using namespace std;

int topla(int sayi1, int sayi2)
{
    int toplam = sayi1 + sayi2;
    return toplam;
}

int main()
{
    int sonuc = topla(5,4);
    cout<< sonuc;
}
```

Kodlayıcı: Parametre olarak gönderilen harfi, yine parametre olarak gönderilen sayı kadar ekrana yazdıran bir fonksiyon tanımlamak isteseydiniz nasıl kodlardınız?
Örnek kullanım: tekrar_yaz('z',5)



Resim 29. Ekran çıktısı

Cevap: `#include <iostream>`

```
using namespace std;
void tekrar_yaz(char a, int adet)
{
    for(int i=0;i<adet;i++)
        cout << a;
}
int main()
{
    tekrar_yaz('z',5);
}
```

Kodlayıcı: Bir kullanıcının sosyal medyada paylaştığı fotoğraflardaki toplam beğeni sayılarının bir dizide saklandığı düşünülürse; paylaşılan fotoğraflardaki beğenilerin toplamını geriye döndüren bir fonksiyonu nasıl kodlardınız?

Cevap: `#include <iostream>`

```
using namespace std;
int dizi_topla(int dizi[5])
{
    int toplam = 0;
    for(int i=0;i<5;i++)
        toplam = toplam + dizi[i];

    return toplam;
}
int main()
{
    int sayilar[5] = {5,6,9,3,2};
    int sonuc = dizi_topla(sayilar);
    cout << sonuc;
}
```

Hafta 8. Ders Materyalleri

EK 1. Fonksiyonların Özelliklerini Keşfediyorum

1

Evimize katı meyve sıkacağı alıyoruz ve biz bunu canımız her ne zaman meyve suyu çektiğinde meyve sıkacağı kullanıyoruz. Yani bu makinenin görevi biz istediğimizde meyve suyu yapmak.

2

İnstagram da milyonlarca fotoğraf paylaşan insan var ve bu fotoğraflar on binlerce insan tarafından beğeniliyor. İnstagramı yazanlar her beğenilen fotoğraf için ayrı ayrı kodlar mı yazdılar acaba.

3

Her gün sabah uyanıp ekmek almaya gitmekten yoruldum. Keşke bir yardımcı robotum olsaydı onu her çağırdığımda gelip benim yerime ekmek alsaydı.

4

Türkçe öğretmenimin bana verdiği kompozisyon ödevini 20 sayfa yazarak bitirdim. Öğretmenim "yalnız" kelimesini yanlış yazdığımı söyledi. Şimdi tüm sayfalara teker teker gidip bu yanlışları düzeltmem gerekecek. Keşke yanlış yazdığım kelimenin birisini düzelttiğimde diğer yanlışlarımda düzelseydi.

5

Kodlamak istediğim web sitesinde sisteme her giriş yapan kullanıcıya Merhaba "kullanıcının ismi yazmak istiyorum. Ne yani binlerce kişi web siteme girerse her isim için ayrı ayrı merhaba mı diyeceğim.

EK 2. Fonksiyonların Kullanım Afişi



Resim 30. Fonksiyon tanımlama afişi

EK 3. C++ Programında Fonksiyon Tanımlama Görevleri

1. Instagram’da işe başlayan bir yazılımcı, kullanıcıların fotoğraflara gönderdikleri yorumlara yönelik bir fonksiyon tanımlamak istiyor. Bu fonksiyonu siz olsaydınız nasıl tanımlardınız.

2. Dikdörtgen şeklinde olan büyük bir arazi üçgensel bölgelere ayrılmak istenmektedir. Bunun içinde araziye ne kadar üçgen sığabileceğini bulmak isteyen bir yazılımcı ihtiyaç duyduğu anda çağırabileceği üçgen alanın hesaplamasına yönelik bir fonksiyon tanımlamak istemektedir. Yazılımcı bu alan fonksiyonunu nasıl tanımlamalıdır?

EK 4. Fonksiyonları Kullanarak Kodlama Yapma**1**

Ekrana 10 kez deneyap ardından 2 kez "Merhaba!" yazan bir ekrana_yaz isimli bir fonksiyon yazalım.

2

Dikdörtgen şeklinde olan büyük bir arazi üçgensel bölgelere ayrılmak istenmektedir. Bunun içinde araziye ne kadar üçgen sığabileceğini bulmak isteyen bir yazılımcı ihtiyaç duyduğu anda çağırabileceği üçgen alanının hesaplamasına yönelik bir fonksiyon yazmak istemektedir. Yazılımcı bu üçgen alan bulma fonksiyonunu nasıl kodlaması gerekmektedir?

3

Fonksiyona gönderilen sayı 5 ile tam bölünüyor ise ekrana "tam bölünür." aksi durumda kalanı yazdıran fonksiyonu yazalım.

4

Fonksiyona gönderilen iki sayıdan küçük olanı geriye döndüren fonksiyonu yazalım.

5

Fonksiyona gönderilen tam sayı tipindeki dizinin en büyük sayısını ekrana yazan fonksiyonu yazalım. (Instagram üzerindeki en fazla beğeni alan fotoğraf)

6

İki dizi içerisindeki en büyük iki sayının toplamını bulan fonksiyonu yazalım.

EK 5. Hafta 8 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 9. Nesneler

Kazanımlar

- K1. C++ programlama dilinde nesne yönelimli programlama mantığını açıklar.
- K2. Nesne yönelimli programlamayı prosedürel programlamadan ayırt eder.
- K3. Nesne yönelimli programlamanın avantaj ve dezavantajlarını ayırt eder.
- K4. Nesne ve sınıf kavramlarını analiz eder.
- K5. Nesne yönelimli programlamada erişim belirteçlerini analiz eder.
- K6. Günlük hayatta karşılaştığı problemlerle ilgili fonksiyon oluşturma işlemini gerçekleştirir.
- K7. C++ programlamada sınıf ve nesne tanımlamasını farklı durumlarda uygular.

Haftanın Amacı ve Materyaller

Bu haftanın amacı, nesne yönelimli programlamanın temel felsefesini, yordamsal programlamadan farkını, avantaj ve dezavantajları ile nesne yönelimli programlamada sınıf, nesne ve fonksiyon tanımlama işlemlerini uygulamaktır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız](#).

Önerilen Ders Akışı (50 + 45 + 55 + 50 dk.)

GİRİŞ (5 dk.)

GELİŞME

Nesneni Çiz (45 dk.)

Ders Arası (10 dk.)

Sınıfının Özelliklerini Tanı (45 dk.)

EK 1. Grup Afişleri

Ders Arası (10 dk.)

Afişini Yeniden Tasarla (45 dk.)

EK 2. Afiş Tasarım Şablonu

SONUÇ (10 dk.)

EK 3. Hafta 9 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Süre: 5 dk.

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 9 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken, *dikkat çekmek için* eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 9 sunumunu ekran paylaşımına açar. Eğitmen bir önceki hafta işledikleri konu olan fonksiyonlar hakkında özet geçer ve öğrencilerin kazandıkları rozetleri sorar. Rozet görevleri ve döngüler üzerinden *ön bilgileri hatırlatır* ve dokuzuncu haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

B. Gelişme

Nesneni Çiz

Süre: 45 dk.

Kazanımlar: K1. C++ programlama dilinde nesne yönelimli programlama mantığını açıklar.

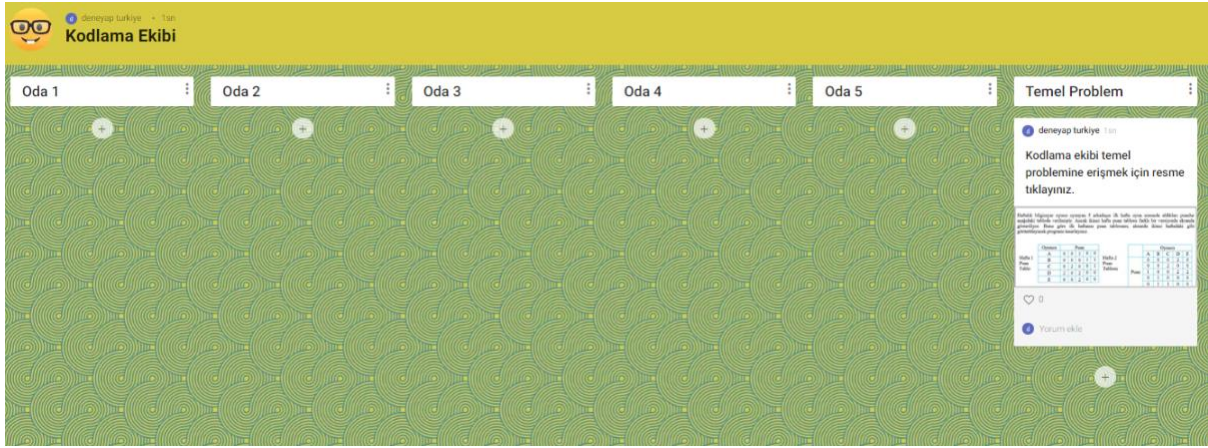
K2. Nesne yönelimli programlamayı prosedürel programlamadan ayırt eder.

K3. Nesne yönelimli programlamanın avantaj ve dezavantajlarını ayırt eder.

Materyaller: [Çarkı Döndür Oyunu](#)

Dijital Tartışma Panosu: Grup Çalışması

Hazırlık: Eğitmen derse girmeden önce beş farklı grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletecektir. Eğitmen padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. Örnek bir tasarım Resim 31’de gösterilmiştir.

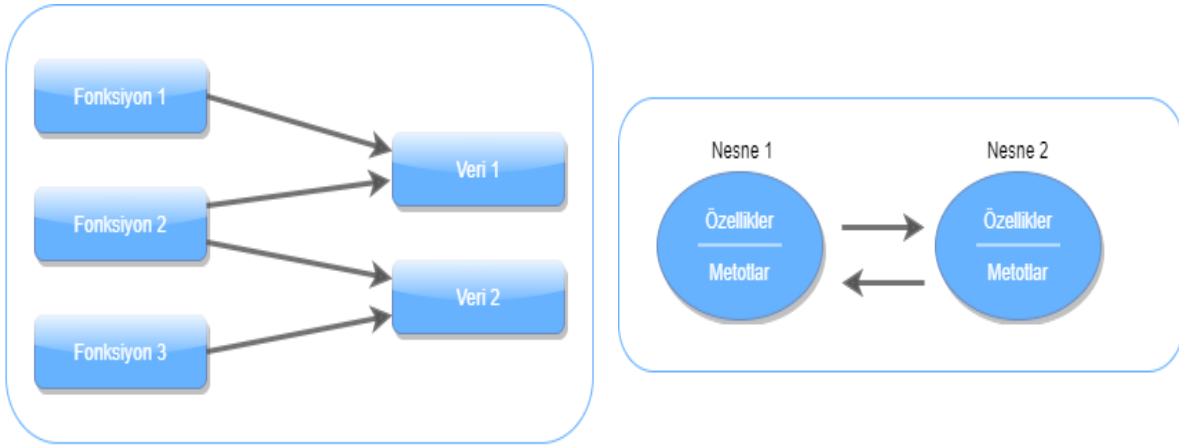


Resim 31. Nesneni Çiz: Grup Çalışması Örnek Padlet Tablosu

Uygulama: Öğretmen öğrencilere konu öncesi çalışma odalarına dağıtarak bir oyun oynayacaklarını söyler. Hafta 9 sunumunda “Nesneni Çiz” başlıklı slaytı kullanarak oyun talimatlarını verir. Gruplara dağıldıktan sonra öğrencilerden bir Web 2.0 aracı olan wordwall ile oluşturulan çarkı döndürerek bir seçim yapmaları istenir. “Meyve, Taşıt, Hayvan, Mobilya ve Kıyafet” olmak üzere beş farklı sınıfın bulunduğu “Çarkı Döndür” oyunu için <https://wordwall.net/resource/18287257> linki öğrencilerle paylaşılan notlar üzerinden iletilir. Paylaşılan notlar kısmından ayrıca oyun sonunda tahminlerin iletileceği dijital tartışma panosunun linki paylaşılır. Öğrenciler çalışma odalarına 15 dk. süre için gönderilir. Grup üyelerinden biri bu çarkı ekran paylaşımı yaparak diğer üyeler için çevirebilir. Seçimlerini belirleyen grup üyeleri, örneğin “Meyve” sınıf kartına ait “elma, kiraz, muz” vb. bir nesne düşünür ve bu nesneyi bir kâğıda çizer. Tüm grup üyeleri çizimini bitirince, sırayla birbirlerine çizimlerini tahmin edebilecekleri sorular sorarlar. Bu sorular rengi ne renk? canlı mı cansız mı? eşya mı? yenilebilir mi? tarzında nesne hakkında çıkarımda bulunabilecekleri türde olmalıdır. Grup üyeleri bu şekilde arkadaşlarının çizdiği nesneyi tahmin etmeye çalışır. Doğru tahmin edilen nesneye ait çizimi grup üyesi, dijital panoya yükler ve tüm arkadaşların görmesini sağlar. Her bir grup kendi üyelerinin çizimlerini tahmin etme süresini 1 dk. olarak belirler. Bu şekilde tüm üyelerin süre içerisinde tahminleri alınmaya çalışılır. Böylece tüm grup üyelerinin çizimlerinin tahmin süresi bittikten sonra, grupların çizim yaptıkları sınıfı dijital panoda grup altında yazmaları ve doğru tahmin edilen nesne çizimlerini dijital panoya iletmeleri istenir. Ana odaya döndükten sonra, öğretmen dijital panoyu ekran paylaşımı yapar ve en çok doğru tahmin çizimini panoya ileten grubu kazanan grup ilan eder. Oyun sonunda öğretmen sınıf ve nesne arasındaki farkı Hafta 9 sunumunu kullanarak özetler. Öğretmen özet yaparken, oyunda birinci gelen grubun çizim yaptığı sınıf ve nesnelerden örnekler verir. Buradan yola çıkarak nesne tabanlı programlamanın önemi, avantaj ve dezavantajları hakkında açıklama yaparak konuyu özetler.

Eğitime Öneriler: Öğretmen konu özeti için aşağıdaki bilgileri kullanabilir.

Konu İçeriği: C++ programlama dilinin amacı, kendi başına en güçlü programlama dillerinden biri olan C programlama diline nesne yönelimi eklemektir. Nesne yönelimli programlama, programları nesnelerin sınıfları açısından tanımlayan bir yazılım tasarım ve programlama yöntemidir. Aşağıdaki şekilde görüldüğü üzere yordamsal (prosedürel) programlama dillerinde veri ve fonksiyonlar birbirinden ayrı iken, nesne yönelimli programlama, veri ve fonksiyonları birleştirerek aralarındaki görevleri gerçekleştirmek için düzenli iletişim sağlayan nesneler kümesi şeklindedir.



Resim 32. Yordamsal ve nesne yönelimli programlama

Yordamsal programlama, veriler üzerinde işlemler gerçekleştiren fonksiyonlar yazmakla ilgiliyken, nesne yönelimli programlama ise hem verileri (özellikler) hem de fonksiyonları içeren nesneler oluşturmakla ilgilidir. Nesne yönelimli programlamanın, yordamsal programlamaya göre birçok avantajı vardır:

- ❖ Daha hızlıdır ve uygulanması daha kolaydır.
- ❖ Programlar için net bir yapı sağlar.
- ❖ C++ kodunun tekrar edilmemesine yardımcı olur ve kodun bakımını, değiştirilmesini ve hata ayıklamasını kolaylaştırır.
- ❖ Daha az kod ve daha kısa geliştirme süresiyle tam yeniden kullanılabilir uygulamalar oluşturmayı mümkün kılar.

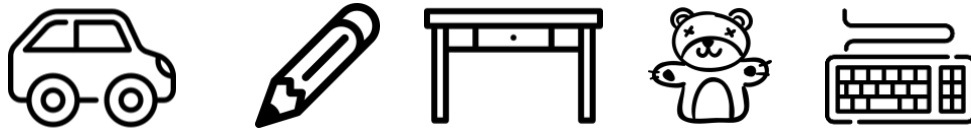
Tablo 21. Prosedürel ve nesne yönelimli programlama farkı

Prosedürel Programlama	Nesne Yönelimli Programlama
Program, fonksiyon adı verilen küçük parçalara bölünmüştür.	Program, nesneler adı verilen küçük parçalara ayrılır.
Yeni veri ve fonksiyon eklemek kolay değildir.	Yeni veri ve fonksiyon eklemek koladır.
Verileri gizlemek için uygun bir yol yoktur, bu nedenle daha az güvenlidir.	Daha güvenli olmak için veri gizleme sağlar.
Fonksiyon veriden daha önemlidir.	Veri, fonksiyondan daha önemlidir.
Gerçek olmayan dünyaya dayanır.	Gerçek dünyaya dayanır.
Örnekler: C, Fortran, Pascal, Basic vb.	Örnekler: C++, Java, Python, C# vb.

C++ modülleri tasarlarırken, tüm dünyayı nesne şeklinde görmeye çalışıyoruz. Örneğin araba; renk, kapı sayısı, hız limiti gibi belirli özelliklere sahip bir nesnedir. Farklı isimler ve markalara sahip birçok araba olabilir, ancak hepsi bazı ortak özellikleri paylaşmaktadır. Ayrıca, hızlanma ve yavaşlama gibi belirli yöntemlere de sahiptir.

Nesne (Object) Kavramı

Nesne yönelimli programlamanın temel birimi olan, bazı özellikleri ve davranışları olan tanımlanabilir bir varlıktır. Yani veriler üzerinde çalışan fonksiyonlar nesne olarak adlandırılan bir birim olarak oluşturulur. Örneğin: araba, kalem, masa, oyuncak ve klavye gibi.

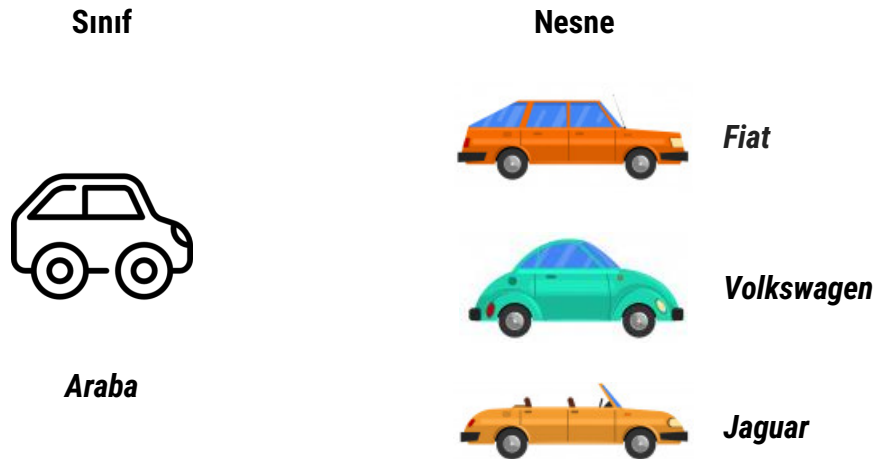


Resim 33. Nesne örnekleri

Nesne soyut bir veri örneğidir. Sınıf tanımlandığında, bellek ayrılmaz, ancak örnek oluşturulduğunda (yani bir nesne oluşturulduğunda) bellek ayrılır.

Sınıf (Class) Kavramı

Bir sınıf tanımladığınızda, nesne için bir taslak tanımlarsınız. Sınıf adının ne anlama geldiği, yani sınıftaki bir nesnenin ne içereceği ve böyle bir nesne üzerinde hangi işlemlerin gerçekleştirilebileceği tanımlanır. Araba bir sınıf ise bundan üretilen farklı araba çeşitleri de nesneleri ifade eder.



Resim 34. Sınıf ve nesne kavramı

Sınıfın Özelliklerini Tanı

Süre: 50 dk

Kazanımlar: K4. Nesne ve sınıf kavramlarını analiz eder.

K5. Nesne yönelimli programlamada erişim belirteçlerini analiz eder.

Materyaller: EK 1. Grup Afişleri

Dijital Tartışma Panosu: Grup Çalışması

Hazırlık: Eğitimci derse girmeden önce dört farklı grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletecektir. Eğitimci padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. EK 1'deki grup afişlerinden birer tane dijital panoda oluşturulan gruplar altına yüklenir. Talimat olarak her grup altına "Grup afişini birlikte inceleyiniz. Sınıf, üye listesi, erişim belirteci, fonksiyon ve nesne oluşturma kavramları hakkında tartışma yapınız. Tartışma sonundaki fikirlerinizi grup altında panoda açıklayınız." ifadesi yazılır. Örnek bir tasarım Resim 35'te gösterilmiştir.

Resim 35. Grup Afişleri Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci Hafta 9 sunumunda "Sınıfın Özelliklerini Tanı" başlıklı slaytı kullanarak talimatları verir. Öğrenciler bu etkinlikte beşerli kişiler halinde çalışmaktadır. Eğitimci her gruba grup afişlerinden birini dijital pano aracılığıyla iletir. Paylaşılan notlar kısmından padlet linkini öğrencilere iletir. Eğitimci öğrencileri çalışma odalarına 10 dk. süre ile rastgele dağıtır.

Öğrencilerin gruplar halinde bu afişleri incelemeleri ve sınıf, üye listesi, erişim belirteci, fonksiyon ve nesne oluşturma konularını aralarında tartışmaları istenir. Ana odaya döndükten sonra eğitmen dijital panoyu ekran paylaşımına açar. İlk grubun afişinin üzerine tıklayarak açar ve “Sınıf” kavramını afiş üzerinden açıklar. Daha sonra topu birinci gruptan birine sözlü olarak atar. Örneğin şimdi birinci gruptan öğrenci ismi vererek, bize sıradaki “Üye listesi” kavramını açıklasın der ve bu öğrenciye söz hakkı verir. Eğitmen bunun için gruplara giren öğrencilerin isimlerini bir kenara yazmalıdır. Daha sonra birinci gruptan söz hakkı alan öğrenci, sıradaki kavramı grup 2 afişini üzerinden açıklaması için topu ikinci gruptan bir başka öğrenciye sözlü olarak atar. Eğitmen burada ikinci grup üyelerinin isimlerini öğrenciye sayabilir. Bu şekilde ikinci gruba söz hakkı verilir ve sıradaki kavram grubun kendi afişini üzerinden öğrenciler tarafından açıklanır. Bu şekilde dört grubun da sırası tamamlanır ve tüm afişler üzerinden kavramlar tartışılmış olur. Eğitmen zaman planlaması için bir çan sesi kullanarak süreci yönetebilir. Eğitmen dijital panoyu ekran paylaşımını kapatır ve sunum üzerinden “Sınıfının Özelliklerini Tanı” başlıklı slayttan devam ederek kavramları tekrar eder.

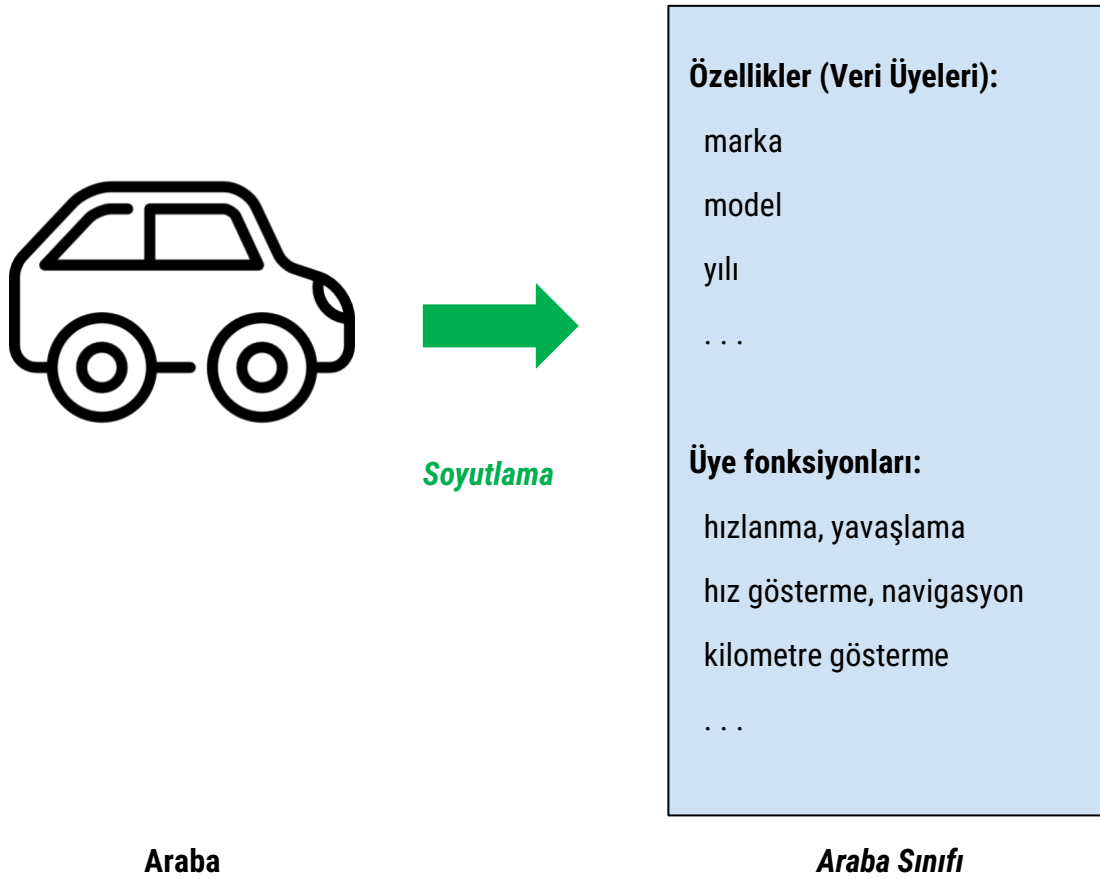
Eğitmene Öneriler: Eğitmen gruplara rastgele dağıttığı öğrencilerin listesini almalıdır. Eğitmen tartışmalar sırasında kavramlar üzerine aşağıdaki bilgilerden yararlanarak detaylar verebilir. Özellikle uyarı niteliğindeki içerikler, afiş tasarımları üzerinden öğrencilerin dikkati çekilerek anlatılabilir.

Konu İçeriği: Sınıf Tanımlama

Sınıflar, nesne yönelimli programlamanın en önemli yapı elemanıdır. Bir sınıf, bir nesnenin özelliklerini ve kapasitelerini tanımlar. Pek çok farklı sınıf tanımlı oluşturabilirsiniz. Bu bölümden itibaren artık nesne yönelimli programlama tasarımına giriş yapıyoruz. Başlangıçta biraz zorlu olacağı öngörülebilir fakat ne kadar değerli bir şey öğrendiğimizi unutmayalım. Kavramları anlamaya başladığınızda ve kod yazmayı ilerledikçe, nesne yönelimli programlamanın ilk bakışta görüldüğü kadar zor olmadığını fark edeceksiniz. İlk aşamada sınıf kavramını anlamak için bir gerçek dünya nesnesi olan arabayı düşünebiliriz.

UYARI:

Her sınıf modeline yalnızca bir kavram (nesne bütünlüğü) sunmalıdır. Örneğin, Araba sınıfını Sürücü sınıfından ayrı tutun.



Resim 36. Araba sınıfı örneği

Yukarıda gördüğünüz üzere arabaya ait bir sınıf oluşturmak istediğimiz zaman bu sınıfa ait özellikleri ve fonksiyonları belirlememiz gerekmektedir. Özellikler sınıfın ayırt edici bilgilerinden oluşan marka, model ve yılı gibi verileri içermektedir. Fonksiyonlar ise sınıfın hızlanma, yavaşlama ve hız ve kilometre gösterme gibi fonksiyonlarını içermektedir. Sınıfın tasarımı tamamen bizim elimizde olup istediğimiz olası tüm yetenekleri sınıfa ekleyebiliriz. Bir sınıfı tanımlamak için aşağıdaki gibi genel bir sözdizimi kullanılmaktadır.

```
class SınıfAdı
{
    public:
        üyeListesi
};
```

Burada **class**, sınıf tanımına ait anahtar kelime, **public** üye listesinin erişim belirteci, **üyeListesi**, sınıf üyelerinin listesi ve **SınıfAdı**, sınıfın adıdır. Sınıf adının büyük harfle başladığına dikkat edelim. Bu en yaygın kullanımdır ve kodunuzun okunabilir olması için önem arz eder. Sınıf bildiriminin fonksiyonlardaki gibi parantezle başlayıp bittiğine dikkat ediniz. Ayrıca kapanış parantezinden sonra noktalı virgül kullanıldığını unutmayınız. Noktalı virgölün unutulması derleyici tarafından yakalanacak bir hataya sebep olacaktır.

UYARI:

Bu eğitim kapsamında sınıfları ve sınıf üyelerini ayırt etmek için standart adlandırma kurallarına uyacağız. Sınıf adları büyük harfle ve üye adları küçük harfle başlatılacaktır. Bu kural sadece C++ 'da değil, aynı zamanda diğer tüm nesne yönelimli programlama dillerinde de kullanılır.

üyeListesi, üye bildirimlerini içeren listeden oluşur. Bunlar üye veri veya üye fonksiyon bildirimleri olabilir. Veri üyesi bildirimleri normal değişken bildirimleri ile aynıdır. Örneğin, sırasıyla karakter, tam sayı ve kesirli sayı türlerinde veri üyeleri oluşturmak istersek aşağıdaki tanımlamaları yaparız.

```
char a;
int b;
double c;
```

Ancak, veri üyelerini bildirdiğiniz yerde ilk değer ataması gerçekleştiremezsiniz. Değişkenlere ilk değer atamasının hazırlanan bir fonksiyonda ya da sınıfın dışında gerçekleştirilmesi gerekir. Tanımlanan bu veri üyelerinin kapsamı, sınıftan oluşturulan nesnenin kapsamı ile aynıdır. Bununla birlikte, veri üyelerine sınıfın dışından her zaman erişilebilmesi mümkün değildir. Yukarıda verdiğimiz araba sınıfında markanın, modelin ve yılın ne olduğu bu veri üyeleri tarafından saklanır. Bir sınıfın veri üyeleri, sınıf özelliklerini tanımlar ve açıklar. Bu nedenle, her bir araba nesnesinin markası, o nesnenin bir özelliğidir. Aşağıda araba sınıfına ait marka, model, yıl, renk ve fiyat veri üyelerinin sınıf içerisinde nasıl tanımlandığını görebilirsiniz.

```
class Araba
{
public:
    char marka[30];
    char model[30];
    int yıl;
    char renk[30];
    float fiyat;
};
```

Erişim Belirteçleri

Erişim belirteçleri, C++ sınıfınızın içerisinde yer alan veri üyelerine nereden erişilebileceğini kontrol etmenizi sağlar. Bu kontrol sayesinde veri üyeleri üzerinde yetkisiz olarak değişiklik yapılmasının önüne geçilmesi sağlanır. Erişim belirteci, sınıftaki veri üyelerine erişimi kontrol eden bir kelimedir. Bir erişim tanımlayıcısının sözdizimi aşağıdaki gibidir:

```
class SınıfAdı
{
    üyeListesi
    erişimBelirteci:
        üyeListesi
    erişimBelirteci:
        üyeListesi
};
```

Bir erişim belirticisi tanımlandıktan sonraki üye listesindeki tüm veriler için geçerlidir. Sınıf içerisinde başka bir erişim belirticisi tanımlanırsa bu tanımlamadan sonraki veri üyeleri bu erişim türüne ait olur. Sınıfın sonuna ulaşıncaya kadar sınıfın tüm üyelerini en son tanımlanan erişim belirticisi etkiler. Bir sınıfın sahip olabileceği farklı erişim belirteçlerinden ikisi:

1) public: Bu anahtar kelime ile tanımlanan tüm üyelere, sınıfın ulaşılabilir olduğu her yerden erişilebilir.

2) private: Bu anahtar kelime ile tanımlanan üyelere yalnızca aynı sınıfın diğer üye fonksiyonları tarafından erişilebilir.

UYARI:

Varsayılan olarak, tüm sınıf üyelerinin erişimi “private” olarak tanımlıdır. Bu nedenle, sınıf bildiriminde erişim belirticisi olmadan görünen tüm üyelerin erişimi “private” olur.

```
class Cep Telefonu
{
    char marka[30];
    int yıl;
    public:
        char model[30];
        char renk[30];
    private:
        float fiyat;
        int imei_no;
};
```

Yukarıda verilen sınıf tanımlamasında fiyat ve imei no veri üyeleri ile birlikte marka ve yıl veri üyelerinin de private olduğunu unutmayınız. Diğer taraftan model ve renk public olarak tanımlanmıştır. Sınıf tanımlamalarınızda istediğiniz kadar erişim belirteciniz olabilir, ancak belirteçleri tek bir grup altında toplamak sınıfın anlaşılabilirliğini artıracaktır.

Fonksiyon Oluşturma

Bir sınıfın üye fonksiyonu, diğer herhangi bir değişken gibi sınıf içinde tanıma sahip olan bir fonksiyondur. Üyesi olduğu sınıfın herhangi bir nesnesi üzerinde çalışır ve bu nesne için bir sınıfın

tüm üyelerine erişim sağlayabilir. Fonksiyon tanımları olmadan bir sınıfın tanımı tam olarak gerçekleştirilmiş sayılmaz. Fonksiyon tanımı yapıldıktan sonra tanımlanan bu fonksiyonlara yalnızca sınıfın bir nesnesi aracılığıyla erişilebilir.

Bir fonksiyonu iki şekilde tanımlayabilirsiniz. Birinci yol, sınıf bildirimi içinde bir fonksiyon bildirmek ve sonra onu dışarıda uygulamaktır. İkinci yol, fonksiyonu aynı zamanda sınıf bildirimi içinde bildirmek ve uygulamaktır. Genelde uygulamalarda birinci yol tercih edilir. Bir sınıf dışında gerçekleşen bir fonksiyon tanımının genel sözdizimi şu şekildedir:

```
dönüş_tipi SınıfAdı::fonksiyon_adi(parametre_listesi)
{
    fonksiyon_gövdesi
}
```

Tanımlanan fonksiyon içinde, sınıfa ait tüm üyelere adları kullanılarak erişilebilir. Sınıf üyeliği otomatik olarak sağlanır ve aynı sınıfa ait fonksiyonlar birbirini doğrudan çağırabilir. Belirli bir nesne için bir fonksiyon çağırıldığında, ilgili fonksiyon bu nesnenin verilerini işleyebilir. Artık Araba sınıfının hızlanma ve yavaşlama fonksiyonlarını uygulayabiliriz.

```
class Araba
{
public:
    char marka[30];
    char model[30];
    float fiyat;
    int hiz;

    void hizlanma();
    void yavaslama();
};

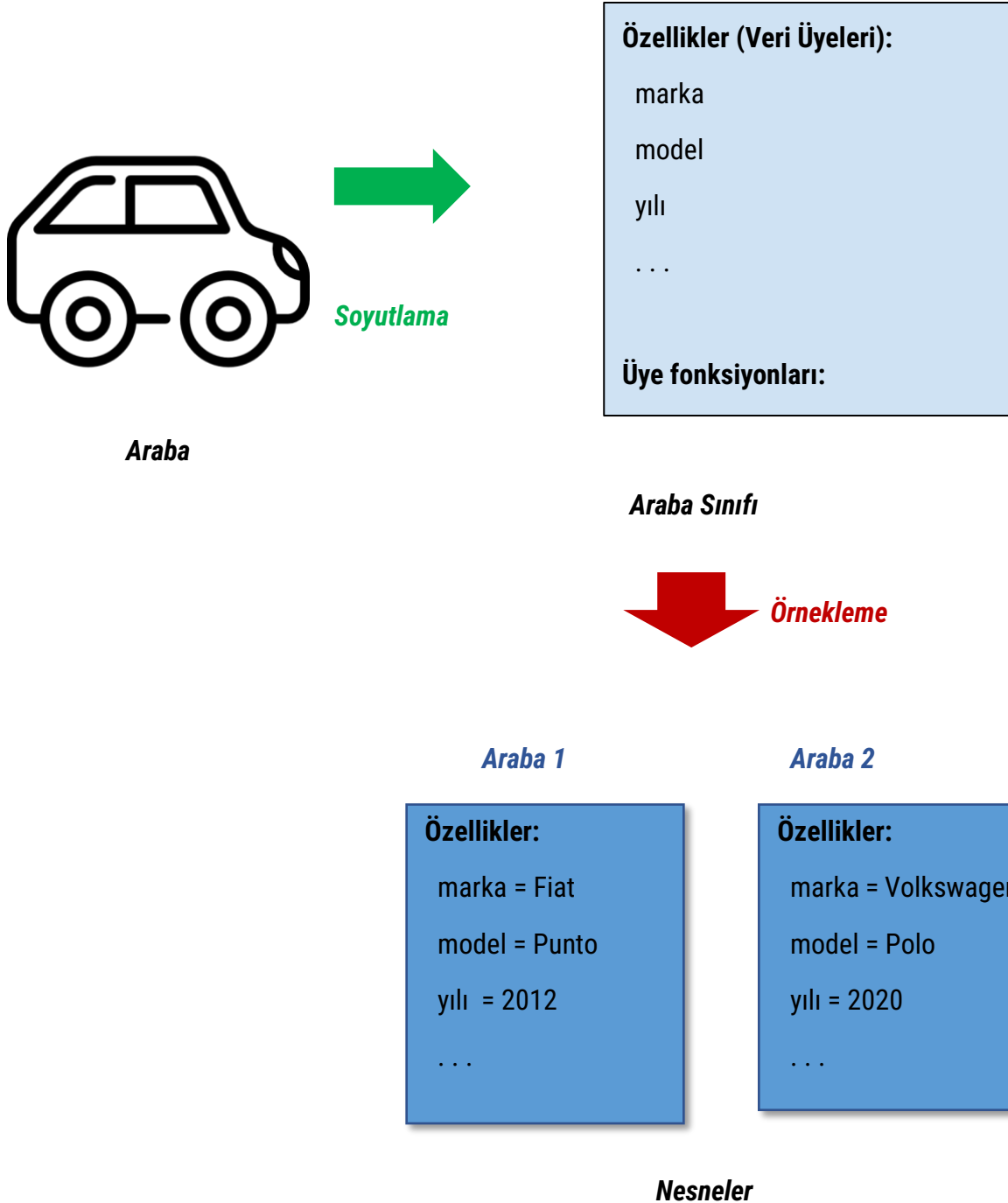
void Araba::hizlanma()
{
    hiz = hiz + 10;
    cout << "Araba hizlanıyor." << endl;
}
void Araba::yavaslama()
{
    hiz = hiz - 10;
    cout << "Araba yavaslıyor." << endl;
}
```

UYARI:

Her yöntemin ve sınıfın başına, kullanıcıya tanımlanan üye veya sınıfın ne iş yaptığını belirten bir yorum satırı ekleyin.

Nesne Tanımlama

Nesne, sınıfın bir örneğidir. Bir sınıf tanımlandığında, bellek tahsis edilmez, ancak bir nesne yaratıldığında (somutlaştırıldığında) bellek ayrılır. Sınıfta tanımlanan verileri kullanmak ve fonksiyonlara erişmek için nesneler oluşturmamız gerekir. Daha önce, bir sınıf oluşturmanın bir veri türü oluşturmaya benzediğini belirtmiştik. Bu ilişki, ilerleyen aşamalarda nesneleri daha iyi öğrenmeye başladığınızda belirginleşecektir. Bir sınıfın adını temel bir veri türünün adı gibi (int, char, float) kullanabilirsiniz. Bir nesne değişkeni, belirli bir sınıfın nesnesini depolayan bir değişkendir. Aşağıdaki görselde araba sınıfı kullanılarak birden fazla nesnenin nasıl oluşturulduğunu görebilirsiniz.



Nesne oluşturmak için öncelikle nesne için şablon olacak sınıf tasarımınızı gerçekleştirin. Sınıf tanımlamanızı gerçekleştirdikten sonra nesne değişkeni için bir tanımlayıcı oluşturun ve hangi sınıftan oluşacağını belirtiniz. Nesne oluşturmak için aşağıda verilen sözdizimlerinden birini kullanabilirsiniz:

SınıfAdı nesneDeğişkeni (parametre_listesi);

SınıfAdı nesneDeğişkeni = SınıfAdı (parametre_listesi);

Afişi Yeniden Tasarla

Süre: 45 dk.

Kazanımlar: K6. Günlük hayatta karşılaştığı problemlerle ilgili fonksiyon oluşturma işlemini gerçekleştirir.

K7. C++ programlama ile sınıf ve nesne tanımlamasını gerçekleştirir.

Materyaller: EK 1. Grup Afişleri

EK 2. Afiş Tasarım Şablonu

Dijital Tartışma Panosu: Grup Çalışması

Hazırlık: Eğitimci “Sınıfının Özelliklerini Tanı” etkinliğinde kullandıkları tartışma panosunu bu etkinlikte de kullanmaktadır. Grup afişleri bu etkinlikte destekleyici bilgi olarak kullanılmaktadır.

Uygulama: Eğitimci sunumda “Afişini yeniden tasarla: Grup Çalışması 1” başlıklı slayt üzerinden talimatları verir. Öğrenciler inceledikleri grup afişlerini ve dijital tartışma panosunu bu etkinlikte de materyal olarak kullanırlar. Bu nedenle eğitimci çalışma odaları öncesi “Grup afişleri dijital tartışma panosu”nun linkini tekrar paylaşılan notlardan iletmelidir. Eğitimci ders başında oynadıkları “Çarkı Döndür” oyununu hatırlatır. Bu oyunda öğrencilerin grup olarak bir sınıf seçimi ve bireysel olarak çizim yaptıkları nesneler vardı. İlk olarak öğrencilerden “Çarkı Döndür” oyununda belirledikleri sınıf ve bu sınıfa ait çizim yaptığı nesneyi düşünmesi istenir. Belirledikleri sınıf ve nesneleri afişteki kodlara benzer şekilde bir program olarak tasarlayacaklar ve kodlarını çalıştırıp test edeceklerdir. Bunun için “Çarkı Döndür” oyununda eğitimcinin de kaydettiği gruplar aynı olacak şekilde öğrenciler dört ayrı çalışma odasına dağıtılır. Grup çalışmalarının ilk aşamasında 15 dk. süre tanınır. Gruplar bu süre içinde doğru çalışan kodlar üzerinde çalışır ve süre sonunda panoya kodlarının ekran görüntüsünü atarlar. Ana odaya dönen öğrencilere ikinci grup çalışması için “Afişini yeniden tasarla: Grup Çalışması 2” başlıklı slayt üzerinden yeni talimatlar açıklanır.

Ana odaya dönen öğrenciler tekrar aynı grup üyeleri ile odalara dağıtılır. Ancak bu sefer odalar farklı olacaktır. Örneğin oda 1, oda 2’ye, oda 2 ise oda 3’e atılır. Bunun nedeni bir önceki grubun kodlarının yeni grup tarafından kontrolünün de sağlanmasıdır. Gruplar ikinci oturumda afiş tasarımları üzerinde 15 dk. süre içinde çalışacaktır. Gruplar afişlerini yeni atandıkları odada panoya iletilen kod satırları üzerinden yeniden tasarlayacaktır. Yeni tasarlanan afişte, grup

afişinin kod satırlarının revize edilmesi ve kod satırları üzerinden temel kavramları açıklayan grup afişindeki gibi notların bulunması istenmektedir. Buradaki amaç tasarladıkları yeni afişler aracılığıyla, birbirlerine öğrendiklerini aktarmak ve konuyu başkalarına öğretmektir. Bu noktada “artık öğretmen sizsiniz” diyerek eğitmen öğrencileri cesaretlendirir. Eğitmen bu şekilde öğrencilerden doğru çalışan kodlar ile grup afişlerini yeniden tasarlamalarını ve panoya kendi tasarladıkları afişleri göndermelerini ister.

Eğitmene Öneriler: Eğitmen öğrencilerine yeni afiş tasarlama aşamasında canva.com afiş tasarlama programını kullanabilir. Bu noktada afişin revize edilebilir şablon dosyası öğrencilerle paylaşılabilir (EK 2). Böylece afiş üzerindeki kodlar ve açıklamalar üzerinde öğrenciler değişiklik yapabilir. Diğer bir alternatif ise, word ortamında afişi oluşturmaları istenebilir.

C. Sonuç

Süre: 10 dk.

Materyal: EK 3. Hafta 9 Süreli Ödev

Hazırlık: EK 3 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.’sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta “Fonksiyonlar ve Sınıf Dosyaları” konusunun işleneceği hakkında bilgi verilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS’den yapacaktır. EK 3 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitmen görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar

Denetleyici: Kardeş olan iki peyzaj ustası kare şeklinde olan bahçelerine peyzaj yapmak istiyor. Her ikisi de kendi bahçelerine çit gerecektir. Çit için kare şeklindeki iki bahçenin kenar bilgilerini ölçerek, bahçelerin çevresini hesaplayan bir program tasarlarlar. Ancak program düzgün çalışmamaktadır. Programı düzeltmek zorundalar çünkü çevrelerini hesaplayacakları daha pek çok bahçe var. Bu iki kardeş programı baştan tasarlamaları için yardım ediniz.

İpucu! Ustanın kullandığı hatalı programda sınıf içerisinde kullanılacak kenar bilgisi değer atama yoluyla gerçekleştirken, fonksiyon tanımlama sınıf dışında yazılmıştır. Tasarlayacağınız kodda bu detayların bulunmasına dikkat ediniz.

```
#include <iostream>
using namespace std;

class Kare {
private:
    float kenar;
public:
    void deger_atama(float);
    float cevre() {
        return 4 * kenar;
    }
};

void Kare::deger_atama (float k) {
    kenar = k;
}

int main () {
    Kare kare;
    kare.deger_atama (4.3);

    cout<<"Kare Cevresi: "<<kare.cevre()<<"\n";
    return 0;
}
```

Kodun Çıktısı:

Kare Cevresi: 17.2

Kodlayıcı: Ev sahipleri bahçelerine bir havuz yaptırmak istiyor. Bahçeyi inceleyen ustanın, havuzun yapılacağı alanı hesaplamaya ihtiyacı var. Ev sahiplerine istedikleri havuzun yarıçapını belirlemelerini istiyor. Ev sahiplerinden bu bilgiyi aldıktan sonra, daire şeklindeki havuzun alanını hesaplayan bir programa bilgileri giriyor. Ustanın kullandığı programın kodlarını tasarlayınız (Dairenin alanı hesaplama formülü: $\pi * r * r$ ve $\pi = 3.14$).

İpucu! Sınıf içerisinde tanımlanacak yarıçap bilgisinin usta tarafından erişilebilir olmasına dikkat ediniz.

```
#include <iostream>
using namespace std;

class Daire {
public:
    float yari_cap;
    float alan_bul(float yari_cap) {
        return 3.14 * yari_cap * yari_cap;
    }
};

int main () {
    Daire daire;
    cout << "Dairenin yaricapini giriniz: " << endl;
    cin >> daire.yari_cap;

    cout<<"Daire Alani: "<<daire.alan_bul(daire.yari_cap)<<"\n";

    return 0;
}
```

Kodun Çıktısı:

Dairenin yaricapini giriniz:

4.2

Daire Alani: 55.3896

Hafta 9. Ders Materyalleri

EK 1. Grup Afişleri

Grup afişlerine erişmek için materyaller klasörüne gidiniz..

EK 2. Afiş Tasarım Şablonu

Canva.com sitesi aracılığıyla tasarlanacak boş afiş tasarım şablonu için [tıklayınız](#).

EK 3. Hafta 9 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 10. Nesne Yönelimli Programlama

Kazanımlar

- K1. Yapıcı ve yıkıcı fonksiyon arasındaki farkı ayırt eder.
- K2. Yapıcı fonksiyonları kullanarak program geliştirir.
- K3. Yıkıcı fonksiyonları kullanarak program geliştirir.

Haftanın Amacı ve Materyaller

Bu haftanın amacı yapıcı, yıkıcı ve sabit fonksiyonları kullanmak, sınıf dosyaları ile büyük boyutlu projeler hazırlamayı anlamaktır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız](#).

Önerilen Ders Akışı (50 + 45 + 55 + 50 dk.)

GİRİŞ (5 dk.)

GELİŞME

Yarış Benimle (45 dk.)

EK 1. Yarışma Kartları

Ders Arası (10 dk.)

Kod Satırlarını Tamamla (45 dk.)

EK 2. Grup Görev Kartları

Ders Arası (10 dk.)

El Kaldır (45 dk.)

SONUÇ (10 dk.)

EK 3. Hafta 10 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 10 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken, dikkat çekmek için eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 10 sunumunu ekran paylaşımına açar. Eğitmen bir önceki hafta işledikleri konu olan “Nesneler” hakkında özet geçer ve öğrencilerin kazandıkları rozetleri sorar. Rozet görevleri ve döngüler üzerinden ön bilgileri hatırlatır ve onuncu haftanın amacını paylaşarak öğrencileri hedeften haberdar eder.

B. Gelişme

Yarış Benimle

Süre: 45 dk.

Kazanımlar: K1. Yapıcı ve yıkıcı fonksiyon arasındaki farkı ayırt eder.

Materyaller: EK 1. Yarışma Kartları

[Yarış Benimle Oyunu](#)

Hazırlık: Eğitmen derse öncesi EK 1’de yer alan yarışma kartlarını kullanarak bir oyun tasarlar. Bu oyunda kura sisteminde gruplar tarafından bir görev kartı seçilir ve karttaki görev tamamlandığında puan kazanılır ya da kaybedilir. Oyunun bir Web 2.0 aracı olan baamboozle.com aracılığıyla oluşturulmuş versiyonuna materyallerden erişebilirsiniz (Resim 37).



Resim 37. Yarış Benimle Oyun Ara Yüzü (<https://www.baamboozle.com/classic/556028>)

Uygulama: Eğitimci öğrencileri iki gruba ayırarak, hep birlikte bir oyun oynayacaklarını belirtir. Hafta 10 sunumunda “Yarış Benimle” başlıklı slaytı kullanarak oyun talimatlarını verir. Gruplara ayrılma işlemi için eğitimci öğrenci isimlerinin alfabetik sıralaması yapılır. İsmi baş harfi A-K arası olanlar birinci grup L-Z arası ikinci grup gibi. Bir diğer gruba ayırma yöntemi ise, öğrenciler rastgele iki çalışma odasına dağıtılır. 3 dk. süre içinde gruplarına bir isim bulup, grup üyelerini kaydetmeleri istenir. Bu yöntemlerden birini kullanan eğitimci iki grubu belirledikten sonra, bir Web 2.0 aracı olan bamboozle ile hazırlanmış oyunu açar ve ekran paylaşımı yapar. Oyunun kuralları aşağıdaki gibidir:

- Oyun iki gruba oynanır.
- Gruplar sırayla bir kart numarası seçer. (Grup üyelerinden sohbette yazan ilk numara grup kartı olarak seçilir.)
- Görev kartı açılır ve grup yanıtı üyelerden toplanan anket ile alınır.
- Oyunda “Check” butonu tıklanarak grup yanıtı kontrol edilir.
- Yanıt doğru ise, “Okay!” değilse, “Oops!” tıklanır. Böylece grup puan kaybeder, kazanır ya da karşı gruba puan kazandırır.

Oyunun EK 1 yarışma kartları ile oluşturulmuş hazır linkine materyaller kısmından ulaşabilirsiniz. Oyun iki gruba klasik tema kullanılarak oynanmaktadır. Her grup sırayla ekranda gördükleri kod örneğini inceleyerek, kendilerine iletilen yarışma kartındaki ifadenin doğru ya da yanlış olduğuna karar verecektir. Gruplar oyunda yanlış yanıtlarından puan kaybederken, bazı kartlarda sürpriz puanlar kazanılıp kaybedebilmektedir. Gruplardan bir kart seçmeleri ve yanıtlamaları beklenir. Bunun için grup üyelerinin her birinin katılımını sağlamak adına anket ya da sohbet aracından faydalanılır.

Örneğin eğitimci “Grup 1 bir kart numarası düşünüp sohbette bana yazın. İlk yazanın numarasını seçeceğim” diyerek, grup 1’deki tüm üyelerin sohbeti kullanmasını ister. Seçilen kart açıldıktan sonra eğitimci kartı sesli şekilde okur ve tekrar grup 1 üyelerine seslenir. “Grup 1 kartınızdaki ifade sizce doğru mu? yanlış mı? anketime yanıt verin.” diyerek tüm grup 1 üyelerinin görüşlerini alarak karta kabul edilecek yanıt oylama yaptırır. Grup 2 bu aşamada sessizdir.

Bu şekilde gruptaki tüm öğrenciler performanslarını yansıtır. Web 2.0 aracı kullanılarak eğlenceli ve merak uyandırıcı bir yöntem ile yeni bilgi sunulur ve öğrenciler oyunda katılıma teşvik edilir. Eğitimci grup yanıtlarını aldıktan ve doğru ifadeyi oyunda açtıktan sonra her bir kart üzerine öğrencilere konu hakkında geri bildirimlerde bulunur. Oyun sonunda yapıcı ve yıkıcı fonksiyon hakkında Hafta 10 sunumunda ilerleyerek, konuyu özetler. Konunun sonunda sabit fonksiyonlara kısaca değinilir.

Eğitime Öneriler: Eğitimci yarışma kartlarının içeriğini revize ederek, oyunun farklı içeriklerle de oluşturabilir. Eğitimci oyun sırasında aşağıdaki içerikten yararlanarak öğrencilere geri bildirimde bulunur. Konu ile ilgili geri bildirimler için aşağıdaki içerikten yararlanılabilir:

Tanımladığınız sınıf içerisinde yapıcı (constructor) ve yıkıcı (destructor) olmak üzere iki özel fonksiyon türü olabilir. Her ikisi de isteğe bağlıdır ve diğer fonksiyonları sağlayamayacağı özel fonksiyonlar sağlarlar. Yapıcı fonksiyonlar sınıftan bir nesne oluşturulduğu zaman otomatik olarak çağrılır ve genellikle veri üyeleri için başlangıç değerlerini atamak için kullanılır. Örneğin cep telefonu sınıfı için düşünecek olursak yapıcı fonksiyon yıl veri üyesinin değerini sıfır yapabilir. Her zaman sınıfla aynı isme sahiptir ve int, void vb. gibi bir dönüş değerine sahip olamaz.

Yıkıcı fonksiyonlar ise, bir nesne yok edildiğinde otomatik olarak çağrılır ve gerekli tüm temizleme görevlerini gerçekleştirir. Yıkıcı fonksiyonlar, her zaman sınıfla aynı adla adlandırılır, ancak başında yaklaşık işareti (~) bulunur. Yıkıcı fonksiyonlar da bir dönüş değerine sahip olamaz. Hem yapıcı hem de yıkıcı fonksiyonlar diğer fonksiyonlar gibi tanımlanır ve uygulanır. Sınıf içinde eş zamanlı olarak tanımlamaları yapılacaksa aşağıdaki sözdizimi kullanılır.

```
class SınıfAdi
{
    SınıfAdi (parametre listesi){
        yapıcı fonksiyon gövdesi
    }
    ~SınıfAdi () {
        yıkıcı fonksiyon gövdesi
    }
};
```

Yapıcı ve yıkıcı fonksiyonların tanımlamaları daha sonra yapılacaksa aşağıdaki sözdizimi kullanılır.

```
class SınıfAdi
{
    SınıfAdi (parametre listesi);
    ~SınıfAdi ();
};

SınıfAdi::SınıfAdi (parametre listesi){
    yapıcı fonksiyon gövdesi
}

SınıfAdi::~SınıfAdi (){
    yıkıcı fonksiyon gövdesi
}
```

Yapıcı fonksiyonun parametre alabileceğine dikkat ediniz. Parametre alabilen bir yapıcı fonksiyon oluşturursanız, sınıfın kullanımı sırasında nesne oluştururken bu parametreler için değerler sağlanmalıdır. Diğer taraftan, yıkıcı fonksiyonların argümanları olamaz. Otomatik olarak çağırıldığından, kullanıcının bağımsız değişken sağlama şansı yoktur.

```
class CepTelefonu
{
public:
    char model[30];
    float fiyat;
    bool aramaDurum;
    bool mesajDurum;

    void arama();
    void mesaj_gonder();

    CepTelefonu(){
        aramaDurum = false;
        mesajDurum = false;
    }

    ~CepTelefonu(){
        cout << "Nesne yok edildi." << endl;
    }
};
```

Kod Satırlarını Tamamla

Süre: 45 dk.

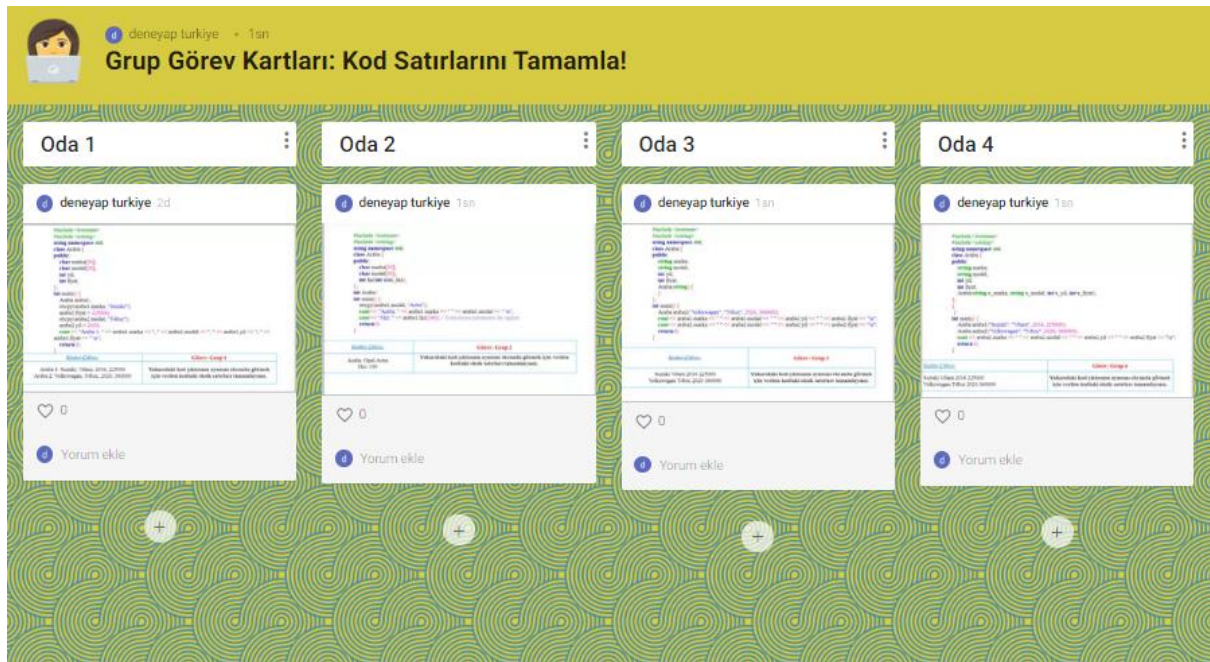
Kazanımlar: K2. Yapıcı fonksiyonları kullanarak program geliştirir.

K3. Yıkıcı fonksiyonları kullanarak program geliştirir.

Materyaller: EK 2. Grup Görev Kartları

Dijital Tartışma Panosu: Grup Çalışması

Hazırlık: Öğitmen derse girmeden önce dört farklı grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletecektir. Öğitmen padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. EK 2'deki grup görev kartlarından birer tane dijital panoda oluşturulan gruplar altına yüklenir. Örnek bir tasarım Resim 38'de gösterilmiştir.



Resim 38. Grup Görev Kartları Örnek Dijital Tartışma Panosu

Uygulama: Öğitmen Hafta 10 sunumunda “Kod Satırlarını Tamamla” başlıklı slaytı kullanarak talimatları verir. Öğrenciler bu etkinlikte beşerli gruplar halinde çalışmaktadır. Öğitmen her gruba grup görev kartını dijital pano aracılığıyla iletir. Öğitmen paylaşılan notlar kısmından padlet linkini öğrencilere gönderir. Öğitmen öğrencileri çalışma odalarına 15 dk. süre ile rastgele dağıtır. Öğrencilerden grup görev kartında yapıcı ve yıkıcı fonksiyon içeren ancak kod satırları eksik olan programı doğru çalışacak şekilde tamamlamaları beklenir. Gruplar tamamladıkları kodları derleyicide çalıştırarak test eder ve ortak bir ekran görüntüsünü grup altına gönderirler. Ana odaya döndükten sonra öğretmen dijital panoyu ekran paylaşımına açar. Öğitmen tüm grupların gönderilerinin altında olacak şekilde doğru kodları padlete yükler ve öğrencilerden kendi grup görevleri ile doğru yanıtı karşılaştırmalarını ister. Bu sırada 2 dakikalık bir sözsüz müzik açar. Müzik sonunda öğretmen grup 1 üyelerine, kendi kodlarında hatalı olan yer var mı? varsa nasıl

düzeltilebileceğini sorar. Cevaplarını padlette grup gönderilerinin altına yorum olarak yazmalarını ister. Grup üyeleri bireysel olarak bunun üzerine düşünüp, padlette grup gönderilerinin altına kendi yorumunu bırakır. Eğitimci yazılan yorumlar ve gönderiler üzerinden sırayla görevler hakkında açıklamalarda bulunur ve öğrencilere geri bildirim verir.

Eğitime Öneriler: Grup görevleri ve yanıtlarını aşağıda bulabilirsiniz.

Grup 1: Basit bir Araba sınıfı oluşturarak, birden fazla nesne tanımlaması yapmak.

```
#include <iostream>
#include <cstring>
using namespace std;
// Bazi niteliklere sahip bir Araba sinifi olusturun
class Araba {
public:
    char marka[30];
    char model[30];
    int yil;
    int fiyat;
};
int main() {
    // İlk Araba nesnesini olusturun
    Araba araba1;
    strcpy(araba1.marka, "Suzuki");
    strcpy(araba1.model, "Vitara");
    araba1.yil = 2016;
    araba1.fiyat = 225000;
    // Ikinci Araba nesnesini olusturun
    Araba araba2;
    strcpy(araba2.marka, "Volkswagen");
    strcpy(araba2.model, "T-Roc");
    araba2.yil = 2020;
    araba2.fiyat = 360000;
    // Nesnelerin ozelliklerini yazdirin
    cout << "Araba 1: " << araba1.marka << ", " << araba1.model << ", " << araba1.yil << ",
" << araba1.fiyat << " \n";
    cout << "Araba 2: " << araba2.marka << ", " << araba2.model << ", " << araba2.yil << ",
" << araba2.fiyat << " \n";
}
```

Kodun Çıktısı:

```
Araba 1: Suzuki, Vitara, 2016, 225000
Araba 2: Volkswagen, T-Roc, 2020, 360000
```

Grup 2: Araba sınıfına bir fonksiyon ekleme ve nesne tanımlama.

```

#include <iostream>
#include <cstring>
using namespace std;
class Araba {
public:
    char marka[30];
    char model[30];
    int hiz;
    int hizlan(int x);
};
int Araba::hizlan(int x) {
    return hiz + x;
}
int main() {
    Araba araba1; // Araba nesnesini olusturun
    strcpy(araba1.marka, "Opel");
    strcpy(araba1.model, "Astra");
    araba1.hiz = 120;
    cout << "Araba: " << araba1.marka << " " << araba1.model << " \n";
    cout << "Yeni Hiz: " << araba1.hizlan(30); // Fonksiyonu parametre ile cagirin
}

```

Kodun Çıktısı:

```

Araba: Opel Astra
Yeni Hiz: 150

```

Grup 3: Araba sınıfına sınıf içi bir yapıcı fonksiyon ekleme ve nesne tanımlama.

```

#include <iostream>
#include <cstring>
using namespace std;
class Araba {
public:
    string marka;
    string model;
    int yil;
    int fiyat;
    //Parametrelili sınıf içi yapıcı fonksiyon
    Araba(string x_marka, string x_model, int x_yil, int x_fiyat) {
        marka = x_marka;
        model = x_model;
        yil = x_yil;
        fiyat = x_fiyat;
    }
};

int main() {
    // Yapıcı fonksiyonu farklı değerlerle çağırarak Araba nesneleri oluşturma
    Araba araba1("Suzuki", "Vitara", 2016, 225000);
    Araba araba2("Volkswagen", "T-Roc", 2020, 360000);

    // Değerleri yazdırma
    cout << araba1.marka << " " << araba1.model << " " << araba1.yil << " " << araba1.fiyat
    << "\n";
    cout << araba2.marka << " " << araba2.model << " " << araba2.yil << " " << araba2.fiyat
    << "\n";
}

```

Kodun Çıktısı:

```

Suzuki Vitara 2016 225000
Volkswagen T-Roc 2020 360000

```

Grup 4: Araba sınıfına sınıf dışı bir yapıcı fonksiyon ekleme ve nesne tanımlama.

```

#include <iostream>
#include <cstring>
using namespace std;
class Araba {
public:
    string marka;
    string model;
    int yil;
    int fiyat;
    // Yapıcı fonksiyon bildirimi
    Araba(string x_marka, string x_model, int x_yil, int x_fiyat);
};
//Sınıf dışında yapıcı fonksiyon tanımlama
Araba::Araba(string x_marka, string x_model, int x_yil, int x_fiyat) {
    marka = x_marka;
    model = x_model;
    yil = x_yil;
    fiyat = x_fiyat;
}
int main() {
    // Yapıcı fonksiyonu farklı değerlerle çağırarak Araba nesneleri oluşturma
    Araba araba1("Suzuki", "Vitara", 2016, 225000);
    Araba araba2("Volkswagen", "T-Roc", 2020, 360000);
    // Değerleri yazdırma
    cout << araba1.marka << " " << araba1.model << " " << araba1.yil << " " << araba1.fiyat
    << "\n";
    cout << araba2.marka << " " << araba2.model << " " << araba2.yil << " " << araba2.fiyat
    << "\n";
}

```

Kodun Çıktısı:

```

Suzuki Vitara 2016 225000
Volkswagen T-Roc 2020 360000

```

El Kaldır

Süre: 45 dk.

Kazanımlar: K2. Yapıcı fonksiyonları kullanarak program geliştirir.

K3. Yıkıcı fonksiyonları kullanarak program geliştirir.

Materyaller: Hafta 10 Sunum

Hazırlık: Eğitimci etkinlik için sunum üzerinden “El Kaldır” slaytlarını kullanır.

Uygulama: Öğrenciler bu etkinlikte biri kırmızı ve diğeri mavi olan iki takıma ayrılır. Gruplara ayrılma işlemi için eğitimci öğrenci isimlerinin alfabetik sıralaması yapılır. İsminin baş harfi A-K arası olanlar kırmızı takım, L-Z arası mavi takım gibi. Takımlara ilk görev sunum üzerinden gösterilir ve görevi kaydetmeleri istenir. Daha sonra takımlar 10 dk. süre ile çalışma odalarına gönderilir. Bu süre içinde takımlar aynı zamanda bir de grup lideri belirlemelidir. Takımlar birinci görev üzerinde birlikte çalışarak eksiksiz ve başarılı bir şekilde kodu tamamlamaya çalışır. Süre sonunda ana odaya dönen takımlar kodu doğru tamamladıklarını düşünüyorlarsa, kullanıcı durumlarını aynı anda “el kaldır” olarak değiştirmelidir. Tüm takım üyelerinin el kaldır işlemini tamamlaması ve diğer gruba göre ilk el kaldırmaları önemlidir. Bu şekilde tüm grup üyeleri el kaldıran ilk takımdan kodlarını ekran paylaşımı yapmaları istenir. Kodlar doğru ise, ilk el takımın olur ve 1 puan alırlar. Kodlar hatalı ise eğitimci hataları diğer gruba düzelterip düzeltemeyeceğini sorar ve söz hakkı tanır. Diğer takım hataları düzeltirse, 1 puan bu takıma yazılır. Eğitimci kod hataları üzerine geri bildirimde bulunur ve sunumda ilerleyerek ikinci görevi ekrana yansıtır. Daha sonra aynı takımlarla ikinci görev için çalışma odalarına geçilir ve benzer süreç tekrar edilir. Etkinlik süresine bağlı olarak üçüncü görev de tamamlanır. Bu şekilde görevlerden en çok puan alan takım birinci seçilir. Tüm katılımcıların birinci takım üyelerini alkışlamaları istenir. Bunun için kullanıcı durumları “alkış” ifadesi ile değiştirilir.

Eğitime Öneriler: Eğitimci öğrencilerin kodlarını düzenlerken görevlerin yanıtlarına aşağıdan erişebilir. Bu görevleri dersin sonunda ÖYS üzerinden paylaşabilir. Eğitimci üçüncü görevi etkinlik süresi içerisinde işlemek için yeterli süre bulamazsa, öğrencilere bu görevi ÖYS üzerinden ödev olarak verebilir.

Birinci görev: “Öğrenci” isimli bir sınıf tanımlayarak, bu sınıfta iki öğrenci nesnesi oluşturmak istenmektedir. Sınıf tanımı içerisinde öğrenci numarası, ad ve soyad üyeleri tutulacaktır. Sınıf içerisinde ana fonksiyondan gelen bilgileri atamak için “deger_ata” ve atanan bilgileri göstermek için de “goster” isimli fonksiyon oluşturulması istenmektedir. Ana fonksiyonda iki öğrenci nesnesi tanımlayarak “deger_ata” fonksiyonu ile iki adet öğrenci bilgisini gönderin ve “goster” fonksiyonu kullanarak da öğrenci bilgilerini ekrana yazdırınız.

```

#include <iostream>
#include <cstring>
using namespace std;
class Ogrenci {
    int ogr_no;
    char ogr_ad[20];
    char ogr_soyad[20];
    void deger_ata(int no, char ad[], char soyad[])
    {
        ogr_no = no;
        strcpy(ogr_ad, ad);
        strcpy(ogr_soyad, soyad);
    }
    void goster(){
        cout<<"Ogrenci Bilgi: " << ogr_no <<" " << ogr_ad <<" " << ogr_soyad << endl;
    }
};
int main(void) {
    Ogrenci ogr1;
    Ogrenci ogr2;
    ogr1.deger_ata(372, "Arda", "Ozcan");
    ogr2.deger_ata(624, "Duru", "Ozen");
    ogr1.goster();
    ogr2.goster();
    return 0;
}

```

İkinci görev: “Ceptelefonu” isimli bir sınıf tanımlayarak, bu sınıfta iki cep telefonu nesnesi oluşturmak istenmektedir. Sınıf tanımı içerisinde marka, model, fiyat, arama durum ve mesaj durum üyeleri tanımlanacaktır. Arama durum ve mesaj durum değişkenleri bool olarak tanımlanacaktır. Sınıf içerisinde ana fonksiyondan gelen fiyat bilgisini atamak için bir yapıcı fonksiyon tanımlayınız. Yine sınıf içerisinde tanımlayacağınız “arama” ve “mesaj” isimli iki fonksiyon ile cep telefonu arama ve mesaj durumu bilgisini ekrana yazdırınız. Ana fonksiyonda iki cep telefonu nesnesi tanımlayarak ürün fiyatlarını ekrana yazdırınız.

```

#include <iostream>
using namespace std;
class Cep telefonu{
public:
    char marka[30];
    char model[30];
    int fiyat;
    bool aramaDurum;
    bool mesajDurum;

    Cep telefonu(int x_fiyat){
        fiyat = x_fiyat;
    }
    ~Cep telefonu(){
        cout << "Nesne yok edildi." << endl;
    }
    void arama();
    void mesaj_gonder();
};

void Cep telefonu::arama()
{
    aramaDurum = true;
    cout << "Istediginiz arama gerceklestiriliyor." << endl;
}

void Cep telefonu::mesaj_gonder()
{
    mesajDurum = true;
    cout << "Istediginiz mesaj gonderiliyor." << endl;
}

int main(){
    Cep telefonu urun1(4500);
    Cep telefonu urun2 = Cep telefonu(3750);
    cout << "Urun 1 baslangic fiyati: " << urun1.fiyat << endl;
    cout << "Urun 2 baslangic fiyati: " << urun2.fiyat << endl;
    return 0;
}

```

Kodun Çıktısı:

```

Urun 1 baslangic fiyati: 4500
Urun 2 baslangic fiyati: 3750

```

Üçüncü görev: Oda sınıfına ait tasarladığınız bir suit odanın ölçü bilgilerini kullanıcıdan alarak suit odanın alan ve hacmini hesaplayan programı yazınız. Program içerisinde tanımlayacağınız Oda sınıfının uzunluk, genişlik ve yükseklik isimli üyeleri olacaktır. Sınıf içerisinde tanımlayacağınız “veriAl” fonksiyonu ile ana fonksiyondan gelen verileri alacaksınız. Sınıf içerisinde yazacağınız “alanHesapla” fonksiyonu ile alanı, “hacimHesapla” fonksiyonu ile hacmi hesaplayınız. Ana fonksiyonda bir adet Oda nesnesi oluşturarak uzunluk, genişlik ve yükseklik bilgilerini “veriAl” fonksiyonunu çağırarak atayınız. Daha sonra odanın alanını ve hacmini ekrana yazdırınız.

```
#include <iostream>

using namespace std;

class Oda {
private:
    double uzunluk;
    double genislik;
    double yukseklik;
public:
    void veriAl(double uzn, double gns, double yks) {
        uzunluk = uzn;
        genislik = gns;
        yukseklik = yks;
    }
    double alanHesapla() {
        return uzunluk * genislik;
    }
    double hacimHesapla() {
        return uzunluk * genislik * yukseklik;
    }
};

int main() {
    Oda suitOda;
    suitOda.veriAl(3.5, 4.3, 3.2);
    cout << "Suit Oda Alan: " << suitOda.alanHesapla() << endl;
    cout << "Suit Oda Hacim: " << suitOda.hacimHesapla() << endl;
    return 0;
}
```

C. Sonuçlar

Süre: 10 dk.

Materyal: EK 3. Hafta 10 Süreli Ödev

Hazırlık: EK 3 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta "C++ Programında Kütüphane Kullanımı ve Dosyalama İşlemleri" konusunun işleneceğini bildirir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. EK 3 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitimci görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar

Tasarlayıcı: Bir futbol takımının teknik direktörü, futbolcuların ad-soyad, forma numarası ve attığı gollerin sayısını tutmak için bir program hazırlamak ister. Teknik direktör bunun için örnek bir program yazar. Programda örnek olarak Futbolcu sınıf yapısı ve bu sınıfa ait 2 futbolcunun bilgileri nesne olarak tutulmaktadır. Bu programın kodlarını tasarlayınız.

```
#include <iostream>
#include <string>
using namespace std;
class Futbolcu{
public:
    string ad_soyad;
    int forma_no;
    int gol_sayisi;
    Futbolcu(string x_ad_soyad, int x_forma_no, int x_gol_sayisi){
        ad_soyad = x_ad_soyad;
        forma_no = x_forma_no;
        gol_sayisi = x_gol_sayisi;
    }
};
```

```

    }
};

int main()
{
    Futbolcu f1("Arda Alp", 21, 13);
    Futbolcu f2("Burakcem", 23, 17);
    cout << f1.ad_soyad << " " << f1.forma_no << " " << f1.gol_sayisi << "\n";
    cout << f2.ad_soyad << " " << f2.forma_no << " " << f2.gol_sayisi << "\n";
    return 0;
}

```

Kodun Çıktısı:

```

Arda Alp 21 13
Burakcem 23 17

```

Kodlayıcı: Grafik programlarında kullanmak üzere nokta nesnelerini tanımlamak için bir Nokta sınıfı oluşturalım. Noktalar iki boyutlu düzlemde yer alacağından özellik olarak x ve y koordinatları olmak üzere iki adet koordinat bilgisine sahiptir. Programınızda noktaların sahip olması gereken yetenekler (davranışlar) ise şunlar olmalıdır:

- Noktalar, düzlemde herhangi bir yere konumlanabilmeli: git fonksiyonu
- Noktalar bulundukları koordinatları ekranda gösterebilmeli: goster fonksiyonu
- Noktalar, sıfır (0,0) koordinatında olup olmadıkları sorusunu yanıtlayabilmeli: sifir_mi fonksiyonu

```

#include <iostream>
using namespace std;
class Nokta{
    int x,y;
public:
    void git(int, int);
    void goster();
    void sifir_mi();
};
void Nokta::git(int yeni_x, int yeni_y)
{
    x = yeni_x;
    y = yeni_y;
}
void Nokta::goster()
{
    cout << "X noktasi: " << x << ", Y noktasi: " << y << endl;
}
void Nokta::sifir_mi()
{
    if ((x == 0) && (y == 0))
        cout << "n1 su anda sifir noktasindadir." << endl;
    else
        cout << "n1 su anda sifir noktasinda degildir." << endl;
}
int main() {
    Nokta n1,n2;
    n1.git(78,34);
    n1.goster();
    n1.git(61,35);
    n1.goster();
    n1.sifir_mi();
    n2.git(0,0);
    n2.sifir_mi();
    return 0;
}

```

Kodun Çıktısı:

```

X noktasi: 78, Y noktasi: 34
X noktasi: 61, Y noktasi: 35
n1 su anda sifir noktasinda degildir.
n1 su anda sifir noktasindadir.

```

Hafta 10. Ders Materyalleri

EK 1. Yarışma Kartları

Yapıcı Fonksiyon
(Constructors), sınıftan
bir nesne oluşturulduğu
anda otomatik çalışır.

Yapıcı Fonksiyon
(Constructors), sınıfla
aynı isimde olamaz.

Rakibin puanından kendi
grubuna 10 puan ekle

Gruba 10 puan
kaybettirdin

Yıkıcı Fonksiyon
(Destructors), nesne
kullanımının bittiği
zaman temizle görevi
için son olarak çalışır.

Yıkıcı Fonksiyon
(Destructors), her zaman
sınıfla aynı isimdedir.

Yapıcı Fonksiyon
(Constructors),
int, void vb. herhangi bir
dönüş tipi alamazken,
yıkıcı Fonksiyonlar
alabilir.

Gruba 10 puan
kazandırdın

Yapıcı Fonksiyon
(Constructors), parametre
alabilir.

Yıkıcı Fonksiyon
(Destructors), parametre
alabilir.

Yapıcı Fonksiyon
(Constructors), başında
yaklaşık işareti (~)
bulunur.

Grup puanından rakibine
10 puan gönder

EK 2. Grup Görev Kartları

```
#include <iostream>

#include <cstring>

using namespace std;

class Araba {
public:
    char marka[30];
    char model[30];
    int yil;
    int fiyat;
};

int main() {
    Araba araba1;

    strcpy(araba1.marka, "Suzuki");
    araba1.fiyat = 225000;
    strcpy(araba2.model, "T-Roc");
    araba2.yil = 2020;

    cout << "Araba 1: " << araba1.marka << ", " << araba1.model << ", " <<
araba1.yil << ", " << araba1.fiyat << " \n";

    return 0;
}
```

Kodun Çıktısı:

```
Araba 1: Suzuki, Vitara, 2016, 225000
Araba 2: Volkswagen, T-Roc, 2020, 360000
```

Görev: Grup 1

Yukarıdaki kod çıktısının aynısını ekranda görmek için verilen koddaki eksik satırları tamamlayınız.

```
#include <iostream>

#include <cstring>

using namespace std;
class Araba {
public:
    char marka[30];
    char model[30];
    int hiz(int max_hiz);
};
int Araba::
int main() {
    strcpy(araba1.model, "Astra");
    cout << "Araba: " << araba1.marka << " " << araba1.model << "\n";
    cout << "Hiz: " << araba1.hiz(30); // Fonksiyonu parametre ile cagirin
    return 0;
}
```

Kodun Çıktısı:

Araba: Opel Astra

Hiz: 150

Görev: Grup 2

Yukarıdaki kod çıktısının aynısını ekranda görmek için verilen koddaki eksik satırları tamamlayınız.

```

#include <iostream>
#include <cstring>
using namespace std;
class Araba {
public:
    string marka;
    string model;
    int yil;
    int fiyat;
    Araba(string) {
    }
};
int main() {
    Araba araba2("Volkswagen", "T-Roc", 2020, 360000);
    cout << araba1.marka << " " << araba1.model << " " << araba1.yil << " "
    << araba1.fiyat << "\n";
    cout << araba2.marka << " " << araba2.model << " " << araba2.yil << " "
    << araba2.fiyat << "\n";
    return 0;
}

```

Kodun Çıktısı:

```

Suzuki Vitara 2016 225000
Volkswagen T-Roc 2020 360000

```

Görev: Grup 3

Yukarıdaki kod çıktısının aynısını ekranda görmek için verilen koddaki eksik satırları tamamlayınız.

```

#include <iostream>
#include <cstring>
using namespace std;
class Araba {
public:
    string marka;
    string model;
    int yil;
    int fiyat;
    Araba(string x_marka, string x_model, int x_yil, int x_fiyat);
};
{
}
int main() {
    Araba araba1("Suzuki", "Vitara", 2016, 225000);
    Araba araba2("Volkswagen", "T-Roc", 2020, 360000);
    cout << araba2.marka << " " << araba2.model << " " << araba2.yil << " "
    << araba2.fiyat << "\n";
    return 0;
}

```

Kodun Çıktısı:

```

Suzuki Vitara 2016 225000
Volkswagen T-Roc 2020 360000

```

Görev: Grup 4

Yukarıdaki kod çıktısının aynısını ekranda görmek için verilen koddaki eksik satırları tamamlayınız.

EK 3. Hafta 10 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 11. C++ Programlama Dilinde Kütüphane Kullanımı ve Dosyalama İşlemleri

Kazanımlar

- K1. Program içinde karakter kütüphanesi kullanarak sonucu test eder.
- K2. Program içinde katar kütüphanesi kullanarak sonucu test eder.
- K3. Dosyalama işleminin gerekliliğini açıklar.
- K4. Dosya kütüphanesi kullanarak program geliştirir.
- K5. Dosya okuma işlemlerini içeren program tasarlar.
- K6. Dosya yazma işlemlerini içeren program tasarlar.

Haftanın Amacı ve Materyaller

Bu haftanın amacı öğrencilerin C++ programlama dili içerisinde bulunan kütüphaneleri kullanma ve dosyalama işlemleri yapabilmesini sağlamaktır.

Haftanın sunumu, süreli ödevi ve ek dokümanlarına erişmek için [tıklayınız](#).

Önerilen Ders Akışı (40+ 60+ 30 + 45+ 50 dk.)

GİRİŞ (10 dk.)

GELİŞME

C++ Programlama Dilinde Yerleşik Kütüphaneleri Keşfediyorum (30 dk.)

EK 1. Karakter Kütüphanesi

EK 2. Metin Kütüphanesi

Ders Arası (10 dk.)

Kütüphanelerdeki Bazı Fonksiyonları Kullanarak Kodluyorum (30 dk.)

EK 3. Kütüphaneleri ve Fonksiyonları Kullanma: Görevler

Ders arası (5 dk.)

Neden Dosyalama İşlemleri Yaparız? (30 dk.)

EK 4. C++ Programlama Dilinde Dosyalama İşlemleri Afişi 1

EK 5. C++ Programlama Dilinde Dosyalama İşlemleri Afişi 2

EK 6. C++ Programlama Dilinde Dosyalama İşlemleri Afişi 3

EK 7. Dosyalama İşlemleri

Ders Arası (5 dk.)

C++ Dilinde Dosyalama İşlemleri Yapıyorum (30 dk.)

EK 8. Kodlar Arasında Farkı Bulma

Ders arası (10 dk.)

Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum (40 dk.)

EK 9. Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum

SONUÇ (5 dk.)

EK 10. Hafta 11 Süreli Ödev (50 dk.)

Etkinlik Uygulama Ortamı

1. Eğitim ÖYS ile entegre çalışan ve “breakout rooms” çalışma odaları oluşturma özelliği bulunan Big Blue Button çevrim içi öğrenme platformu aracılığıyla gerçekleştirilir.
2. Sorumlu eğitmen öğrencilerin çalışma odalarını ziyaret ederek, öğrenme ortamına katılım sağlayacaktır.
3. Çalışma odalarında yer alacak öğrenciler uygulama üzerinden rastlantısal olarak gruplara dağıtılacaktır. Eğitmen tarafından belirlenen grup çalışma süresi sonunda ana odaya geri dönen grupların ürünleri, dijital tartışma panosu aracılığıyla toplanacaktır. Bu rehberde dijital tartışma panosu olarak padlet.com kullanılmaktadır.
4. Big Blue Button içerisinde yer alan katılımcı durum özelliklerinin değiştirilmesi, anket sorusu iletme, paylaşılan notlar ve sohbet araçları da eğitimde aktif şekilde kullanılmaktadır.

A. Giriş

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 11 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 11 sunumunu ekran paylaşımına açar. Eğitmen bir önceki hafta işledikleri konu olan nesne tabanlı programlamanın prensipleri hakkında özet geçer ve öğrencilerin kazandıkları rozetleri sorar. Rozet görevleri üzerinden *ön bilgileri hatırlatır* ve on birinci haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

Eğitime Öneriler: Verilen süreli ödevlerde sorunla karşılaşmış öğrencilere eğitmen söz hakkı verebilir. Benzer hatalar ile karşılaşmış ve sorunu çözmüş öğrencilere de söz hakkı tanınır. Aksi durumda eğitmen hatanın nasıl giderileceği konusunda geri bildirimde bulunur. Bu şekilde diğer öğrencilerin de benzer sorunlara karşı çözüm üretmeleri sağlanmış olur. Zamana bağlı olarak eğitmen öğrencilere sağlayacağı bu desteği ders dışında sorunla karşılaşan öğrencilere yardımcı olarak da tamamlayabilir.

B. Gelişme

C++ Programlama Dilinde Yerleşik Kütüphaneleri Keşfediyorum

Süre: 30 dk

Kazanımlar: K1. Program içinde karakter kütüphanesi kullanarak sonucu test eder.

K2. Program içinde katar kütüphanesi kullanarak sonucu test eder.

Materyaller: Hafta 11 Sunum

Hazırlık: Hafta 11 Sunumunda “C++ Programında Kütüphane Kullanımı ve Dosyalama İşlemleri” adlı slaytlar sırası geldikçe açılır.

Uygulama: Eğitimci bu ders için kütüphanelere giriş yapmak için hazırlanan “C++ Programlama Dilinde Kütüphaneler” adlı slaytta gelerek öğrencilere kütüphanelerin ne olduğuna yönelik açıklamalar yapar. Daha sonra EK 1, EK 2 de yer alan ve sunuma da eklenmiş kütüphane ve o kütüphanedeki kullanılacak hazır fonksiyonları öğrencilerin gerek sunu ile gerek chat ortamında tartışma yaparak gerek birbirleriyle etkileşime girerek öğrenmesi amaçlanmaktadır. Bu bağlamda slaytta her bir fonksiyon için boş bırakılan *sonuç* kısmının yanıtlarını öğrencilerin chat kısmına yazması istenir. Sunuda yer alan her fonksiyon için ayrı ayrı 20 saniyelik yanıt hakkı verilir ve her fonksiyon için ayrı ayrı verilen cevaplara göre geri bildirim verilir.

Not: Her fonksiyon değişiminde eğitimci sizce bu görevi yapacak bir fonksiyonun çıktısı ne olabilir şeklinde sorar. Verilen cevaplara göre geri bildirim verilerek bir sonraki fonksiyona geçilir.

Öğrencilere verilen görevlerin cevapları aşağıdaki gibidir. Cevaplar kırmızı renkte verilmiştir.

Tablo 22. Karakter Fonksiyonları ve Örnek Kullanımları

Fonksiyon	İşlevi	Örnek Kullanım	Sonuç
isalpha(c)	c karakteri eğer bir harf ise geriye true değilse false döndürür.	isalpha('2')	F
isdigit(c)	c karakteri eğer bir rakam ise geriye true değilse false döndürür.	isdigit('2')	T
isalnum(c)	c karakteri eğer bir rakam veya harf ise geriye true değilse false döndürür.	isalnum('*')	F
islower(c)	c karakteri eğer bir küçük harf ise geriye true değilse false döndürür.	islower('d')	T
isupper(c)	c karakteri eğer bir büyük harf ise geriye true değilse false döndürür.	isupper('H')	T
tolower(c)	c karakterini küçük harfe çevirir.	tolower('E')	e
toupper(c)	c karakterini büyük harfe çevirir.	toupper('g')	G
strlen (s1)	s1 katarının uzunluğunu döndürür.	s1 = "Merhaba" strlen (s1)	7
strcpy (s1, s2)	s2 katarını s1 katarına kopyalar.	s1 = "Merhaba" s2 = "Dunya" strcpy (s1, s2)	Dunya

Tablo 22 (devamı). Karakter Fonksiyonları ve Örnek Kullanımları

Fonksiyon	İşlevi	Örnek Kullanım	Sonuç
strcat (s1, s2)	s2 katarını s1 katarının sonuna ekler.	s1= "Merhaba" s2= "Dunya" strcat (s1, s2)	MerhabaDunya
strcmp (s1, s2)	s1 ve s2 aynı ise 0 değerini döndürür; Eğer alfabetik olarak s1, s2 metninden önce geliyorsa -1, sonra geliyorsa 1 değerini döndürür.	s1= "Merhaba" s2= "Dünya" strcmp (s1, s2)	M harfi D harfinden sonra geldiği için 1 değerini döndürür.

Kütüphanelerdeki Bazı Fonksiyonları Kullanarak Kodluyorum

Süre: 30 dk.

Kazanımlar: K1. Program içinde karakter kütüphanesi kullanarak sonucu test eder.

K2. Program içinde katar kütüphanesi kullanarak sonucu test eder.

Materyaller: Hafta 11 Sunum

EK 4. Kütüphaneleri ve Fonksiyonları Kullanma: Görevler

Hazırlık: Eğitimden bu bölüm için çözülmesi gereken iki görevin bulunduğu slaytları sırası geldikçe açması beklenir. Ayrıca kodlama yapılacağı için Code::Blocks programı açılır.

Uygulama: Bu bölüm 3 görevden oluşmaktadır. Her bir görev için eğitmenin o görev esnasında neler yapacağı aşağıda sırasıyla verilmektedir.

Görev 1: Klavyeden karakterleri sırasıyla girilen "Arda" ismini "a" değişken adıyla karakter dizisinde, "Duru" ismini ise "b" değişken adıyla katarda tutarak ekrana yazdıran kodu oluşturunuz.

Yukarıdaki görev için "Görev 1" başlıklı slayt açılır. Daha sonra eğitmen Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğitmen her satır kodun ne anlama geldiğini ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 7 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eğitmen öğrencilerin kodları kendi bilgisayarlarında yazdıklarından emin olduktan ve ihtiyacı olan öğrencilere gerekli geri bildirimleri verdikten sonra bir sonraki görev ve slayda geçer. Görev 1'in cevabı aşağıdaki gibidir.

Cevap 1:

```

#include<iostream>
using namespace std;
int main()
{
    char a[4];
    char b[5];
    int i;
    cout << "Ilk ismin karakterlerini giriniz: " << endl;
    for(i=0; i < 4; i++) {
        cin >> a[i];
    }
    cout << "Ikinci ismin karakterlerini giriniz: " << endl;
    for(i=0; i < 4; i++) {
        cin >> b[i];
    }
    b[4] = '\0';
    cout << "Ilk isim: ";
    for(i=0; i < 4; i++) {
        cout << a[i];
    }
    cout << "\nIkinci isim: ";
    cout << b;
    return 0;
}

```

Görev 2: Büyük küçük karışık hâlde yazılmış bir cümleyi her harfi büyük olacak şekilde yazdıralım. “BuGun Hava Cok GUZE!” cümlesini “Bugun Hava Cok Guzel!” şekline getirelim.

Yukarıdaki görev için “Görev 2” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğunu bildirir ve öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodları tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma

işlemi yapması istenir. Öğitmen bitirip el kaldıran öğrencilerden rastgele birini seçerek ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da öğretmen kodu nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 2'in cevabı aşağıdaki gibidir.

Cevap 2:

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
    char mesaj[] = "BuGun Hava Cok GUZEL!";

    for(int i=0; i<strlen(mesaj);i++)
    {
        if(i==0 || mesaj[i-1] == ' ')
            mesaj[i] = toupper(mesaj[i]);
        else
            mesaj[i] = tolower(mesaj[i]);
    }

    cout << mesaj;
    return 0;
}
```

Görev 3: Yazdığınız bir programda kullanıcıya bir karakter katarı içerisinde, kaç tane kelimedenden oluştuğunu sayabilecek bir program yazınız.

Yukarıdaki görev için “Görev 3” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğunu bildirir ve öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodları tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma

işlemi yapması istenir. Öğitmen bitirip el kaldıran öğrencilerden rastgele birini seçerek ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da öğretmen kodu nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 3'ün cevabı aşağıdaki gibidir.

Cevap 3:

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
    char mesaj[] = "Bugun hava cok guzel!";
    int kelimeSayisi = 0;
    for(int i=0; i<strlen(mesaj);i++)
    {
        if(mesaj[i] == ' ')
            kelimeSayisi++;
    }
    cout << "Bu cumlede " << kelimeSayisi+1 << " kelime bulunmaktadır.";
    return 0;
}
```

Neden Dosyalama İşlemleri Yaparız?

Süre: 30 dk.

Kazanımlar: K3. Dosyalama işleminin gerekliliğini açıklar.

K4. Dosya kütüphanesi kullanarak program geliştirir.

Materyaller: Hafta 11 Sunum

EK 5. C++ Programlama Dilinde Dosyalama İşlemleri Afişi 1

EK 6. C++ Programlama Dilinde Dosyalama İşlemleri Afişi 2

EK 7. C++ Programlama Dilinde Dosyalama İşlemleri Afişi 3

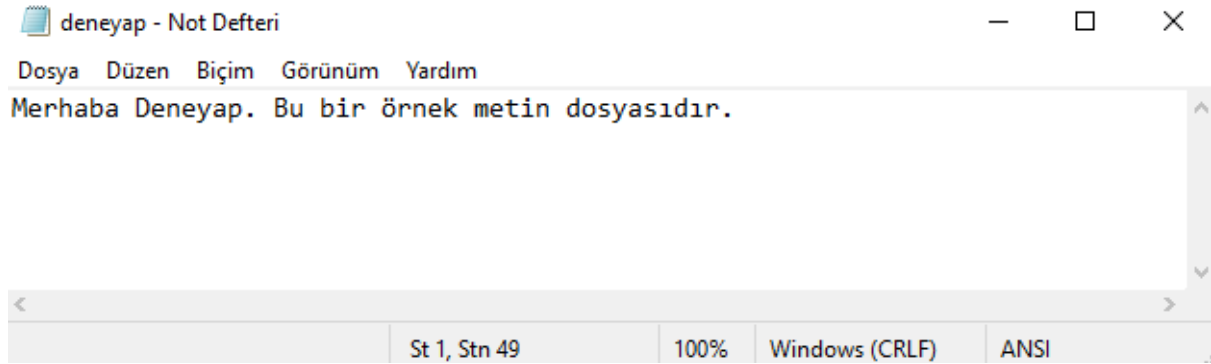
EK 8. Dosyalama İşlemleri

Hazırlık: Eğitimci sunum üzerinden dosyalama işlemleri adlı slayttı veya ekte bulunan afişin yüksek çözünürlüklü halini açar.

Uygulama: Eğitimci dosyalama işlemleri için hazırlanan afiş ekran paylaşarak gösterir ve dosyalama işlemleri hakkında aşağıdaki gibi bir özetleme yapar.

DOSYALAMA

Programlama dilleri için dosya, verilerin kalıcı olarak saklanması için kullanılır. Dosya, program yardımıyla veya kullanıcılar tarafından oluşturularak depolama biriminde tutulur. Not defteri ile kolayca oluşturabiliriz.



Resim 39. Dosya içeriği

Program verilerinin aksine, dosyadaki veriler bilgisayar kapansa bile silinmez. Bu sebeple ihtiyaç duyulan önemli bilgiler veya kullanıcılardan alınan bilgiler dosyalar yardımıyla tutulur. Genellikle “txt” uzantılı metin dosyaları kullanılır. “txt” uzantılı dosyalar hem kullanıcılar tarafından hem de program tarafından kolayca oluşturulabilir, okunabilir ve üzerine yazılabilir.

Var olan dosyalar üzerinde yapılacak metinsel işlemler okuma veya yazmadır. Dosya işlemleri ise, yeni dosya oluşturma ve mevcut dosyanın silinmesidir. Tüm bu işlemler C++ programlama dili ile hızlıca yapılmaktadır. Dosya işlemleri için C++ içerisinde üç temel sınıf bulunmaktadır.

ifstream	Okuma amaçlı açılacak dosya işlemleri için kullanılır.
ofstream	Yazma amaçlı açılacak dosya işlemleri için kullanılır.
fstream	Hem okuma hem de yazma amaçlı dosya işlemleri için kullanılır.

Dosyalar üzerinde yapılacak temel işlemler ve fonksiyonları ise aşağıdaki gibidir.

Dosyayı Aç	open()
Dosyadan veri oku	read()
Dosyaya veri yaz	write()
Dosyayı kapat	close()

Dosyayı açma sırasında eğer dosya mevcut değil ise, varsayılan olarak boş bir dosya oluşturulacaktır. Dosyayı farklı modlarda açabiliriz. Bu modlar Tablo 23'te verilmiştir.

Tablo 23. Dosya açma modları

Açıklama	mod
Normal dosya okuma modudur. Dosya en baştan okunmaya başlanır. Bu mod ifstream için varsayılan moddur.	ios::in
Normal dosya yazma modudur. Dosyaya en baştan yazılmaya başlanır. Bu mod ofstream için varsayılan moddur.	ios::out
Dosya yazma modudur. Dosyaya yazım işleminde, veriler dosyanın son karakterinden sonra eklenir.	ios::app
Dosya açıldığında içindeki tüm veriler silinir.	ios::trunc
Sadece dosya mevcut ise dosya açılacaktır. Eğer yoksa dosya oluşturulmayacaktır.	ios::nocreate

Öğrencilere verilen görevlerin cevapları aşağıdaki gibidir. Öğrenci gruplarının doldurdıkları görev kâğıtlarının aşağıdaki gibi olması beklenir.

Tablo 24. Dosya açma yapıları

Dosya Açma Kipi	Dosya açılma kontrolü kodu	Dosya mevcutsa ne olur?	Dosya yoksa ne olur?
ifstream sınıfının varsayılan modu okuma için olan ios::in modudur.	Bir dosyanın açılıp/açılmadığı " is_open() " fonksiyonu ile kontrol edilir. Eğer hatalı bir durum olduysa "0" değeri döndürecektir. Bu şekilde dosyanın düzgün biçimde açıldığı kontrol edilebilir.	Dosya mevcut ise açma moduna göre içerik silinebilir. ios::out modunda dosya içeriği silinecektir. ios::app modunda ise dosya içeriği korunacaktır.	Eğer dosya mevcut değil ise belirtilen isimde yeni dosya oluşturulacaktır. ios::nocreate modunda ise yeni dosya oluşturulmayacak sadece dosya mevcut ise açılacaktır.
ofstream sınıfının varsayılan modu yazma için olan ios::out modudur.			
fstream ise varsayılan modu ios::in ve ios::out modlarıdır.			
Örnek Kod: fstream dosya; dosya.open("deneyap.txt", ios::in ios::out);	Örnek Kod: if(!dosya.is_open()) cout << "Dosya acilamadi!";		Örnek Kod: dosya.open("deneyap.txt", ios::nocreate); if(!dosya.is_open()) cout << "Dosya mevcut değil!";

C++ Dilinde Dosyalama İşlemleri Yapıyorum

Süre: 30 dk.

Kazanımlar: K5. C++ programlama dilinde dosya okuma işlemlerini yapar.

K6. C++ programlama dilinde dosya yazma işlemlerini yapar.

Materyaller: Hafta 11 Sunum

EK 9. Kodlar Arasında Farkı Bulma

Hazırlık: Öğitmen dersin bu bölümü için hazırlanan "Kodlar Arası Farkı Bulma" adlı slayttı açar. Kodlama yapılacağı için Code::Blocks programı açılır.

Uygulama: "Kodlar Arası Farkı Bulma" adlı slayt açıldıktan sonra iki sütunda yer alan kodlar arasındaki fark öğrencilere sorulur. Zaman olarak öğrencilere 5 dk. verilir. Cevabı bulan öğrencilerin chat kısmına cevabı yazması istenir.

Eğitmen öğrencilerden gelen yanıtlara göre öğrencilere aşağıdaki ipucu bilgilerini kullanarak ve kodları öğrencilerle birlikte yazarak onların verdikleri yanıtların doğru olup olmadığını kendilerinden kontrol etmesini ister. Öğrencilerin kodları yazarak farkı anlamaya çalışması önem arz etmektedir.

İpucu Bilgileri:

C++ programlama dili üzerinde dosya işlemleri <fstream> kütüphanesi aracılığıyla yapılmaktadır. ifstream ve ofstream sınıfları bu amaçlar için kullanılmaktadır. Dosya okuma işlemleri için ifstream, dosyaya yazma işlemleri için ofstream kullanılır.

Boş dosya oluşturma: İlk olarak kütüphaneyi projemize dahil ettik. Ardından dosya isimli bir nesne oluşturduk. Parantez içerisinde de dosyanın ismini verdik. Eğer dosya yok ise projenin bulunduğu klasörde boş bir metin dosyası oluşturulmuş oldu.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ofstream dosya("deneyap.txt");
}
```

İstersek yapıcı fonksiyonu kullanmadan dosya.open() fonksiyonu ile de dosyayı açabiliriz. Varolan dosyanın üstüne bilgi ekleyebiliriz bunun için dosya isminden sonra ikinci parametre olarak mod bilgisini (ios::app) vermemiz gerekmektedir.

Dosya içerisine yazı yazmak için;

```
dosya << "Merhaba Deneyap!";
```

Satırını ekleyelim. Böylece dosyamızın içerisine “Merhaba Deneyap!” yazmış olduk. Son olarak dosyamızı kullanmayı bitirmek için;

```
dosya.close();
```

yazarak dosyamızı kapatıyoruz. Dosyamızı kapatarak geçici hafızayı da temizlemiş oluyoruz.

Bu bilgiler ışığında dağıttığım materyal üzerindeki kod farklılıklarını grup arkadaşlarımızla inceleyip, daha sonra hep beraber üzerinde tartışalım.

Eğitime Öneriler: Öğitmenlere bu derse yönelik öneriler yukarıda (uygulama başlığı altında) verilmiştir.

Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum

Süre: 40 dk.

Kazanımlar: K5. C++ programlama dilinde dosya okuma işlemlerini yapar.

K6. C++ programlama dilinde dosya yazma işlemlerini yapar.

Materyaller: Hafta 11 Sunum

EK 10. Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum

Hazırlık: Eğitimden bu bölüm için çözülmesi gereken üç görevin bulunduğu slaytları sırası geldikçe açması beklenir. Ayrıca kodlama yapılacağı için Code::Blocks programı açılır.

Uygulama: Bu bölüm 3 görevden oluşmaktadır. Her bir görev için eğitmenin o görev esnasında neler yapacağı aşağıda sırasıyla verilmektedir.

Görev 1: Klavyeden girilen “Merhaba Deneyap!” adlı cümleyi direkt string nesnesine aktarıp sonucu ekrana yazdıran kodu yazalım.

Yukarıdaki görev için “Görev 2” başlıklı slayt açılır. Daha sonra eğitmen Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğitmen her satır kodun ne anlama geldiğini ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 10 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eğitmen öğrencilerin kodları kendi bilgisayarlarında yazdıklarından emin olduktan ve ihtiyacı olan öğrencilere gerekli geri bildirimleri verdikten sonra bir sonraki görev ve slayda geçer. Görev 2’in cevabı aşağıdaki gibidir.

Cevap 1:

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string cumle = "Merhaba Deneyap!";
    cout << "Mesaj:" << cumle;
}
```

Klavyeden okuduğumuz cümleleri doğrudan string nesnesine aktarabiliriz. Bunun için aşağıdaki örneğe bakalım.

```

C:\Users\Win7\Documents\Deneyap\bin\Release\Deneyap.exe
Merhaba Deneyap!
Mesaj:Merhaba Deneyap!
Process returned 0 (0x0) execution time : 5.326 s
Press any key to continue.

```

Resim 40. Ekran çıktısı

```

#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
    string cumle;
    getline(cin, cumle);
    cout << "Mesaj:" << cumle;
}

```

Cin komutu kullanıldığında hafızada tutulan önceki girişlerin temizlenmediği durumlarda sorun oluşabilmektedir. Bu yüzden *cin.ignore()* fonksiyonu kullanılarak giriş hafızası temizlenebilir.

Görev 2: Klavyeden girilen 10 sayıyı dosyaya yazalım.

Yukarıdaki görev için “Görev 2” başlıklı slayt açılır. Burada ekranda gösterilen kodların belirtilen görevi gerçekleştirmek için eksik olduğunu bildirir ve öğrencilerden bu eksik kodları nasıl tamamlayacağını kendi bilgisayarında kodlama yaparak bulmaları istenir. Kendilerine 7 dk’lık süre verildiği belirtilerek yukarıdaki belirtilen görevdeki eksik kodları tamamlanması beklenir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapılması istenir. Eğitimci bitirip el kaldıran öğrencilerden rastgele birini seçerek ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitimci kodu nasıl yazıldığını kısaca özetleyerek, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister ve bir sonraki göreve geçer. Görev 2’in cevabı aşağıdaki gibidir.

Cevap 2:

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ofstream dosya("deneyap.txt");

    for(int i=0;i<10;i++)
    {
        int sayi;
        cin >> sayi;
        dosya << sayi << endl;
    }

    dosya.close();
}
```

Görev 3: Klavyeden girilen öğrenci sayısı kadar sınav notlarını klavyeden okuyup dosyaya yazdıran programı oluşturunuz.

Yukarıdaki görevi gerçekleştirmek için öğrencilerin kendi bilgisayarlarında 10 dk. içerisinde kodlaması beklenir. Bu esnada eğitimciler öğrencilerin sıkıştığı noktalarda ipuçları vererek onlara destek olabilir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapılması istenir. Eğitimci bitirip el kaldıran öğrencilerden rastgele birini seçer ve seçilen öğrencinin ekran paylaşımı yaparak nasıl kodlama gerçekleştirdiğini anlatmasını ister. Daha sonra da eğitimci kodu nasıl yazıldığını kendisi de yazarak özetler, görevi gerçekleştiremeyen öğrencilerin kodu tamamlamasını ister.

Cevap 3:

```

C:\Users\Win7\Documents\Deneyap\bin\Release\Deneyap.exe
Kac ogrenci olacak:
3
1. ogrenci sonucu:99.9
2. ogrenci sonucu:55.6
3. ogrenci sonucu:85.5

Process returned 0 (0x0)   execution time : 7.699 s
Press any key to continue.

```

Resim 41. Ekran çıktısı

```

#include <iostream>
#include <fstream>
using namespace std;
int main(){
    int ogrenciSayisi;
    float sonuc;

    ofstream dosya("sinav.txt");
    if(!dosya.is_open()) {
        cout << "Dosya Okunamadi!";
        return 0;
    }
    cout << "Kac ogrenci olacak:" << endl;
    cin >> ogrenciSayisi;
    for(int i=0;i<ogrenciSayisi;i++) {

        cout << i+1 << ". ogrenci sonucu:";
        cin >> sonuc;
        dosya << sonuc << endl;
    }
    dosya.close();
}

```

C. Sonuç

Süre: 5 dk.

Materyal: Hafta 11 Süreli Ödev

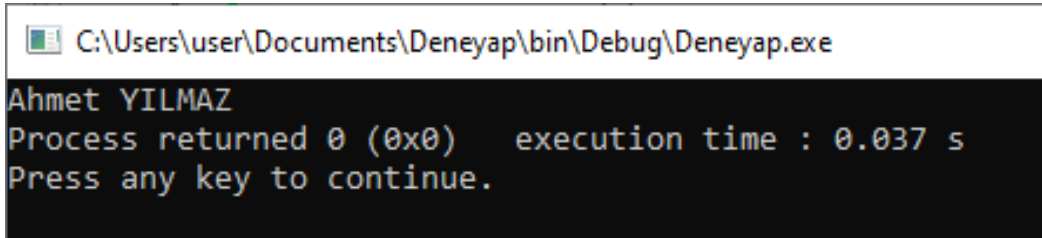
Hazırlık: EK 10 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 5 dk'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. "Hafta 11 Süreli Ödev" adlı dosya ödev sırasında öğrencilere açılır. Ödevin süresi dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitmen görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar:

1. **Kodlayıcı:** Kişinin soyadını büyük harfe çevirelim.



```
C:\Users\user\Documents\Deneyap\bin\Debug\Deneyap.exe
Ahmet YILMAZ
Process returned 0 (0x0)   execution time : 0.037 s
Press any key to continue.
```

Resim 42. Ekran çıktısı

```

#include <iostream>
#include <cstring>
using namespace std;
int main(){
    char mesaj[] = "Ahmet Yilmaz";
    bool bosluk = false;
    for(int i=0; i<strlen(mesaj);i++) {
        if(bosluk)
            mesaj[i] = toupper(mesaj[i]);
        if(mesaj[i] == ' ')
            bosluk = true;
    }
    cout << mesaj ;
    return 0;
}

```

2. **Kodlayıcı:** Klavyeden girilen cümledeki 'a' veya 'A' karakterlerini sayan programı yazınız.

```

#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char mesaj[100];
    cin.getline(mesaj,100);
    int sayac = 0;

    for(int i=0;i<strlen(mesaj);i++)
    {
        if(mesaj[i]=='a' || mesaj[i]=='A')
            sayac++;
    }

    cout << "Bu cumlede " << sayac << " adet a vardır.";
    return 0;
}

```

3. **KODLAYICI** 1-100 arası sayıların toplamını dosyaya yazınız.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    int toplam = 0;
    for(int i=1;i<100;i++)
        toplam += i;
    ofstream dosya("sonuc.txt");
    if(dosya.is_open())
    {
        dosya << toplam;
    }
    else
    {
        cout << "Dosya Okunamadi!";
    }
}
```

Hafta 11. Ders Materyalleri

EK 1. Karakter Kütüphanesi

Tek bir karakter için hazırlanmış fonksiyonlar ctype kütüphanesi içerisinde bulunur. Standart kütüphane olarak projemizde bulunmaktadır.

“

`#include<ctype>`

satırı ile projemize dahil ederiz.

”

Lütfen şimdi aşağıda yer alan noktalı yerleri grup arkadaşlarımızla dolduralım.

Tablo 25. Matematiksel fonksiyonlar ve örnek kullanımları

Fonksiyon	İşlevi	Örnek Kullanım	Sonuç
isalpha(c)	c karakteri eğer bir harf ise geriye true değilse false döndürür.	isalpha(11.4)	
isdigit(c)	c karakteri eğer bir rakam ise geriye true değilse false döndürür.	isdigit(11.4)	
isalnum(c)	c karakteri eğer bir rakam veya harf ise geriye true değilse false döndürür.	isalnum(/*)	
islower(c)	c karakteri eğer bir küçük harf ise geriye true değilse false döndürür.	islower ('d')	
isupper(c)	c karakteri eğer bir büyük harf ise geriye true değilse false döndürür.	isupper('H')	
tolower(c)	c karakterini küçük harfe çevirir.	tolower('E')	
toupper(c)	c karakterini büyük harfe çevirir.	toupper('g')	

EK 2. Metin Kütüphanesi

C++ programlama dilinde metinler üzerinde işlem yapan kullanıma hazır fonksiyonları içerisinde barındıran kütüphanenin adı **cstring** kütüphanesidir.

“

#include<cstring>

satırı ile projemize dahil ederiz.

”

Lütfen şimdi aşağıda yer alan noktalı yerleri grup arkadaşlarımızla dolduralım.

Tablo 26. Katar fonksiyonları ve örnek kullanımları

Fonksiyon	İşlevi	Örnek Kullanım	Sonuç
strlen (s1)	s1 katarının uzunluğunu döndürür.	s1 = "Merhaba" strlen (s1)	
strcpy (s1, s2)	s2 katarını s1 katarına kopyalar.	s1= "Merhaba" s2= "Dünya" strcpy (s1, s2)	
strcat (s1, s2)	s2 katarını s1 katarının sonuna ekler.	s1= "Merhaba" s2= "Dünya" strcat (s1, s2)	
strcmp (s1, s2)	s1 ve s2 aynı ise 0 değerini döndürür; s1 < s2 ise 0'dan küçük değer döndürür; s1 > s2 ise 0'dan büyük değer döndürür.	s1= "Merhaba" s2= "Dünya" strcmp (s1, s2)	

EK 3. Programa Dilinde Bulunan Kütüphaneleri ve Fonksiyonları Kullanma

Görev 1

Klavyeden karakterleri sırasıyla girilen "Arda" ismini "a" değişken adıyla karakter dizisinde, "Duru" ismini ise "b" değişken adıyla katarde tutarak ekrana yazdıran kodu oluşturunuz.

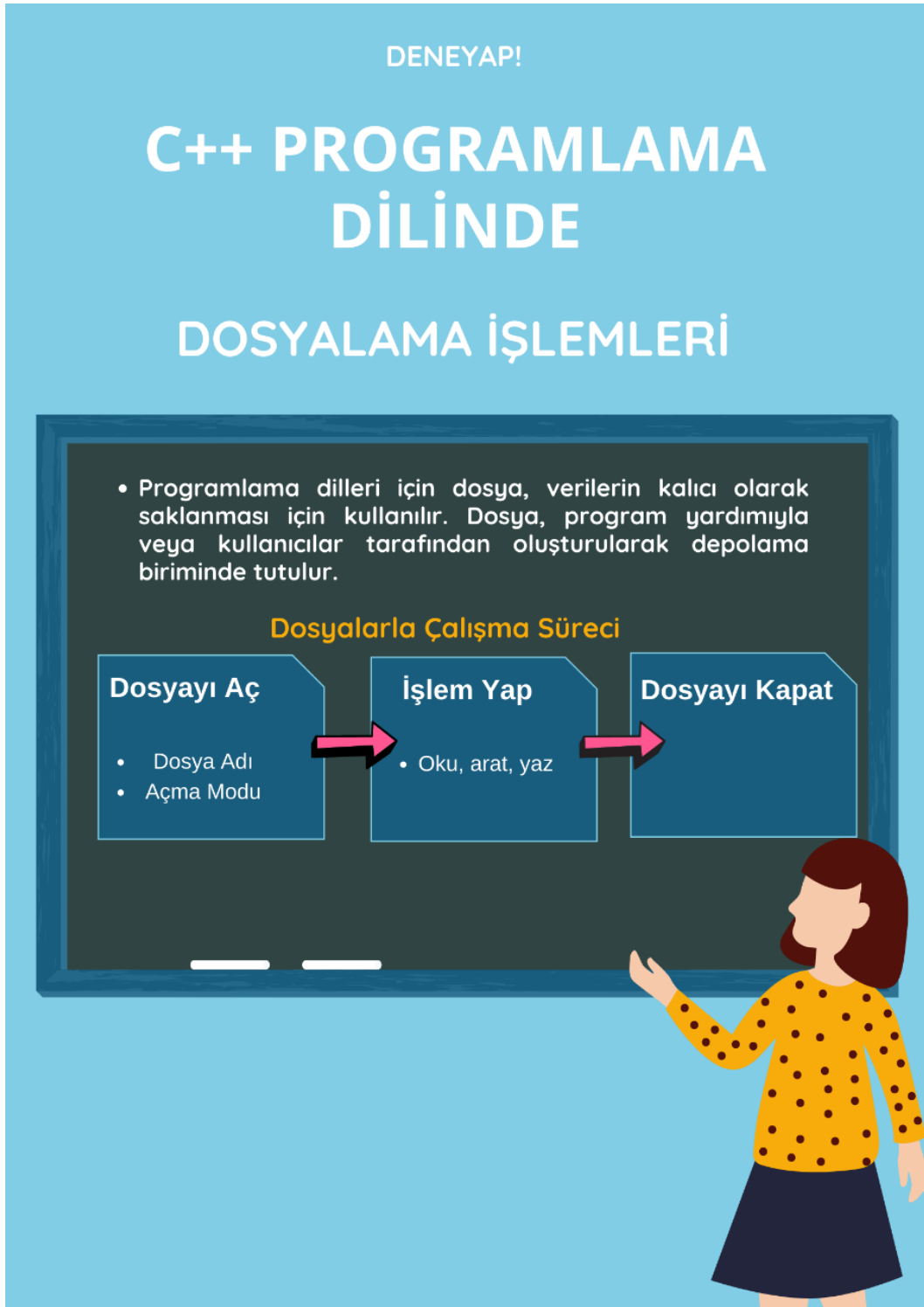
Görev 2

Büyük küçük karışık halde yazılmış bir cümleyi her harfi büyük olacak şekilde yazdıralım. "BuGun Hava Cok GUZEL!" cümlesini "Bugun Hava Cok Guzel!" şekline getirelim.

Görev 3

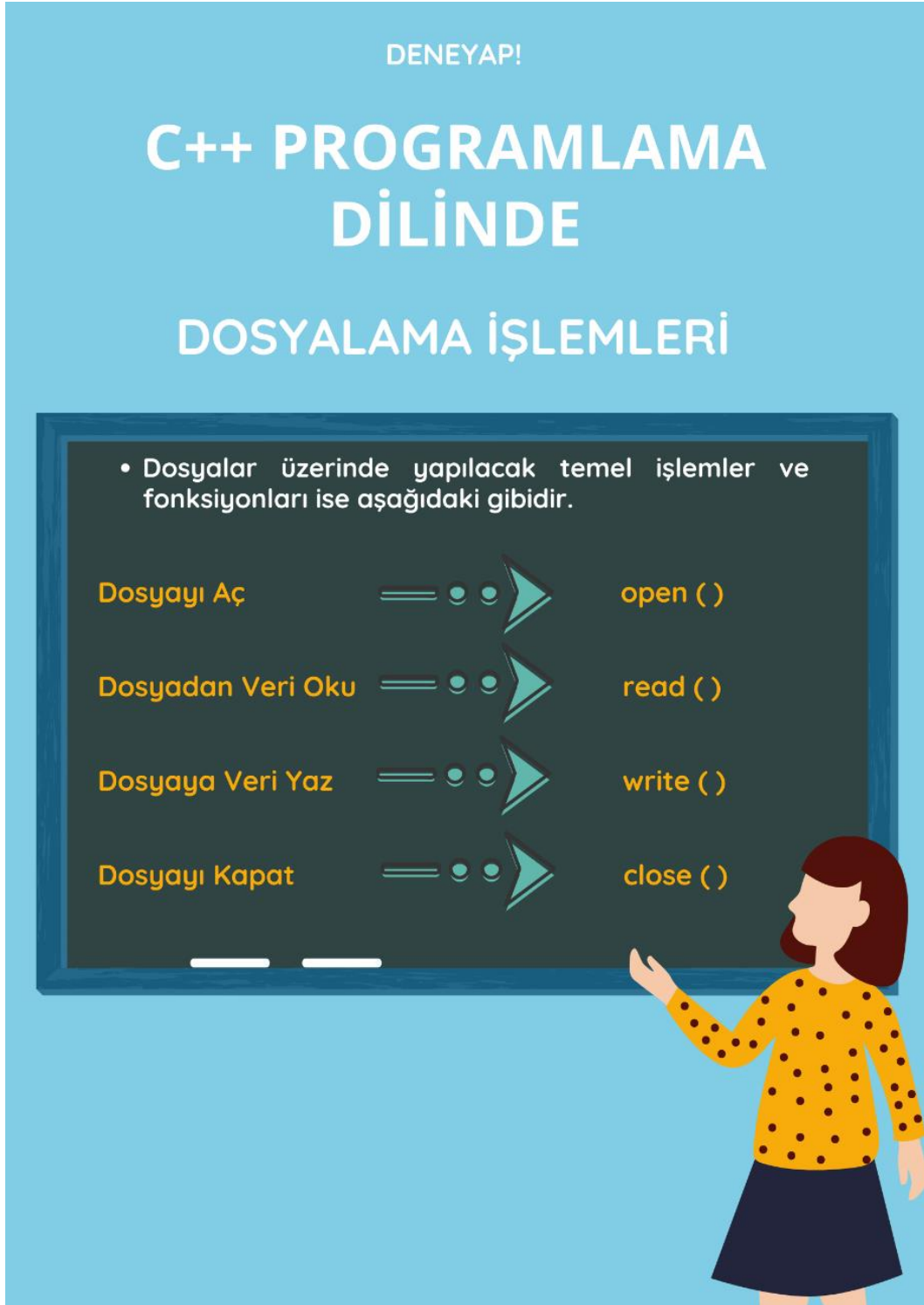
Yazdığınız bir programda kullanıcıya girişte karşılamak isteyeceğiniz mesajın, kaç tane kelimedenden oluştuğunu sayabilecek bir program yazınız.

EK 4. C++ Programlama Dilinde Dosyalama İşlemleri Yapıyorum Afişi 1



Resim 43. Dosyalama işlemleri afişi 1

EK 5. Programlama Dilinde Dosyalama İşlemleri Yapıyorum Afişi 2



Resim 44. Dosyalama işlemleri afişi 2

EK 6. C++ Programlama Dilinde Dosyalama İşlemleri Yapıyorum Afişi 3

DENEYAP!

C++ PROGRAMLAMA DİLİNDE

DOSYALAMA İŞLEMLERİ

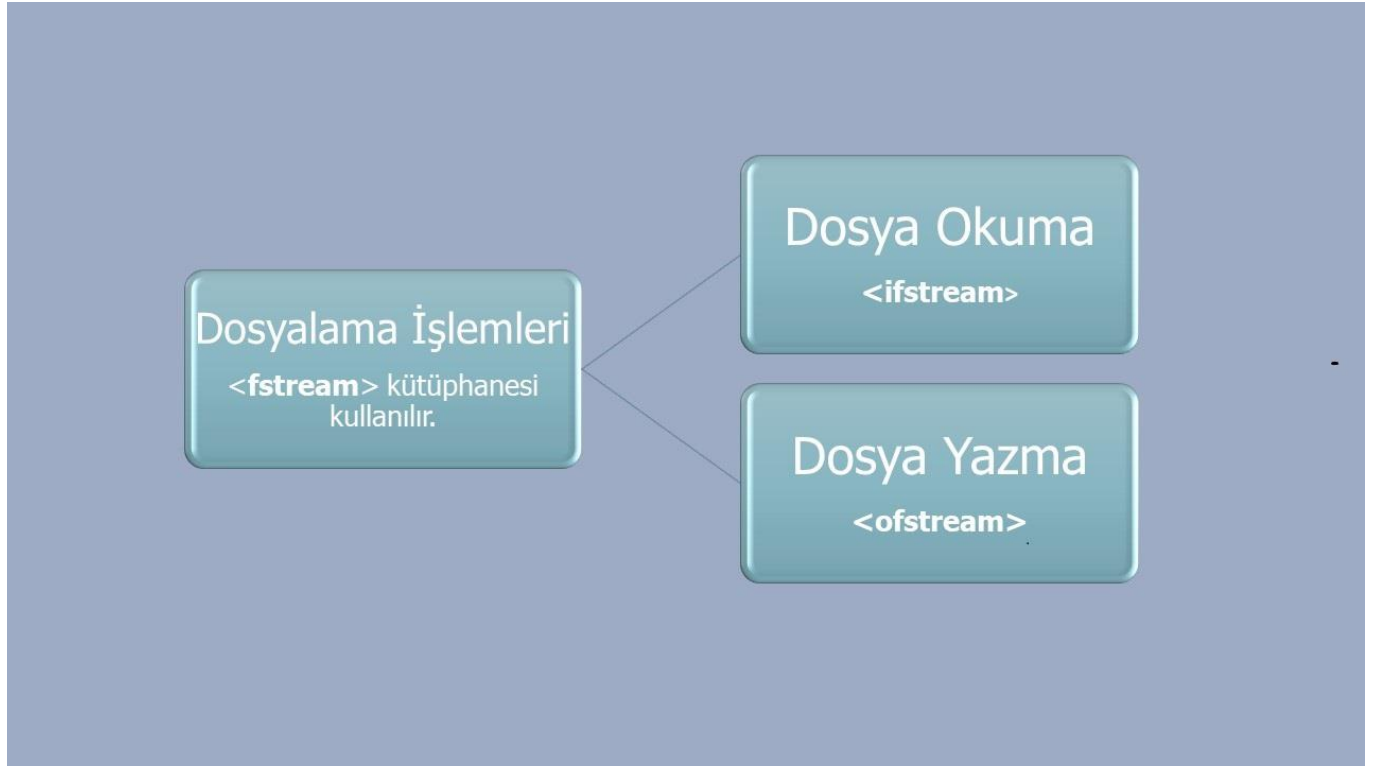
- Dosyayı açma sırasında eğer dosya mevcut değil ise, varsayılan olarak boş bir dosya oluşturulacaktır. Dosyayı farklı modlarda açabiliriz. Bu modlar;

Mod	Açıklama
<code>ios::in</code>	Normal dosya okuma modudur. Dosya en baştan okunmaya başlanır.
<code>ios::out</code>	Normal dosya yazma modudur. Dosya en baştan okunmaya başlanır.
<code>ios::app</code>	Dosya yazma modudur. Veriler dosyanın son karakterinden sonra eklenir.
<code>ios::trunc</code>	Dosya açıldığında içindeki tüm veriler silinir.
<code>ios::nocreate</code>	Sadece dosya mevcut ise dosya açılacaktır.



Resim 45. Dosyalama işlemleri afişi 3

EK 7. Dosyalama İşlemleri



Resim 46. Dosyalama işlemleri

EK 8. Kodlar Arasındaki Farkı Bulma

İşlem

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ofstream dosya;

dosya.open("deneyap.txt");

 dosya << "Merhaba Deneyap!" <<
 endl;

dosya.close();

}

İşlem

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ofstream dosya;

dosya.open("deneyap.txt",ios::app);

dosya<< "Merhaba Deneyap!" << endl;

dosya.close();

}

Dikkat!

Yukarıda verilen kodları dikkatlice gözlemleyerek aralarında ne gibi bir fark olduğunu kodları bilgisayarda yazarak bulmaya çalışalım.

EK 9. Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum

Görev 1

Klavyeden girilen "Merhaba Deneyap!" cümleyi direk string nesnesine aktarıp sonucu ekrana yazdıran koda yazdıralım.

Görev 2

Klavyeden girilen 10 sayıyı dosyaya yazalım.

Görev 3

Klavyeden girilen öğrenci sayısı kadar sınav notlarını klavyeden okuyup dosyaya yazdıran programı oluşturunuz.

EK 10. Hafta 11 Süreli Ödev

Süreli ödev bağlantıda verilen afişteki görevlerin ÖYS sistemine süreli sınav olarak yüklenmesini gerektirir. Süreli ödev afişine erişmek için materyaller klasörüne gidiniz.

Hafta 12. Proje Yarışması

Bu bölümde önceki dönemlerde yapılan örnek bir proje yarışmasına ait bir planlama sunulmuştur. Her bir yeni eğitim dönemindeki proje yarışmasında burada verilen içerikten farklı olarak yeni bir proje yarışması eğitmenlere güncel olarak sunulacaktır. Aşağıda verilen bilgiler bu yarışmalar öncesinde yapılması gerekenleri detaylı bir şekilde açıklamaktadır.

Yarışma için öğrenci gruplarının oluşturulmasında öğrencilerin eğitim boyunca kazandıkları beceri rozetleri dikkate alınmalıdır. Buradaki amacımız 4 farklı beceri rozetine farklı sayıda sahip olan öğrencilerin gruplara dengeli olarak dağıtılmasıdır. Grupların oluşturulmasında aşağıda belirtilen süreç uygulanamazsa, eğitmen öğrenci bilgi seviyelerini göz önünde bulundurarak dengeli gruplar oluşturacak şekilde dağılım yapmalıdır. Öğrenci bilgi seviyelerinin çok yakın olduğu ya da ayırım yapılamadığı durumlarda ise öğrencilerin gruplara rastgele olarak dağıtılması sağlanabilir.

Rozetler:

1. Analizci Rozeti (AR): Verilen problem için üretilen çözümlerin uygunluğunu kontrol eder ve varsa mantık hatalarının giderilmesini sağlar.



2. Tasarlayıcı Rozeti (TR): Probleme uygun çözümlerin uygulamaya geçebilmesi için kodlanmasını sağlar.



3. Kodlayıcı Rozeti (KR): Verilen probleme uygun çözümün nasıl olabileceği ile ilgili ön hazırlıkları yaparak gerekli algoritma ve akış diyagramlarının hazırlanmasını sağlar.



4. Denetleyici Rozeti (DR): Verilen problem için üretilen kodlamaların uygunluğunu kontrol eder ve varsa derleyici hatalarının giderilmesini sağlar.



Grupların belirlenmesi için aşağıdaki 5 adım izlenmelidir.

1. Listedeki tüm öğrenciler Kodlayıcı Rozeti (KR) sayılarına göre en büyükten en küçüğe doğru sıralanır ve ilk beş öğrencinin ismi KR1, KR2, KR3, KR4, KR5 olarak etiketlenir.
2. Listedeki kalan öğrenciler Tasarlayıcı Rozeti (TR) sayılarına göre en büyükten en küçüğe doğru sıralanır ve ilk beş öğrencinin ismi TR1, TR2, TR3, TR4, TR5 olarak etiketlenir.
3. Listedeki kalan öğrenciler Analizci Rozeti (AR) sayılarına göre en büyükten en küçüğe doğru sıralanır ve ilk beş öğrencinin ismi AR1, AR2, AR3, AR4, AR5 olarak etiketlenir.
4. Listedeki kalan öğrenciler Denetleyici Rozeti (DR) sayılarına göre en büyükten en küçüğe doğru sıralanır ve ilk beş öğrencinin ismi DR1, DR2, DR3, DR4, DR5 olarak etiketlenir.
5. Öğrenciler verilen etiket bilgileri dikkate alınarak aşağıdaki tabloya yerleştirilir.

Tablo 27. Öğrenci grupları oluşturulması

Grup 1	Grup 2	Grup 3	Grup 4	Grup 5
KR1	KR2	KR3	KR4	KR5
TR5	TR4	TR3	TR2	TR1
AR2	AR1	AR3	AR5	AR4
DR4	DR5	DR3	DR1	DR2

HAZİNE AVINA GİRİŞ

Maceracı arkadaşlar hazine arayışındadır. Bir gün çok eski bir harita bulurlar. Haritada etrafı sularla kaplı üç küçük hazine adasından bahsediliyor. Ancak bu adalarda hazine avlamak o kadar da kolay değil. Maceracılar hazineye ulaşabilmeleri için adalara giriş şifrelerini bulmaları gerekiyor. Adalar arasında ilerledikçe çeşitli problemlerle karşılaşılır. Maceracılar bu problemleri çözdükçe adalara giriş şifresini de çözmüş olacaklar. Maceracıların her soruda gerekli kod görevlerini yaparak bir sonraki adıma geçmeleri gerekmektedir.

DENEYAP SAYISI

Maceracıların haritadaki ilk durağı Deneyap adasıdır. Bu adada çözülmesi gereken bir problem bulunmaktadır. Bu probleme göre bir sayı, kendisini oluşturan rakamların karelerinin toplamından küçük ise bu sayıya “Deneyap sayısı” sayısı denilmektedir.

Örneğin, 54 sayısı $5^2+4^2= 41$ Deneyap sayısı değildir. Çünkü 54, 41’den küçük değildir.

Örneğin, 48 sayısı $4^2+8^2= 80$ Deneyap sayısıdır.

Klavyeden girilen iki basamaklı bir sayının Deneyap sayısı olup/olmadığını belirleyen ve sonucu ekrana yazan programın akış diyagramını oluşturunuz ve C++ kodunu yazınız.

Not 1: Akış diyagramı çiziminizi görüntü olarak kaydedip göndermeniz gerekmektedir.

Not 2: C++ kodunuzu metin olarak göndermeniz gerekmektedir.

ÇIKTI BUL

Maceracıların haritadaki ikinci durağı Çıktı adasıdır. Bu adada çözülmesi gereken problem verilen C++ kodunun çıktısının bulunmasıdır. Çıktının bulunması için verilen fonksiyonların çağırılarak iki adet sonucun ekrana yazdırılması gerekmektedir.

Not: C++ kodunun çıktısını metin olarak göndermeniz gerekmektedir.

```
#include <iostream>
#include <cstring>
using namespace std;

int karakter_say(char mesaj[])
{
    int c = 0;
    for(int i=0; i<strlen(mesaj); i++)
        if(mesaj[i] == toupper(mesaj[i]))
            c++;
    return c;
}

int main()
{
    char metin[] = "DeneYap";
    int k = karakter_say(metin);
    cout << k << " ve" << strlen(metin)-k;
    return 0;
}
```

HAZİNE ADASI

Hazine Adası'na hoş geldiniz.

Maceracılar, hazine sandığının anahtarını açmak için adada buldukları üçgen şekline benzeyen farklı boyutlardaki cisimlerin alanını hesaplamaları gerekmektedir. Onlara yardım edebilmen için öncelikle bir *Üçgen* sınıfı tanımlaman gerekmektedir. Daha sonra ana fonksiyonda bir örnek üçgen tanımlayarak kullanıcıdan alacağınız taban ve yükseklik bilgisine göre üçgenin alanını hesaplayıp ekrana yazdırınız.

1. *Üçgen* sınıfının *taban* ve *yükseklik* isminde *float* tipinde iki üyesi olmalıdır.
2. *Üçgen* sınıfının *alanHesapla()* isminde bir üye fonksiyonu olmalıdır.
3. *alanHesapla()* fonksiyonu kendisine gönderilen taban ve yükseklik bilgisine göre farklı cisimlerin alanını hesaplayabilmelidir.
4. Üçgenin alanını hesaplamak için formülümüz: $alan = \frac{taban \times yükseklik}{2}$

Not: C++ kodunuzu metin olarak göndermeniz gerekmektedir.

YARIŞMANIN BİLİMSEL KAZANIMLARI

Görev 1: Hafta 1, 2, 3, 4, 5

Görev 2: Hafta 4, 5, 6, 7, 11

Görev 3: Hafta 4, 5, 6, 7, 8, 9

Görevlerin Değerlendirilmesi

Her bir yarışma sürecinde jürilerin kullanması için aşağıdaki linkte bulunana benzer bir Jüri Değerlendirme Formu oluşturulacaktır. Bu proje yarışmasına ait değerlendirme formu için link buradadır:

<https://forms.gle/Q18SaowLy8GVWmCU8>

- Birinci görev için verilmesi gereken süre 30 dakikadır.
- İkinci görev için verilmesi gereken süre 15 dakikadır.
- Üçüncü görev için verilmesi gereken süre 30 dakikadır.
- Problem çözümleri ekip tarafından daha kısa sürede teslim edilebilir.
- Problem çözümleri teslim edildikten sonra tekrar çözüm üzerinde bir değişiklik yapılamaz.
- Ekipler yeni problemi çözerken jüri teslim edilen problem çözümünü cevap kâğıdına uygun olarak değerlendirerek 0-100 arası puanı oluşturur.
- Her bir problemdeki çözümlerin parçalı olarak puanlaması "**Problem Çözümleri**" başlığı altında detaylı olarak verilmiştir.
- Puan hesaplaması aşağıdaki gibi yapılacaktır.

$$GP = P1 + P2 + P3$$

GP: Genel Puan

P1 = Problem 1 Çözümünden Alınan Puan (Min:0 Max:100)

P2 = Problem 2 Çözümünden Alınan Puan (Min:0 Max:100)

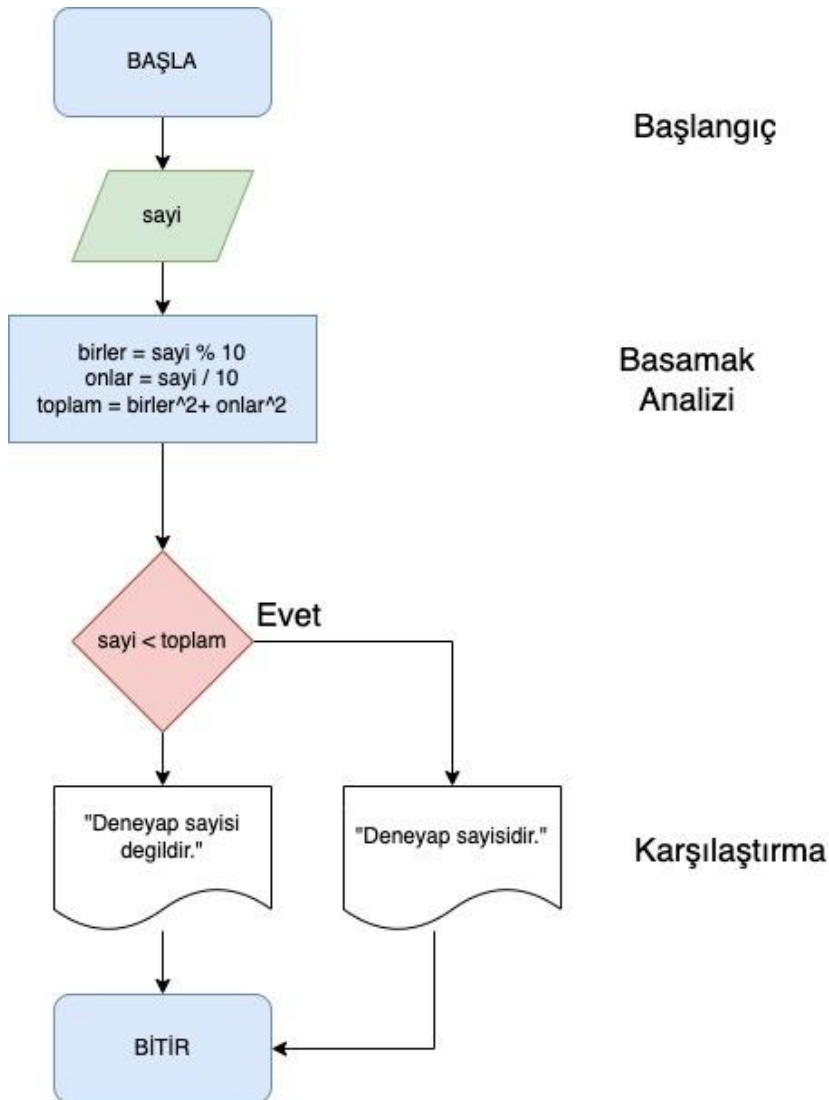
P3 = Problem 3 Çözümünden Alınan Puan (Min:0 Max:100)

Görevlerin Çözümleri

Çözüm 1:

- Puanlama:**
- Akış Diyagramı Başlangıç Kısmı: 10 Puan
 - Akış Diyagramı Basamak Analizi Kısmı: 25 Puan
 - Akış Diyagramı Koşulların Tanımlanması (Karşılaştırması): 15 Puan
 - Kod İçerisinde Değişken Tanımlama: 10
 - Kod İçerisinde Basamak Analizi Yapma: 20
 - Kod İçerisinde Karşılaştırma Yapma: 15
 - Yazılan Kodun Çalışması: 5

Akış Diyagramı:



C++ Kodu:

```
#include <iostream>
using namespace std;
int main()
{
    int sayi, birler, onlar, toplam=0;
    cin >> sayi;
    birler = sayi%10;
    onlar = sayi/10;
    toplam = (birler*birler) + (onlar*onlar);
    if(sayi < toplam)
        cout << "Deneyap sayisidir!";
    else
        cout << "Deneyap sayisi degildir!";
}
```

Çözüm 2:

Puanlama: Birinci çıktı değeri: 50 Puan

İkinci çıktı değeri: 50 Puan

Çıktı: 2 ve 5

Çözüm 3:

Puanlama: Sınıf ve Değişken Tanımlama Bölümü: 40 Puan

Alan Fonksiyonu Tanımlama: 40 Puan

Ana Fonksiyonda Örnek Tanımlama ve Kullanma: 20 Puan

Çıktı:

Ucgenin tabani = 5

Ucgenin yuksekligi = 7

Ucgenin alani=17.5

Kod:

```
// YARISMA GOREVI 3:
#include<iostream>
using namespace std;
// bir sinif tanimlayiniz
class Ucgen
{
    private:
        float taban, yukseklik;

    public:

        float alanHesapla(float taban, float yukseklik)
        {
            float alan;
            alan= 1.0 / 2.0 * taban * yukseklik;
            return alan;
        }

};

int main()
{
    // Ucgen sinifindan bir nesne tanimlayiniz
    Ucgen ucgen1;
    float t,y;
    cout<<"Ucgenin tabani = ";
    cin>>t;
    cout<<"Ucgenin yuksekligi = ";
    cin>>y;
    // Ucgenin alanini taban ve yukseklik bilgilerini kullanarak cagiriniz
    cout<<"Ucgenin alani="<<ucgen1. alanHesapla(t,y);

    return 0;
}
```

ÇEVİRİM İÇİ YAZILIM TEKNOLOJİLERİ

ORTAOKUL

