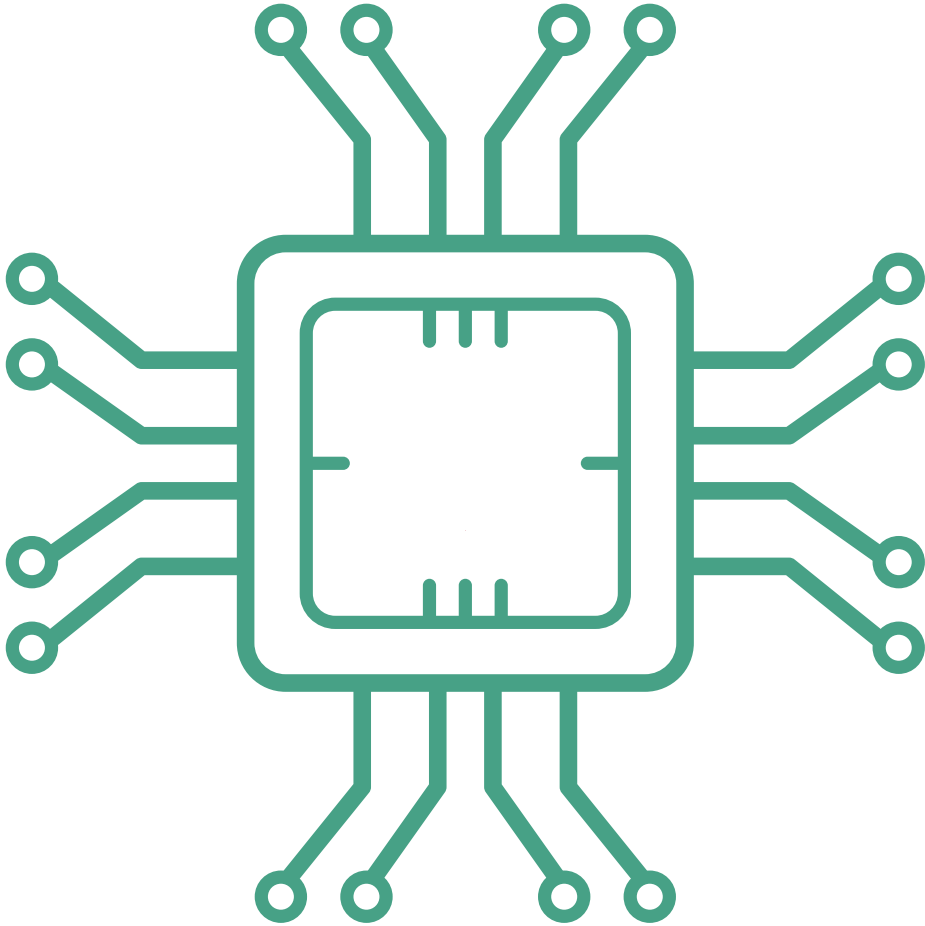


DENEYAP
TÜRKİYE

ÇEVİRİM İÇİ YAZILIM TEKNOLOJİLERİ

LİSE



T.C. SANAYİ VE
TEKNOLOJİ BAKANLIĞI



T.C. GENÇLİK VE
SPOR BAKANLIĞI



TÜBİTAK



TÜRKİYE
TEKNOLOJİ
TAKIMI

TÜBİTAK Deneyap Kitapları 17

**ÇEVİRİM İÇİ YAZILIM
TEKNOLOJİLERİ**
LİSE

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 materyaller, 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: 978-605-312-533-4
Yayıncı Sertifika No: 47703

Yayın Tarihi: 2023

TÜBİTAK Başkanı: Prof. Dr. Hasan MANDAL
Bilim ve Toplum Başkanı: Ömer KÖKÇAM
Genel Yayın Yönetmeni: Fatma BAŞAR
Editör: Dr. İpek PİRİROĞLU GENCER
Düzelti: 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

LİSE

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	13
Hafta 1. Programlamaya Giriş	14
A. Giriş	15
B. Gelişme	18
Bilgisayarın Çalışması Neye Benzer?	18
Programlama Türlerini Karşılaştıralım	19
Bilgisayar Verileri Nasıl Saklar?.....	21
Beni Bul ve Değiştir.....	23
Farklı Atama Türlerini Tanıyalım.....	27
Sayı Sistemlerini Keşfedelim.....	29
C. Sonuç.....	34
Hafta 1. Ders Materyalleri	36
Hafta 2. Algoritma Tasarımı	45
A. Giriş	46
B. Gelişme	47
Algoritmayı Tanıyalım	47
Algoritmaları Eşleştirelim	47

Çiftleri Toplayalım	48
Algoritma Terimlerini Keşfedelim	50
Otomatik Park Edelim.....	51
Algoritmayı Test Edelim	52
C. Sonuç.....	57
Hafta 2. Ders Materyalleri	61
Hafta 3. C++ Dilinde Değişken ve Veri Tipleri	71
A. Giriş	72
B. Gelişme	72
Değişkenlere İsim Verelim	72
Sabitleri Ayıralım	75
Örnekleri Çoğaltalım	76
Veri Tipini Dönüştürelim	76
Kaçış Dizgelerini Tanıyalım.....	78
Çıktıları Karşılaştıralım	79
Operatörlerle Yüzleşelim	80
Listeyi Dolduralım.....	81
C. Sonuç.....	83
Hafta 3. Ders Materyalleri	87
Hafta 4. Karar Mantık Yapıları	103
A. Giriş	104
Biliyor muydunuz?.....	104
B. Gelişme	105
Karar Yapılarını Tanıyalım.....	105
Görevleri Kodlayalım	107
İç içe Koşula Farklı Bir Bakış (Switch)	109
C. Sonuç.....	111
Hafta 4. Ders Materyalleri	115
Hafta 5. Döngü Yapıları.....	122
A. Giriş	123
B. Gelişme	123
Döngüleri Tanıyalım	123
Döngüleri Ne İçin Kullanıldığını Keşfetme.....	124
Döngü Görevlerini Kodlayalım	125

C. Sonuç.....	129
Hafta 5. Ders Materyalleri	133
Hafta 6. Diziler ve Katarlar	139
A. Giriş	140
B. Gelişme	140
Dizileri Tanıyalım	140
Dizilere Değer Verelim.....	142
Döngülerle Diziler	142
Dizilerle Kodlayalım	144
Kodlama Ekibi	150
Farkı Keşfedelim.....	153
C. Sonuç.....	154
Hafta 6. Ders Materyalleri	158
Hafta 7. Fonksiyonlar	163
A. Giriş	164
B. Gelişme	164
Fonksiyonları Tanıyalım.....	164
Fonksiyonların Nasıl Kullanıldığını Keşfediyorum	166
Fonksiyonları Kullanarak Kodlama Yapalım	167
C. Sonuç.....	174
Hafta 7. Ders Materyalleri	177
Hafta 8. Nesneler	182
A. Giriş	183
B. Gelişme	183
Nesneleri Çiz.....	183
Sınıfının Özelliklerini Tanı	188
Afişi Yeniden Tasarla	193
C. Sonuç.....	194
Hafta 8. Ders Materyalleri	198
Hafta 9. Nesne Yönelimli Programlama.....	199
A. Giriş	200
B. Gelişme	200
Yarış Benimle	200
Kod Satırlarını Tamamla	204

Sınıfları Dosyala	209
C. Sonuç.....	212
Hafta 9. Ders Materyalleri	215
Hafta 10. Nesne Yönelimli Programlamanın Prensipleri	221
A. Giriş	222
B. Gelişme	222
Adam Asmaca	222
Kalıtımı Sürdür	228
Aşırı Yüklenenler.....	232
Geçersiz Olanlar	235
C. Sonuç.....	238
Hafta 10. Ders Materyalleri	242
Hafta 11. C++ Programlama Dilinde Kütüphane Kullanımı ve Dosyalama İşlemleri.....	250
A. Giriş	251
B. Gelişme	251
C++ Programlama Dilinde Yerleşik Kütüphaneleri Keşfediyorum.....	251
Eksik Kodları Dolduruyorum	253
Kütüphanelerdeki Bazı Fonksiyonları Kullanarak Kodluyorum	256
Neden Dosyalama İşlemleri Yaparız?	258
C++ Dilinde Dosyalama İşlemleri Yapıyorum.....	260
Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum	262
C. Sonuç.....	266
Hafta 11. Ders Materyalleri	269
Hafta 12. Proje Yarışması	280

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.

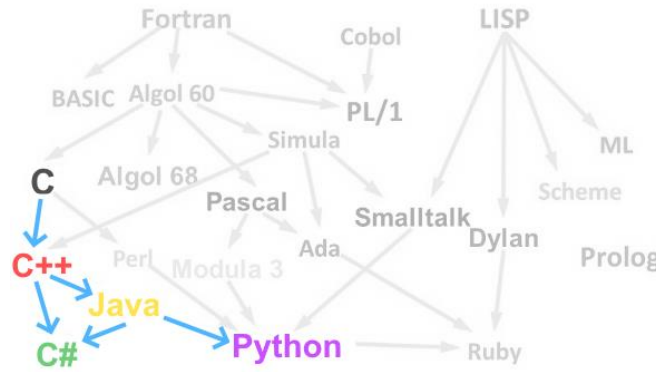
Dersin Çıktıları

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.
- 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 programlamaya 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

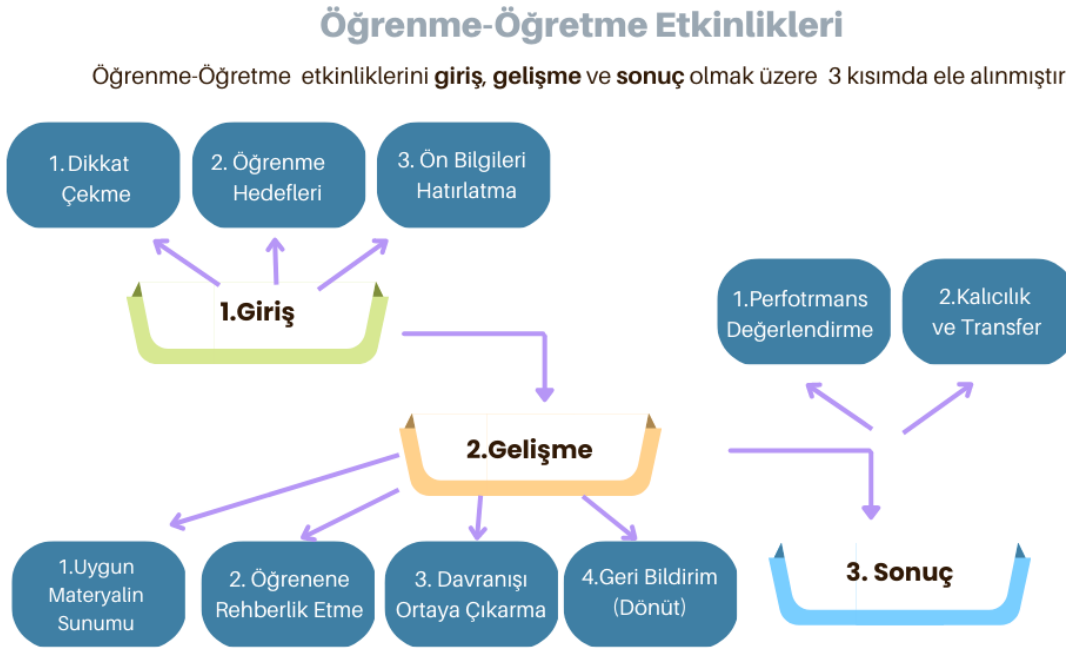
YouTube, Google, Amazon, Twitter, Facebook gibi uygulamaların yapımında C++ programlama dili de kullanılmıştır. Dünya çapında popülerliğini korumaktadır. Yüksek performanslı uygulamalar, oyunlar ve karmaşık araçlar yazmak için ve yazılan uygulamanın direkt olarak donanım ile haberleşmesini için C++ dili 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, Örnek Akış Şemaları, Kodlamaya Geçiş, Program Yazmanın Adımları (Hafta İçi Uygulama: C++ Programlama Dilinin Özellikleri, IDE (Derleyici) Kurulumu ve İlk Programlama)
- Hafta 3: C++ Tanımlamalar ve Operatörler, Değişken Tanımlama, Veri Tipleri, Sabitlerin Tanımlanması, Veri Tipi Dönüşümleri, Kaçış Dizgeleri, Operatörlerin Kullanımı, Bit İşlemler
- Hafta 4: Karar Verme Komutları (if-else, switch), If ifadesi, Switch İfadesi, Mantıksal Operatörler, Operatör Öncelikleri
- Hafta 5: Döngü Komutları (for, while, do-while), İç İçe Döngüler, Rastgele Sayılar
- Hafta 6: Diziler, Karakter Katarları, Örnek Problem Çözümleri
- Hafta 7: Fonksiyonlar, Fonksiyon Sözdizimi, Fonksiyon Çağırma
- Hafta 8: Nesne Yönelimli Programlama, Sınıf Tanımlama, Metot Tanımlama, Nesne Değişkenleri Oluşturma, Yapıcı Yöntemler
- Hafta 9: Nesne Yönelimli Programlama, Nesne Değişkenleri Oluşturma, Yapıcı ve Yıkıcı Metotlar
- Hafta 10: Nesne Yönelimli Programlama, Veri Soyutlama, Kapsülleme, Kalıtım, Polimorfizm, Aşırı Yükleme, Geçersiz Kılma
- 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, eğitimciler dijital araçlardan faydalanarak öğrencilerin performanslarını yansıtmalarını ister. Eğitimciler 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 İçi 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 İçi 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 İçi 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 İçi istasyon tekniği ve çevrim İçi 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 İçi 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.



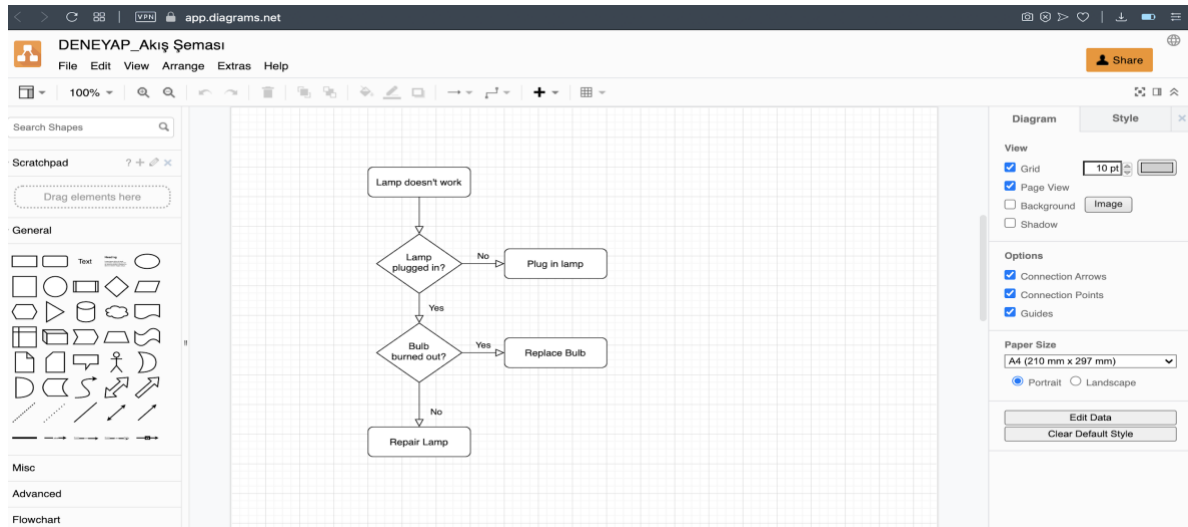
Beceri rozetleri verilen süreli görevleri süreçte değerlendirmeyi sağlamak amacıyla kullanılmıştır. Bu şekilde öğrencilerin etkinliklere katılımını artırmak hedeflenmektedir. Ayrıca öğrenciler tüm hafta boyunca birden fazla analizci ya da kodlayıcı rozeti kazanabilir. Örneğin dönem sonunda edindikleri 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
- Gerekliğinde senkron (canlı) ders ortamının oluşturulması (Oluşturulan canlı ders çalışma odaları destekli olup grup çalışmalarına imkân vermedir.)
- Eğitmen eğitimleri sürecinde gerekli kaynakların, sunumların ve dokümanların eğitimcilerle paylaşılabilmesi
- Eğitimcilerin fikir alışverişi yapabilmesi için tartışma ortamlarının kurulabilmesi

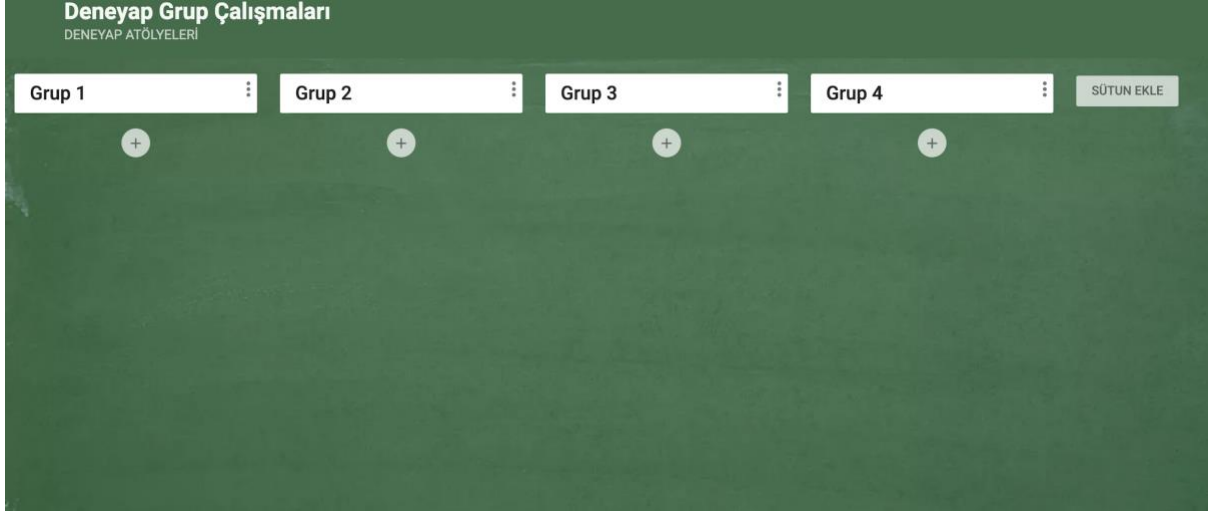
GitHub: Derste aktif şekilde kullanılacak olan arşivleme portfolyo oluşturma ortamlarından biri Github'tır. GitHub açık kaynaklı projeler tarafından tercih edilen ve yazılım geliştirme projeleri için kullanılan web tabanlı popüler bir depolama servisi. GitHub ile dünya çapında herkes tarafından görüntülenebilen 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 portfolyo 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 İçi ortak çalışma alanı kurma
- Çevrim İçi tartışma ya da beyin fırtınası oluşturma
- Grup ürünlerini paylaşma
- Akranların yaptıklarını çevrim İçi 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 İçi grup ya da bireysel öğrenci paylaşımlarının kaydını tutma
- Tüm öğrencilere birbirlerinin paylaşımlarını çevrim İçi anlık inceleme fırsatı sunma
- Çevrim İçi 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.
- 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 açıklar.
- K2. Bilgisayarın temel birimlerini açıklar.
- K3. Programlama türlerini ayırt eder.
- K4. Veri saklama birimlerini ayırt eder.
- K5. Problem çözme sürecinde aritmetik, mantıksal ve karşılaştırmalı işleçleri kullanır.
- K6. Değişkenlere farklı türlerde değer ataması yapar.
- K7. İkili, sekizli, onlu ve onaltılı sayı sistemlerini kullanır.

Haftanın Amacı ve Materyaller

Bu haftanın amacı, programlamaya dünyasına giriş yapılarak öğrencilerin temel düzeyde bildikleri matematiksel, karşılaştırma ve mantıksal işleçleri (operatörleri) kullanırmak ve sayı sistemlerinin keşfedilmesidir.

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

Önerilen Ders Akışı (60 + 40 + 50 + 50 dk.)

GİRİŞ

Hoş Geldin Panosu (10 dk.)

Aklına Ne Geliyor? (10 dk.)

GELİŞME

Bilgisayarın Çalışması Neye Benzer? (20 dk.)

Programlama Türlerini Karşılaştıralım (20 dk.)

EK 1. Bilgi Kartları

Ders Arası (10 dk.)

Bilgisayar Verileri Nasıl Saklar? (15 dk.)

Bul ve Değiştirmeyi Dene (25 dk.)

EK 2. İşleç bilgi kartları

EK 3. Örnek Olay Sunum Kartı

Ders Arası (10 dk.)

Farklı Türleri Tanıyalım (20 dk.)

Sayı Sistemlerini Keşfet (20 dk.)

EK 4. Veri Dönüştürme Örnek Olay Kartları

SONUÇ (10 dk.)

EK 5. 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

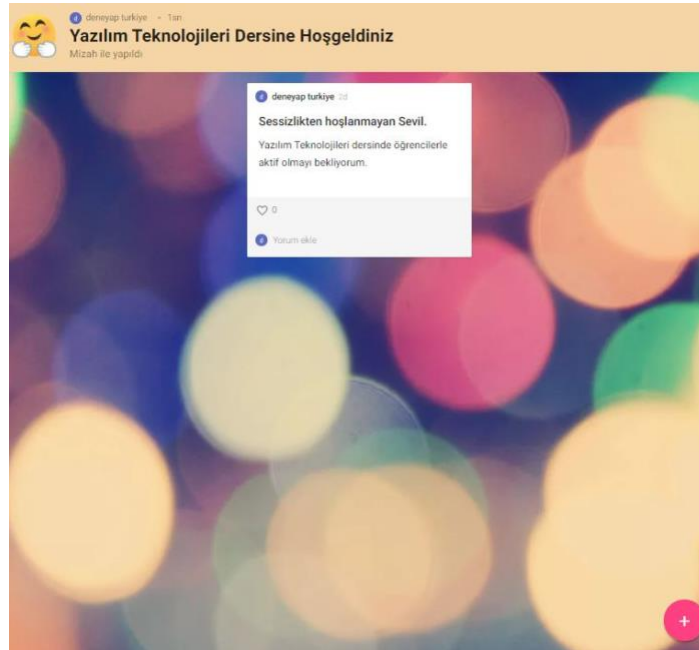
Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 1 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır.

Hoş Geldin Panosu

Süre: 10 dk.

Materyal: Hafta 1 Sunum

Hazırlık: Eğitmen derse girmeden önce duvar temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletecektir. Örnek bir tasarım Resim 6’da gösterilmiştir.



Resim 6. Örnek Hoş Geldin Panosu

Uygulama: Öğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar. Hafta 1 sunumunu sanal sınıfta kullanmak üzere yükler. Öğrenciler tamamlandığında, öğretmen ders öncesi oluşturduğu Hoş Geldin Panosunun linkini öğrencilere paylaşılan notlar kısmından iletir. Bu ortamı öğrencilere göstermek için aynı zamanda ekran paylaşımı da yapar. Buraya ilk olarak öğretmen kendi ismini ve isminin baş harfi ile başlayan bir özellik yazmıştır. Örneğin Sevil. Özelliğim ise “Sessizlikten hoşlanmaz.” Öğitmen öğrencilerden de kendilerini padlet ortamında bu şekilde tanıtmasını ister. Ayrıca öğrencilerden Yazılım Teknolojileri dersine yönelik beklentilerini de burada paylaşımları istenir. Öğrenciler bu ortamda fotoğraf da paylaşabilirler. Öğitmen öğrencilerin padlet ortamını doldurma aşamasında eğlenceli sözsüz bir müzik açar. Mümkünse müzik süresi 3 dk.’lık olabilir. Müzik sona erdiğinde etkinlik biter. Bu şekilde dikkati toplayan öğretmen, padlet paylaşımları üzerinden kısaca özet geçer. Slaytta ilerleyerek haftanın konu ve kazanımlarını açıklar ve öğrencileri hedeften haberdar eder.

Aklına Ne Geliyor?

Süre: 10 dk.

Kazanımlar: K1. Temel programlama terimlerini öğrenir.

Materyal: Hafta 1 Sunum

Uygulama: Bu görevde öğretmen öğrencilerden sanal sınıfta “sohbet” kısmını kullanmalarını ister. Öğitmen 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, öğretmen kelimeyi bu sefer değiştirir ve öğrencilere şunu söyler.

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

Öğitmen tekrar 1 dk.’lık bir müzik açar. Müzik sonunda öğretmen 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.

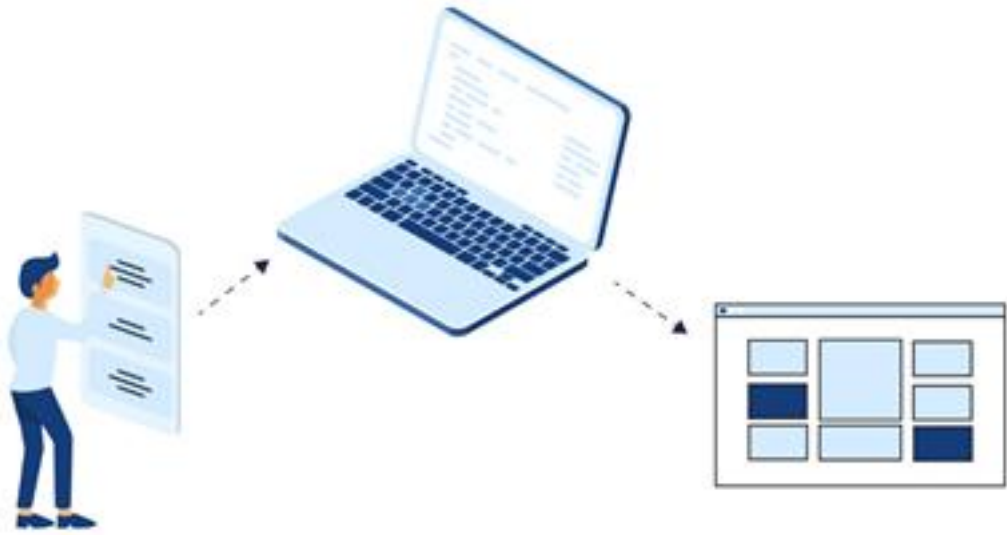
Tablo 1. Programla ve Programlama Dili Kavramları

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

Eğitime Öneriler: Eğitimci 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 7. 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. Eğitimci sunumun anket slaytında bekler. Uyarıcı materyal olarak anket yoluyla *yeni bilginin sunulması* sağlanacaktır. Eğitimci 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 verdiğiniz bu ifadeyi ekranda gördüğünüz insan organlarından hangisi ile eşleştirebilirsiniz? Anketi cevaplayınız.”*

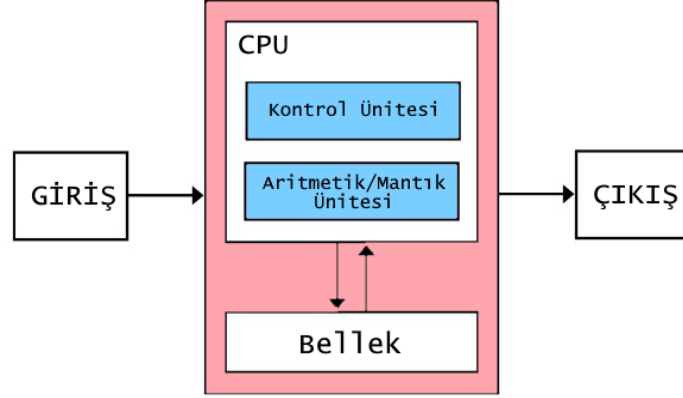
Daha sonra öğrencilerin yanıtlarını anket aracılığıyla iletmesini ister. Bu şekilde öğrencilerin performansını yansıtmaları beklenir. Öğrencilerin her bir anket sorusuna verdiği yanıttan sonra eğitimci öğrenci yanıtlarından yararlanarak yapılan eşleşme hakkında *öğrenmeye kılavuzluk eder*. Eğitimci birinci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

Anket Soruları:

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

Eğitimci 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 içeriği: 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 (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 8. 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 iletme 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.

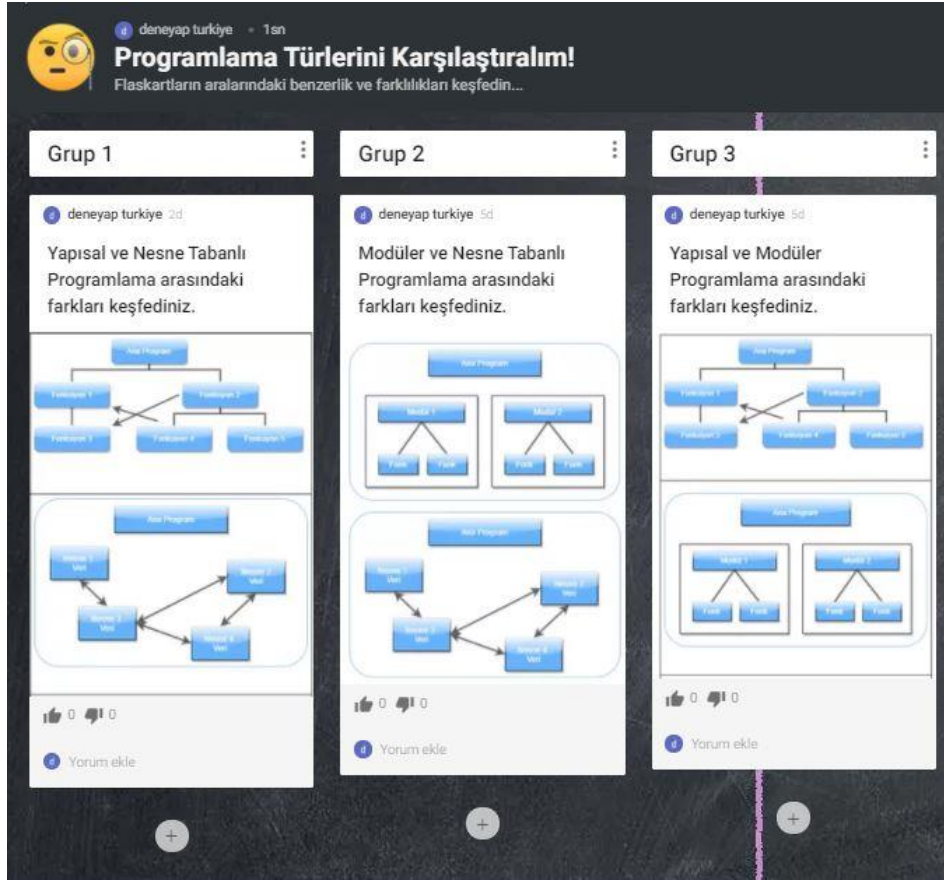
Programlama Türlerini Karşılaştıralım

Süre: 20 dk.

Kazanımlar: K3. Programlama türlerini ayırt eder.

Materyaller: EK 1: Bilgi kartları

Hazırlık: Öğitmen derse girmeden önce 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. Örnek bir tasarım Resim 9'da gösterilmiştir. EK1'deki materyalde yer alan her bir kart ayrı ayrı gruplara yüklenecektir. Örnek bir tasarım Resim 9'da gösterilmiştir.



Resim 9. Programlama Türlerini Karşılaştıralım Örnek Padlet Panosu

Uygulama: Öğitmen, sunumda anket slaytından sonra “Programlama Türlerini Karşılaştıralım” etkinliğine gelir. Öğrencilere padlet linkini paylaşılan notlar kısmından iletir. Öğrenciler burada üç ayrı grup sütunu ve her sütun altında eğitimci tarafından eklenmiş programlama türünü gösteren bilgi kartları görecektir. Öğrencileri üç farklı çalışma odasına rastgele göndereceğini bu şekilde gruplara ayrılacaklarını belirtir. Burada birinci odadakiler, padlet ortamındaki birinci görev sütununu dolduracaktır. Odalara dağılan öğrenciler grup içinde bu bilgi kartları inceleyip karşılaştırma yapacaktır. Bu karşılaştırmada iki programlama türü arasındaki farklılıkları keşfetmeleri beklenir. Örneğin yapısal ve modüler programlama bilgi kartını alan grup, bu iki programlama türü arasındaki farklı noktaları padlet ortamında grup sütununa yazar. Bu şekilde her grup kendi içerisinde 10 dk. tartışır ve görev sütunlarını doldururlar. Çalışma odası süresi 10 dk. olarak ayarlanır. Süre bitiminde ana ekrana dönen öğrencilerin padlet ortamı eğitimci tarafından ekranda paylaşılmış haldedir. Eğitimci padlet paylaşımları üzerinden konuyu aşağıdaki içeriğe benzer şekilde özetlemeye çalışır.

Konu içeriği: Programlama, talimatları bir makineye veya bilgisayara göndermek için tasarlanmış bir gösterim sunar. Programlama temel olarak bir makinenin performansını kontrol etmek veya algoritmaları ifade etmek için kullanılır. Programlamada genel olarak üç farklı tür mevcuttur.

1. Yapısal programlama, programın tek bir yapı olarak oluşturulduğu ve genellikle programlamaya yeni başlayanların küçük ve basit kodlar yazarak başladığı programlama

yaklaşımıdır. Talimatlar seri ve yapılandırılmış bir şekilde yürütülecektir. Yapısal programlama tekniğinde kod içerisinde var olan alt programlar tekrarlı olarak çağrılabilir. Böylece gereksiz kod tekrarı ortadan kalkmış olur. Yapısal bir program kolay anlaşılır ve düzenlenebilir bir programdır.

2. Modüler programlama, programın ayrı ayrı modüller içinde gruplandırıldığı bir yazılım tasarım tekniğidir. Programın daha küçük parçalara ayrılarak büyük yazılım programlarının ve sistemlerinin oluşturulmasını sağlar. Her modül ana program içinde tanımlı değişkenleri kullanabilir ve kendi verisine de sahiptir.
3. Nesne yönelimli programlama, nesnelere ve nesneler üzerindeki işlemlere dayanan programlama yapısıdır. Programcının kendi sınıfını ve prosedürlerini oluşturup bunun üzerinde işlemler yapar. Programlar, nesnelerin birbirlerine mesaj göndererek etkileşime geçmeleri üzerine kurgulanmıştır. Çok sayıdaki nesneler önceden programcıya hazır bir şekilde sunulur. Nesneye yönelik programlama, gerçek dünya varlıklarını programlamada uygulamayı amaçlar.

Bilgisayar Verileri Nasıl Saklar?

Süre: 15 dk

Kazanımlar: K4. Veri saklama birimlerini anlar.

Materyaller: Hafta 1 Sunum

Uygulama: Eğitimci sunumda “Bilgisayar Veriyi Nasıl Saklar?” başlıklı slayta gelir ve öğrencilere burada 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 sunumda ilerleyerek açıklamalarda bulunur ve konuyu aşağıdaki gibi özetler.

Konu içeriği:

Bilgisayar sistemlerindeki bütün bilgiler ikili sistemde “0” ve “1” ile temsil edilerek saklanır. İkili sistemdeki her bir basamağa bit denir. Bu durumlar, elektrik akımının ortamda var olup olmamasına karşılık gelir. Eğer elektrik akımı varsa, anlık durum 1 değerini alır, yoksa 0 değerini alır. Bit, nicelik ifade edebilmek için yeterli bir birim olmadığı için en yaygın dijital veri depolama birimi olarak 8 bitten oluşan bayt kullanılır. Yani ikili sayı sistemi özünde sadece iki seçenek sunabildiği için, çalışma mekaniklerini bir sonraki aşamaya taşıyabilmek için dizgileri kullanırız. Örneğin sadece bir bit kullanmak yerine, sekiz tane biti bir araya getirip 1 byte oluşturabiliriz.

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

8 bit = 1 bayt

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),

petabayt (PB), exabyte (EB), zettabayt (ZB) ve yottabyte (YB) olarak ifade edilir ve 1 KB 1024 bayttır.

Birim	Kısaltma	Kapasite
Bit	b	1 ya da 0
Bayt	B	8 bits
Kilobayt	KB	1024 bayt
Megabayt	MB	1024 kilobayt
Gigabayt	GB	1024 megabayt
Terabayt	TB	1024 gigabayt
Petabayt	PB	1024 terabayt
Eksabayt	EB	1024 petabayt
Zettabayt	ZB	1024 eksabayt
Yottabayt	YB	1024 zettabayt

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:

20 MB = 20480 KB

8192 MB = 8 GB

17 GB = 17408 MB

9 MB = 9216 KB

7168 EB = 7 ZB

Beni Bul ve Değiştir

Süre: 25 dk.

Kazanımlar: K5. Programlamadaki aritmetik işleçlerini kullanır.

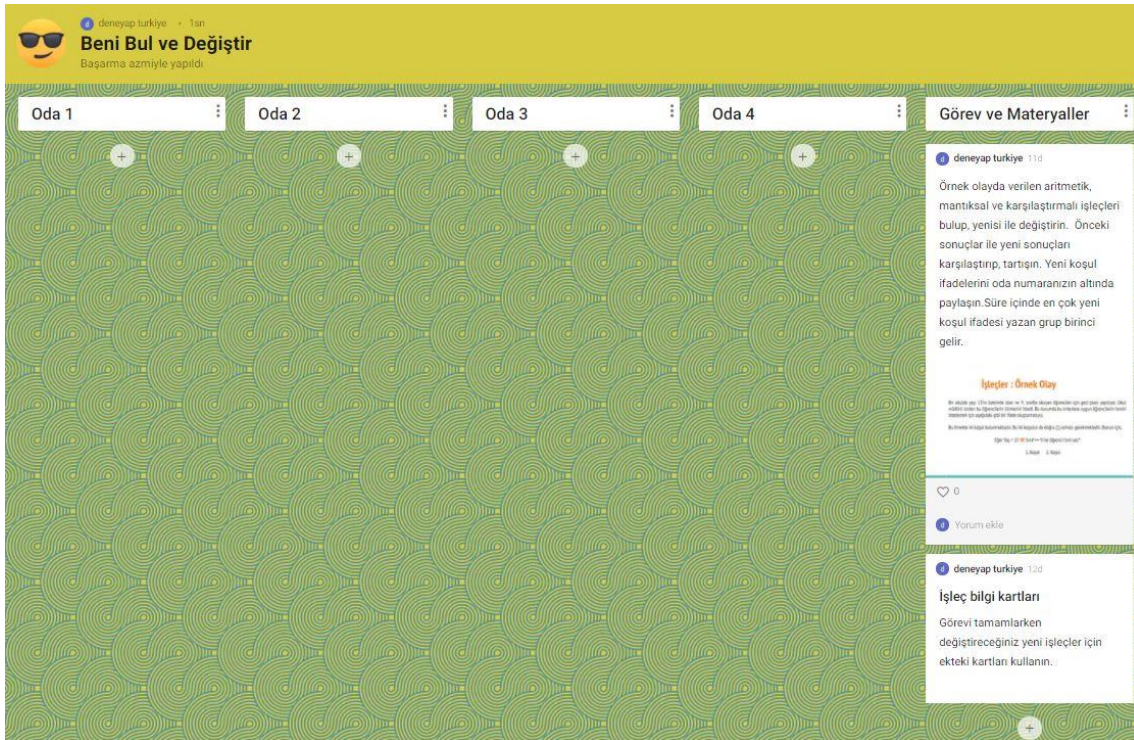
K6. Programlamadaki karşılaştırma işleçlerini kullanarak koşul ifadeleri oluşturur.

K7. Programlamadaki mantıksal işleçleri kullanır.

Materyaller: EK 2. İşleç bilgi kartları

EK 3. Örnek olay sunum kartı

Hazırlık: Örnek linkteki gibi materyalleri barındıran raf temalı ve materyallerin erişilebilen padlet linki oluşturulur. Örnek bir tasarım Resim 10'da gösterilmiştir. EK2 ve EK3'teki materyaller görev ve materyal sütununda ayrı olacak şekilde yüklenecektir. Örnek bir tasarım Resim 10'da gösterilmiştir.



Resim 10. Beni Bul ve Değiştir Örnek Padlet Panosu

Uygulama: Eğitimci işleçler slaytında işleç kavramı hakkında kısa bir bilgi verir. Eğitimci slaytta ilerleyerek öğrencilere bir örnek olay verir. Bu örnek olay üzerinde öğrencilerin yapacakları bir görev bulunmaktadır. Eğitimci bu örnek olay üzerinde gruplar halinde birtakım görevler yapacaklarını belirtir. Her grup padlet ortamında bir görev sütunu ile karşılaşır. Örneğin birinci çalışma odasında yer alan öğrenciler birinci sütundaki görevi örnek olaya bağlı olarak 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 2). Bu kartlarda örnek olayda yer alan aritmetik, karşılaştırma ve mantıksal işleçleri bulup değiştirecekleri yeni işleç hakkında bilgi ve örnek kullanımlar yer alır.

Öğrencilere paylaşılan notlar kısmında raf temasıyla oluşturulmuş padlet linkini atar. Öğrenciler 10 dk.'lık çalışma odalarına gönderilmek üzere beşerli dört gruba ayrılır. Gruplar örnek olaydaki aritmetik, mantıksal ve karşılaştırmalı işlemleri bulup, EK 2'de yer alan farklı işlemler ile değiştirecektir. Öğrenciler grup olarak padlette, ilgili grup sütunu altında işlemler bilgi kartlarını kullanarak yeni koşul ifadesini 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 erişmesini sağlar.

Eğitime Öneriler: Ders sonunda eğitmen padlet etkinliğinin pdf çıktısını alarak ÖYS üzerinde ilgili hafta altına ekler ve bu şekilde öğrenci ürünlerini ders sonrası erişimine açar.

Konu içeriği:

Aritmetik İşlemler

Matematiksel hesaplamalar bilgisayar programlarında yaygın olarak kullanılmaktadır. Programlama dilinde bir işlem, bir değer veya değişken üzerinde çalışan bir semboldür. İki sayıyı birbirine eklemek (3+5) gibi basit bir hesaplama yapabilen ya da $P(x) = 2x^3 - 3x^2 + 6x + 4$ gibi karmaşık bir denklemi çözebilen bir program yazabiliriz. Programlama dilinde bu iki ifade aritmetik ifadeler ve bu ifadelerde kullanılan artı (+), eksi (-) gibi semboller de aritmetik işlemler olarak adlandırılır. Bu ifadelerdeki 3, 5 ve x gibi değerler de işlenen olarak adlandırılır. İfadelerin matematiksel yazılımlarının uygun bir şekilde bilgisayar kodlanmasının gerçekleştirilmesi gerekmektedir. Aşağıdaki tablodaki örnekleri inceleyiniz.

Not: Söзде kod bilgisayar kodlamasında '^' sembolü üst alma olarak kullanılmaktadır.

Tablo 2. Matematiksel ifadeler ve bilgisayar kodlamaları

Matematiksel Yazılım	Bilgisayar Kodlanması
$x - 2y + 4xy$	$x-2*y+4*x*y$
$2x^2 - 3xy + y^2 + x^2y^3$	$2*(x^2)-3*x*y+(y^2)+(x^2)*(y^3)$
$x^2 + \frac{x}{5} + 3x^2y - \frac{x}{y}$	$x*x+x/5+3*(x^2)*y-x/y$
$\frac{2x^3 + 4y^2}{xy}$	$(2*(x^3)+4*(y^2)) / (x*y)$
$\sqrt{x+y} + \frac{3xy}{xy^2 - y}$	$(x+y)^(1/2)+(3*x*y) / (x*(y^2)-y)$

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

Tablo 3. Aritmetik işlemler

İşleç	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 bölümünden kalanı verir.	$x \% y$
++	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.

++ veya -- işlemleri değişken değerlerini 1 arttırmak ya da 1 azaltmak 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
```

Karşılaştırma İşlemleri

Bu işlemlerin amacı iki değişken ya da değişken grubunu belirtilen şarta göre karşılaştırmaktır. Bu karşılaştırmalar aynı türdeki değişkenler arasında olmalıdır. Bu işlemleri özellikle ilerleyen haftalarda göreceğimiz koşul yapıları ve döngülerde kullanılır. Yapılan karşılaştırmalar doğruysa “1” yanılsa “0” sonucunu döndürür.

Tablo 4. Karşılaştırma işlemleri

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 İşlemler

Mantıksal işlemler 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 işlemleri kullanılmalıdır.

Tablo 5. Mantıksal işlemler

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

Mantıksal işlemlerin 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

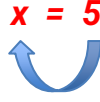
Süre: 20 dk.

Kazanımlar: K8. Değişkenlere değer atama türlerini tanır.

Materyaller: Hafta 1 Sunum

Uygulama: Öğretmen 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 değişkendir ve atama operatörünün sağ taraftaki işleneni bir değerdir.



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 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.	$x \% = y$ $(x = x \% y)$

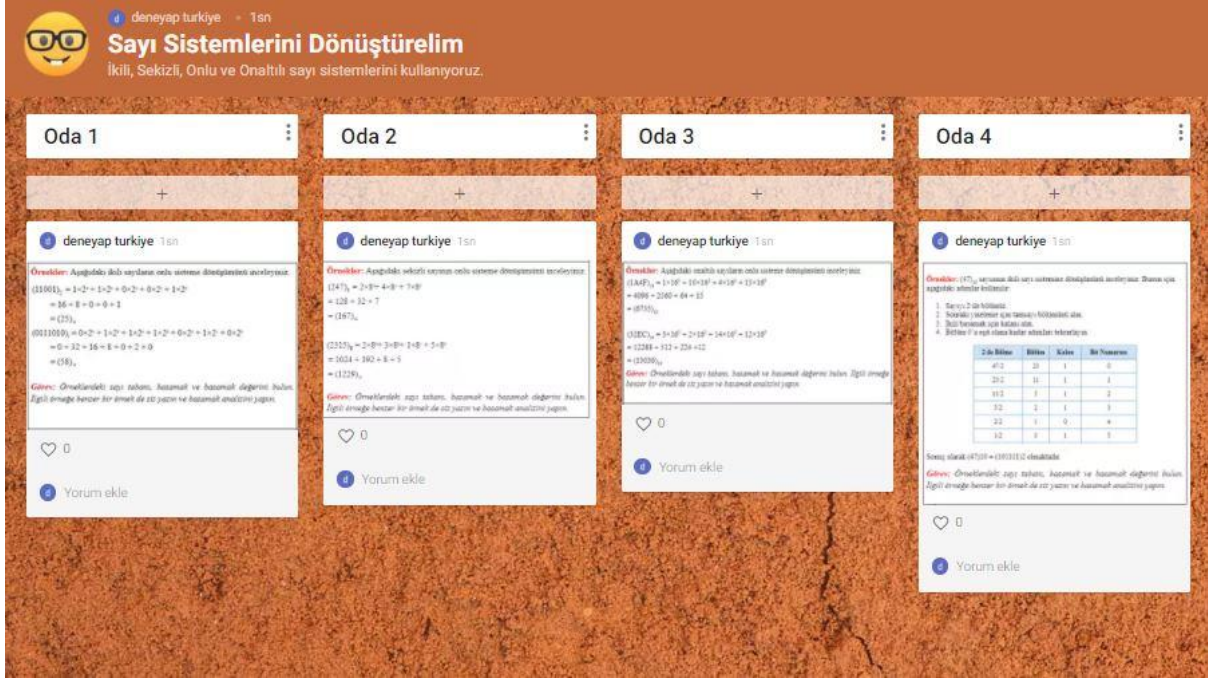
Sayı Sistemlerini Keşfedelim

Süre: 20 dk.

Kazanımlar: K9. İkili, sekizli, onlu ve onaltılı sayı sistemlerini kavrar.

Materyaller: EK 4. Veri Dönüştürme Örnek Olay Kartları

Hazırlık: Öğrenci ürünlerinin paylaşımı için raf temalı bir padlet ortamı oluşturulur. Örnek bir tasarım Resim 11'de gösterilmiştir. Ek4'teki materyalde yer alan her bir örnek olay kartı gruplara yüklenecektir. Örnek bir tasarım Resim 11'de gösterilmiştir.



Resim 11. 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. Eğitimci padlet ortamında verilen yanıtlar üzerinden 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.

$$5429 = 5 \times 10^3 + 4 \times 10^2 + 2 \times 10^1 + 9 \times 10^0$$

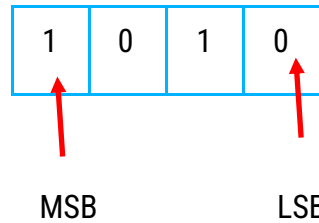
$$387 = 3 \times 10^2 + 8 \times 10^1 + 7 \times 10^0$$

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.



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

Örnek: Aşağıdaki ikili sayının basamak analizini inceleyiniz.

$$\begin{aligned}
 (11001)_2 &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= 16 + 8 + 0 + 0 + 1 \\
 &= (25)_{10}
 \end{aligned}$$

Örnek: Aşağıdaki ikili sayının basamak analizini inceleyiniz.

$$\begin{aligned}(0111010)_2 &= 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 0 + 32 + 16 + 8 + 0 + 2 + 0 \\ &= (58)_{10}\end{aligned}$$

Sekizli Sayı Sistemi

Sekizli sayı sisteminde 0 ile 7 arasında sadece sekiz basamak vardır. Her sayı, bu sayı sisteminde 0, 1, 2, 3, 4, 5, 6 ve 7 ile temsil edilir. Sekizli sayı sisteminin tabanı 8'dir, çünkü sadece 8 basamaklıdır.

8^7	8^6	8^5	8^4	8^3	8^2	8^1	8^0
-------	-------	-------	-------	-------	-------	-------	-------

Herhangi bir sekizli sayının onlu eşdeğeri, basamak değeri ile her bir basamak ağırlığının çarpımının toplamıdır.

Örnek: Aşağıdaki sekizli sayının basamak analizini inceleyiniz.

$$\begin{aligned}(247)_8 &= 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 \\ &= 128 + 32 + 7 \\ &= (167)_{10}\end{aligned}$$

Örnek: Aşağıdaki sekizli sayının basamak analizini inceleyiniz.

$$\begin{aligned}(2315)_8 &= 2 \times 8^3 + 3 \times 8^2 + 1 \times 8^1 + 5 \times 8^0 \\ &= 1024 + 192 + 8 + 5 \\ &= (1229)_{10}\end{aligned}$$

Onaltılı Sayı Sistemi

Onaltılı bir sayı sistemi, 0'dan 9'a ve A'dan F'ye kadar on altı alfasayısal değere sahiptir. Her sayı, bu sayı sisteminde 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E ve F ile temsil edilir. Onaltılı sayı sisteminin tabanı 16'dır. Çünkü 16 alfasayısal değere sahiptir. Burada A:10, B:11, C:12, D:13, E:14 ve F:15 değerine sahiptir.

16^7	16^6	16^5	16^4	16^3	16^2	16^1	16^0
--------	--------	--------	--------	--------	--------	--------	--------

Herhangi bir onaltılı sayının onlu eşdeğeri, basamak değeri ile her bir basamak ağırlığının çarpımının toplamıdır.

Örnek: Aşağıdaki onaltılı sayının basamak analizini inceleyiniz.

$$\begin{aligned}(1A4F)_{16} &= 1 \times 16^3 + 10 \times 16^2 + 4 \times 16^1 + 15 \times 16^0 \\ &= 4096 + 2560 + 64 + 15 \\ &= (6735)_{10}\end{aligned}$$

Örnek: Aşağıdaki onaltılı sayının basamak analizini inceleyiniz.

$$\begin{aligned}(32EC)_{16} &= 3 \times 16^3 + 2 \times 16^2 + 14 \times 16^1 + 12 \times 16^0 \\ &= 12288 + 512 + 224 + 12 \\ &= (13036)_{10}\end{aligned}$$

Aşağıdaki tabloda ikili, sekizli, onlu ve onaltılı sayı sistemleri arasındaki ilişki gösterilmektedir.

Tablo 10. Sayı sistemleri dönüşümü

Onaltılı	Onlu	Sekizli	İkili
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111

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: $(47)_{10}$ sayısını ikili sayı sistemine dönüştürelim.

2 ile Bölme	Bölüm	Kalan	Bit Numarası
47/2	23	1	0
23/2	11	1	1
11/2	5	1	2
5/2	2	1	3
2/2	1	0	4
1/2	0	1	5

Sonuç olarak $(47)_{10} = (101111)_2$ olmaktadır.

Örnek: $(173)_{10}$ sayısını ikili sayı sistemine dönüştürelim.

2 ile Bölme	Bölüm	Kalan	Bit Numarası
173/2	86	1	0
86/2	43	0	1
43/2	21	1	2
21/2	10	1	3
10/2	5	0	4
5/2	2	1	5
2/2	1	0	6
1/2	0	1	7

Sonuç olarak $(173)_{10} = (10101101)_2$ olmaktadır.

C. Sonuç

Süre: 10 dk.

Materyal: EK 5 Hafta 1 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 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 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 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 ya da ÖYS üzerinden öğrencilerle paylaşır.

Tasarlayıcı: Otonom bir otomobil tasarlamak istiyoruz. Yoldaki çizgileri takip eden, hız sınırlarına dikkat eden, Otonom bir araç için giriş-çıkış, CPU, Bellek görevleri neler olmalıdır? Çizerek gösterelim.

Yanıt: Bu soru öğrencilerin hayal yeteneklerini geliştirmek için verilmiştir. Tek bir doğru cevabı yoktur. Giriş olarak: Kamera Çıkış olarak: Direksiyon, Gaz ve Fren pedalları düşünülebilir. CPU: görüntü işlemek için, bellek ise hız sınırlarının tutulması için kullanılabilir.

Tasarlayıcı: Aşağıdaki ifadeyi bilgisayar kodlanmasına çeviriniz.

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Yanıt: $x = (-b + ((b^2 - 4*a*c)^{(1/2)})) / (2*a)$

Analizci: Aşağıda verilen Fizik dersini geçme kriterlerini mantıksal koşul olarak tek satırda ifade ediniz.

Tüm deneyleri okulda (A) ya da evde (B) yapmak

Tüm ödevleri teslim etmek (C)

Sınav ortalaması (D) 50'den büyük

Devamsızlığı (E) 20 günden az olmak

Yanıt: $(A \parallel B) \&\& C \&\& (D > 50) \&\& (E < 20)$

Kodlayıcı: 8-bitlik siyah/beyaz görüntülerin her piksel (en küçük görüntü parçası) değeri 0-255 arasında bir değer alır. Genişliği 200 piksel ve yüksekliği 300 piksel olan bir görüntü için kaç KB hafıza gerekir?

Yanıt: $200 * 300 * 1 \text{ Bayt} = 60000 \text{ Bayt} = 58.6 \text{ KBayt}$

Tasarlayıcı: Aşağıdaki tabloyu bilgisayarda sayısal olarak tutmak istersek; ikili ve onlu tabanda olması gereken değer kaçtır?

(0:0 ve X:1 olarak alınmalı.)

0	X	X	0	X	0	X	0
---	---	---	---	---	---	---	---

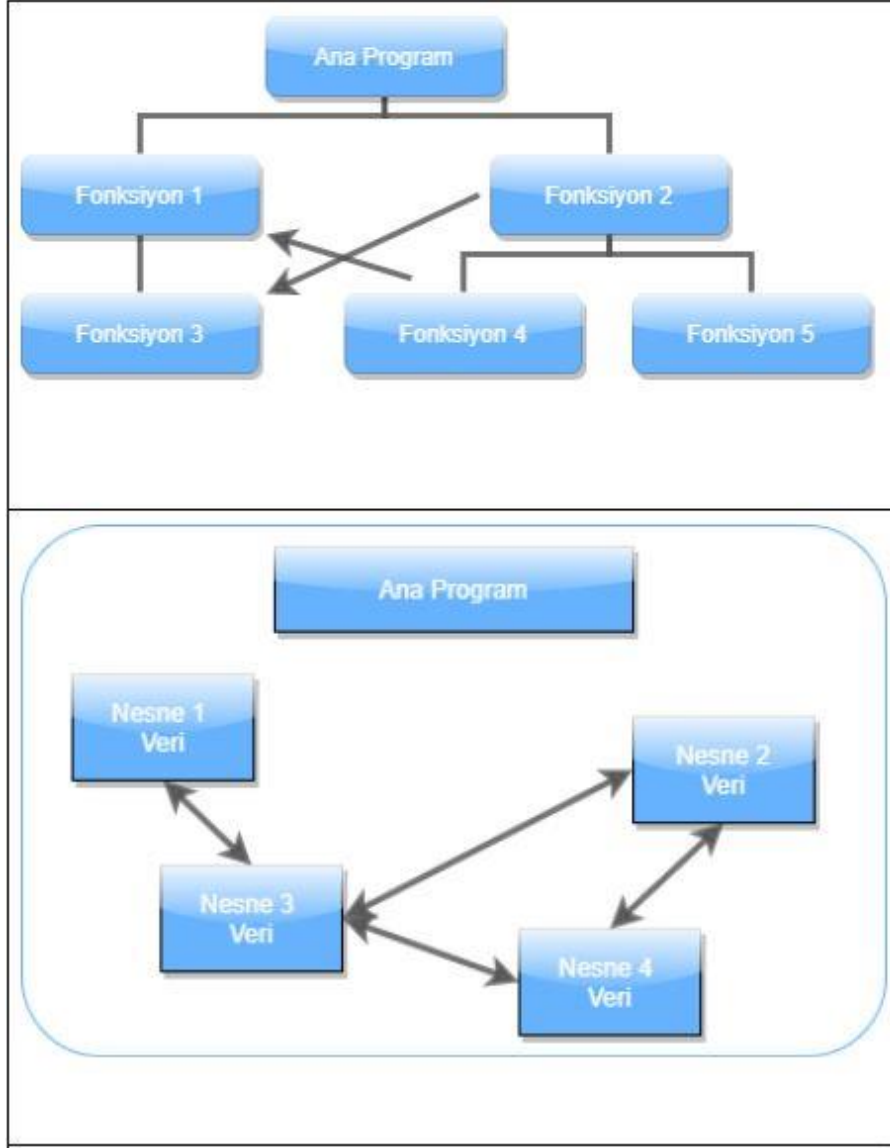
Yanıt: $(01101010)_2 = (106)_{10}$

Hafta 1. Ders Materyalleri

EK 1. Bilgi Kartları

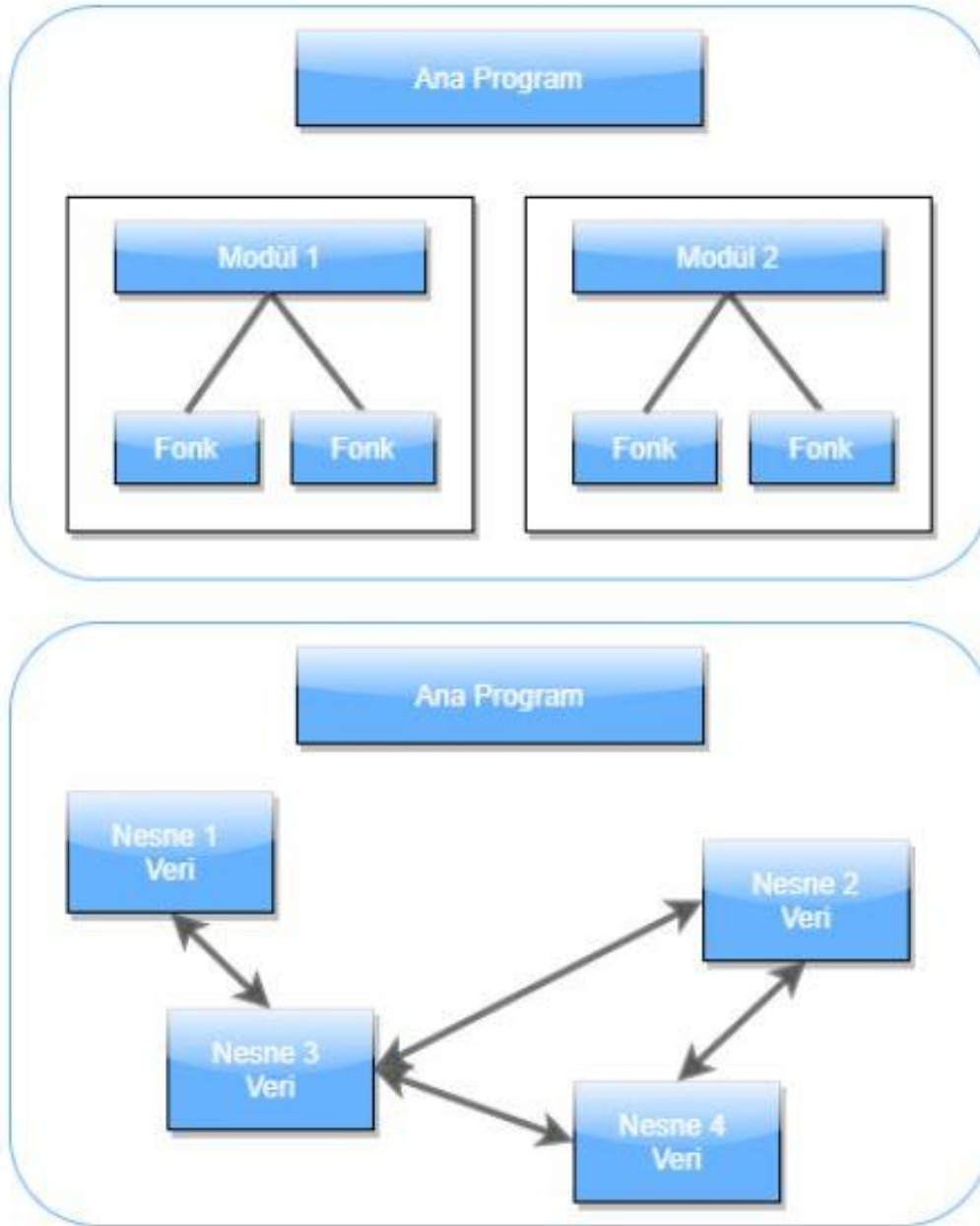
Materyal 1. Grup 1 Materyali

Grup 1 Görev: Yapısal ve Nesne Yönelimli Programlama arasındaki farkları keşfediniz.



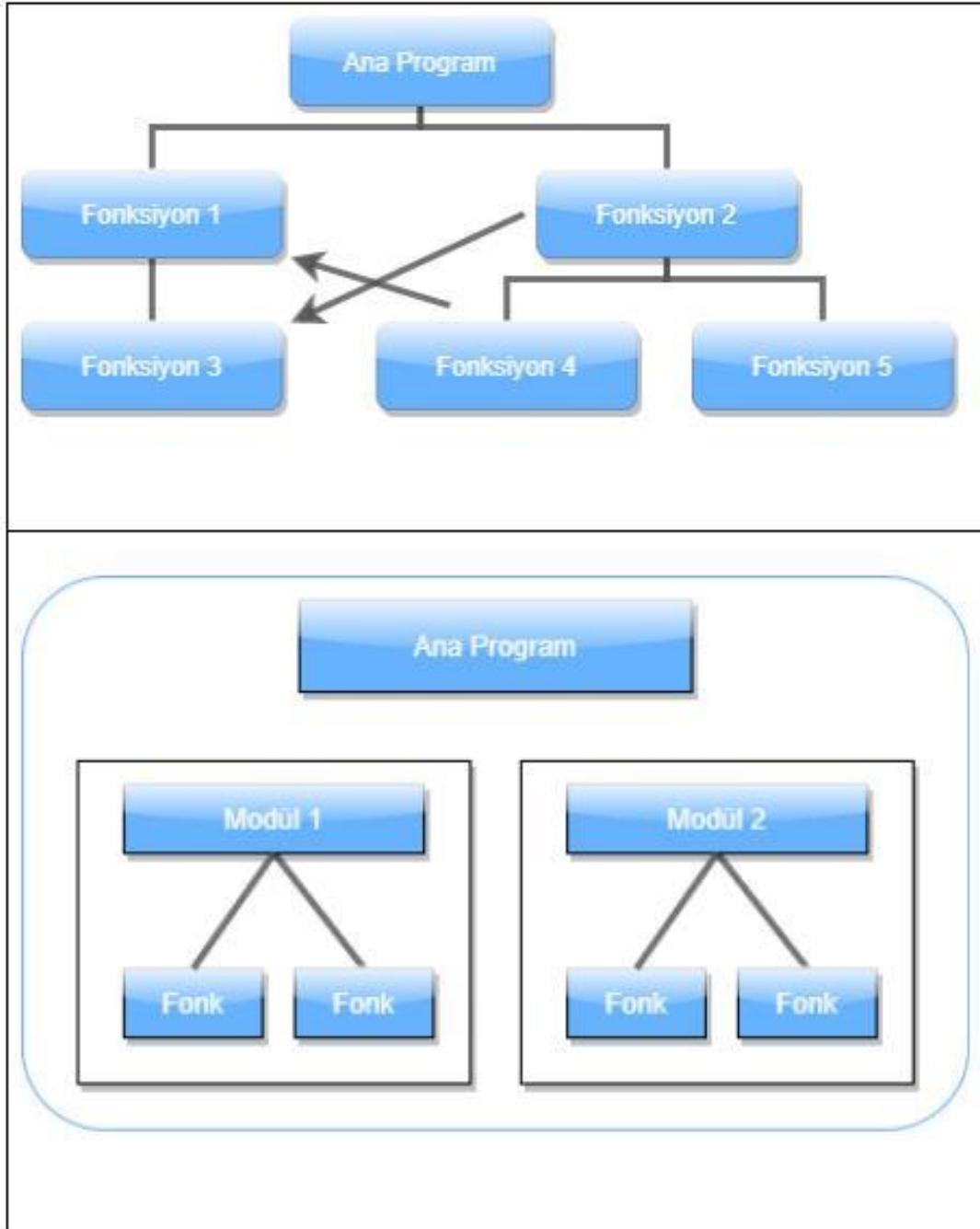
Materyal 2. Grup 2 Materyali

Grup 2 Görev: Modüler ve Nesne Yönelimli Programlama arasındaki farkları keşfediniz.



Materyal 3. Grup 3 Materyali

Grup 3 Görev: Yapısal ve Modüler Programlama arasındaki farkları keşfediniz.



EK 2. İşleç Bilgi Kartları**Aritmetik İşlemler**

Arda'nın yaşı $a=5$ ve Duru'nun yaşı $d=4$ 'tür. Arda ve Duru'nun yaşları arasında tanımlanabilecek aritmetik işlemleri inceleyiniz.

Tablo 11. Aritmetik İşlemler ve Örnekleri

İşlem	Açıklama	Sonuç
$a + d$	Arda ve Duru'nun yaşlarını toplama	$15+4=19$
$a - d$	Arda'nın yaşından Duru'nun yaşını çıkarma	$15-4=11$
$a * d$	Arda ve Duru'nun yaşlarını çarpma	$15*4=60$
a / d	Arda'nın yaşını Duru'nun yaşı ile bölme	$15/4=3$
$a \% d$	Arda'nın yaşını Duru'nun yaşına göre mod alma	$15\%4=3$
$a++$	Arda'nın yaşını bir artırma	16
$d--$	Duru'nun yaşını bir azaltma	3

Karşılaştırma İşlemleri

Arda'nın yaşı $a=5$ ve Duru'nun yaşı $d=4$ 'tür. Arda ve Duru'nun yaşları arasında tanımlanabilecek karşılaştırma işlemleri inceleyiniz.

Tablo 12. Karşılaştırma İşlemleri ve Örnekleri

İşlem	Açıklama	Sonuç
$a < d$	Arda'nın yaşı, Duru'nun yaşından küçüktür.	0 (Yanlış)
$a > d$	Arda'nın yaşı, Duru'nun yaşından büyüktür.	1 (Doğru)
$a \leq d$	Arda'nın yaşı, Duru'nun yaşına küçük eşittir.	0 (Yanlış)
$a \geq d$	Arda'nın yaşı, Duru'nun yaşına büyük eşittir.	1 (Doğru)
$a == d$	Arda'nın yaşı, Duru'nun yaşına eşittir.	0 (Yanlış)
$a != d$	Arda'nın yaşı, Duru'nun yaşına eşit değildir.	1 (Doğru)

Mantıksal İşlemler

Arda'nın yaşı $a=5$ ve Duru'nun yaşı $d=4$ 'tür. Arda ve Duru'nun yaşları arasında tanımlanabilecek mantıksal işlemleri inceleyiniz.

Tablo 13. Mantıksal İşlemler ve Örnekleri

İşlem	Açıklama	Sonuç
$((a==5) \&\& (d>5))$	Arda 5 yaşındadır VE Duru 5 yaşından büyüktür.	0 (Yanlış)
$((a==5) \parallel (d>5))$	Arda 5 yaşındadır VEYA Duru 5 yaşından büyüktür.	1 (Doğru)
$!(a==4)$	Arda 4 yaşında değildir.	1 (Doğru)

EK 3. Örnek Olay Sunum Kartı

İşleçler : 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 VE Sınıf == 9 ise öğrenci ismi yaz*

1. Koşul 2. Koşul

GÖREV: Örnek olayda verilen aritmetik, mantıksal ve karşılaştırma işleçleri bulup, yenisi ile değiştirin. Önceki sonuçlar ile yeni sonuçları karşılaştırıp, tartışın. Yeni koşul ifadelerini oda numaranızın altında paylaşın. Süre içinde en çok yeni koşul ifadesi yazan grup birinci gelir.

EK 4. Veri dönüştürme örnek olay kartları

Örnekler: Aşağıdaki ikili sayıların onlu sisteme dönüşümünü inceleyiniz.

$$(11001)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 16 + 8 + 0 + 0 + 1$$

$$= (25)_{10}$$

$$(0111010)_2 = 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 0 + 32 + 16 + 8 + 0 + 2 + 0$$

$$= (58)_{10}$$

Görev: Örneklerdeki sayı tabanı, basamak ve basamak değerini bulun. İlgili örneğe benzer bir örnek de siz yazın ve basamak analizini yapın.

Örnekler: Aşağıdaki sekizli sayının onlu sisteme dönüşümünü inceleyiniz.

$$(247)_8 = 2 \times 8^2 + 4 \times 8^1 + 7 \times 8^0$$

$$= 128 + 32 + 7$$

$$= (167)_{10}$$

$$(2315)_8 = 2 \times 8^3 + 3 \times 8^2 + 1 \times 8^1 + 5 \times 8^0$$

$$= 1024 + 192 + 8 + 5$$

$$= (1229)_{10}$$

Görev: Örneklerdeki sayı tabanı, basamak ve basamak değerini bulun. İlgili örneğe benzer bir örnek de siz yazın ve basamak analizini yapın.

Örnekler: Aşağıdaki onaltılı sayıların onlu sisteme dönüşümünü inceleyiniz.

$$(1A4F)_{16} = 1 \times 16^3 + 10 \times 16^2 + 4 \times 16^1 + 15 \times 16^0$$

$$= 4096 + 2560 + 64 + 15$$

$$= (6735)_{10}$$

$$(32EC)_{16} = 3 \times 16^3 + 2 \times 16^2 + 14 \times 16^1 + 12 \times 16^0$$

$$= 12288 + 512 + 224 + 12$$

$$= (13036)_{10}$$

Görev: Örneklerdeki sayı tabanı, basamak ve basamak değerini bulun. İlgili örneğe benzer bir örnek de siz yazın ve basamak analizini yapın.

Örnekler: $(47)_{10}$ sayısının ikili sayı sistemine dönüşümünü inceleyiniz. Bunun için aşağıdaki adımlar kullanılır:

Sayıyı 2 ile bölünüz.

Sonraki yineleme için tamsayı bölümünü alın.

İkili basamak için kalanı alın.

Bölüm 0'a eşit olana kadar adımları tekrarlayın.

2 ile Bölme	Bölüm	Kalan	Bit Numarası
47/2	23	1	0
23/2	11	1	1
11/2	5	1	2
5/2	2	1	3
2/2	1	0	4
1/2	0	1	5

Sonuç olarak $(47)_{10} = (101111)_2$ olmaktadır.

Görev: Örneklerdeki sayı tabanı, basamak ve basamak değerini bulun. İlgili örneğe benzer bir örnek de siz yazın ve basamak analizini yapın.

EK 5. 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. Algoritma temel kavramlarını bilir.
- K2. İhtiyaç duyulan algoritmanın tasarımını sağlar.
- K3. Bir algoritmanın akış şemasını bilgisayar ortamında çizer.
- K4. Değişken kavramının özelliklerini bilir.
- K5. Akış diyagramlarını sözde koda dönüştürür.
- K6. Akış diyagramında değişkenleri ayırt eder.

Haftanın Amacı ve Materyaller

Bu haftanın amacı bir problemi çözmek için kod yazmada gerekli algoritmik düşünme yeteneğini geliştirmek ve problemin çözümünü bilgisayar kullanarak aktarmaktır.

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

Önerilen Ders Akışı (50 + 50 + 40 + 50 dk.)

GİRİŞ

Aklına Ne Geliyor? (10 dk.)

GELİŞME

Algoritmayı Tanıyalım (20 dk.)

Algoritmaları Eşleştirelim (20 dk.)

EK 1. Algoritmaları Eşleştirelim Çalışma Kâğıdı

Ders Arası (10 dk.)

Çiftleri Toplayalım (30 dk.)

EK 2. Çiftleri Toplama Algoritması

EK 3. Akış Diyagramları Bilgi Afişi

EK 4. App.diagrams.net Kullanım Kılavuzu

Algoritma Terimlerini Keşfedelim (20 dk.)

Ders Arası (10 dk.)

Otomatik Park Edelim (20 dk.)

Algoritmayı Test Edelim (20 dk.)

EK 5. Kod Çalıştırma Tablosu

SONUÇ (10 dk.)

EK 6. Hafta 2 Süreli Ödev (50 dk.)

EK 7 Ders Dışı Aktivite Sunusu

EK 8 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 2 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır.

Aklına Ne Geliyor?

Süre: 10 dk.

Kazanımlar: K1. Algoritma temel kavramlarını bilir.

Materyal: 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 yazması ile *ön bilgileri hatırlamaya* çalışırlar. Eğitmen algoritmanın ne olduğu ile ilgili konuya öğrenci paylaşımları üzerinden *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 Tanıyalım

Süre: 20 dk.

Kazanımlar: K1. Algoritma temel kavramlarını bilir.

Materyaller: Hafta 2 Sunum

Hazırlık: Eğitimci 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. Eğitimci öğrencilere sunum üzerinden aşağıda verilen soruyu sözlü olarak sorarak düşüncelerini ister ve anketi başlatır. Ankete verilen yanıtlar ile öğrencilerin performanslarını göstermeleri beklenir.

Anket Sorusu: Verilen kek algoritmaları arasındaki benzerlik ve farklılıkları düşünün. Sizce en iyi yazılan kek algoritması hangisidir?

Öğrencilerden ankete yanıtları alındıktan sonra, doğru algoritma algoritmanın özellikleri üzerine düşünceleri istenir. “Diğer algoritmalara bakarak, doğru algoritma olduğuna nasıl karar verdiniz” şeklinde sorular yönelterek beyin fırtınası yaptırır. Eğitimci doğru algoritma yanıtını veren öğrencilere söz hakkı tanır. Bu şekilde onların algoritmanın özelliklerini keşfetmelerini sağlar. Slaytı ilerleterek iyi bir algoritmanın özelliklerini aşağıdaki gibi özetler.

- Bir başlangıç noktası olmalı
- Basit olmalı
- Algoritma içindeki ifadeler herkes tarafından aynı şekilde anlaşılabilir olmalı
- Yorum gerektirmemeli ve belirsiz ifadelerle 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

Algoritmaları Eşleştirelim

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 yansıtma beklentisi bekler. Örnek anket sorusu öğrencilere aşağıdaki gibi yöneltilir.

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

- A. P1. İki sayıyı toplama
- B. P2. Bir sayının karesini alma

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

- A. MY: Metinsel Yazım
- B. SK: Sözde Kod
- C. 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.

Eğitime 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.

Çiftleri Toplayalım

Süre: 30 dk.

Kazanımlar: K2. İhtiyaç duyulan algoritmanın tasarımını sağlar.

K3. Bir algoritmayı akış şemasında gösterir.

K4. Bir algoritmanın akış şemasını bilgisayar ortamında çizer.

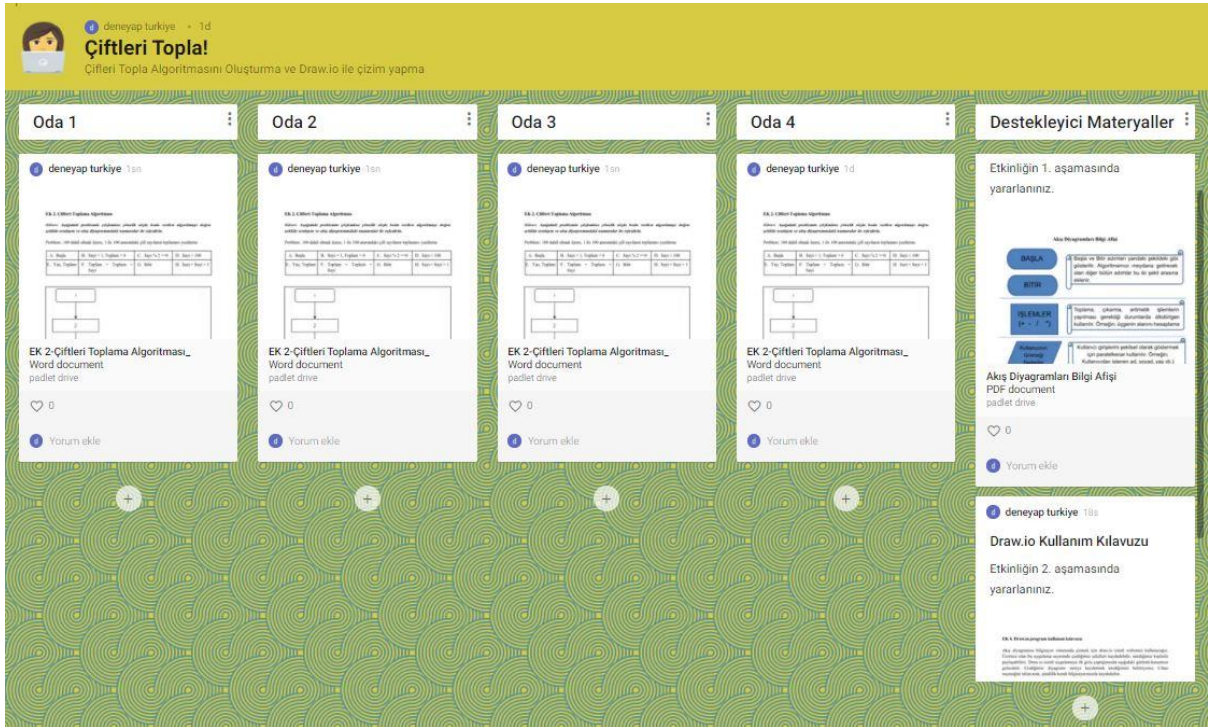
Materyaller: Dijital Tartışma Panosu: Grup Çalışması

EK 2. Çiftleri Toplama Algoritması

EK 3. Akış Diyagramları Bilgi Afişi

EK 4. App.diagrams.net Kullanım Kılavuzu

Hazırlık: Öğitmen derse girmeden önce grup çalışması için raf temalı bir dijital tartışma panosu padlet ortamda oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletacaktır. Öğitmen linkin gönderilmesi için sohbet ortamını da kullanabilir. EK 2, EK 3 ve EK 4 materyali dijital tartışma panosuna yüklenecektir. Örnek bir tasarım Resim 12’de gösterilmiştir.



Resim 12. Çiftleri Topla Örnek Dijital Tartışma Panosu

Uygulama: Bu etkinlik toplam iki aşamadan oluşmaktadır. Öğitmen bu etkinlikte öğrencileri dört gruplu çalışma odalarına gönderecektir. Süre olarak 10 dk. belirlenir. Öğrencilere çalışma odalarına gitmeden önce dijital tartışma panosunun linki paylaşılan notlar kısmından iletir. Panoda öğrenciler EK 2’de yer alan çiftleri toplama algoritmasını akış şemasındaki numaralarla eşleştirmelidir. Bunu yaparken EK 3’te yer alan akış diyagramları bilgi afişini incelemeleri ve şekiller üzerinde tartışmaları istenir. Bu şekilde öğrencilere uygun materyal sunulması ve öğrenene rehberlik edilmektedir. Öğrenciler tartışma sonrası eşleştirmelerini grup sütunu altında yazarak performanslarını gösterir. Ana odaya dönen öğrenciler için öğretmen ekran paylaşımında panoyu göstermektedir. Burada öğrencilerin cevapları üzerinden gruplar arası benzerlik ve farklılıklara dikkat çekilerek, doğru bilgi keşfettirilir. Öğrencilere geri bildirim verir. Akış diyagramı bilgi afişini açarak konuyu 5 dk. içinde özetler.

Eğitmen ikinci bir grup çalışması daha yapacaklarını belirtir. Öğrenciler tekrar dörderli gruplara ayrılacaktır. İlk grupların bu aşamada değişeceği belirtilir. Yeni gruplardan bir önceki grupların eşleştirmelerini değerlendirerek doğru çiftleri toplama algoritmasının akış diyagramını çizmelerini beklenmektedir. Bunun için öğrenciler aynı tartışma panosuna tekrar 10 dk. süre ile gönderilir. Öğrenciler atandığı odada yer alan bir önceki grubun algoritma sıralamasında hata varsa sıralamayı düzelterek app.diagrams.net programını açar ve diyagramları çizmeye çalışır. Tüm gruplar kendi içlerinde bir lider seçer. Grup lideri app.diagrams.net sayfasını ekran paylaşımına açar ve grup olarak tek bir çizim tamamlarlar. Öğrencilerden bu çizim aşamasında EK 4'teki kullanım kılavuzundan yardım almaları beklenir. Ana odaya döndükten sonra grupların çizimleri kontrol edilir. Eğitmen çizimlerde hatalı olanlar varsa program üzerinden çizimi tekrar yaparak konuyu özetler.

Algoritma Terimlerini Keşfedelim

Süre: 20 dk.

Kazanımlar: K4. Değişken kavramının özelliklerini bilir.

Materyaller: Hafta 2 Sunum

Hazırlık: Hafta 2 Sunumunda algoritma terimlerini keşfet slayt açılır.

Uygulama: Eğitmen bu etkinlikte sunum üzerinden ilerleyecektir. Öğrencilere sunumda bir önceki etkinlikte yürüttükleri çiftleri toplama algoritması 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 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, yukarıda sayılan 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 ikiye 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 değer eklenmesi ya da var olan değer yeni bir değerlerle çarpılarak oluşan bu yeni değer 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 2 Sunum

Hazırlık: Hafta 2 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: Eğitmen öğrencilere algoritmadaki 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.

Algoritmayı Test Edelim

Süre: 20 dk.

Kazanımlar: K4. Değişken kavramının özelliklerini bilir.

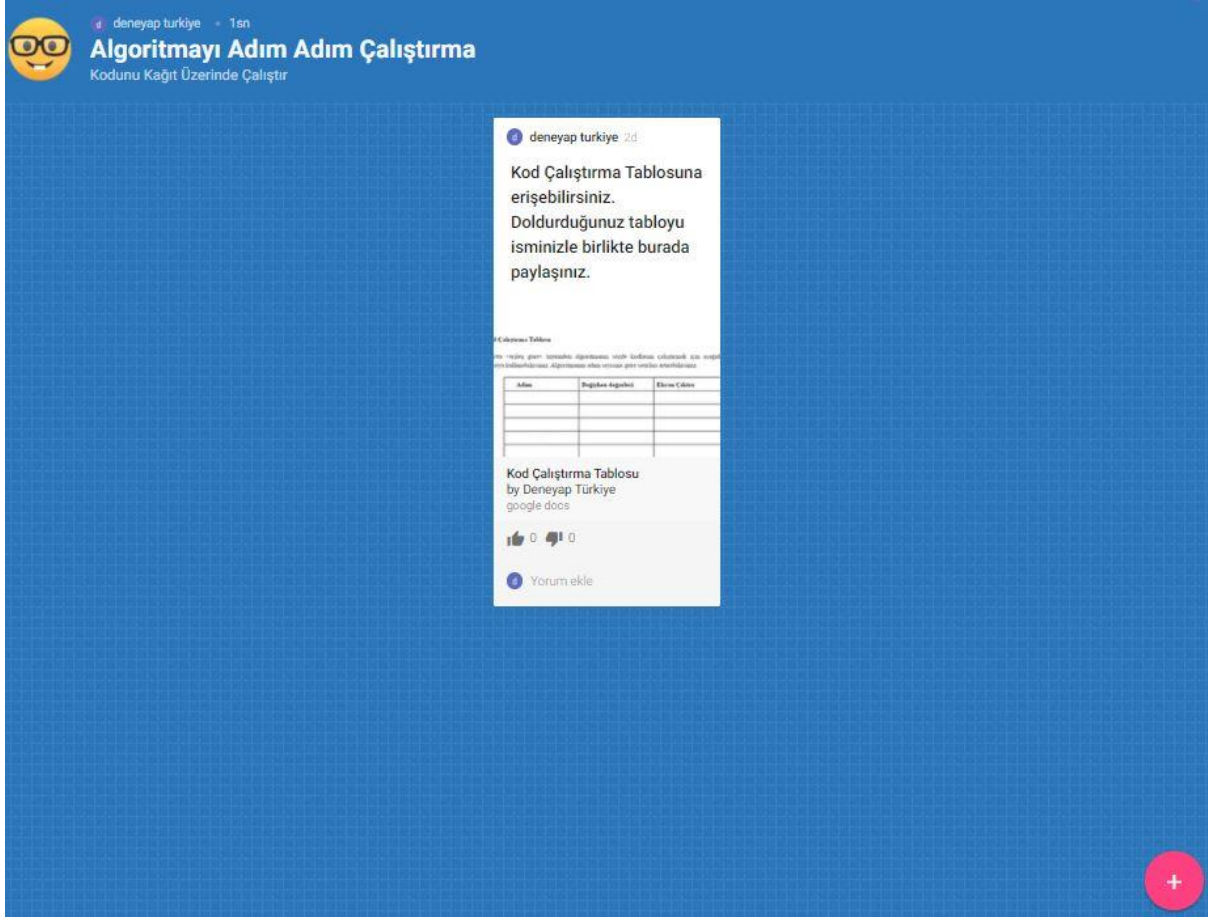
K5. Akış diyagramlarını sözde koda dönüştürür.

K6. Akış diyagramında değişkenleri ayırt eder.

Materyaller: Dijital Tartışma Panosu: Bireysel Çalışma

EK 5. Kod Çalıştırma Tablosunu Doldurma

Hazırlık: Hafta 2 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 13'te gösterilmiştir. EK 5 materyali dijital tartışma panosuna yüklenecektir.



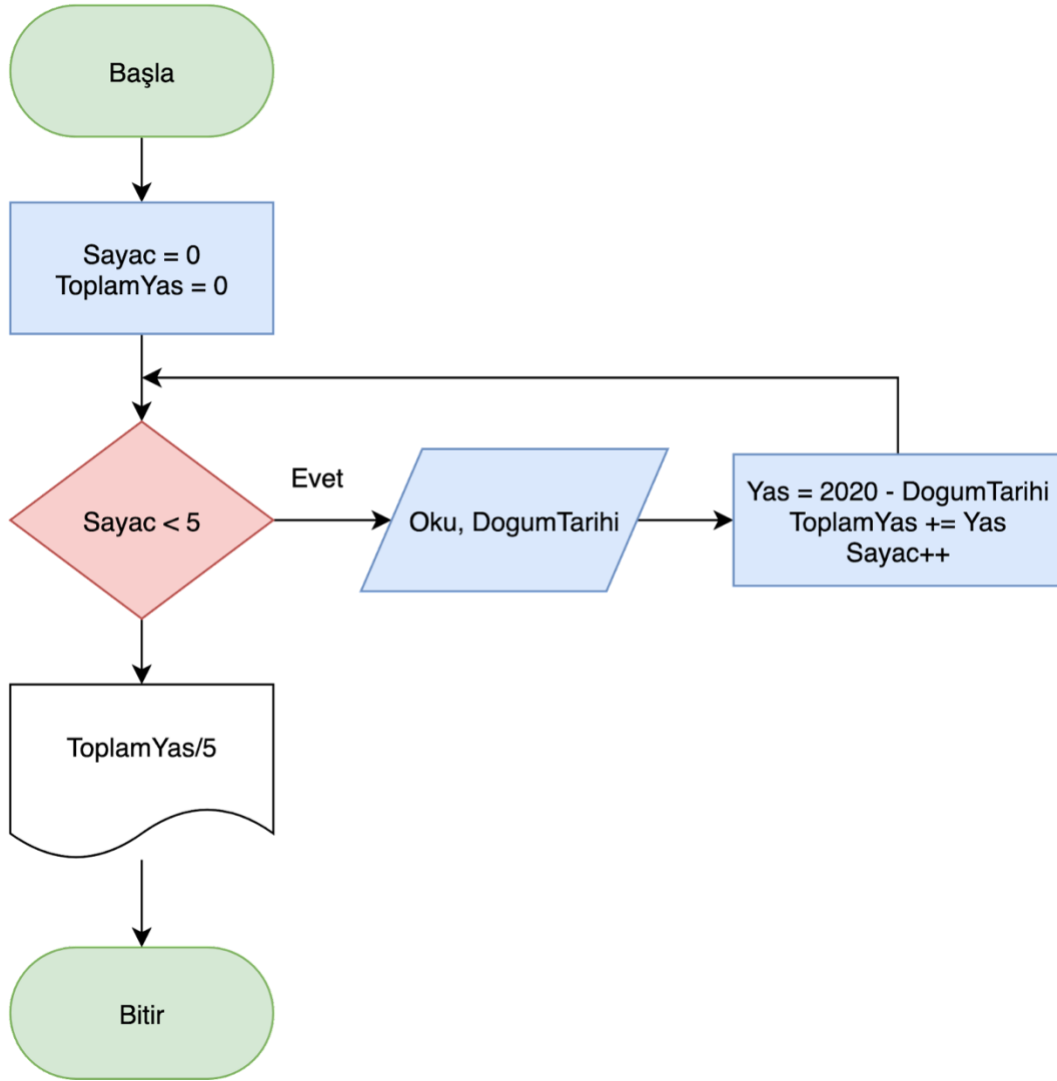
Resim 13. Algoritma Testi Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci slayta ilerleyerek örnek bir sözde kodun adım adım çalıştırıldığı kod çalıştırma tablosunu açıklar. Devamında öğrencilere bir sonraki aşamada yapacakları işlemi özetler. Eğitimci slaytta ilerleyerek öğrencilere akış şeması çizilmiş bir algoritma görevi gösterir. Eğitimci öğrencilere bu akış şemasındaki sözde kodları ayrıştırıp, kodları adım adım çalıştırmayı deneyeceklerini belirtir. Bu şekilde öğrencilerin performanslarını yansıtılmaları beklenir. Eğitimci bunun için öğrencilerden ilk olarak EK 5'i bilgisayarlarına indirmelerini ister. Eğitimci ders sırasında materyalinin bağlantı linkini ve materyali doldurduktan sonra gönderi yapacakları dijital tartışma panosunun linkini paylaşılan notlar kısmından iletir. EK 5'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: Sevdiğiniz 5 kişinin doğum tarihlerini girerek yaşlarının ortalamasını bulan bilgisayar programının akış diyagramı aşağıda verilmektedir. Bu akış şemasındaki sözde kodları adım adım çalıştırıp, test etmek için Kod Çalıştırma tablosunu doldurun.



Hazırlanacak kod çalıştırma tablosu aşağıdaki gibi olmalıdır.

Tablo 14. Kod çalıştırma tablosu

Adım	Değişken değerleri	Çıktı
Başla	Programın Başlangıcı	
Sayac=0, ToplamYas=0	Sayac=0 ToplamYas=0	
Eğer 0<5	evet	
3.1) Oku, DogumTarihi	DogumTarihi=2001	
3.2) Yas = 2020 - 2001	Yas = 19	
3.3) ToplamYas += Yas	ToplamYas = 19	
3.4) Sayac ++	Sayac=1	
Eğer 1 <5	evet	
4.1) Oku, DogumTarihi	DogumTarihi=2003	
4.2) Yas = 2020 - 2003	Yas = 17	
4.3) ToplamYas += Yas	ToplamYas = 36	
4.4) Sayac ++	Sayac=2	
Eğer 2<5	evet	
5.1) Oku, DogumTarihi	DogumTarihi=2005	
5.2) Yas = 2020 - 2005	Yas = 15	
5.3) ToplamYas += Yas	ToplamYas = 51	
5.4) Sayac ++	Sayac=3	
Eğer 3<5	evet	
6.1) Oku, DogumTarihi	DogumTarihi=2007	
6.2) Yas = 2020 - 2007	Yas = 13	
6.3) ToplamYas += Yas	ToplamYas = 64	
6.4) Sayac ++	Sayac=1	
Eğer 4<5	evet	
7.1) Oku, DogumTarihi	DogumTarihi=2004	
7.2) Yas = 2020 - 2004	Yas = 16	
7.3) ToplamYas += Yas	ToplamYas = 80	
7.4) Sayac ++	Sayac=5	
Eğer 5<5	hayır	
Yaz, ToplamYas/5	80/5	16
Bitir.	Programın sonu	

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

Tablo 15. Algoritma adımları

Adım	Açıklama
Başla	Programın Başlangıcı
Sayac=0, ToplamYas=0	Burada bize iki adet değişken gerekli. Kaç kişinin bilgisi girildiğini Sayac isimli değişkende tutuyoruz. İstedğimiz sayıların toplamını ise ToplamYas isimli değişkende tutuyoruz.
Sayac<5 Olduğu Sürece	Buradaki koşulumuz 5 değerine ulaşmış olmamız. Sayac değişkenimiz 5'e ulaşana kadar devam edeceğiz.
3.1) Oku, DogumTarihi	Klavyeden kişinin doğum yılını al.
3.2) Yas = 2020 - DogumTarihi	Yaşını hesapla
3.3) ToplamYas += Yas	Kişinin yaşını toplam yaş değerine ekliyoruz
3.4) Sayac ++	Elimizdeki Sayac değerini bir arttırıyoruz.
Yaz, ToplamYas/5	Ekrana ortalama yaş değerini (Toplam/5) değerini yazdırıyoruz.
Bitir.	Programın sonu

C. Sonuç

Süre: 10 dk.

Materyal: EK 6 Hafta 2 Süreli Ödev

EK 7 Ders Dışı Aktivite Sunusu

EK 8 Derleyici Kullanım Kılavuzu

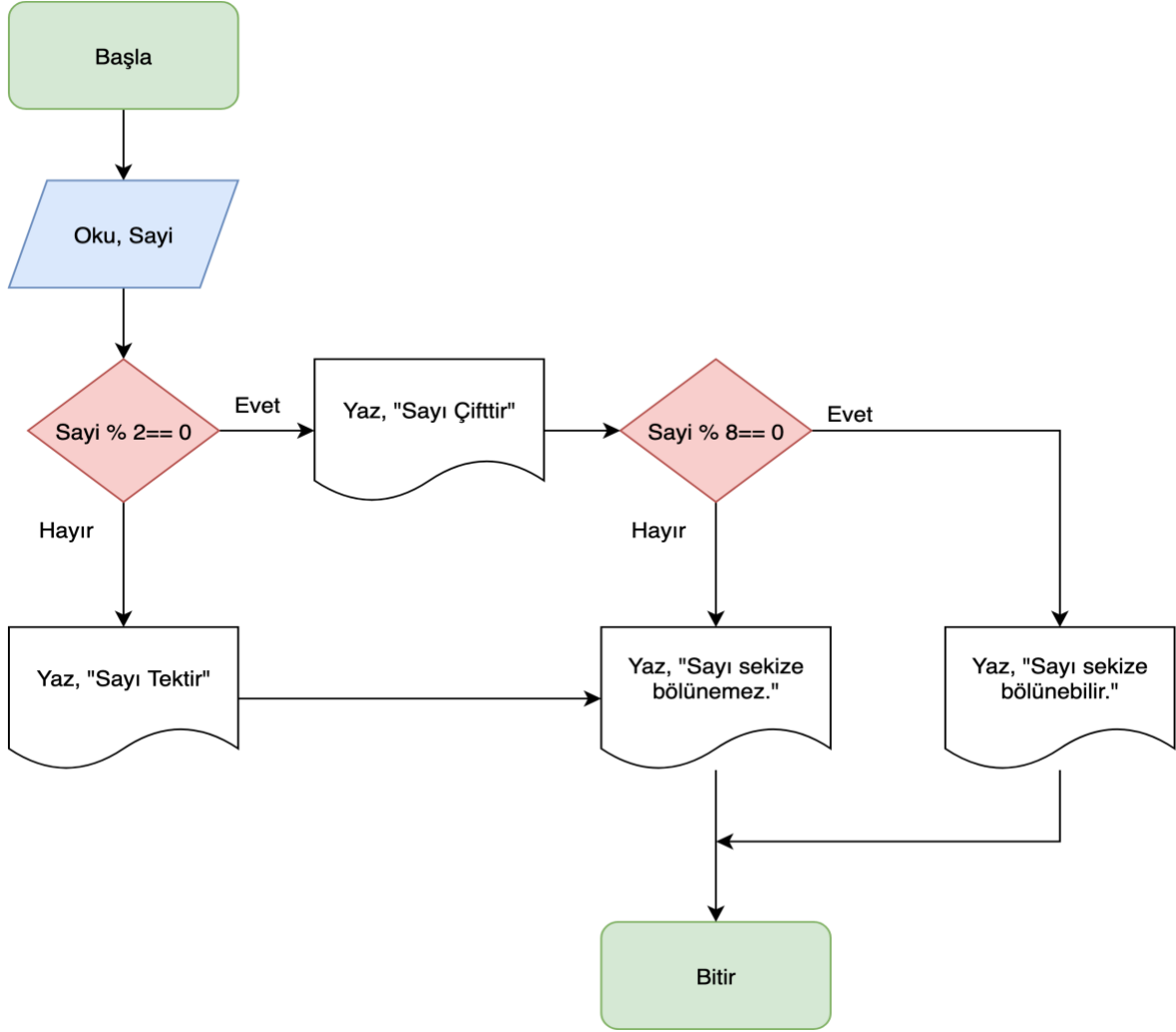
Hazırlık: EK 6 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

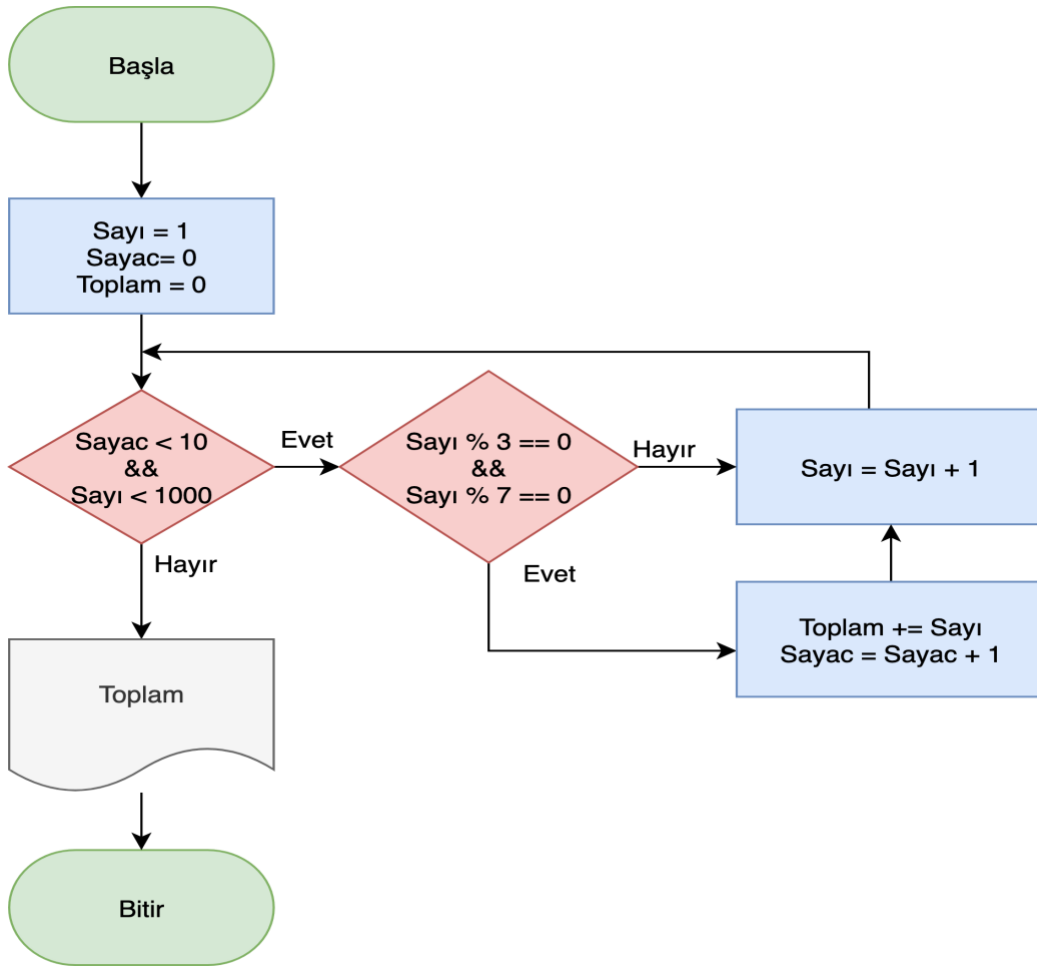
Ders içi uygulama: Dersin son 5 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek haftanın konusu hakkında bilgi verilir.

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 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 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 7 sunusunu inceleyecektir. Bu sunumun sonunda yer alan EK 8 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. Eğitmen bu yanıtları uygulama süresinin sonunda GitHub ya da ÖYS üzerinden öğrencilerle paylaşır.

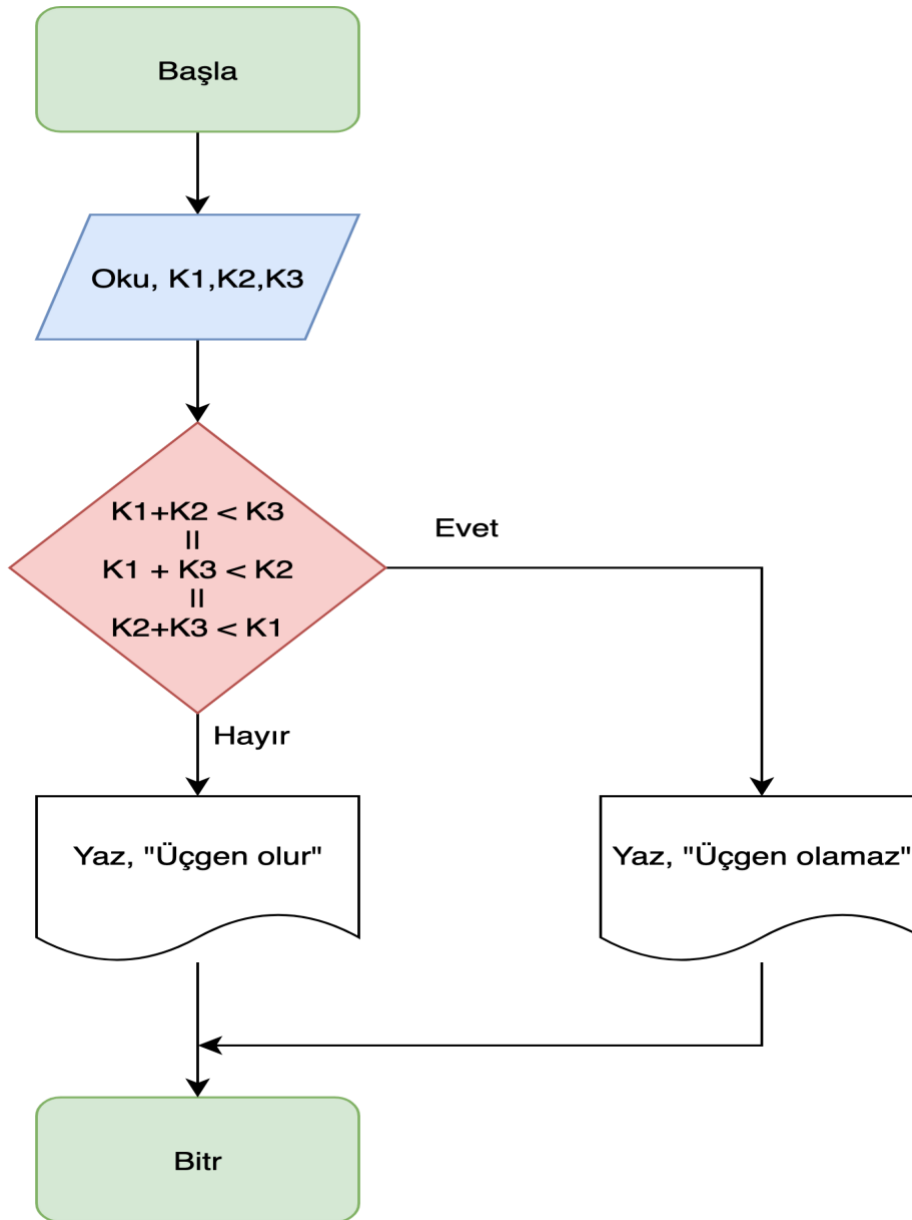
Tasarlayıcı 1:

Tasarlayıcı 2:**Analizci:**

Eğer sayaç 10 değerinden başlatılırsa, girilen sayı 10'dan küçük olursa program doğrudan sonlanacaktır. 10'dan büyük olduğu durumlarda ise 10 ile girilen sayı arasındaki tüm sayıları çarpacaktır.

Denetleyici:

40320

Tasarlayıcı 3:

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 altına yazınız.

Problemler

P1. Çemberin alanını hesaplama

P2. İki sayıyı toplama

P3. Bir sayının karesini hesaplama

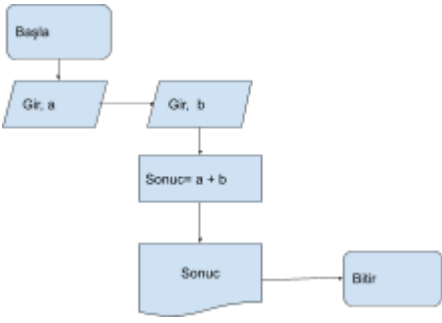
Algoritma İfade Şekli

MY: Metinsel Yazım

SK: Sözde Kod

AD: Akış Diyagramı

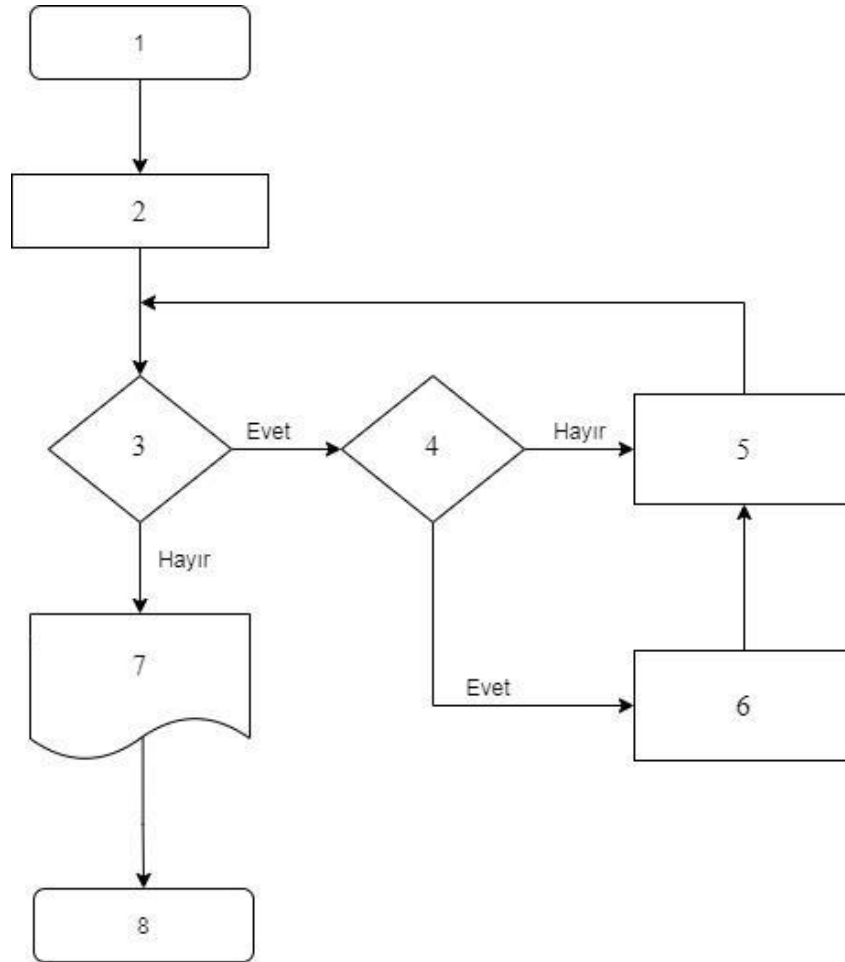
1. Başla. 2. Birinci sayıyı gir (a) 3. İkinci sayıyı gir (b) 4. İşlemi yap. (sonuc = a+b) 5. Ekrana yaz (sonuc) 6. Bitir.	<pre> graph TD A[Başla] --> B[/Gir, x/] B --> C[Kare = x * x] C --> D[Kare] D --> E[Bitir] </pre>	1. Başla. 2. Sayıyı (x) gir 3. Sayının karesini hesapla (Kare=x*x işlemini yap) 4. Sonucu (Kare) yaz 5. Bitir.
Problem: Algoritma İfadesi:	Problem: Algoritma İfadesi:	Problem: Algoritma İfadesi:
<pre> graph TD A[Başla] --> B[/Gir, r/] B --> C{r<0} C -- Hayır --> E[Alan] C -- Evet --> D[Alan= pi * r * r] D --> E E --> F[Bitir] </pre>	1. Başla. 2. Yarıçap gir (r). 3. Eğer yarıçap sıfırdan küçükse ikinci adıma git. 4. Alanı hesapla (alan= $\pi \times r^2$) 5. Sonucu ekrana yazdır. 6. 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:

1. Başla. 2. Gir, r 3. Eğer $r < 0$ ikinci adıma git. 4. $alan = \pi \times r \times r$ 5. Yaz, Sonuc 6. Bitir	 <pre> graph TD A[Başla] --> B[/Gir, a/] B --> C[/Gir, b/] C --> D[Sonuc= a + b] D --> E[Sonuc] E --> F[Bitir] </pre>	1. Başla 2. Gir, x 3. $Kare = x \times x$ 4. Yaz, Kare 5. Bitir
<i>Problem:</i> <i>Algoritma İfadesi:</i>	<i>Problem:</i> <i>Algoritma İfadesi:</i>	<i>Problem:</i> <i>Algoritma İfadesi:</i>

EK 2. Çiftleri Toplama Algoritması

Görev: Aşağıdaki problemin çözümüne yönelik sözde kodu verilen algoritmayı doğru şekilde sıralayın ve akış diyagramındaki boşluklara yazın.

Problem: 1 ile 100 arasındaki (100 dahil olmak üzere) çift sayıların toplamını yazdırma



Sözde Kod	Numarası	Sözde Kod	Numarası
Başla		Sayi=1, Toplam=0	
Yaz, Toplam		Sayi % 2 ==0	
Bitir		Sayi = Sayi + 1	
Toplam = Toplam + Sayi		Sayi <= 100	

EK 3. Akış Diyagramları Bilgi Afişi

BAŞLA

BİTİR

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

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.)



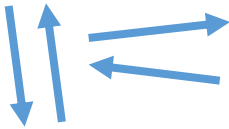
ÇIKTI

Yukarıda kullanıcıdan alınan bilgileri ekrana yazdırılması gerektiğinde yandaki şekil kullanılır.



Karar
Yapıları

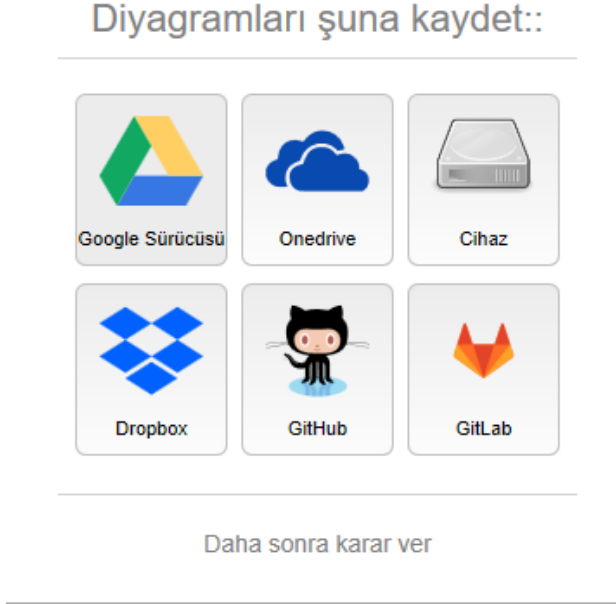
Algoritma akışında işlemler tek yönlü iken karar yapılarında iki farklı ok çıkabilir ve bazı işlemlerin tekrarlanmasını sağlayabilir. Bu şekilde bir



Yandaki oklar algoritmanın akış yönünü bize belirtir.

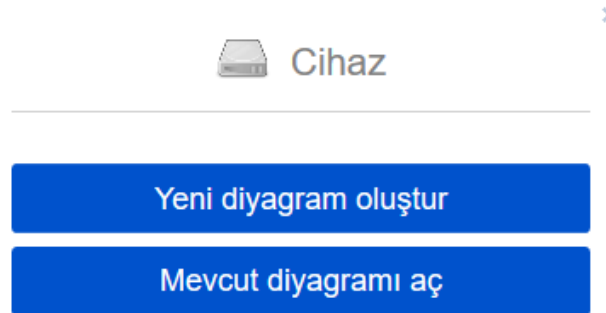
EK 4. App.diagrams.net Kullanım Kılavuzu

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.



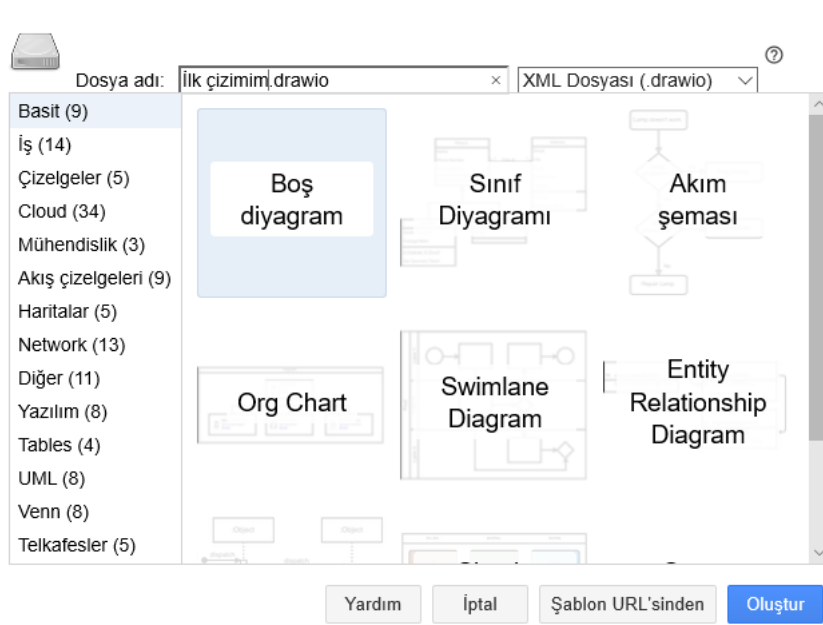
Resim 14. App.diagrams.net giriş ekranı

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



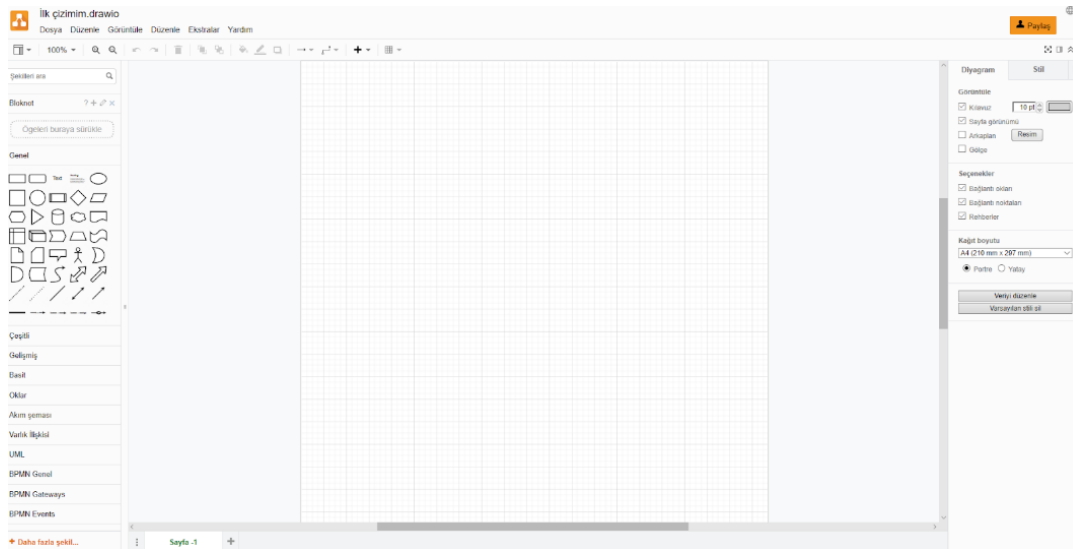
Resim 15. App.diagrams.net yeni diyagram oluşturma ekranı

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



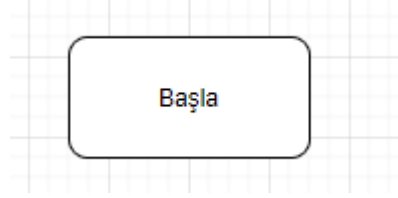
Resim 16. App.diagrams.net yeni diyagram tipi seçme ekranı

Karşımıza aşağıdaki çizim ekranı gelecektir.



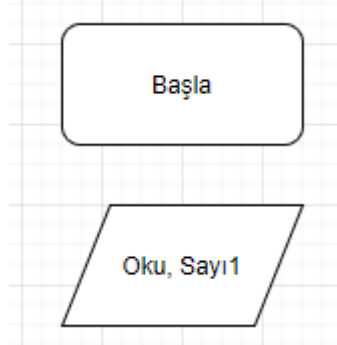
Resim 17. App.diagrams.net çizim ekranı

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.



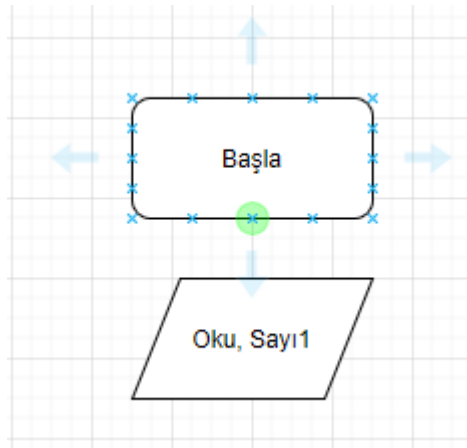
Resim 18. Genel bloğu

Ardından bir tane de paralel kenar ekleyelim.



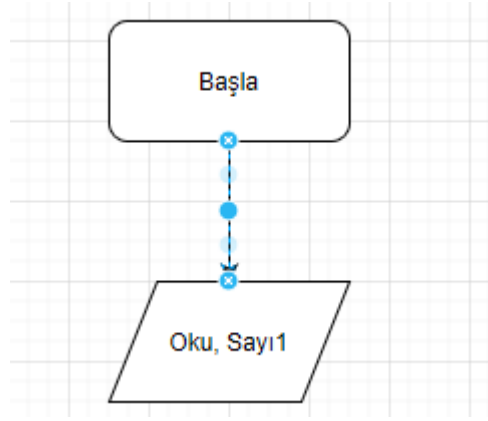
Resim 19. Paralel kenar bloğu

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



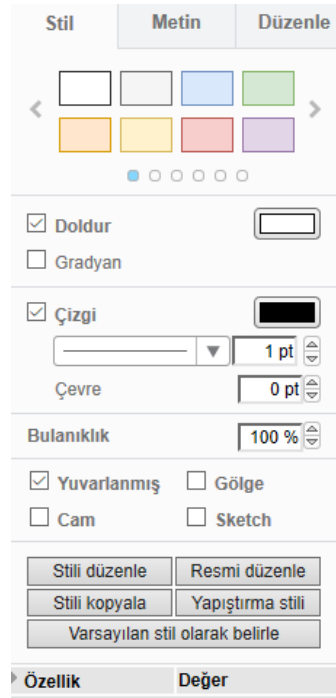
Resim 20. Blokları bağlama öncesi

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



Resim 21. Blokları bağlama sonrası

Eğer bloğun renklerini değiştirmek istersek sağ paneli kullanabiliriz.



Resim 22. Blok özellikleri ekranı

EK 5. Kod Çalıştırma Tablosu

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

Tablo 16. Algoritma adımları ve ekran çıktısı

Adım	Değişken değerleri	Ekran Çıktısı

EK 6. 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.

Diğer Ekler: Ders Dışı Aktivite Materyalleri

Eklere (EK 7 ve EK 8) erişmek için materyaller klasörüne gidiniz.

Hafta 3. 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. Veri tipi dönüşüm işlemlerini yapar.
- K5. Kaçış dizgelerini kullanır.
- K6. C++ temel giriş/çıkış akışlarını bilir.
- K7. C++ işleçlerinin kullanımını bilir.
- K8. Bit işlemleri ile veriler üzerinde hesaplamalar yapar.

Haftanın Amacı ve Materyaller

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

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

Önerilen Ders Akışı (60 + 40 + 50 + 60 dk.)

GİRİŞ (10 dk.)

GELİŞME

Değişkenlere İsim Verelim (20 dk.)

EK 1. Kural Kimde Saklı

Sabitleri Ayıralım (10 dk.)

Örnekleri Çoğaltalım (20 dk.)

Ders Arası (10 dk.)

Veri Tipini Dönüştürelim (20 dk.)

Kaçış Dizgelerini Tanıyalım (10 dk.)

EK 2. Kaçış Dizgeleri Listesi ve Örnek Kod

Çıktıları Karşılaştıralım (10 dk.)

Ders Arası (10 dk.)

Operatörlerle Yüzleşelim (30 dk.)

EK 3. Destekleyici Bilgiler

EK 4. Görev Kodları

Listeyi Dolduralım (20 dk.)

EK 5. Keşfetme Kartları

SONUÇ (10 dk.)

EK 6. Hafta 3 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 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 ikinci 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 üçüncü 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

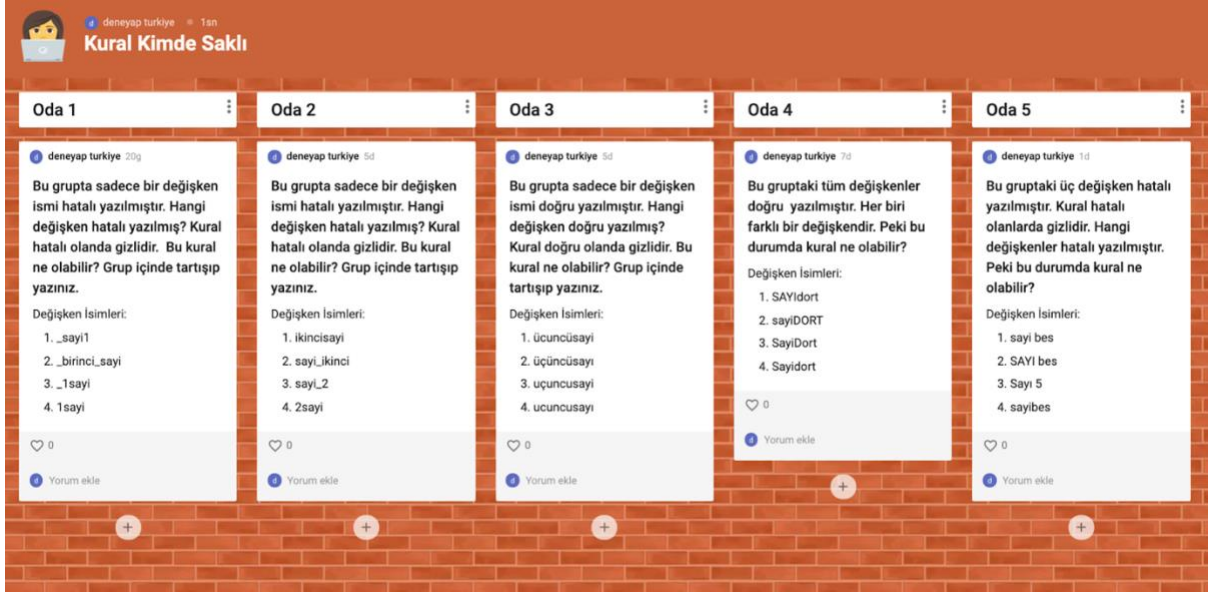
Süre: 20 dk.

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

Materyaller: Dijital tartışma panosu: Grup Çalışması

EK 1. Kural Kimde Saklı

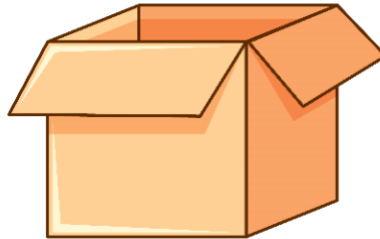
Hazırlık: Eğitimci derse girmeden önce grup çalışması için raf temalı bir dijital panoyu padlet ortamında oluşturur. Link ders sırasında öğrencilere paylaşılan notlar kısmından iletilecektir. Eğitimci padlet linkinin gönderilmesi için sohbet ortamını da kullanabilir. Örnek bir tasarım Resim 23'te gösterilmiştir. EK 1 materyali dijital tartışma panosuna yüklenecektir.



Resim 23. Kural Kimde Saklı Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci Hafta 3 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 değişken_adi;
veri-tipi deg_adi1, deg_adi2, deg_adi3;

Daha sonra, eğitimci grup etkinliği yapacaklarını ve bu 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. Etkinlik sonunda eğitmen değişkenlere değer atama türlerinden kısaca bahseder.

"Bir değişkene bir değer atandığında "ilk değeri ataması" ifadesi kullanılır. İsteğe bağlı olarak, bu atama değişken tanımlanırken ya da daha sonra kullanımda yapılabilir. Herhangi bir değişkende saklanan değerini görüntülemek için "cout" fonksiyonu kullanılır."

Eğitmen aşağıdaki kodları bilgisayarından tüm öğrencilerin görebileceği şekilde ekrana yansıtır ve buradaki dördüncü satıra dikkat etmelerini ister. Dördüncü satır ve sonrası kod üzerindeki farklılıkların ne anlama geleceği hakkında kısaca bilgi verilir.

İlk değer atamasız değişken tanımlama

```
1. #include <iostream>
2. int main()
3. {
4.     int sayi;
5.     sayi = 5;
6.     std::cout << sayi;
7.     return 0;
8. }
```

İlk değer atamalı değişken tanımlama

```
1. #include <iostream>
2. int main()
3. {
4.     int sayi = 5;
5.     std::cout << sayi;
6.     return 0;
7. }
```

Eğitmene Öneriler: Eğitmen etkinlikte öğrencileri bilmeseler de bir tahminde bulunmaları için 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 ayrıca öğrencilere aşağıdaki ipucunu vererek önerilerde bulunabilir.

İPUCU: C++ programlamada, değişkenlerin adları için küçük harfler kullanmak standart bir uygulamadır. Bazı değişkenlerin adları belirli bir kullanımla ilişkilendirilme eğilimindedir. Örnek olarak;

c, ch: karakterler için kullanılır.

i, j, k, l, m, n: tam sayılar ve indisler için kullanılır.

x, y, z: tam ve virgüllü sayılar için kullanılır.

Ayrıca programlarınızın okunabilirliğini artırmak için, *ogrenci_yas*, *sinav_notu* vb. gibi daha uzun ve daha açıklayıcı adlar seçebilirsiniz.

Sabitleri Ayırılım

Süre: 10 dk.

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

Materyaller: Hafta 3 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.”

Eğitimci 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:

1. “const” anahtar sözcüğünü kullanarak tanımlarız.
2. Sabit adları, değişken adlarından ayırt etmek için büyük harfle yazılır.
3. Sabitler her zaman tanımlama sırasında değerlerini almalıdır.

Eğitimci 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. Sonunda konuyu aşağıdaki bilgilerden yararlanarak özetler:

`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: 20 dk.

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

Materyaller: Hafta 3 Sunum

Hazırlık: Eğitimci Hafta 3 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 bir 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 soruları 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.

Eğitime 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.

Veri Tipini Dönüştürelim

Süre: 20 dk.

Kazanımlar: K4. Veri tipi dönüşüm işlemlerini yapar.

Materyaller: Hafta 3 Sunum

Hazırlık: Eğitimci Hafta 3 sunusunda “Veri Tipini Dönüştürelim” başlıklı slayta devam eder.

Uygulama: Bu etkinlikte eğitimci veri tiplerinin bazı durumlarda birbirine dönüştürülmesinin gerektiğini belirtmek için sunu üzerinden ilerlemeye devam eder. Eğitimci neden veri tipleri arasında dönüşüme ihtiyaç duyulduğuna ilişkin uyarıcı materyal olarak yeni bilginin verilmesinde sunuyu kullanır ve aşağıdaki gibi bilgiyi verir:

Bir değişkende saklanan herhangi bir veri, dönüşüm olarak bilinen bir işlemle farklı bir veri tipindeki bir değişkene dönüştürülebilir. Dönüşüm işleminde, tipi dönüştürülecek değişkenin adından önce parantez içinde kullanılacak veri tipi belirtilir. İki farklı sözdizimi kullanılabilir.

`degisken_adi = (veri-tipi) degisken_adi;`

`degisken_adi = static_dönüşüm <veri-tipi> degisken_adi;`

Bir tam sayının başka bir tamsayıya bölünmesi her zaman bir tamsayı sonucu üreteceğinden, dönüşüm genellikle aritmetik bir işlemin sonucunu doğru bir şekilde saklamak için gereklidir. Örneğin, 15/2 tam sayı bölümü, kesilmiş 7 tam sayı sonucunu üretir. Bu işlemde 7,5 gibi bir virgüllü sonuç istiyorsak float tip dönüşümü kullanılması gerekir.

`float sayi = (float) 15/2;`

`float sayi = static_cast <float> 15/2;`

İPUCU: Bir tamsayının başka bir tamsayı ile bölünmesinin sonucu, yuvarlanmaz, kesilir. Bu nedenle 9,9'un kesilmesi sonucu 9 elde edilir.

Ardından aşağıdaki iki örnek kod üzerinden öğrencilere kod çıktısını tahmin etmeleri istenir. Bunun için sohbet üzerinden öğrenci performanslarının iletilmesi istenir. Eğitmen iki çıktı arasındaki farklılığa dikkat çekerek, veri dönüştürme konusunda öğrencilere geri bildirim verir.

```
#include <iostream>
using namespace std;
int main()
{
    int a = 5;
    char b = 'A';

    a = a + b;
    float c = a + 3.0;

    cout << "a = " << a << endl
         << "b = " << b << endl
         << "c = " << c << endl;

    double d = 3.4;
    int e = (int)d + 2;

    cout << "d = " << d << endl;
    cout << "e = " << e << endl;

    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int a = 5;
    char b = 'A';

    a = a + b;
    float c = a + 3.0;

    cout << "a = " << a << endl
         << "b = " << b << endl
         << "c = " << c << endl;

    double d = 3.4;
    double e = (double)d + 2;

    cout << "d = " << d << endl;
    cout << "e = " << e << endl;

    return 0;
}
```

Kodun çıktısı nasıldır? Tahminleri sohbette yazınız. a = 70 b = A c = 73 d = 3.4 e = 5 İpucu! ‘A’ karakterinin değeri ASCII tablosunda 65’tir.	Kodun çıktısı nasıldır? Tahminleri sohbette yazınız. a = 70 b = A c = 73 d = 3.4 e = 5.4 İpucu! ‘A’ karakterinin değeri ASCII tablosunda 65’tir.
--	--

Kaçış Dizgelerini Tanıyalım

Süre: 15 dk.

Kazanımlar: K5. Kaçış dizgelerini kullanır.

Materyaller: EK 2. Kaçış Dizgeleri Örnek Kod

Hazırlık: Eğitimci Hafta 3 sunusunda “Kaçış Dizgelerini Tanıyalım” başlıklı slaytla devam eder. Eğitimci etkinlik öncesi EK 2’de yer alan örnek kodun derleyici üzerinde yazılı dosyasını oluşturur. ÖYS üzerinden EK 2 paylaşılır.

Uygulama: Eğitimci slaytta yer alan kaçış dizgeleri ve örnek kodu üzerinden uyarıcı materyal ile yeni bilgiyi verir. Öğrenciler EK 2’yi bilgisayarlarına indirerek kaçış dizgeleri listesine erişim sağlamalıdır. Eğitimci bu listedekilerden birkaçının denendiği EK 2’deki örnek kodu, derleyici üzerinden açar ve kod üzerinde değişiklik yapacağı bu dosyayı ekran paylaşımı yapar. Önce kodu çalıştırarak ekran çıktısını gösterir. Ardından eğitimci listedeki kaçış dizgelerinden birkaçını kod üzerinde deneyeceklerini belirtir. Bunun için eğitimci rastgele bir öğrenciye hak verir. “Hangi kaçış dizgesini değiştirelim istersin?” diye sorar. Eğitimci kod üzerinde değişiklik yapar. Daha sonra söz hakkı verdiği öğrenciye, “Şimdi topu kime atmak istersin?” diye sorar ve rastgele seçilen öğrencinin bir başka öğrenciye söz hakkı tanımasını sağlar. Eğitimci ikinci öğrenciye ekran çıktısında neyin değiştiğini sorar. Bu şekilde öğrencilerden her bir değişiklikte program çıktısını karşılaştırıp, kaçış dizgelerinin anlamlarını keşfetmeleri istenir. Böylece en az iki kaçış dizgesi kod üzerinde test edilir. Eğitimci öğrencilerden listedeki diğer kaçış dizgelerini de evde bireysel olarak kodu değiştirip denemelerini ister. Eğitimci ayrıca EK 2’yi öğrencilerle ÖYS üzerinden de paylaşmıştır. Bu şekilde gerektiğinde kaçış dizgeleri listesini istediklerinde açıp bakabileceklerini belirtir.

Çıktıları Karşılaştıralım

Süre: 10 dk.

Kazanımlar: K6. C++ temel giriş/çıkış akışlarını bilir.

Materyaller: Hafta 3 Sunum

Hazırlık: Eğitimci Hafta 3 sunusunda "Çıktıları Karşılaştıralım: Cin-Cout Arasındaki Fark" başlıklı slaytla 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 çıktısının ne olacağını tahmin etmeleri istenir. 1 dk. sohbet üzerinden öğrenciler performanslarını gösterir. Bu süre içerisinde eğitimci sözsüz kısa bir müzik başlatır. Sonunda eğitimci sohbet üzerinden gelen yanıtlardan da yararlanarak öğrencilerin çıktılar arasındaki farkı keşfetmelerini sağlar. Bunun için beyin fırtınası yöntemi kullanılır. Eğitimci cin ve cout komutlarının görevlerini özetleyerek açıklar.

Eğitime Öneriler: Eğitimci 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. Eğitimci derse girmeden önce bireysel çalışma 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 24'te gösterilmiştir. EK 3, dijital tartışma panosunda bağlantı olarak verilirken, EK 4 materyali dört grup halinde ayrı ayrı dijital tartışma panosuna yüklenecektir.

The screenshot shows a digital discussion board with a brick wall background. At the top, there's a header with a user profile 'deneyap turkiye' and the title 'Operatörlerle Yüzleşelim'. Below the header, there are five tabs: 'Oda 1', 'Oda 2', 'Oda 3', 'Oda 4', and 'Destekleyici Bilgiler'. Each tab contains a task card for 'Görev Kodu 1' through 'Görev Kodu 4'. Each task card has a title, a description, a code block with C++ code, and a 'Yorum ekle' (Add comment) button. The 'Destekleyici Bilgiler' tab contains two cards: 'Aritmetik Operatörler' and 'Karşılaştırma Operatörleri'. Each of these cards has a table of operators and their symbols, and a 'Yorum ekle' button.

Operatör	Anlam	Kullanım
+	Eklemeye	a + b
-	Çıkartmaya	a - b
*	Çarpmaya	a * b
/	Bölme	a / b
%	Bölme kalanı	a % b
++	Artırma	a++
--	Azaltma	a--

Operatör	Anlam	Kullanım
<	Küçük	a < b
<=	Küçük veya eşit	a <= b
>	Büyük	a > b
>=	Büyük veya eşit	a >= b
==	Eşit	a == b
!=	Eşit değil	a != b

Resim 24. Operatörlerle Yüzleşelim Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci bu etkinliği öğrencilerin bireysel olarak çalışacağı dijital tartışma panosu üzerinde tamamlar. Bu nedenle eğitimci 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 öğrencilere 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 eğitimci 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. Eğitimci 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 eğitimci en çok kod görevine katkı gösteren arkadaşların ödüllendirileceğini belirtir. Etkinlik sonunda eğitimci dört göreve ilişkin kod çıktılarını 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. Eğitimci 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.

Listeyi Dolduralım

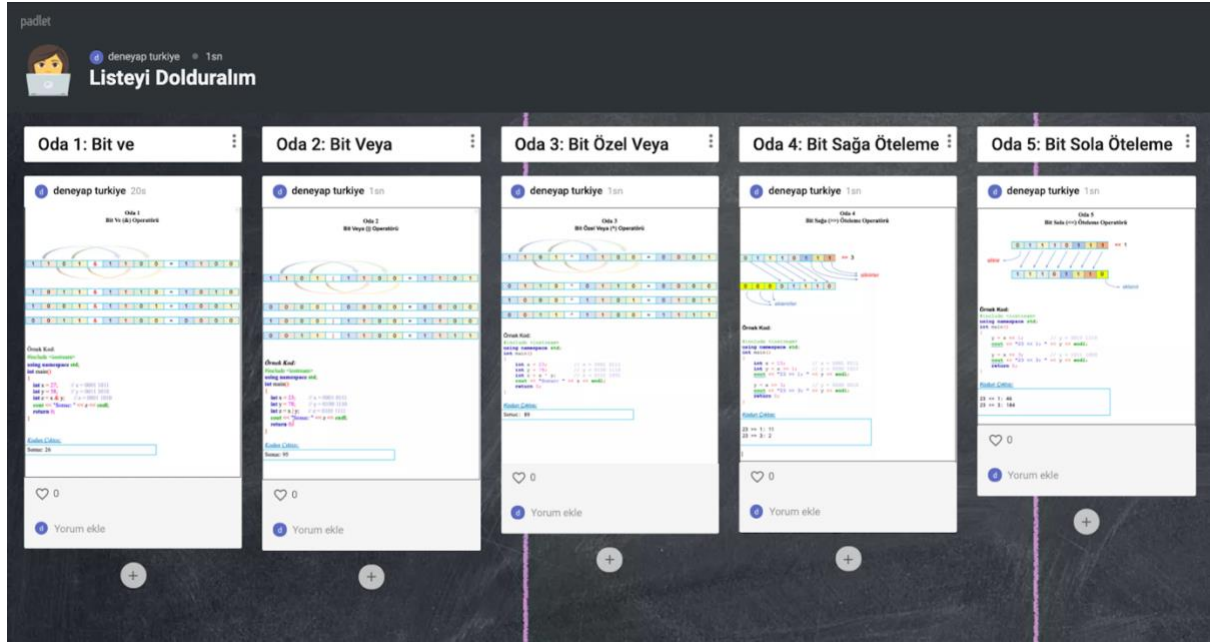
Süre: 20 dk.

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

Materyaller: Dijital tartışma panosu: Grup Çalışması

EK 5. Keşfetme Kartları

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 25'te gösterilmiştir. EK 5 materyali beş grup halinde dijital tartışma panosuna yüklenecektir. Her grup bir farklı keşfetme kartını almaktadır.



Resim 25. Listeyi Dolduralım Örnek Dijital Tartışma Panosu

Uygulama: Öğretmen tarafından Hafta 3 sunusunda “Listeyi Dolduralım” başlıklı slaytına devam edilir. Bu slaytta “Bit Değil” işleminin görevini keşfettirecek bilgiler bulunmaktadır. Öğretmen bu bilgiler üzerinden “Bit Değil” işleminin görevini birlikte keşfeder ve açıklar. Daha sonra öğretmen slaytta ilerler ve diğer Bit işlemlerinin görevlerini de benzer şekilde verilen bilgilerden grup olarak keşfetmeleri gerektiğini belirtir. Öğretmen öğrencilere ders öncesi hazırladığı dijital panonun linkini gönderir. Öğrenciler bu aşamadan sonra dörderli beş grup halinde 10 dk. süre ile çalışma odalarına dağıtılır.

Tablo 17. Bit işlem operatörleri

Operatör	İşlem	Görev
~	Bit Değil	Bit dizisindeki 0 olan bitleri 1, 1 olan bitleri 0 yapar.
<<	Bit Sola öteleme	En soldaki bit kaybolurken en sağdan bir adet 0 eklenir. Öteleme sayısı değişebilir.
>>	Bit Sağa öteleme	En sağdaki bit kaybolurken en soldan bir adet 0 eklenir. Buradaki öteleme sayısı değişebilir.
&	Bit Ve	İki bitin de 1 olduğu durumda sonuç 1 olurken, diğer durumlarda sonuç 0 olur.
^	Bit Özel Veya	İki bitin de aynı anda 0 ya da 1 olduğu durumda sonuç 0 olurken, diğer durumlarda sonuç 1 olur.
	Bit Veya	İki bitin de 0 olduğu durumda sonuç 0 olurken, diğer durumlarda sonuç 1 olur.

Etkinlikte her grup bir tane Bit işleminin görevini tanımlamalıdır. Bu şekilde öğrenci performanslarının sergilenmesi beklenir. Öğrenciler ana odaya geri döndüklerinde, eğitmen dijital panoyu ekran paylaşımı yaparak öğrenci ürünleri üzerinden Bit işlem operatör görevlerini Tablo 17'deki gibi özetler. Öğrenciler tarafından keşfedilemeyen ya da hatalı olan görevler dijital pano üzerinde eğitmen tarafından düzeltilir, doğru görev ilgili grup görevi altına yazılır. Eğitmen bu şekilde geri bildirimde bulunmaktadır.

Eğitmene Öneriler: Eğitmen panodaki öğrenci ürünlerinde hata varsa, öğrencilere söz hakkı verebilir. Eğitmen bu durumda doğru görevi dijital panoda ilgili grup altında paylaşmalıdır. Öğrenciler tarafından oluşturulan dijital panonun pdf çıktısı alınarak, ders sonrası ÖYS üzerinden öğrenci erişimine açılması sağlanabilir. Bu şekilde öğrenciler ders sonrası beyin fırtınası ile yaptıkları ürünlerine kalıcı olarak erişim sağlayacaktır.

C. Sonuç

Süre: 10 dk.

Materyal: EK 6 Hafta 3 Süreli Ödev

Hazırlık: EK 6 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 "Karar Mantık Yapıları" konusunun işleneceği bildirilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak bir süre içinde 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 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 ya da ÖYS üzerinden öğrencilerle paylaşır.

Kodlayıcı: Kullanıcıdan aldığınız iki sayının değerlerini değiştiren programın kodunu herhangi ekstra bir değişken kullanmadan yazınız (*ipucu: sadece toplama ve çıkarma işlemlerini kullanabilirsiniz*).

```

#include <iostream>
using namespace std;
int main()
{
    int x, y;
    cout << "Birinci sayiyi girin: ";
    cin >> x;
    cout << "Ikinci sayiyi girin: ";
    cin >> y;

    x = x + y;
    y = x - y;
    x = x - y;

    cout << "X'in yeni degeri: " << x << endl;
    cout << "Y'nin yeni degeri: " << y << endl;
    return 0;
}

```

Denetleyici: Aşağıdaki değişkenlerden isim ve değer atamasına göre uygun olmayan değişken tanımları ve değer atamalarını yuvarlak içine alınız.

int __xyz5;	int _xyz5;	const int XYZ;	short xyz = 34452;
int __5xyz;	int xyz_5;	int xyz = '*';	int xyz = 34452;
int __XYZ5;	int 5_xyz;	char xyz = '\192';	float xyz = 12345.12345;
int xyz=5.2;	int _5xyz;	unsigned xyz = '\192';	bool xyz = -1

Kodlayıcı: Aritmetik ve karşılaştırma işleçlerini kullanmadan kullanıcıdan aldığınız iki sayının eşit olup olmadığını kontrol eden ve eşitse ekrana 1, değilse 0 yazan bir program hazırlayınız (*ipucu: bit Özel Veya ve Bit değil operatörlerini kullanabilirsiniz*).

```
#include <iostream>

using namespace std;

int main()
{
    int x, y, z;

    cout << "Birinci sayiyi girin: ";

    cin >> x;

    cout << "Ikinci sayiyi girin: ";

    cin >> y;

    z = !(x ^ y);

    cout << "Sonuc: " << z << endl;
}
```

Analizci: 3 kişinin yaşları toplamını hesaplayan, program kodu verilmiştir. Kod üzerinde eksik yerleri tamamlayınız.

Eksik Kodlar

```
#include <iostream>

using namespace std;

int main()
{
    int yas1, yas2;

    cout << "1. kisinin yasini giriniz:";

    cin >> yas2;

    cout << "3. kisinin yasini giriniz:";
}
```

Tamamlanmış Kodlar

```
#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 << "3. kisinin yasini giriniz:";
    cin >> yas3;
    cout << "\nYaslarinizin toplami: " << yas1+yas2+yas3;
}
```

Tasarlayıcı: Aşağıdaki gibi bir ekran çıktısını kaçış dizgelerini kullanarak oluşturunuz.

```
\DENEYAP\           Atolyesi'ni

        "cok" ama "cok" seviyorum!
```

```
#include <iostream>

using namespace std;

int main()
{
    cout << "\\DENEYAP\\ \t\t Atolyesi'\n\n"
           " \t\"cok\" ama \"cok\" seviyorum!\n";
}
```

Hafta 3. 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	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. _sayi1 _birinci_sayi _1sayi 1sayi	Bu grupta sadece bir değişken ismi hatalı yazılmıştır. Hangi değişken hatalı yazılmış? Kural hatalı olanda gizlidir. Bu kural ne olabilir? Grup içinde tartışıp yazınız. ikincisayi sayi_ikinci sayi_2 2sayi
<i>Kural 1: Değişken ismi alt çizgi ile başlayabilir. Ancak numara ile değil.</i>	<i>Kural 2: Değişken ismi bir harf ile başlayabilir. Ancak numara ile değil.</i>
Oda 3: Görev	Oda 4: 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. üçüncüsayi üçüncüsayı uçuncusayi ucuncusayı	Bu gruptaki tüm değişkenler doğru yazılmıştır. Her biri farklı bir değişkendir. Peki bu durumda kural ne olabilir? Grup içinde tartışıp yazınız. sayiDORT SAYIdort SayiDort Sayidort
<i>Kural 3: Değişken isminde Türkçe karakter kullanılmaz.</i>	<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 bes

sayi bes

Sayı 5

sayibes

Kural 5: Değişken ismi yazılırken boşluk kullanılmaz.

EK 2. Kaçış Dizgeleri Örnek Kod

Tablo 18. Kaçış dizgeleri

Kaçış dizgesi	Anlamı
\a	Ses ve uyarı üretir.
\b	İmleci bir pozisyon geri hareket ettirir.
\f	İmleci bir sonraki sayfanın ilk pozisyonuna getirir.
\n	İmleci bir sonraki satırın ilk pozisyonuna getirir.
\r	İmleci mevcut satırın ilk pozisyonuna getirir.
\t	İmleci bir sonraki yatay tab pozisyonuna getirir.
\v	İmleci bir sonraki dikey tab pozisyonuna getirir.
\'	Tek tırnak işareti oluşturur.
\"	Çift tırnak işareti oluşturur.
\?	Soru işareti oluşturur.
\\	Ters eğik çizgi işareti oluşturur.
\0	Boş karakter oluşturur.

```

#include <iostream>
using namespace std;
int main()
{
    cout << "\nBu \t cumlede\n\t\t"
           "  cok \"fazla\" kacis dizgesi vardır!\n";
    return 0;
}

```

Kodun Çıktısı:

```

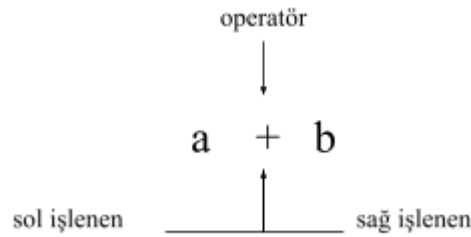
Bu   cümlede
      cok “fazla” kacis dizgesi vardır!

```

EK 3. 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 19. 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 -- işlemleri değişken değerlerini 1 arttırmak ya da 1 azaltmak 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 doğrusa "1" yanlışsa "0" sonucunu döndürür.

Tablo 20. Önemli aritmetik operatörler

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 21. 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.	$x\%=y$ $(x=x\%y)$

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 22. 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 23. 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 24. 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 = 23, b = 5;
    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:

```
#include <iostream>
using namespace std;
int main(){
    int x = 5, y = 4;
    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;
}
```

Kodun Çıktısı:

```
#include <iostream>
using namespace std;
int main()
{
    int x = 5, y = 0;
    cout << ((x <= y) || (y>0)) << endl;
    cout << ((x > 4) && (y==0)) << endl;
    cout << (x && !y) << endl;
    cout << (!(x-2) || (y+2 > 2)) << endl;
}
```

Kodun Çıktısı:

```
#include <iostream>
using namespace std;
int main()
{
    float x, y, z;

    x = 2.4;

    y = x;

    z = x + 1.3 + (y * 2.0);

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

    x = y = 3.4;

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

    x+= 5.1;
    y+= y * 2.0;

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

    x/= 2.0;
    y*= 3.0;

    cout << "x: " << x << endl;
    cout << "y: " << y << endl << endl;
}
```

Kodun Çıktısı:

EK 5. Keşfetme Kartları

Bit Ve (&) Operatörü



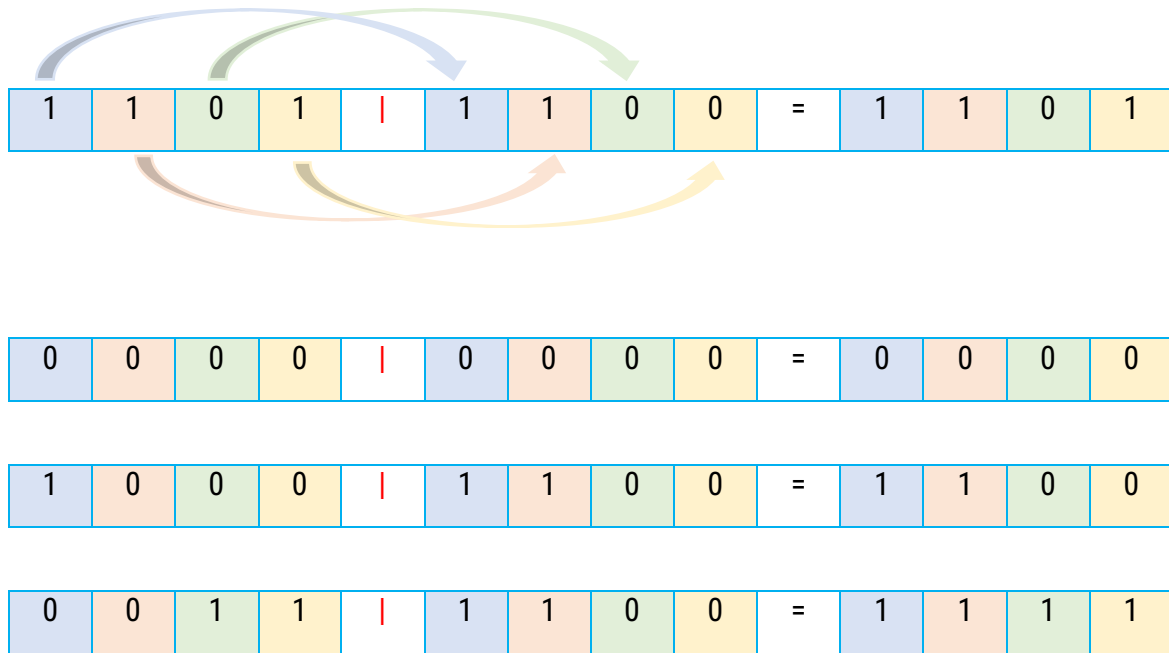
Örnek Kod:

```
#include <iostream>
using namespace std;
int main()
{
    int x = 27;           // x = 0001 1011
    int y = 58;           // y = 0011 1010
    int z = x & y;        // z = 0001 1010
    cout << "Sonuc: " << z << endl;
}
```

Kodun Çıktısı:

Sonuc: 26

Bit Veya (|) Operatörü



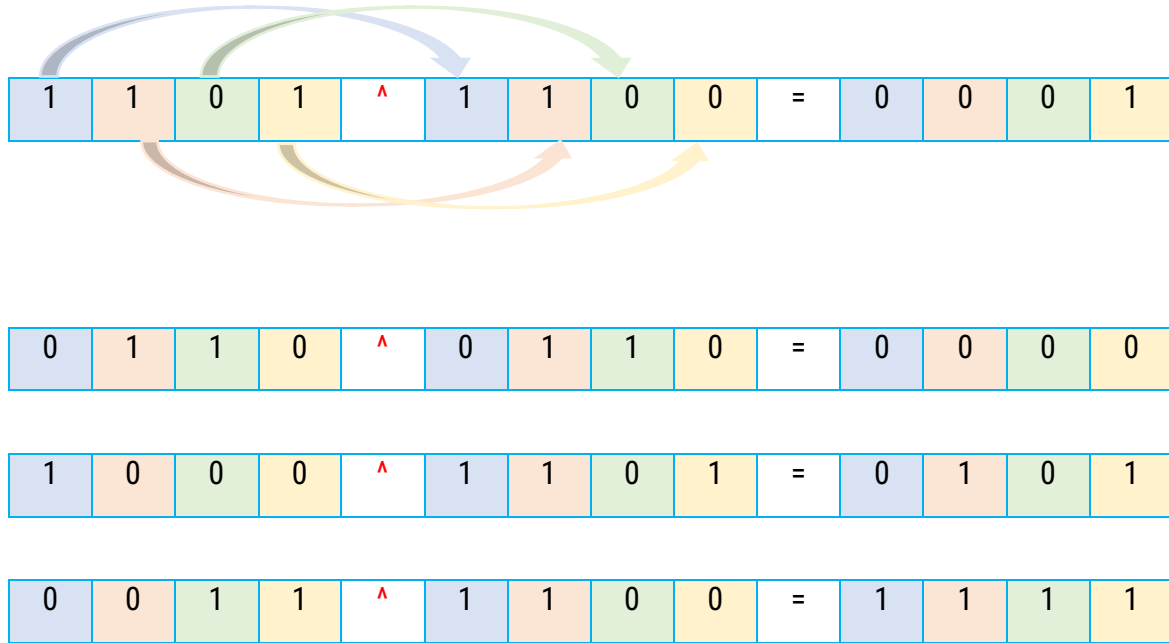
Örnek Kod:

```
#include <iostream>
using namespace std;
int main()
{
    int x = 23;           // x = 0001 0111
    int y = 78;           // y = 0100 1110
    int z = x | y;        // z = 0101 1111
    cout << "Sonuc: " << z << endl;
}
```

Kodun Çıktısı:

Sonuc: 95

Bit Özel Veya (^) Operatörü

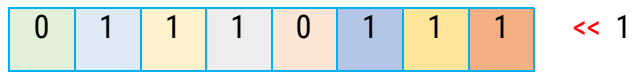


Örnek Kod:

```
#include <iostream>
using namespace std;
int main()
{
    int x = 23;           // x = 0001 0111
    int y = 78;           // y = 0100 1110
    int z = x ^ y;        // z = 0101 1001
    cout << "Sonuc: " << z << endl;
}
```

Kodun Çıktısı:

Sonuc: 89

Bit Sola (<<) Öteleme Operatörü**Örnek Kod:**

```
#include <iostream>

using namespace std;

int main()
{
    y = x << 1;           // y = 0010 1110
    cout << "23 << 1: " << y << endl;

    y = x << 3;           // y = 1011 1000
    cout << "23 << 3: " << y << endl;
    return 0;
}
```

Kodun Çıktısı:

```
23 << 1: 46
```

```
23 << 3: 184
```

Bit Değil (~) Operatörü

~	1	0	0	1	=	0	1	1	0
---	---	---	---	---	---	---	---	---	---

~	1	1	0	0	=	0	0	1	1
---	---	---	---	---	---	---	---	---	---

~	1	0	0	0	=	0	1	1	1
---	---	---	---	---	---	---	---	---	---

~	1	1	1	1	=	0	0	0	0
---	---	---	---	---	---	---	---	---	---

Örnek Kod:

```
#include <iostream>
using namespace std;
int main()
{
    unsigned char x = 23;    // x = 0001 0111
    unsigned char y = 78;    // y = 0100 1110
    unsigned char z = ~x;     // z = 1110 1000
    cout << "~23: " << (int)z << endl;

    z = ~y;                  // z = 1011 0001
    cout << "~78: " << (int)z << endl;
    return 0;
}
```

Kodun Çıktısı:

```
~23: 232
~78: 177
```

EK 6. 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.

Hafta 4. 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 + 20 + 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 (50 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

Biliyor muydunuz?

Süre: 10 dk.

Materyal: Hafta 4 Sunum

Hazırlık: Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 4 Sunum).

Uygulama: Eğitmen öğrencilere sunum üzerinden aşağıda verilen soruyu sözlü olarak sorarak düşünmelerini 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 4 Sunum

EK 1. Karar Yapılarını Tanıyalım Görev Kâğıtları

Hazırlık: Hafta 4 Sunumunda “Algoritma Çıktıları Ne Olur” adlı slaytlar sırası geldikçe açılır.

Uygulama: Eğitimci bu ders için hazırlanan beş görevi sunum üzerinden ikişer 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 aşağıdaki gibi kısa bir bilgilendirme yaparak özetleme yapması beklenir.

Programlama dillerinin olmazsa olmazı karar ifadeleridir. Karar ifadeleri program akışına yön vermek için kullanılır. Belirli 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 ‘{’ ve ‘}’ arasına yazılır. Tek satırlık işlemler için ‘{’ ve ‘}’ 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” kullanmaktayız.

```

if(koşul1)
    koşul1 için yapılacak işlem
else if(koşul2)
    koşul2 için yapılacak işlem
else if(koşul3)
    koşul3 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.

Materyaller: Hafta 4 Sunum

EK 1. Karar Yapılarını Tanıyalım Görev Kâğıtları

Hazırlık: Eğıtmenden 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 eğıtmenin 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 eğıtmen Code::Blocks programında yukarıdaki görevi öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eğıtmen 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. Eğıtmen 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 yazdınız? 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. Eğıtmen 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: 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 $\geq$ 50 ise öğrenci geçer

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

ortalama $\geq$ 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. Eğitmen 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: 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. Eğitmen 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 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 içe Koşula Farklı Bir Bakış (Switch)

Süre: 20 dk.

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

Materyaller: Hafta 4 Sunum

EK 2. 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 (Tablo 1) 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” koşulunu 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 hâlinde çalışır.

Switch kullanımı yukarıdaki gibi özetlendikten sonra eğitmen 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 4 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 bir süre içinde 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.

1)Kodlayıcı:

```
#include <iostream>

using namespace std;

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

    if(sayi %2 == 0)
    {
        if(sayi %4 == 0)
            cout << "Sayi cifttir ve 4 ile bolunur";
        else
            cout << "Sayi cifttir ve 4 ile bolunmez";
    }
    else
    {
        if(sayi %3 == 0)
            cout << "Sayi tektir ve 3 ile bolunur";
        else
            cout << "Sayi tektir ve 3 ile bolunmez";
    }
}
```

2)Analizci

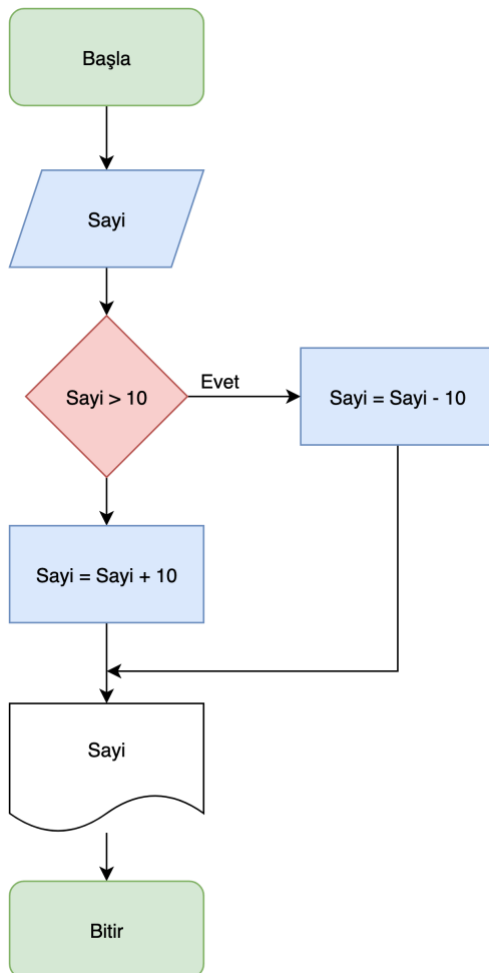
```
#include <iostream>

using namespace std;

int main()
{
    int sayi = 13;
    if(sayi % 2 == 0)
        cout << sayi /2;
    else
        cout << (sayi-1) /2;
}
```

Cevap: 6

3)Tasarlayıcı



Cevap:

```
#include <iostream>

using namespace std;

int main()
{
    int sayi;

    cin >> sayi;

    if(sayi>10)
        sayi = sayi -10;

    else
        sayi = sayi +10;

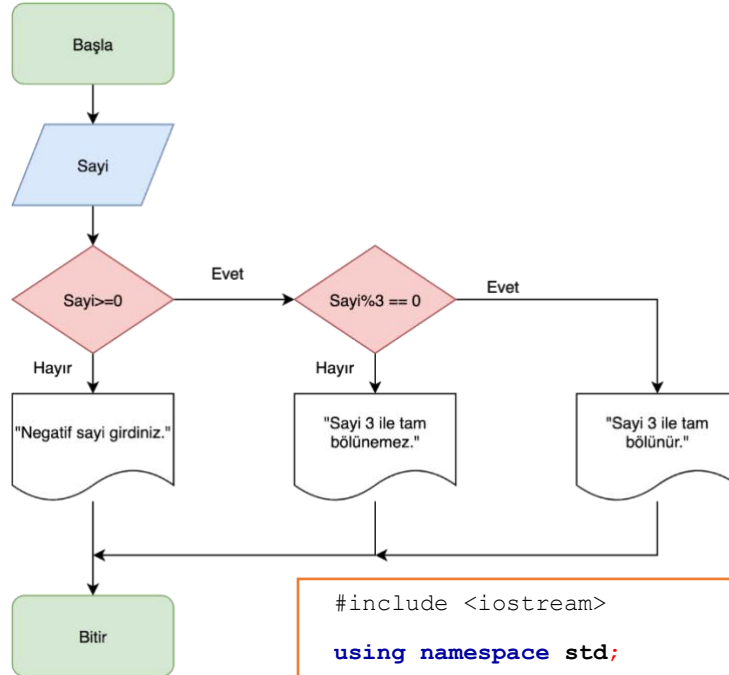
    cout << sayi;

}
```

Hafta 4. Ders Materyalleri

EK 1. Görevler

1. Görev



1. Grup Ekran Çıktısı:

.....

2. Grup Ekran Çıktısı:

.....

3. Grup Ekran Çıktısı:

.....

4. Grup Ekran Çıktısı:

.....

5. Grup Ekran Çıktısı:

.....

```

#include <iostream>

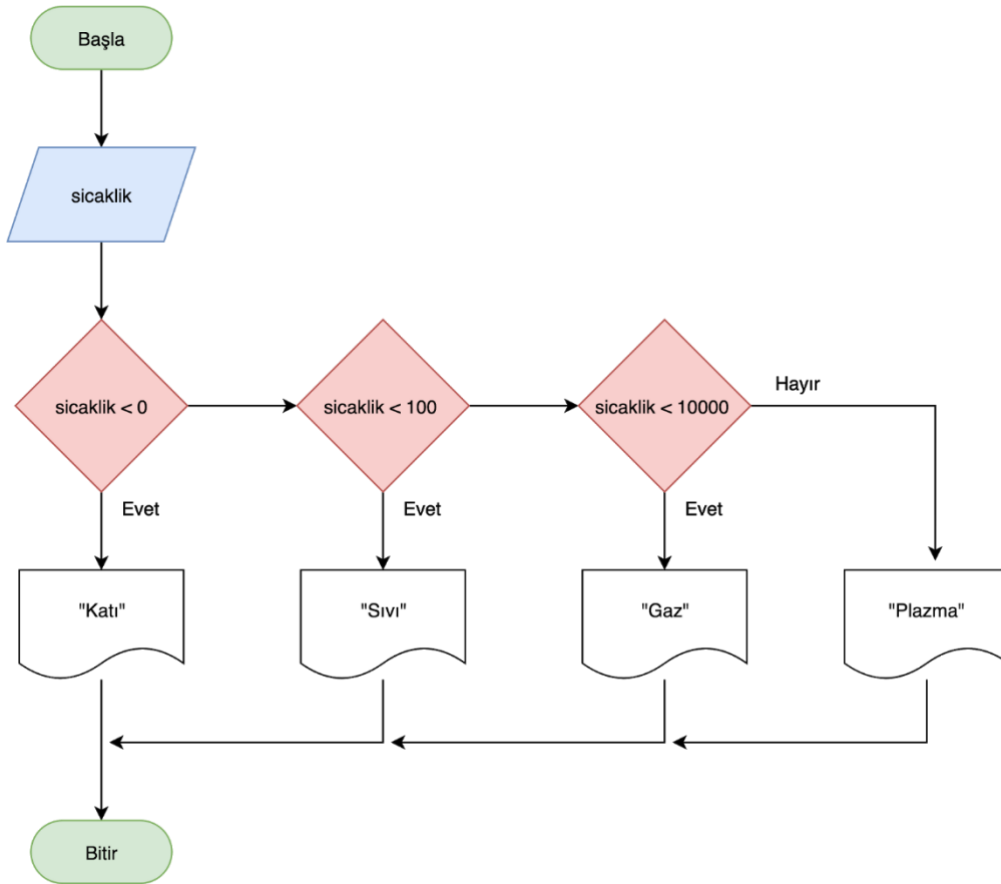
using namespace std;

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:

.....

```

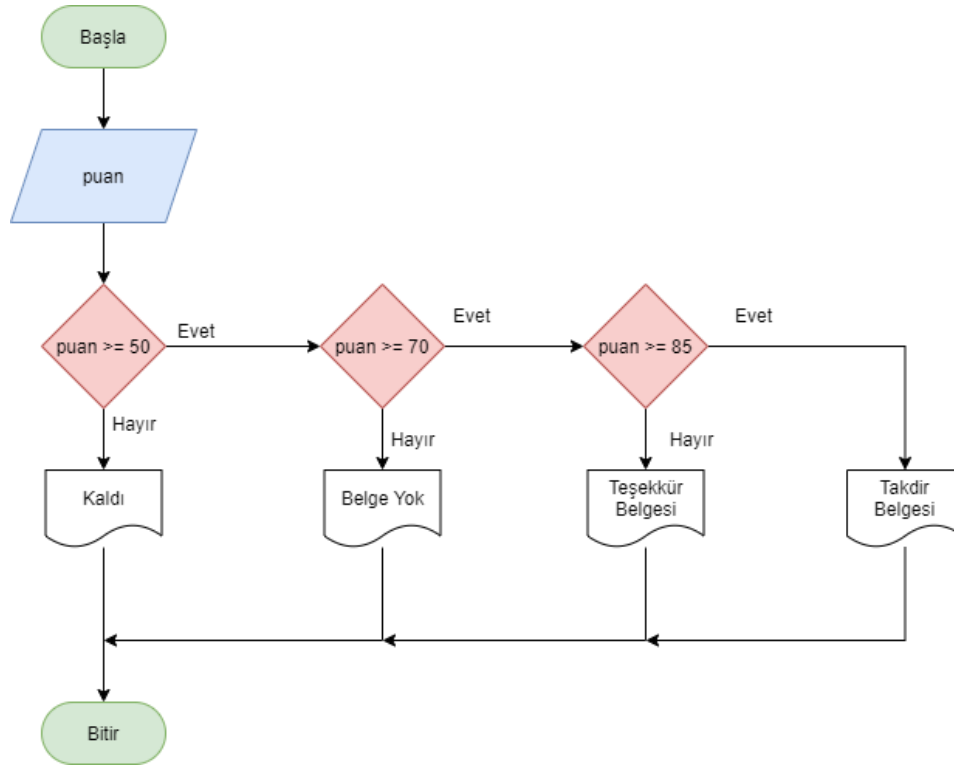
#include <iostream>
using namespace std;

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:

.....

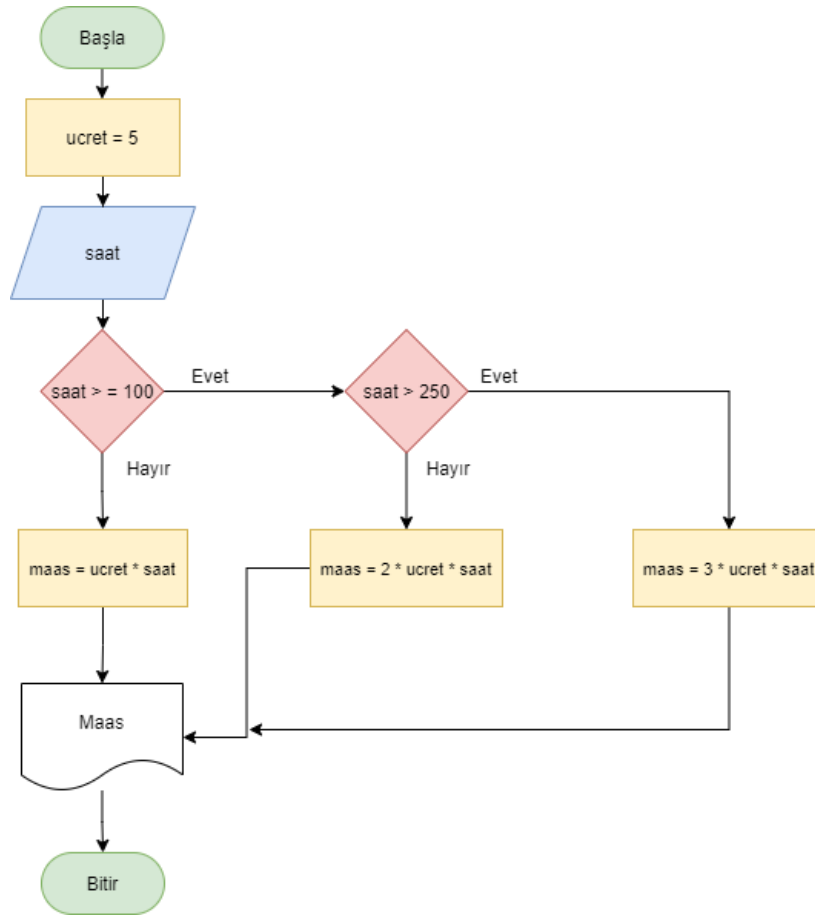
5. Grup Ekran Çıktısı:

.....

```

#include <iostream>
using namespace std;
int main()
{
    int puan;
    cout << "notu giriniz:";
    cin >> puan;
    if(puan<50)
        cout << "kaldı";
    else if(puan<70)
        cout << "belge yok";
    else if(puan<85)
        cout << "teşekkür 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:

.....

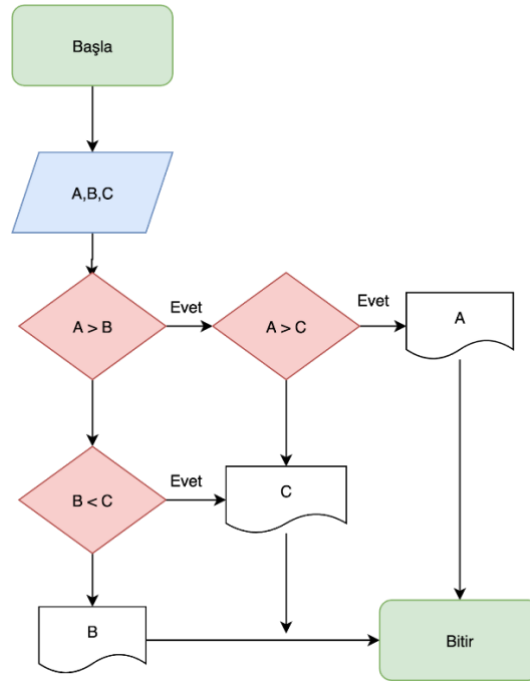
5. Grup Ekran Çıktısı:

.....

```

#include <iostream>
using namespace std;
int main()
{
    int 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;
}
  
```

5. Görev



1. Grup Ekran Çıktısı:

.....

2. Grup Ekran Çıktısı:

.....

3. Grup Ekran Çıktısı:

.....

4. Grup Ekran Çıktısı:

.....

5. Grup Ekran Çıktısı:

.....

```

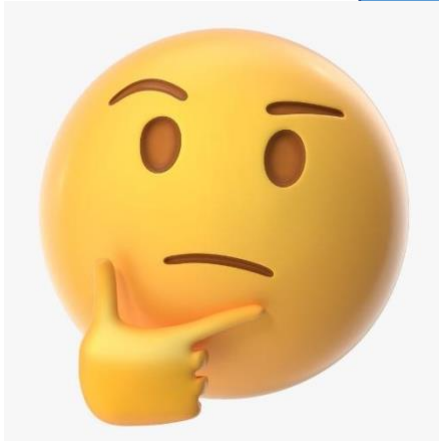
#include <iostream>
using namespace std;

int main()
{
    int A,B,C;
    cout << "Sayi 1:";
    cin >> A;
    cout << "Sayi 2:";
    cin >> B;
    cout << "Sayi 3:";
    cin >> C;

    if(A>B)
    {
        if(A>C)
            cout << A;
        else
            cout << C;
    }
    else if(B<C)
    {
        cout << C;
    }
    else
    {
        cout << B;
    }
    return 0;
}

```

EK 2. SWITCH Kullanımı



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

Cevap: Birden çok koşul olduğu durumlarda if/if-else yapılarını kullanabileceğimiz gibi “switch” bloğunu da kullanabiliriz. Gelin if ile switch kullanımını hesap makinesi yapımı için gereken kodlamayı yaparak kıyaslayalım. Ardından verilen diğer örneği de beraber inceleyelim.

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 == '*')
```

Switch/Case Kullanımı

```
#include <iostream>
using namespace std;
int main()
{
    char islem;
    cin >> islem;
    switch(islem){
        case '+':
            cout << "Toplama islemi";
            break;
        case '-':
```

```

        cout << "Carpma islemi";
    else if(islem == '/')
        cout << "Bolme islemi";
    else
        cout << "Hatali giris.";
}

```

```

        cout << "Cikarma islemi";
        break;
    case '*':
        cout << "Carpma islemi";
        break;
    case '/':
        cout << "Bolme islemi";
        break;
    default:
        cout << "Hatali giris.";
}

```

```

}

```

EK 3. 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. Döngü Yapıları

Kazanımlar

- K1. C++ programında döngü mantığını kavrar.
- K2. C++ programında döngü oluşturabilir.
- K3. C++ programında iç içe döngü yapısını bilir.

Haftanın Amacı ve Materyaller

Bu haftanın amacı öğrencilerin programlamada döngüleri 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

Ders Arası (10 dk.)

SONUÇ (10 dk.)

EK 5. Hafta 5 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: 10 dk.

Uygulama: Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 5 Sunum). Aşağıdaki içeriğe uygun şekilde sunuyu süreci yönetmek için kullanır. Eğitmen tüm öğrencilerin derse katılımını beklerken eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 5 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 beşinci 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 5 Sunum

EK 1. Döngüleri Tanıyalım

Hazırlık: Hafta 5 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 kullanıldınız sorusu yine anket yoluyla 1 dk. zaman verilerek sorulur. Etkinliklerin cevabı aşağıda verilmiştir.



Resim 26. Çözüm 1



Resim 27. Çözüm 2

Eğitmene 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 aldırarak 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 5 Sunum

EK2. 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.

Bir döngüde olması gerekenler şunlardır;

1. Kontrol değişkeni: Döngü başlangıç ve bitişi hangi değişken üzerinden kontrol edilecek.
2. Değişken başlangıç değeri
3. Bitiş koşulu
4. Değişken güncellemesi: Her tekrar sonrasında kontrol değişkeni nasıl etkilenecek?
5. 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 5 Sunum

EK 4. Döngü Görevlerini Bilgisayarlarımızda Kodlayalım

Hazırlık: Eğitmenen bu bölüm için döngüler ile ilgili çözülmesi gereken beş görevin bulunduğu 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: Ahmet okul kütüphanesindeki raflara herkesin kolayca kitapları bulabilmesi için sayı etiketleri yapıştırmak istiyor. Kütüphanede 100 tane raf olduğu düşünülürse Ahmet'in 1 den 100'e kadar sayıları sıralayıp ekranda göstermesi gerekmektedir. Buradan hareketle Ahmet'in nasıl bir kod yazması gereklidir, bilgisayarımızda kodlayalım.

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.

Cevap 1:

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

Görev 2: Ali kardeşi Buğra'nın 1'den 100'e kadar 7'şerli olarak sayı saymasını istemektedir. Buradan hareketle kardeşinin doğru sayıp saymadığını kontrol etmesi için bilgisayarda 1'den 100'e kadar küçükten büyüğe olacak şekilde 7'ye bölünen sayıları ekranda göstermek istiyor. Bunun için Ali 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 yapılması istenir. Eğitmen bitirip el kaldıran öğrencilerden rastgele birini seçerek 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 2'in cevabı aşağıdaki gibidir.

Cevap 2:

```
#include <iostream>
using namespace std;

int main()
{
    for(int i=1; i < 100; i++)
    {
        if(i%7==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 ve 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 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 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: Duru öğretmen öğrencilerine 1'den 9'lara kadar olan çarpım tablosunu öğretmek için bilgisayardan ekran çıktısı alıp yazdırmak istemektedir. Bunun için program yazmak isteyen Duru öğretmen bilgisayarda nasıl kodlamalıdır?

Bu görev öncesi öğrencilere iç içe döngülerin kullanımı hakkında giriş yapılarak nasıl kodlanacağı hakkında temel bilgiler gösterilir.

Not: Birden fazla kontrol değişkeni kullanmamız gerekir ise, iç içe döngüleri kullanırız. Bunun için en güzel örnek çarpım tablosu olabilir.

Yukarıdaki görev için "Görev 5" başlıklı slayt açılır. Daha sonra öğretmen Code::Blocks programında yukarıdaki görevi iç içe döngülerin kullanımına yönelik giriş yaparak, öğrencilerin göreceği şekilde ekran paylaşımı yapılarak kodlamaya başlar. Öğitmen 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. Öğitmen 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:

```
#include <iostream>
using namespace std;

int main()
{
    for(int i=1;i<10;i++)
    {
        for(int j=1; j<10;j++)
        {
            cout << i << "*" << j << "=" << i*j << "\t";
        }
        cout << endl;
    }
}
```

Eğitmene 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.

Eğitmen 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.”

C. Sonuç

Süre: 10 dk.

Materyal: EK 5 Hafta 5 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 Ödev Yanıtları:

1) Kodlayıcı

Klavyeden girilen 10 tam sayının toplamını ekrana yazdıran kodu nasıl yazarsınız?

```
#include <iostream>
#include<cstdlib>
using namespace std;
int main()
{
    int toplam=0;
    for(int i=0;i<10;i++)
    {
        int sayi;
        cin >> sayi;
        toplam += sayi;
    }
    cout << "toplam:" << toplam;
}
```

2) Kodlayıcı

0-100 arasında rastgele üretilen 10 tam sayıdan tek olanların adedini ve toplamını ekrana yazdıran kodu nasıl yazardınız?

```
#include <iostream>
#include<cstdlib>
#include<ctime>
using namespace std;
int main()
{
    srand(time(0));
    int adet=0,toplam=0;
    for(int i=0;i<10;i++)
    {
        int rastgele_sayi = rand()%100;
        if(rastgele_sayi % 2 == 1)
        {
            adet++;
        }
    }
}
```

```

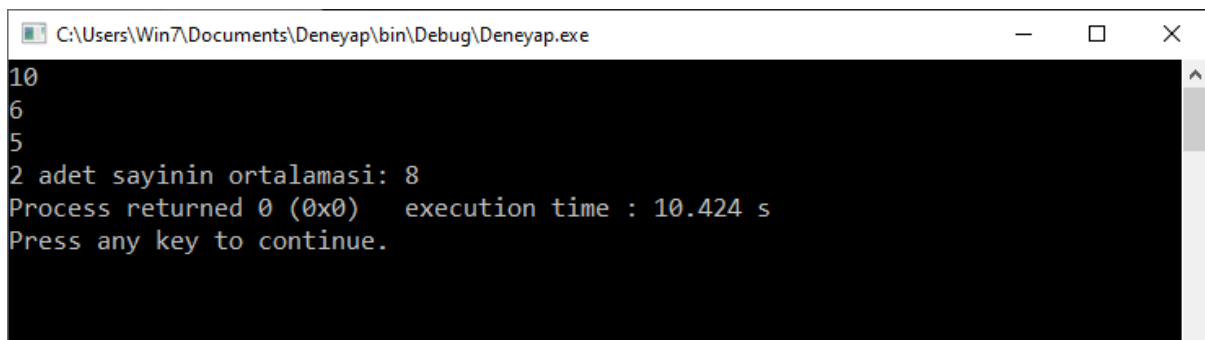
        toplam += rastgele_sayi;
    }
}

cout << adet << " adet tek sayinin toplami:" << toplam;
}

```

3) Kodlayıcı

Klavyeden çift sayı girildikçe toplama işlemi yapan, tek sayı girildiği durumda ise girilen çift sayıların ortalamasını gösteren programı yazalım.



Resim 28. Ekran çıktısı

```

#include <iostream>
using namespace std;
int main()
{
    int sayi, toplam = 0, sayac = 0;
    do
    {
        cin >> sayi;
        if(sayi %2 == 0)
        {
            toplam += sayi;
            sayac++;
        }
    }while(sayi %2 == 0);
    cout << sayac << " adet sayinin ortalamasi: "<<toplam/sayac;
}

```

4) Kodlayıcı

1-100 arasında üçe bölünüp yediye bölünmeyen sayıların miktarını bulan kodu nasıl yazardınız?

```
#include <iostream>
using namespace std;
int main()
{
    int adet=0;
    for(int i=0;i<100;i++)
    {
        if (i%3 == 0 && i%7 != 0)
            adet++;
    }
    cout << adet << " sayi vardır.";
}
```

Hafta 5. Ders Materyalleri

EK 1. Döngüleri Tanıyalım



* 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?

CEVAP:

EK 2. Koddaki Döngülerin Ne İçin Yazıldığını Keşfediyorum

Kod	Problem
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)	
<pre>#include <iostream> using namespace std; int main() { for(int i=0;i<10;i++) cout << "DENEYAP" << endl; return 0; }</pre>	
<pre>#include <iostream> using namespace std; int main() { for(int i=1; i<10; i++) { cout << i <<endl; } return 0; }</pre>	
<pre>int sayi = 1; while(sayi<100) { cout << sayi <<endl; sayi ++; }</pre>	

<pre>int i; for(i=1; i<20; i++) if(i%2==1) cout << i << endl;</pre>	
<pre>int i=1; while(i<20) { if(i%2==1) cout << i << endl; i++; }</pre>	
<pre>int i=1; do { if(i%2==1) cout << i << endl; i++; }while(i<20);</pre>	

EK 3. Döngü Çeşitleri Afişi (Yüksek çözünürlüklü hali materyaller klasöründe bulunmaktadır.)

#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(degisken=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.
}
```

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);
```

İç içe döngü kullanmak istediğimizde birden fazla döngü değişkeni kullanabiliriz.

EK 4. Döngü Görevlerini Bilgisayarlarımızda Kodlayalım

1

Ahmet okul kütüphanesindeki raflara herkesin kolayca kitapları bulabilmesi için sayı etiketleri yapıştırmak istiyor. Kütüphanede 100 tane raf olduğu düşünülürse Ahmet'in 1'den 100'e kadar sayıları sıralayıp ekranda göstermesi gerekmektedir. Buradan hareketle Ahmet'in nasıl bir kod yazması gereklidir, bilgisayarımızda kodlayalım.

2

Ali kardeşi Buğra'nın 1'den 100'e kadar 7'şerli olarak sayı saymasını istemektedir. Buradan hareketle kardeşinin doğru sayıp saymadığını kontrol etmesi için bilgisayarda 1'den 100'e kadar küçükten büyüğe olacak şekilde 7'e bölünebilecek sayıları ekranda göstermek istiyor. Bunun için nasıl bir kod yazmalı?

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?

4

Defne öğretmen aldığı 10 tane kitabı sınıfındaki öğrencilere çekiliş yoluyla dağıtmak istemektedir. Sınıftaki öğrencilerin okul numaraları 50 ile yüz arasındadır. 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.

5

Duru öğretmen öğrencilerine 1'den 9'lara kadar olan çarpım tablosunu öğretmek için bilgisayardan ekran çıktısı alıp yazdırmak istemektedir. Bunun için program yazmak isteyen Duru öğretmen bilgisayarda nasıl kodlamalıdır?

EK 5. 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. Diziler ve Katarlar

Kazanımlar

- K1. C++ programlama dilinde dizi kavramını anlar.
- K2. C++ programlama dilinde tek boyutlu ve çok boyutlu diziler arasında karşılaştırma yapar.
- K3. C++ programlama dilinde dizilere değer atama konusunda uygulama yapar.
- K4. Dizileri döngü içinde kurgular.
- K5. Diziler üzerinde istenen işlemleri gerçekleştirir.
- K6. Diziler ve katarlar arasındaki farkı anlar.

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 + 40 + 60 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 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

Süre: 5 dk.

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, *dikkat çekmek için* eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 6 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 altıncı haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

B. Gelişme

Dizileri Tanıyalım

Süre: 25 dk.

Kazanımlar: K1. C++ programlama dilinde dizi kavramını anlar.

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

Materyaller: Hafta 6 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. Eğitimci 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 eğitimci sunumda aşağıdaki talimatlarla verilmektedir:

Elinizdeki madeni paraları ya da kâğıt ve kalem kullanarak,

- A. *Tek boyutlu bir dizi oluşturun. Dizin boyutunu ve dizi indislerini dizi elemanlarının altına yazın.*
- B. *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 eğitimci 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 eğitimci 5 dk.’lık sözsüz bir odaklanma müziği başlatır. Müzik sona erdiğinde etkinlik de sona erecektir. Eğitimci 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 eğitimci 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 eğitimci konuyu özellikle kendi ilettiği EK 1 örneği üzerinden açıklar ve yanlış öğrenmeleri düzeltir.

Eğitime Öneriler: Etkinlik sırasında müzik sona erdiğinde bireysel olarak oluşturulan dizilerin fotoğrafları öğrenciler tarafından dijital panoya gönderilmiş olmalıdır. Eğitimci 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 6 Sunum

Hazırlık: Eğitimci Hafta 6 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.

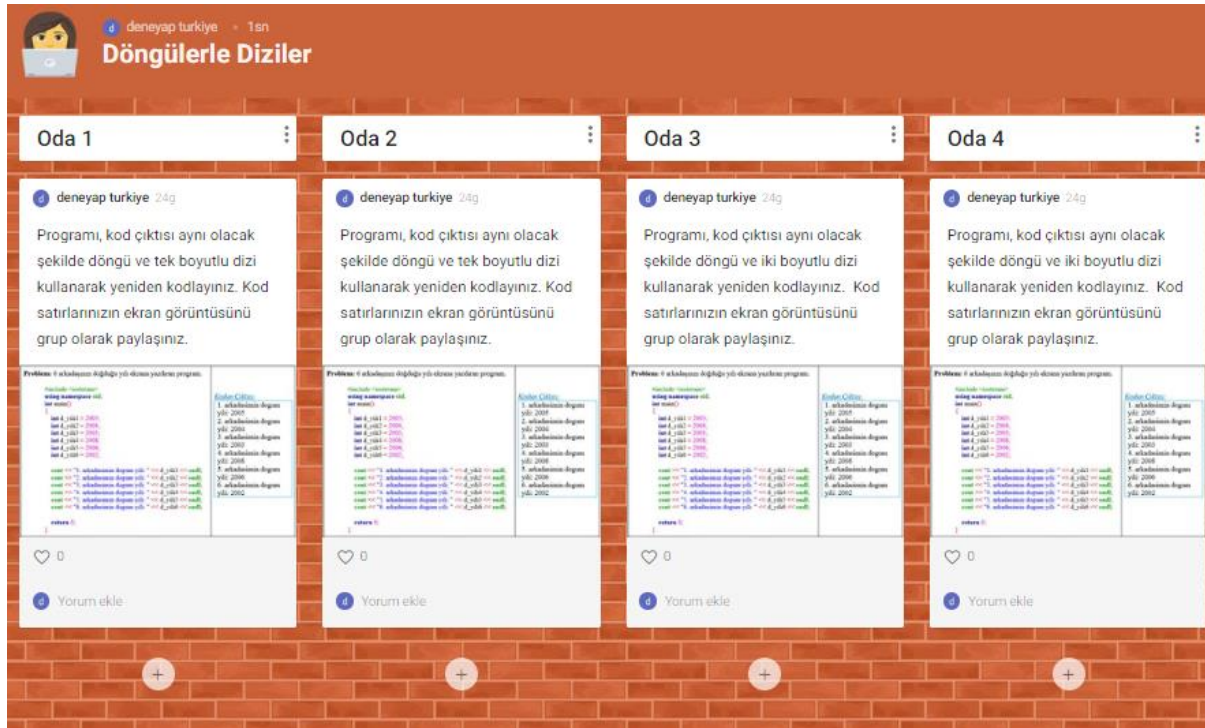
Kazanımlar: K4. Dizileri döngü içinde kullanır.

Materyaller: Hafta 6 Sunum

Dijital Tartışma Panosu: Grup Görevleri

EK 2: Grup Görevleri

Hazırlık: Eğitimci Hafta 6 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 iletecektir. 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 29’da gösterilmiştir.



Resim 29. 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. Öğitmen 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 eğitmen tarafından ekranda paylaşılmış haldedir. Eğitmen 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.

Eğitmene 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	Görev 2
<pre>#include <iostream> using namespace std; int main () { int d_yili[6] = {2005, 2004, 2003, 2008, 2006, 2002}; int i; for(i=0; i<6; i++){ cout << i+1 << ". arkadasimin dogum yili: " << d_yili[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>

Dizilerle Kodlayalım

Süre: 30 dk.

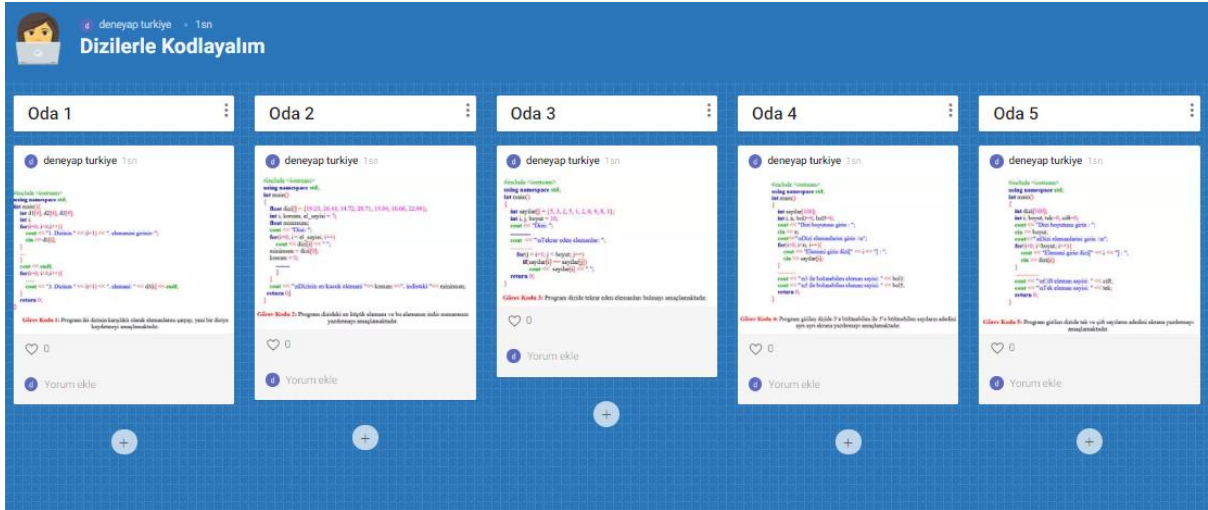
Kazanımlar: K5. Diziler üzerinde istenen işlemleri gerçekleştirir.

Materyaller: Hafta 6 Sunum

Dijital Tartışma Panosu: Grup Görevleri

EK 3: Kod Satırlarını Tamamla

Hazırlık: Eğitmen Hafta 6 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 30'da gösterilmiştir.



Resim 30. 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: Öğitmen 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 d1[4], d2[4], d3[4];
    int i;
    for(i=0; i<4;i++){
        cout << "1. Dizinin " << (i+1) << ".
        elemanini giriniz: ";
        cin >> d1[i];
    }
    cout << endl;
    for(i=0; i<4;i++){
        cout << "2. Dizinin " << (i+1) << ".
        elemanini giriniz: ";
        cin >> d2[i];
    }
    cout << endl;
    for(i=0; i<4;i++){
        d3[i] = d1[i] * d2[i];
        cout << "3. Dizinin " << (i+1) << ".
        elemani: " << d3[i] << endl;
    }
    return 0;
}
```

Görev Kodu 1

Yandaki program iki dizinin karşılıklı olarak elemanlarını çarpıp yeni bir diziye kaydetmeyi amaçlamaktadır.

```
#include <iostream>
using namespace std;

int main(){
    float dizi[] = {19.23, 26.43, 14.72,
28.71, 15.04, 10.06, 22.96};

    int i, y, el_sayisi = 7;
    float x;

    cout << "Dizi: ";
    for(i=0; i < el_sayisi; i++){
        cout << dizi[i] << " ";
    }
    x = dizi[0];
    y = 0;
    for(i=1; i < el_sayisi; i++){
        if(x > dizi[i]){
            x = dizi[i];
            y = i;
        }
    }

    cout << x <<" ve " << y;

    return 0;
}
```

Görev Kodu 2

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

```
#include <iostream>
using namespace std;

int main(){
    int sayilar[] = {5, 3, 2, 5, 1, 2, 6, 9,
8, 1};

    int i, j, boyut = 10;

    cout << "Dizi: ";
    for(i = 0; i < boyut; i++){
        cout << sayilar[i] << " ";
    }
    cout << "\nElemanlar: ";
    for(i = 0; i < boyut-1; i++){
        for(j = i+1; j < boyut; j++){
            if(sayilar[i] == sayilar[j])
                cout << sayilar[i] << " ";
        }
    }

    return 0;
}
```

Görev Kodu 3

Yandaki program dizide tekrar eden elemanları bulmayı amaçlamaktadır.

```
#include <iostream>
using namespace std;
int main()
{
    int sayilar[100];
    int i, n, bol3=0, 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]%3==0)
            bol3++;
        if(sayilar[i]%5==0)
            bol5++;
    }
    cout << bol3 << endl;;
    cout << bol5 << endl;;
}
```

Görev Kodu 4

Yandaki program girilen dizide 3'e bölünebilen ile 5'e bölünebilen sayıların adedini ayrı ayrı ekrana yazdırmayı 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 << cift << endl;;
    cout << tek << endl;;
}
```

Görev Kodu 5

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

Kodlama Ekibi

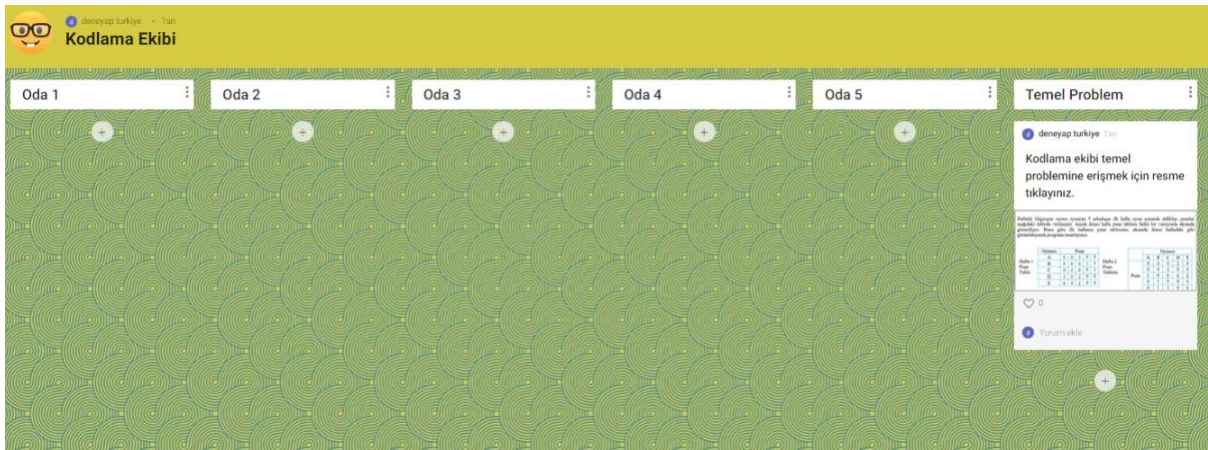
Süre: 30 dk.

Kazanımlar: K5. Diziler üzerinde istenen işlemleri gerçekleştirir.

Materyaller: Hafta 6 Sunum

Dijital Tartışma Panosu: Grup Görevleri

Hazırlık: Eğitimci Hafta 6 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 31’de gösterilmiştir.



Resim 31. 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 Tablo 1’de verilen problemi öğrencilerle paylaşır.

Temel Problem:

Haftalık bilgisayar oyunu oynayan 5 arkadaşın ilk hafta oyun sonunda aldıkları puanlar aşağıdaki tabloda verilmiştir. Ancak ikinci hafta puan tablosu farklı bir versiyonla ekranda gösteriliyor. Buna göre ilk haftanın puan tablosunu, ekranda ikinci haftadaki gibi görüntüleyecek programı tasarlayınız.

Hafta 1
Puan Tablosu

Oyuncu	Puan				
A	0	0	1	0	0
B	0	0	0	1	1
C	0	2	0	0	1
D	2	0	2	0	0
E	0	0	2	0	0

Hafta 2
Puan Tablosu

	Oyuncu				
	A	B	C	D	E
Puan	0	0	0	2	0
	0	0	2	0	0
	1	0	0	2	2
	0	1	0	0	0
	0	1	1	0	0

Eğitmen öğrenci isimlerini alfabetik sıralayarak, sınıfı ikiye ayırır ve grupların görevlerini açıklar. 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 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 oluşturmada önce sunumda “Kodlama Ekibi: Grup Çalışması” başlıklı slayta ilerleyerek öğrencilere açıklama yapar.

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.

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 iki kod satırını birleştirip, diğer kod satırları ile birlikte problem çözümünü tamamlamalıdır. Temel problemin kodlarını aşağıda bulabilirsiniz.

```

1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int i, j, sat, sut, matris1[5][5], matris2[5][5];
6.     sat = 5;
7.     sut = 5;

```

```

8.     cout << "Matrisin elemanlarini sirayla giriniz: " << endl;
9.     for(i=0; i < sat; i++) {
10.         for(j=0; j < sut; j++) {
11.             cin >> matris1[i][j];
12.         }
13.     }

```

```

14.     for(i=0; i < sat; i++) {
15.         for(j=0; j < sut; j++) {
16.             matris2[j][i] = matris1[i][j];
17.         }
18.     }

```

```

19.     cout<<"Matrisin Yer Degistirilmis Hali: " << endl;
20.     for(i=0; i < sut; i++) {
21.         for(int j=0; j < sat; j++) {
22.             cout << matris2[i][j] << " ";
23.         }
24.         cout << endl;
25.     }
26.     return 0;
27. }

```

Farkı Keşfedelim

Süre: 10 dk.

Kazanımlar: K6. Diziler ve katarlar arasındaki farkı anlar.

Materyaller: Hafta 6 Sunum

Hazırlık: Öğitmen Hafta 6 sunumunda “Farkı Keşfedelim” başlıklı slaytı kullanır.

Uygulama: Bu etkinlikte öğretmen sunum üzerinden yeni bilginin sunumunu yapar. Sunumda “Farkı Keşfedelim” başlıklı slayt üzerinde verilen örnek kod öğrencilerle birlikte incelenir. Öğitmen 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 yansıtmaları beklenir. Daha sonra öğretmen 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. Öğitmen öğrenci yanıtlarına bağlı olarak aradaki farkı açıklar ve konuyu özetler.

Eğitmene Öneriler: Öğitmen 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, "Deneyap" 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 katar [] = {'D', 'e', 'n', 'e', 'y', 'a', 'p', '\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 katar [] = "Deneyap";

Aslında yukarıdaki tanımlamada gördüğümüz üzere null karakteri bir katar sabitinin sonuna yerleştirmesiniz. C++ derleyicisi diziyi oluşturduğunda '\0' değerini dizinin sonuna otomatik olarak ekler.

C. Sonuç

Süre: 10 dk.

Materyal: EK 4 Hafta 6 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 "Fonksiyonlar" 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 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: Okul müdürü, Kimya ve Biyoloji dersini alan 6 öğrencinin not ortalamalarını merak etmektedir. Bunun için Bilişim Teknolojileri uzmanından kendisi için bir program yazmasını ister. Bilişim teknolojileri uzmanı bu programı yazmak için iki boyutlu dizilerden yararlanır. Buna göre hazırlanan programı tasarlayınız.

```
#include <iostream>
using namespace std;
int main()
{
    int notlar[2][6] = {{85, 73, 92, 95, 80, 78},
                       {69, 76, 87, 65, 90, 50}};

    int topl = 0, top2 = 0, n = 6;
    float ort1, ort2;
    cout << "Kimya Notlari: " << endl;
    for (int i=0; i < n; i++) {
        cout << notlar[0][i] << " ";
        topl += notlar[0][i];
    }
```

```

cout << "\nBiyoloji Notlari: " << endl;
for (int i=0; i < n; i++) {
    cout << notlar[1][i] << " ";
    top2 += notlar[1][i];
}
ort1 = (float)top1 / n;
cout << "\nKimya ortalamasi: " << ort1 << endl;
ort2 = (float)top2 / n;
cout << "\nBiyoloji ortalamasi: " << ort2 << 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 bulunduğunu 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 kazandiniz!" << endl;
    } else {

```

```

        cout << "Uzgunum, 3 hakkınızda bilemediniz!" << endl;
    }
    return 0;
}

```

Kodlayıcı: Aşağıdaki tabloda voleybol oyuncularının numaraları verilmektedir. Koç, oyuncuları maç öncesi çıktındaki gibi sıraya dizmek istiyor. Bu sırayı oluşturacak kodu tasarlayınız. Program içerisinde tanımlama bölümünde oyuncuların ilk dizilimi aşağıdaki matristeki gibi olmalıdır.

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

Oyuncu Sırası:

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

```

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

    int m = 4, n = 4, i, j = 0, k = 0;
    while (k < m && j < n) {
        for (i = j; i < n; ++i) {
            cout << a[k][i] << " ";
        }
        k++;
        for (i = k; i < m; ++i) {
            cout << a[i][n - 1] << " ";
        }
        n--;
        if (k < m) {
            for (i = n - 1; i >= j; --i) {

```

```
        cout << a[m - 1][i] << " ";
    }
    m--;
}

if (j < n) {
    for (i = m - 1; i >= k; --i) {
        cout << a[i][j] << " ";
    }
    j++;
}

}

return 0;
}
```

Hafta 6. Ders Materyalleri

EK 1. Dizileri Tanıyalım

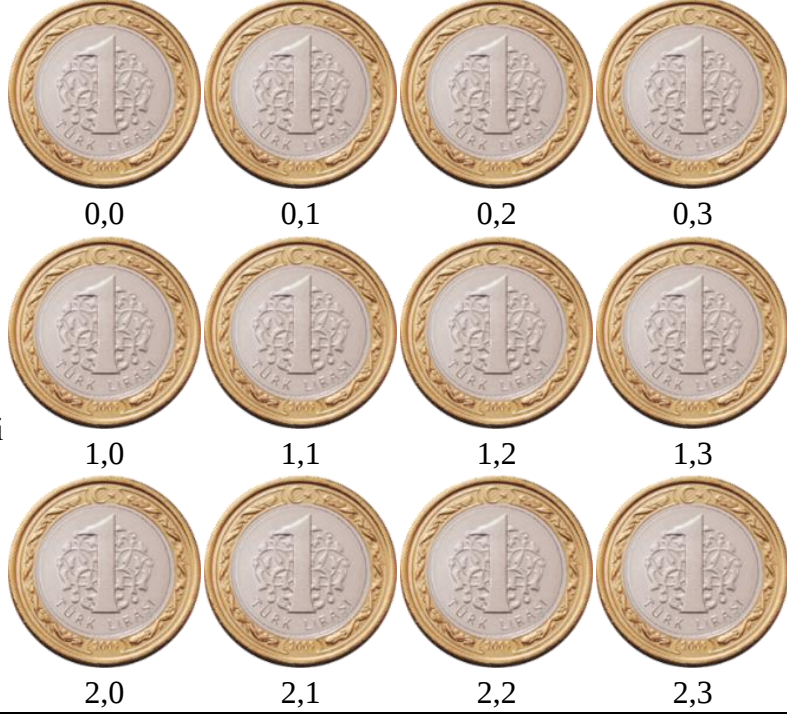
Talimat 1

4 elemanlı tek boyutlu dizi



Talimat 2

4*3'lük
12 elemanlı 3 boyutlu bir dizi



EK 2. Grup Görevleri

Problem: 6 arkadaşının doğduğu yılı ekrana yazdıran program.

```
#include <iostream>
using namespace std;
int main()
{
    int d_yili1 = 2005;
    int d_yili2 = 2004;
    int d_yili3 = 2003;
    int d_yili4 = 2008;
    int d_yili5 = 2006;
    int d_yili6 = 2002;

    cout << "1. arkadasimin dogum yili: " << d_yili1
    << endl;
    cout << "2. arkadasimin dogum yili: " << d_yili2
    << endl;
    cout << "3. arkadasimin dogum yili: " << d_yili3
    << endl;
    cout << "4. arkadasimin dogum yili: " << d_yili4
    << endl;
    cout << "5. arkadasimin dogum yili: " << d_yili5
    << endl;
    cout << "6. arkadasimin dogum yili: " << d_yili6
    << endl;

    return 0;
}
```

Kodun Çıktısı:

1. arkadasimin dogum yili: 2005
2. arkadasimin dogum yili: 2004
3. arkadasimin dogum yili: 2003
4. arkadasimin dogum yili: 2008
5. arkadasimin dogum yili: 2006
6. arkadasimin dogum yili: 2002

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 d1[4], d2[4], d3[4];
    int i;
    for(i=0; i<4;i++){
        cout << "1. Dizinin " << (i+1) << ". elemanini giriniz: ";
        cin >> d1[i];
    }
    ....
}
cout << endl;
for(i=0; i<4;i++){
    .....
    cout << "3. Dizinin " << (i+1) << ". elemani: " << d3[i] << endl;
}
return 0;
}

```

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

```

#include <iostream>
using namespace std;
int main()
{
    float dizi[] = {19.23, 26.43, 14.72, 28.71, 15.04, 10.06, 22.96};
    int i, konum, el_sayisi = 7;
    float minimum;
    cout << "Dizi: ";
    for(i=0; i < el_sayisi; i++)
        cout << dizi[i] << " ";
    minimum = dizi[0];
    konum = 0;
    .....
}
}
cout << "\nDizinin en küçük elemani " << konum << ". indisteki " << minimum;
return 0;
}

```

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

```

#include <iostream>
using namespace std;
int main()
{
    int sayilar[] = {5, 3, 2, 5, 1, 2, 6, 9, 8, 1};
    int i, j, boyut = 10;
    cout << "Dizi: ";
    .....
    cout << "\nTekrar eden elemanlar: ";
    .....
    for(j = i+1; j < boyut; j++)
        if(sayilar[i] == sayilar[j])
            cout << sayilar[i] << " ";
    return 0;
}

```

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

```

#include <iostream>
using namespace std;
int main()
{
    int sayilar[100];
    int i, n, bol3=0, 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 << "\n3 ile bolunebilen eleman sayisi: " << bol3;
    cout << "\n5 ile bolunebilen eleman sayisi: " << bol5;
    return 0;
}

```

Görev Kodu 4: Program girilen dizide 3'e bölünebilen ile 5'e bölünebilen sayıların adedini ayrı ayrı ekrana yazdırmayı 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 5: Program girilen dizide tek ve çift sayıların adedini ekrana yazdırmayı amaçlamaktadır.

EK 4. 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. 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) + 30 dk.)

EK 4. Fonksiyonları Kullanarak Kodlama Yapma

Ders arası (10 dk.)

SONUÇ (5 dk.)

EK 5. 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 eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 7 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 diziler üzerinden *ön bilgileri hatırlatır* ve yedinci 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 7 Sunum

EK 1. Fonksiyonların Özelliklerini Keşfediyorum

Hazırlık: Hafta 7 Sunumunda “Fonksiyon Görevlerini Keşfetme” adlı slaytlar sırası geldikçe açılır.

Uygulama: Eğitmen bu ders için hazırlanan beş görevi sunum üzerinden ikiye dakika arayla teker teker öğrencilerine gösterir. Eğitmen 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 eğitmen 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 eğitmen bazı fonksiyon özelliklerini ipucu kullanarak öğrencilerin keşfetmesine olanak verir.

Eğitmen 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.

Eğitmene 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 7 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 ekranda paylaşarak gösterir ve fonksiyonlar hakkında aşağıdaki gibi bir özetleme yapar.

C++ programlama dilinde fonksiyon yazmak için üç kısım vardır. Bunlardan ilki geriye döndürülecek değişkenin tipi (dönüş tipi), ikinci olarak fonksiyonun ismi, son olarak fonksiyon içerisinde ihtiyaç duyulan bilgiler yani parametrelerdir.

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
}
```

Fonksiyonları amacını belirtecek şekilde isimlendirmeye özen gösterilmelidir. Bu sayede bir fonksiyonu kullanılmak istendiğinde amacı kolayca anlaşılabilir. 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.

Eğitmene 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 fonksiyon tanımlayabilir.

K2. C++ programlama dilinde fonksiyon oluşturmayı bilir.

K3. C++ programlama dilinde fonksiyon çağırmaı bilir.

Materyaller: Hafta 7 Sunum

EK 4 Fonksiyonları Kullanarak Kodlama Yapma

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

Uygulama: Bu bölüm 5 görevden oluşmaktadır. Her bir görev için eđitmenin 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 eđitmen Code::Blocks programında yukarıdaki görevi öğrencilerin göreceđi şekilde ekran paylaşımı yapılarak kodlamaya başlar. Eđitmen her satır kodun ne anlama geldiđini ne için yazıldığını öğrencilere aktarır. Daha sonra da öğrencilerden bu kodları kendi bilgisayarlarında 5 dk. içinde kodlaması istenerek, öğrenciden performansını göstermesi beklenir. Eđitmen 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 dönmeyeceđi için “void” yani “boş” olarak belirtiyoruz. Eğer bir geri dönüş tipi belirtirsek (int, double vs.), kesinlikle geriye o türde bir değeri 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 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 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 yerde 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 tam sayı tipindeki dizinin en büyük sayısını ekrana yazan 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 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’ün cevabı aşağıdaki gibidir.

Cevap 3:

```
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 4: İki dizi içerisindeki en büyük iki sayının toplamını bulan 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:

```

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;
}

```

Görev 5: Bir bilgisayar programında, iki adet fonksiyon bulunmaktadır. İlk fonksiyonda, yaş bilgisi alınmakta ve fonksiyon içerisinde güncellenmektedir. Bu güncellenen değer ana program bloğunu etkilememektedir. İkinci fonksiyonda alınan yaş bilgisi fonksiyon içerisinde güncellenmekte ve değişiklik ana programda etkili olmaktadır. Bunun için nasıl bir program yazmalıyız?

Cevap 5:

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 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 5'in cevabı aşağıdaki gibidir.

Bu görev öncesi öğrencilere fonksiyonlarda referans gönderme hakkında giriş yapılarak nasıl kodlanacağı hakkında temel bilgiler gösterilir.

Referans gönderme:

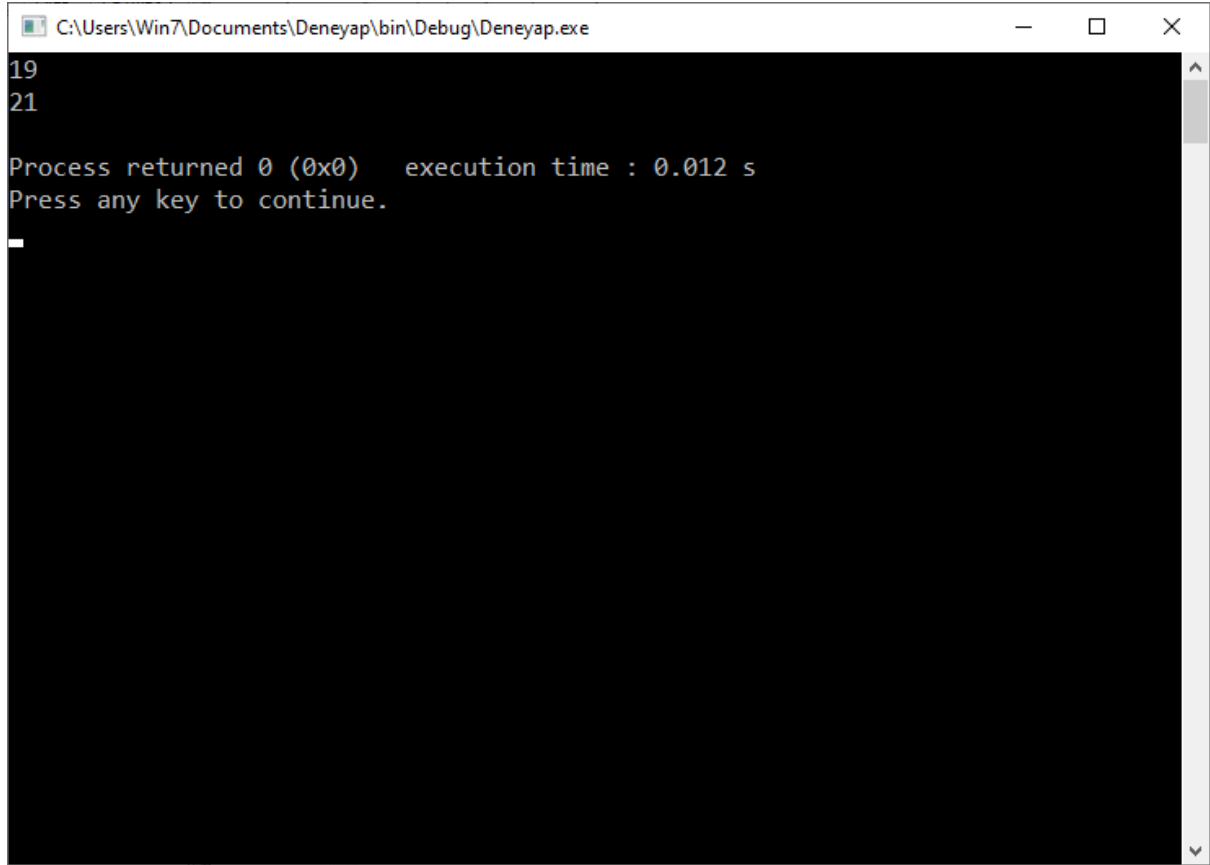
Eğer fonksiyon içerisinde parametre olarak gönderilen değişkenin değeri değişecek ise, değişkenin değeri yerine adresini göndeririz. Böylece fonksiyon içerisindeki değişiklikler değişken üzerine yansiyacaktır. Aşağıdaki örnek bu konuyu açıklayacaktır.

Örnek: Değişken değeri değiştirme referanslı ve referanssız.

```
void fonksiyon1(int sayi)
{
    sayi = 20;
}
void fonksiyon2(int& sayi)
{
    sayi = 21;
}
int main()
{
    int yas = 19;
    fonksiyon1(yas);
    cout<<yas <<endl;

    fonksiyon2(yas);
    cout<<yas <<endl;
}
```

fonksiyon1'e yas isimli değişkenin değerini yani 19 değerini gönderiyoruz. fonksiyon içerisinde sayi isimli değişken üzerinde yapılan değişiklik ana programdaki "yas" isimli değişkeni etkilemeyecektir. fakat fonksiyon2'de fonksiyona "&" işareti kullanarak değişkenin adresini gönderiyoruz. Dolayısıyla fonksiyon içerisindeki tüm değişiklikler ana programa yansiyacaktır. Sonuç olarak program çıktısı aşağıdaki gibi olacaktır. Bu sayede birden fazla değişkenin değerini fonksiyon içerisinde değiştirebiliriz.



```

C:\Users\Win7\Documents\Deneyap\bin\Debug\Deneyap.exe
19
21
Process returned 0 (0x0)   execution time : 0.012 s
Press any key to continue.

```

Resim 32. Ekran çıktısı

Görev 6: Harita mühendisi olan Ali kendisine gönderilen arazinin kenarları bir tane gönderildi ise karenin, iki tane gönderildi ise dikdörtgenin çevresini bulduracak bir fonksiyon tanımlamak istemektedir. Sizce Ali nasıl bir kod yazmalıdır?

Yukarıdaki görev için “Görev 6” 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 6’nın cevabı aşağıdaki gibidir.

Cevap 6:

Bu görev öncesi öğrencilere fonksiyonlarda aşırı yükleme hakkında giriş yapılarak nasıl kodlanacağı hakkında temel bilgiler gösterilir.

Aşırı Yükleme:

Bir fonksiyonu birden farklı şekilde kullanabilmek için fonksiyonları farklı şekilde tanımlayabiliriz. Örneğin: bir fonksiyon ile dikdörtgen/kare çevresini hesaplayalım. Eğer kullanıcı fonksiyona iki değişken gönderir ise bu bir dikdörtgen, tek değişken gönderirse kare olarak ele alır ve işlemleri buna göre yaparız.

```
#include <iostream>
using namespace std;
int cevre_hesapla(int a)
{
    int cevre = a*4;
    return cevre;
}
int cevre_hesapla(int a, int b)
{
    int cevre = 2*a + 2*b;
    return cevre;
}
int main()
{
    int cevre1 = cevre_hesapla(5);
    int cevre2 = cevre_hesapla(5,4);

    cout << cevre1 <<endl;
    cout << cevre2 <<endl;
}
```

Görev 7: Matematik Öğretmeni Hasan, öğrencilerine gösterebilmek adına ekrana girilen sayının faktöriyelini bulduran bir program oluşturmak istemektedir. Bunun için nasıl bir kod yazmalıdır?

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 7:

```
#include <iostream>
using namespace std;
int faktoriyel(int sayi)
{
    if(sayi == 1)
```

```

    return 1;
else
    return sayi*faktoriyel(sayi-1);
}
int main()
{
    int sonuc = faktoriyel(5);
    cout << sonuc << endl;
}

```

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 7 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 7 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 ileildikten sonra öğretmen 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.

Kodlayıcı: Parametre olarak gönderilen iki sayının büyüğünün küçüğüne bölümünden kalanı geriye döndüren bir fonksiyon yazınız.

```

#include <iostream>
using namespace std;
int kalan_bul(int sayi1, int sayi2)

```

```

{
    if(sayi1 > sayi2)
        return sayi1 % sayi2;
    else
        return sayi2%sayi1;
}

int main()
{
    int sonuc = kalan_bul(9,12);
    cout << sonuc;
}

```

Kodlayıcı: Parametre olarak gönderilen dizi içerisindeki sıfırdan büyük sayıların toplamını bulup geriye döndüren bir fonksiyon yazınız.

```

#include <iostream>
using namespace std;
int dizi_topla(int dizi[5])
{
    int toplam = 0;

    for(int i=0;i<5;i++)
        if(dizi[i] > 0)
            toplam = toplam + dizi[i];

    return toplam;
}

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

```

Kodlayıcı: Parametre olarak gönderilen sayının asal sayı olup olmadığını belirleyen fonksiyon yazınız. (asal ise 1 değilse 0 döndürsün.)

```
#include <iostream>

using namespace std;

int asal_sayi_mi(int sayi)
{
    for(int i=2;i<sayi;i++)
        if( sayi% i == 0)
            return 0;

    return 1;
}

int main()
{
    int sonuc = asal_sayi_mi(23);

    cout << sonuc;
}
```

Analizci: Aşağıdaki programın ekran çıktısı nedir?

```
#include <iostream>

using namespace std;

int sayi=2;

void fonksiyon1()
{
    sayi = 5;
}

void fonksiyon2()
{
    int sayi = 7;
}

int main()
{
    fonksiyon1();

    fonksiyon2();

    cout << sayi;
}
```

Cevap:5

Hafta 7. 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. Instagramı 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? (Yüksek çözünürlüklü hali materyaller klasöründe bulunmaktadır.)



Resim 33. Fonksiyon tanımlama afişi

EK 3. C++ Programlama Dilinde 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 alanının hesaplanmasına 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 tam sayı tipindeki dizinin en büyük sayısını ekrana yazan fonksiyonu yazalım. (Instagram üzerindeki en fazla beğeni alan fotoğraf)

4

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

5

Bir bilgisayar programında, iki adet fonksiyon bulunmaktadır. İlk fonksiyonda, yaş bilgisi alınmakta ve fonksiyon içerisinde güncellenmektedir. Bu güncellenen değer ana program bloğunu etkilememektedir. İkinci fonksiyonda alınan yaş bilgisi fonksiyon içerisinde güncellenmekte ve değişiklik ana programda etkili olmaktadır. Bunun için nasıl bir program yazmalıyız?

6

Harita mühendisi olan Ali kendisine gönderilen arazinin kenarları bir tane gönderildi ise karenin, iki tane gönderildi ise dikdörtgenin çevresini bulduracak bir fonksiyon tanımlamak istemektedir. Sizce Ali nasıl bir kod yazmalıdır?

7

Matematik Öğretmeni Hasan, öğrencilerine gösterebilmek adına ekrana girilen sayının faktöriyelini bulduran bir program oluşturmak istemektedir. Bunun için nasıl bir kod yazmalıdır?

EK 5. 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. 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, prosedürel programlamadan farkını, avantaj ve dezavantajlarını analiz etmek ve sınıf, nesne, fonksiyon tanımlama işlemlerini uygulamaktır.

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

Önerilen Ders Akışı (50 + 45 + 45 + 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 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, *dikkat çekmek için* eğlenceli bir müzik açarak öğrencilerini karşılar ve Hafta 8 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 sekizinci haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

B. Gelişme

Nesneni Çiz

Süre: 45 dk

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

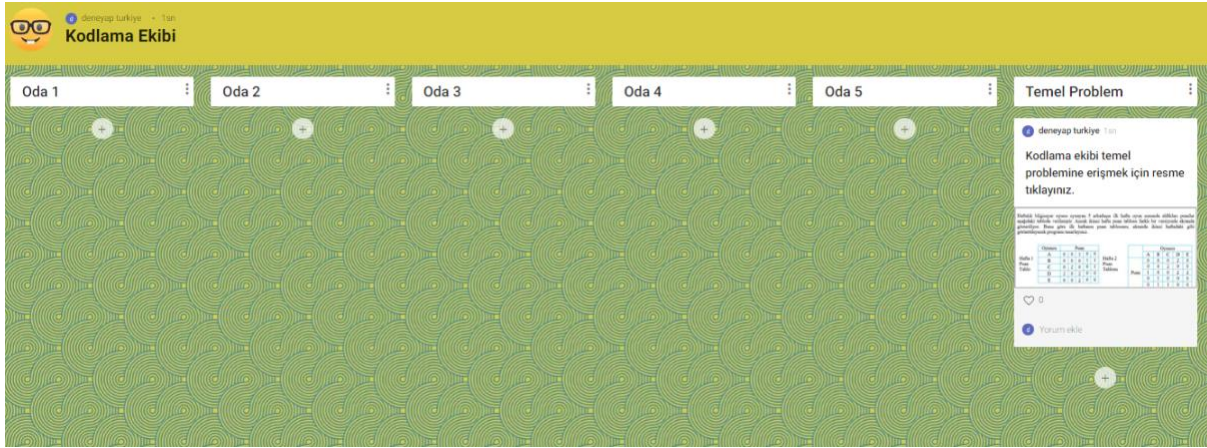
K2. Nesne yönelimli programlamayı prosedürel programlamadan ayırt eder.

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 34’te gösterilmiştir.



Resim 34. Nesneni Çiz: Grup Çalışması Örnek Padlet Tablosu

Uygulama: Öğretmen öğrencilere konu öncesi çalışma odalarına dağıtarak bir oyun oynayacaklarını söyler. Hafta 8 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 8 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.

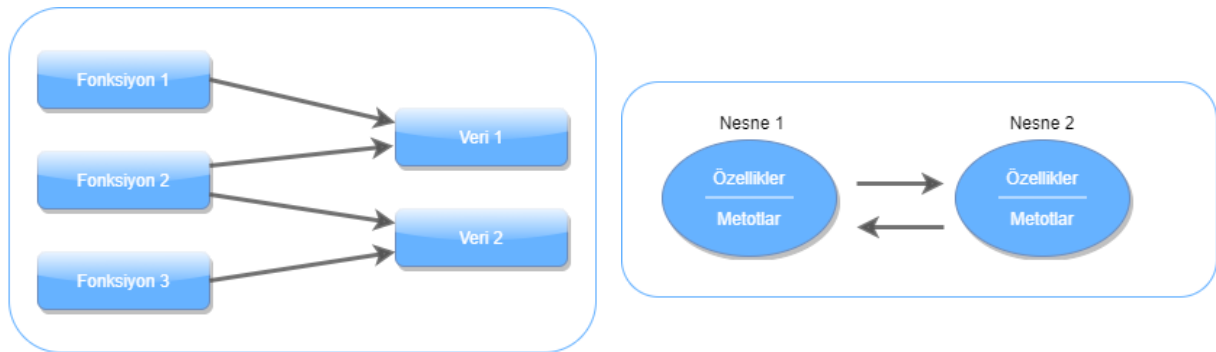
Nesne yönelimli programlama, gerçek dünya ortamına dayalı modeller oluşturmak için sınıfları ve nesneleri kullanan bir programlama dilidir. Diğer bir ifadeyle, nesne yönelimli programlama, programları nesnelerin sınıfları açısından tanımlayan bir yazılım tasarım ve programlama yöntemidir. Bir nesne yönelimli programlama uygulaması, belirli bir hizmeti veya bilgiyi talep

etmek için çağrıldığında mesajları iletecek bir nesneler koleksiyonunu kullanabilir. Nesneler, verileri veri biçiminde iletebilir, alabilir veya bilgileri işleyebilir.

Tablo 25. Prosedürel Programlama ve Nesne Yönelimli Programlama Arasındaki Farklar

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.

Prosedürel programlama, veriler üzerinde işlemler gerçekleştiren fonksiyonlar yazmakla ilgiliyken, nesne yönelimli programlama ise hem verileri (özellikler) hem de fonksiyonları içeren nesneler oluşturmakla ilgilidir. Aşağıdaki şekilde görüldüğü üzere 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 35. Yordamsal ve nesne yönelimli programlama

Nesne (Object) Kavramı

Nesne yönelimli programlamanın temel birimi olan, bazı özellikleri ve davranışları olan tanımlanabilir bir varlıktır. Yani hem veriler hem de veriler üzerinde çalışan fonksiyonlar nesne olarak adlandırılan bir birim olarak oluşturulur. Örneğin: bisiklet, sandalye, kitap, top ve cep telefonu gibi.



Resim 36. Nesne örnekleri

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

Sınıf (Class) Kavramı

Bir sınıf tanımladığınızda, bir 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. Cep telefonu bir sınıf ise bundan üretilen farklı cep telefonu çeşitleri de nesneleri ifade eder.



Resim 37. Sınıf ve nesne kavramı

Nesne Yönelimli Programlamanın Avantaj ve Dezavantajları

Nesne yönelimli programlamayı kullanmamızın bir nedeni, yeni nesneler mevcut olanlardan özellikleri miras alarak oluşturulurken mevcut kodu korumayı ve değiştirmeyi kolaylaştırmasıdır. Bu, geliştirme süresini önemli ölçüde azaltır ve programlamayı çok daha basit hâle getirir. Diğer bir neden, geliştirme kolaylığı ve diğer geliştiricilerin geliştirmeden sonra programı anlama becerisidir. İyi yorumlanmış nesneler ve sınıflar, geliştiriciye programın geliştiricisinin izlemeye

çalıştığı süreci söyleyebilir. Ayrıca yeni geliştirici için programa eklemeleri çok daha kolay hâle getirebilir.

Nesne yönelimli programlamanın, prosedürel programlamaya göre birçok avantajı vardır:

- ❖ Daha hızlı 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.

Nesne Yönelimli Programlamanın Temelleri

C++ programlamanın temel amacı, benzerleri arasında en güçlü ve en yaygın kullanılan programlama dillerinden biri olan C programlama diline nesne yönelimi eklemektir. Nesne yönelimli programlamanın özü, kod içerisinde belirli özelliklere ve yöntemlere sahip bir nesne oluşturmaktır. C++ modüllerini tasarlarken tüm dünyayı nesneler şeklinde tasarlamaya çalışıyoruz. Örneğin bir araba, renk, kapı sayısı ve benzeri gibi belirli özelliklere sahip bir nesnedir. Ayrıca hızlanma, frenleme gibi belirli yöntemlere de sahiptir. Nesne yönelimli programlamanın temelini oluşturan altı ana kavram vardır. Bunlardan ilk ikisi yukarıda detaylı olarak açıklanmıştı. Ama tekrar olması açısından hepsini listelersek:

1) Nesne (Object): Nesne yönelimli programlamanın temel birimidir. Veri ve veriler üzerinde çalışan fonksiyonlar, nesne olarak adlandırılan bir birim olarak paketlenmiştir.

2) Sınıf (Class): Nesne yönelimli programlamaya yapı taşıdır. Sınıfın bir örneğini oluşturarak erişilebilen ve kullanılabilen kendi veri üyelerini ve fonksiyonlarını tutan kullanıcı tanımlı bir veri türüdür.

3) Soyutlama (Abstraction): Dış dünyaya yalnızca gerekli bilgilerin sağlanması ve arka plan ayrıntılarının gizlenmesi, yani gerekli bilgilerin programda ayrıntıları verilmeden temsil edilmesidir.

4) Kapsülleme (Encapsulation): Veri ve bilgilerin tek bir birim altında toplanması olarak tanımlanır. Verileri ve onları işleyen fonksiyonları birbirine bağlamak olarak tanımlanır.

5) Miras (Inheritance): Bir sınıf oluşturulurken başka bir sınıfın özellikleri ve karakteristiklerinin türetilerek kullanılmasıdır. Türetilmiş sınıf olarak adlandırılan yeni bir sınıf oluşturulur.

6) Polimorfizm (Polymorphism): Birden fazla form anlamına gelir. Bir üye fonksiyonun onu çağıran nesneye göre farklı davranabilmesidir.

Sınıfının Özelliklerini Tanı

Süre: 45 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 38'de gösterilmiştir.

The image displays a digital discussion board titled "Grup Afişleri: Sınıfının Özelliklerini Tanı" (Group Posters: Know the Features of the Class). It consists of four panels, each representing a different class: Cep Telefonu, Kış, Öğrenci, and Çalışan. Each panel contains C++ code snippets and a diagram illustrating the class structure, including attributes, methods, and relationships. The panels are arranged in a 2x2 grid, and each has a "Yorum ekle" (Add Comment) button at the bottom.

Oda 1: Cep Telefonu

```
#include <iostream>
#include <string>
using namespace std;

class CepTelefonu {
public:
    char marka[20];
    int fiyat;
    bool aramaDurum;
    void arama();
};

void CepTelefonu::arama() {
    aramaDurum = true;
    cout << "Telefonu arama başlatılıyor..." << endl;
}

int main() {
    CepTelefonu urun;
    strcpy(urun.marka, "Samsung");
    urun.fiyat = 4500;
    urun << "Sıra:" << urun.marka << " Fiyat: " << urun.fiyat << endl;
    urun.arama();
    return 0;
}
```

Oda 2: Kış

```
#include <iostream>
#include <string>
using namespace std;

class Kış {
public:
    char tur[20];
    char ad[20];
    void uuma();
};

void Kış::uuma() {
    cout << "Kışlar kanatlarını kullanan uçaklar" << endl;
}

int main() {
    Kış k1;
    strcpy(k1.tur, "Muhafazat Kuşu");
    strcpy(k1.ad, "Stonok");
    cout << "Kışın türü: " << k1.tur << "Adı: " << k1.ad << endl;
    k1.uuma();
    return 0;
}
```

Oda 3: Öğrenci

```
#include <iostream>
#include <string>
using namespace std;

class Öğrenci {
private:
    int ogr_no;
    char ogr_ad[20];
    char ogr_soyad[20];
    void ekle(int no, char ad[], char soyad[]);
};

ogr_no = 0;
strcpy(ogr_ad, "Ad");
strcpy(ogr_soyad, "Soyad");

void göster() {
    cout << "Öğrenci Bilgi:" << ogr_no << " " << ogr_ad << " " << ogr_soyad << endl;
}

int main() {
    Öğrenci ogr;
    ogr.ekle(172, "Arda", "Özcan");
    ogr.goster();
    return 0;
}
```

Oda 4: Çalışan

```
#include <iostream>
#include <string>
using namespace std;

class Çalışan {
public:
    int sicil_no;
    char ad[20];
    float maaş;
    void ekle(int s, char a[], float m);
};

sicil_no = 0;
strcpy(ad, "a");
strcpy(maas, "m");

void göster() {
    cout << "Çalışan Bilgi:" << sicil_no << " " << ad << " " << maas << endl;
}

int main() {
    Çalışan c1;
    c1.ekle(1002, "Emre", 4000);
    c1.goster();
    return 0;
}
```

Resim 38. Grup Afişleri Örnek Dijital Tartışma Panosu

Uygulama: Eğitimci Hafta 8 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 eğitimci 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. Eğitimci 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. Eğitimci 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. Eğitimci zaman planlaması için bir çan sesi kullanarak süreci yönetebilir.

Eğitime Öneriler: Eğitimci gruplara rastgele dağıttığı öğrencilerin listesini almalıdır. Eğitimci 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 fonksiyonlarını tanımlar. Cep telefonuna ait bir sınıf oluşturmak istendiğinde bu sınıfa ait özellikleri ve fonksiyonları belirlememiz gerekmektedir. Özellikler sınıfın ayırt edici bilgilerinden oluşan marka, model ve imei no gibi verileri içermektedir. Fonksiyonlar ise sınıfın arama, mesaj gönderme ve internete bağlanma 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

{

 üyeListesi

};

Burada **class**, sınıf tanımına ait anahtar kelime, **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 virgülün unutulması derleyici tarafından yakalanacak bir hataya sebep olacaktır.

UYARI:

Her sınıf ismini büyük harfle başlatın. Bu kural sadece C++ 'da değil, aynı zamanda diğer tüm nesne yönelimli programlama dillerinde de kullanılır.

üyeListesi, üye bildirimlerini içeren listeden oluşur. Bunlar veri üyesi veya 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.

```
int x;
```

```
float y;
```

```
char z;
```

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 cep telefonu sınıfında markanın, modelin ve imei numarasının ne olduğu bu veri üyeleri tarafından saklanır. Bir sınıfın veri üyeleri, sınıf özelliklerini tanımlar ve açıklar. Bu nedenle, her bir cep telefonu nesnesinin markası, o nesnenin bir özelliğidir. Aşağıda cep telefonu sınıfına ait marka, model, imei numarası, renk ve fiyat veri üyelerinin sınıf içerisinde nasıl tanımlandığını görebilirsiniz.

```
class CepTelefonu
```

```
{
    char marka[30];
    char model[30];
    int imei_no;
    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 belirteci etkiler. Bir sınıfın üç farklı erişim belirleyicisi bulunmaktadır:

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.

3) protected: Bu anahtar kelime ile tanımlanan üyelere yalnızca aynı sınıfın diğer üye fonksiyonları ve bu sınıftan türetilen sınıfların 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 Ceptelevonu
{
    char marka[30];
    int yil;
public:
    char model[30];
    char renk[30];
private:
    float fiyat;
protected:
    int imei_no;
};

```

Yukarıda verilen sınıf tanımlamasında fiyat veri üyesi ile birlikte marka ve yıl veri üyelerinin de “private” olduğunu unutmayınız. Diğer taraftan model ve renk “public” olarak tanımlı iken imei numarası ise “protected” olarak tanımlanmıştır. Sınıf tanımlamalarınızda istediğiniz kadar erişim belirteciniz olabilir, ancak belirteçleri tek bir grup altında toplamak sınıfın anlaşılabilirliğini artıracaktır.

UYARI:

Sınıflarınızı projeye eklemekten önce çalıştığından emin olmak için test edin. Bir sınıfın çalıştığını gözlemledikten sonra, sınıflar arasındaki etkileşimin doğru şekilde çalıştığından emin olmanız yeterlidir.

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 ancak sınıfın nesneleri kullanılabilir. Tanımlanan bu fonksiyonlara yalnızca sınıfın bir nesnesi aracılığıyla erişilebilir. Bir fonksiyon tanımladığınızda, kapsam çözümleme operatörü “::” aracılığıyla fonksiyon adından ayıracak şekilde sınıf adını da eklemeliyiz. Sınıf adını eklemezsek global fonksiyon tanımı çağrılmaya çalışılır.

UYARI:

Tanımladığınız her bir fonksiyonun yalnızca bir iş yaptığından emin olun. Fonksiyon içerisinde çok fazla kod satırınız var ise, çok fazla şey yapmaya çalışıyorsunuzdur.

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
}
```

Gördüğünüz gibi, bu sözdizimi fonksiyon tanımı sözdizimine çok benzemektedir. Tanımlanan fonksiyon içinde, sınıfa ait tüm üyelere adları kullanılarak erişilebilir. Sınıf üyeliği otomatik olarak sağlanır ve aynı sınıfa ait fonksiyonlar birbirini doğrudan çağırabilir. “private” üyelere erişim yalnızca aynı sınıfa ait fonksiyonlar ile mümkündür. Böylece “private” üyeler tamamen sınıf tarafından kontrol edilir. Bir sınıf tanımladığınızda, sınıfa ait veri üyeleri için bellekten otomatik olarak yer ayrılmaz. Bellek ayırmak için bir nesne tanımlanması gerekir. Belirli bir nesne için bir fonksiyon çağırıldığında, ilgili fonksiyon bu nesnenin verilerini işleyebilir. Artık Cep telefonu sınıfının arama ve mesaj gönderme fonksiyonlarını uygulayabiliriz.

Nesne Oluşturma

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 belirtin. Nesne oluşturmak için aşağıda verilen sözdizimlerinden birini kullanabilirsiniz:

SınıfAdı nesneDeğişkeni;

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 eğitimci öğrencileri cesaretlendirir. Eğitimci bu şekilde öğrencilerden doğru çalışan kodlar ile grup afişlerini yeniden tasarlamalarını ve panoya kendi tasarladıkları afişleri göndermelerini ister.

Eğitime Öneriler: Eğitimci 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 afiş oluşturulmaları istenebilir.

C. Sonuç

Süre: 10 dk.

Materyal: EK 3. Hafta 8 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 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

Denetleyici: Kardeş olan iki peyzaj ustası kare şeklinde olan bahçelerine peyzaj yapmak istiyor. Her ikisi de kendi bahçelerine hem çit gerekcek hem de belli bir alana çiçek ekecektir. Çit ve çiçek alışverişi için kare şeklindeki iki bahçenin kenar bilgilerini ölçerek, bahçelerin alan ve çevresini hesaplayan bir program tasarlarlar. Ancak program hata vermektedir. Programı düzeltmek zorundalar çünkü hesaplayacakları daha pek çok bahçe var. Bu iki kardeşe programı baştan tasarlamaları için yardım ediniz.

İPUCU:

İki kardeşin kullandığı hatalı programda sınıf içerisinde kullanılacak kenar bilgileri değer atama yoluyla gerçekleşirken, 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;
    }
    float alan() {
        return kenar * kenar;
    }
}
```

```

};

void Kare::deger_atama (float k) {
    kenar = k;
}

int main () {
    Kare kare1, kare2;
    kare1.deger_atama (6.2);
    kare2.deger_atama (4.3);

    cout<<"Kare 1 Cevresi: "<<kare1.cevre()<<" Alani: "<<kare1.alan()<<"\n";
    cout<<"Kare 2 Cevresi: "<<kare2.cevre()<<" Alani: "<<kare2.alan()<<"\n";
}

```

Kodun Çıktısı:

```

Kare 1 Cevresi: 24.8 Alani: 38.44
Kare 2 Cevresi: 17.2 Alani: 18.49

```

Kodlayıcı: Ev sahipleri bahçelerine biri yetişkin diğeri çocuk için iki havuz yaptırmak istiyor. Bahçeyi inceleyen ustanın, havuzların yapılacağı alanı hesaplamaya ihtiyacı var. Ev sahiplerine istedikleri havuzların yarıçaplarını belirlemelerini istiyor. Ev sahiplerinden bu bilgiyi aldıktan sonra, daire şeklindeki iki havuzun alanını hesaplayan bir programa bilgileri giriyor. Ustanın kullandığı programın kodlarını tasarlayınız.

İPUCU:

Sınıf içerisinde tanımlanacak yarıçap bilgilerinin 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 daire1, daire2;  
  
cout << "1. dairenin yaricapini giriniz: " << endl;  
cin >> daire1.yari_cap;  
  
cout << "2. dairenin yaricapini giriniz: " << endl;  
cin >> daire2.yari_cap;  
  
cout<<"Daire 1 Alani: "<<daire1.alan_bul(daire1.yari_cap)<<"\n";  
cout<<"Daire 2 Alani: "<<daire2.alan_bul(daire2.yari_cap)<<"\n";  
}
```

Kodun Çıktısı:

```
1. dairenin yaricapini giriniz:  
4.2  
2. dairenin yaricapini giriniz:  
3.6  
Daire 1 Alani: 55.3896  
Daire 2 Alani: 40.6944
```

Hafta 8. 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 materyaller klasörüne gidiniz.

EK 3. 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. 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.
- K4. Kod içinde sınıf dosyaları ile proje 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 + 50+ 50+ 50 dk.)

GİRİŞ (5 dk.)

GELİŞME

Yarış Benimle (45 dk.)

EK 1. Yarışma Kartları

Ders Arası (10 dk.)

Kod Satırlarını Tamamla (50 dk.)

EK 2. Grup Görev Kartları

Ders Arası (10 dk.)

Sınıfları Dosyala (40 dk.)

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

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 “Nesneler” 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

Yarış Benimle

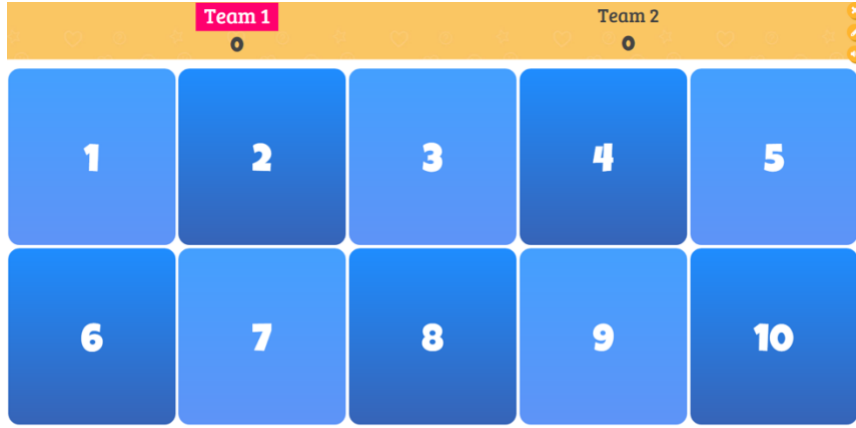
Süre: 45 dk

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 39).



Resim 39. Yarış Benimle Oyun Ara Yüzü (<https://www.baamboozle.com/classic/556028>)

Uygulama: Eğitimci öğrencileri iki gruba ayırarak, hep birlikte bir oyun oynayacaklarını belirtir. Hafta 9 sunumunda “Yarış Benimle” başlıklı slaytı kullanarak oyun talimatlarını verir. Gruplara ayrılma işlemi için eğitimci öğrenci isimlerinin alfabetik sıralamasından yararlanır. 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 sohbetten 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 yanıtı 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 9 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 İçeriği: Yapıcı ve Yıkıcı Fonksiyonlar

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 ayarlamak 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 SınıfAdı
{
    SınıfAdi (parametre listesi){
        yapıcı fonksiyon gövdesi
    }
    ~SınıfAdi (){
        yıkıcı fonksiyon gövdesi
    }
};
```

Yapıcı ve yıkıcı fonksiyonların tanımlamaları daha sonra yapılacaksa aşağıdaki sözdizimi kullanılır.

```
class SınıfAdı
{
    SınıfAdi (parametre listesi);
    ~SınıfAdi ();
};

SınıfAdi::SınıfAdi (parametre listesi){
    yapıcı fonksiyon gövdesi
}

SınıfAdi::~SınıfAdi (){
    yıkıcı fonksiyon gövdesi
}
```

Yapıcı metodun parametre alabileceğine dikkat ediniz. Parametre alabilen bir yapıcı fonksiyon oluştursanız, 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.

Sabit Fonksiyonlar

Veri üyelerinin değerinin sabit olduğu fonksiyonlar oluşturmanız mümkündür. Sabit fonksiyonlarda veri üyelerinin değerlerinin değiştirilmesi imkânsızdır. Bunu yapmaya çalışmak sözdizimi hatasına neden olur. Sabit bir fonksiyon bildirmek için, const anahtar sözcüğünü parametre listesinden sonra aşağıdaki sözdiziminde gördüğünüz gibi ekleriz:

```
dönüş_tipi fonksiyon_adi(parametre_listesi) const;
```

Sınıf tanımlamasının dışında sabit bir fonksiyon oluşturmak isterseniz, const anahtar sözcüğünün hem fonksiyon prototipinde hem de tanımına yerleştirilmesi gerekir.

Kod Satırlarını Tamamla

Süre: 50 dk.

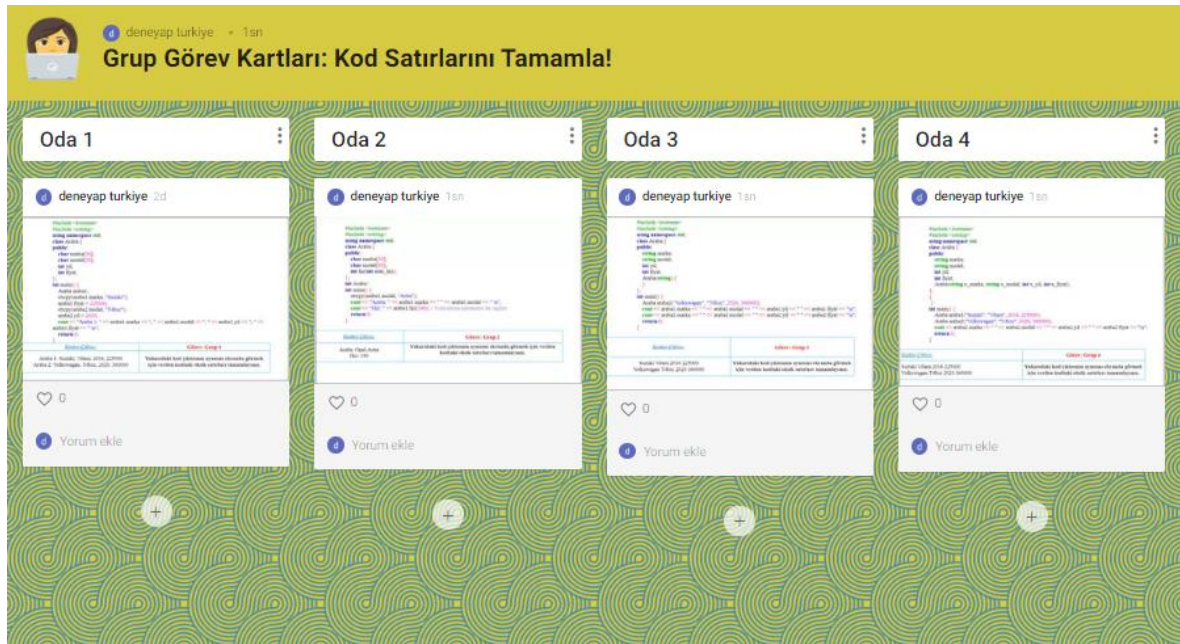
Kazanımlar: K2. Yapıcı fonksiyonları kullanarak program geliştirir.

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 40'da gösterilmiştir.



Resim 40. Grup Görev Kartları Örnek Dijital Tartışma Panosu

Uygulama: Öğitmen Hafta 9 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 yanıt karşılaştırmalarını ister. Bu sırada 2 dk.lık bir sözsüz müzik açar. Müzik sonunda öğretmen grup 1 üyelerine, kendi kodlarında hatalı olan yer var mı? varsa nasıl

düzeltilbileceğini 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() {
    // Ilk 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";

    return 0;
}
```

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 max_hiz);
};

int Araba::hiz(int max_hiz) {
    return max_hiz;
}

int main() {
    Araba arabal; // Araba nesnesini olusturun
    strcpy(arabal.marka, "Opel");
    strcpy(arabal.model, "Astra");
    cout << "Araba: " << arabal.marka << " " << arabal.model << " \n";
    cout << "Hiz: " << arabal.hiz(190); // Metodu parametre ile cagirin
    return 0;
}
```

Kodun Çıktısı:

Araba: Opel Astra

Hiz: 190

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 sinif ici 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 metodu farklı değerlerle çağırarak Araba nesneleri oluşturma
    Araba arabal("Suzuki", "Vitara", 2016, 225000);
    Araba araba2("Volkswagen", "T-Roc", 2020, 360000);

    // Değerleri yazdırma
    cout << arabal.marka << " " << arabal.model << " " << arabal.yil << " " <<
arabal.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

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 metodu farklı değerlerle çağırarak Araba nesneleri oluşturma
    Araba arabal("Suzuki", "Vitara", 2016, 225000);
    Araba araba2("Volkswagen", "T-Roc", 2020, 360000);
    // Değerleri yazdırma
    cout << arabal.marka << " " << arabal.model << " " << arabal.yil << " " <<
arabal.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

```

Sınıfları Dosyala

Süre: 40 dk.

Kazanımlar: K4. Kod içinde sınıf dosyaları ile proje geliştirir.

Materyaller: Dijital Tartışma Panosu: Bireysel Çalışma

Hazırlık: Öğitmen derse girmeden önce dört farklı grup çalışması için duvar temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletecektir.

Uygulama: Öğitmen sunumda "Sınıfları Dosyala" başlıklı slayt üzerinden "Cep telefonu sınıfını, sınıf dosyaları kullanarak tanımlama" ile ilgili örnek kod dosyasını gösterir. Örnek kod üzerinde iki farklı sütunda aynı problemi çözen iki farklı kod bulunmaktadır. Öğitmen öğrencilere bu iki sütunu incelemelerini ve onlara farklı gelen ya da daha önce görmedikleri kod satırlarının hangileri olduğunu sorarak sohbetten kendisine yazmalarını ister. Burada `#include "Cep telefonu.h"` kod satırına odaklanmaları sağlanır. Bu kod satırının ne işe yaradığını tahmin etmeleri istenir. Bu şekilde öğrencilerde soru işareti bırakılarak, sınıf dosyalarına neden ihtiyacımız olduğunu onlara açıklanır. Daha sonra sunumda devam edilerek öğrencilere kod örneğinin içeriğinin farklı bir görev için revize etmelerini ister. Öğrencilere aşağıdaki talimat verilir ve dijital tartışma panosunun linki paylaşılan notlar kısmından iletilir:

Şimdi bir daire sınıfı tasarlayıp oluşturacağınız iki adet dairenin çevresini ve alanını hesaplayan kod satırlarını yazınız. Bunun için 10 dk. süremiz var. Herkes bireysel olarak çalışacak. Süre sonunda kod satırlarının ekran görüntüsünü dijital panoya gönderiniz.

Etkinlik sonunda öğretmen padleti ekran paylaşımı yapar. Doğru kodları padlete yükler ve 3 dk.'lık sözsüz bir müzik başlatır. Müzik sırasında öğretmen öğrencilerden diğerlerinin kod satırlarını inceleyip doğru kod satırlarına en çok yaklaşan gönderiyi oylamalarını ister. Müzik sonunda en çok oy alan kod satırı üzerinden öğretmen sınıf dosyalarının nasıl oluşturulduğunu, neden gerekli olduğunu tekrar özetler ve öğrencilere bu şekilde geri bildirimde bulunur.

Eğitime Öneriler: Sürenin belirlenmesi için 10 dk'lık odaklanma müziği başlatılabilir ya da ekranda görünecek bir kronometre açılabilir. Süre sonunda etkinliğin sonlandırılması için bir çan kullanılabilir. Öğitmen konuyu açıklarken aşağıdaki içerikten faydalanabilir.

Sınıf Dosyaları Oluşturma

Hazırladığınız programlar çok fazla sayıda ve uzun kod satırlarına sahip sınıflar ve fonksiyonlar içerebilir. Böyle durumlarda kodların tek bir dosya üzerinde yönetimi çok zor olur ve belli bir aşamadan sonra yönetilemez hâle gelir. Ayrıca, hazırladığınız sınıflarınızı başka programlarda yeniden kullanmak isteyebilirsiniz. Bu durumda, bunları her bir proje dosyasına tekrar tekrar kopyalayıp eklemeniz gerekir. Bu süreç de oldukça zor olacaktır.

Oluşturmuş olduğunuz sınıfları ayrı bir dosyada tutmanız çok faydalı olacaktır. Bunun için hazırlamış olduğunuz sınıf yapısını `SinifAdi.h` isimli ve uzantılı bir dosya içerisine kaydedebilirsiniz. Normalde C++ dosyaları için `.cpp` uzantısı kullanılmaktadır, fakat `.h` uzantılı

dosyalar da C++ için geçerli bir dosya uzantısıdır. Fakat biz bu uzantılara sahip dosyalarda sınıflarımızı ve varsa kütüphanelerimizi tanımlayabiliriz. Sınıf tanımlaması, kullanılan kod dosyasından ayrı bir dosyaya kaydedildiği için, #include ifadesi ile programa dahil edilmelidir. #include yönergesinin sözdizimi şu şekildedir:

```
#include "dosya_adi"
```

Destekleyici Kod Bilgisi: Cep telefonu sınıfını sınıf dosyaları kullanarak tanımlama.

ornek.cpp	Ceptelevonu.h
<pre>#include <iostream> #include "Ceptelevonu.h" using namespace std; int main() { Ceptelevonu urun1(4500); Ceptelevonu urun2 = Ceptelevonu(3750); cout << "Urun 1 baslangic fiyatı: " << urun1.fiyat << endl; cout << "Urun 2 baslangic fiyatı: " << urun2.fiyat << endl; return 0; }</pre>	<pre>class Ceptelevonu { public: char marka[30]; char model[30]; int fiyat; bool aramaDurum; Ceptelevonu(int x_fiyat){ fiyat = x_fiyat; } ~Ceptelevonu(){ cout << "Nesne yok edildi." << endl; } void arama(); }; void Ceptelevonu::arama() { aramaDurum = true; }</pre>

Kodun Çıktısı:

Urun 1 baslangic fiyatı: 4500

Urun 2 baslangic fiyatı: 3750

Görev kodu ve yanıtı: Daire sınıfı kullanarak oluşturacağınız iki adet dairenin çevresini ve alanını hesaplayınız.

Alancevre.cpp	Daire.h
<pre> #include <iostream> #include "Daire.h" using namespace std; int main () { Daire daire1, daire2; daire1.deger_atama (4.2,3.14); daire2.deger_atama (2.1,3.14); cout << "Daire 1 Cevresi: " << daire1.cevre() << "\n"; cout << "Daire 1 Alani: " << daire1.alan ()<< "\n\n"; cout << "Daire 2 Cevresi: " << daire2.cevre() << "\n"; cout << "Daire 2 Alani: " << daire2.alan() << "\n"; return 0; } </pre>	<pre> class Daire { float yaricap; float pi_sayisi; public: void deger_atama(float ,float); float cevre() { return 2 * pi_sayisi * yaricap; } float alan() { return pi_sayisi * yaricap * yaricap; } }; void Daire::deger_atama (float x_yaricap, float x_pi_sayisi) { yaricap = x_yaricap; pi_sayisi = x_pi_sayisi; } </pre>

Kodun Çıktısı:

Daire 1 Cevresi: 26.376

Daire 1 Alani: 55.3896

Daire 2 Cevresi: 13.188

Daire 2 Alani: 13.8474

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 "Nesne Tabanlı Programlamanın Prensipleri" 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 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

Tasarlayıcı: Bir basketbol takımının koçu, oyuncularının ad-soyad, forma numarası ve attığı basket bilgilerini tutmak için bir program hazırlamak ister. Koç bunun için örnek bir program yazar. Programda örnek olarak Basketbolcu sınıf yapısı ve bu sınıfa ait 2 basketbolcu nesnesi bulunmaktadır. Koçun yazdığı örnek kodu tasarlayınız.

```
#include <iostream>
#include <string>
using namespace std;
class Basketbolcu
{
public:
    string ad_soyad;
    int forma_no;
    int basket_sayisi;
    Basketbolcu(string x_ad_soyad, int x_forma_no, int x_basket_sayisi) {
        ad_soyad = x_ad_soyad;
        forma_no = x_forma_no;
        basket_sayisi = x_basket_sayisi;
    }
}
```

```

};

int main()
{
    Basketbolcu b1("Mirsad Turkcan", 5 , 21);
    Basketbolcu b2("Semih Erden", 9, 15);

    cout << b1.ad_ soyad << " " << b1.forma_no << " " << b1.basket_sayisi << "\n";
    cout << b2.ad_ soyad << " " << b2.forma_no << " " << b2.basket_sayisi << "\n";

    return 0;
}

```

Kodun Çıktısı:

```

Mirsad Turkcan 5 21
Semih Erden 9 15

```

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 9. Ders Materyalleri

EK 1. Yarışma Kartları

Yapıcı fonksiyon (Constructors), sınıftan bir nesne oluşturulduğu anda otomatik çalışır.	Yapıcı fonksiyon (Constructors), sınıfla aynı isimde olamaz.	Yapıcı fonksiyon (Constructors), int, void vb. herhangi bir dönüş tipi alamazken, yıkıcı fonksiyonlar olabilir.	Gruba 10 puan kazandırdın
Rakibin puanından kendi grubuna 10 puan ekle	Gruba 10 puan kaybettirdin	Yapıcı fonksiyon (Constructors), parametre alabilir.	Yıkıcı fonksiyon (Destructors), parametre alabilir.
Yıkıcı fonksiyon (Destructors), nesne kullanımının bittiği zaman temizle görevi için son olarak çalışır.	Yıkıcı fonksiyon (Destructors), her zaman sınıfla aynı isimdedir.	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 arabal;
    strcpy(arabal.marka, "Suzuki");
    arabal.fiyat = 225000;
    strcpy(araba2.model, "T-Roc");
    araba2.yil = 2020;
    cout << "Araba 1: " << arabal.marka << ", " << arabal.model << ", " << arabal.yil
    << ", " << arabal.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(190); // Fonksiyonu parametre ile
    cagirin
    return 0;
}

```

Kodun Çıktısı:

Araba: Opel Astra
Hiz: 190

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 9 Süreli Ödev

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

Hafta 10. Nesne Yönelimli Programlamanın Prensipleri

Kazanımlar

- K1. Veri soyutlama, kapsülleme, kalıtım ve polimorfizm tekniklerini açıklar.
- K2. Kalıtım tekniğini farklı problemler içinde uygular.
- K3. Verilen programda kullanılan aşırı yükleme (overloading) tekniğini analiz eder.
- K4. Verilen programdaki geçersiz kılma/çiğneme (overriding) tekniğini yeniden tasarlar.

Haftanın Amacı ve Materyaller

Bu haftanın amacı, C++ nesne tabanlı programlamanın prensipleri olan veri soyutlama, veri kapsülleme, kalıtım, polimorfizm, aşırı yükleme ve geçersiz kılma kavramlarını öğrenmektir.

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

Önerilen Ders Akışı (45 + 60+ 45+ 50 dk.)

GİRİŞ (5 dk.)

GELİŞME

Adam Asmaca (40 dk.)

EK 1. Adam Asmaca Soruları

Ders Arası (10 dk.)

Kalıtımı Sürdür (30 dk.)

EK 2. Kalıtımı Sürdür Görevleri

Aşırı Yüklenenler (30 dk.)

EK 3. Aşırı Yüklenenler

Ders Arası (10 dk.)

Geçersiz Olanlar (35 dk.)

SONUÇ (10 dk.)

EK 4. 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

Süre: 5 dk.

Eğitmen derse girer girmez haftanın sunusunu ekranda öğrenciler ile paylaşır (Hafta 10). 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 dokuzuncu haftanın amacını paylaşarak öğrencileri *hedeften haberdar eder*.

B. Gelişme

Adam Asmaca

Süre: 40 dk

Kazanımlar: K1. Veri soyutlama, kapsülleme, kalıtım ve polimorfizm tekniklerini açıklar.

Materyaller: EK 1. Adam Asmaca Soruları

[Adam Asmaca Oyunu](#)

Hazırlık: Eğitmen derse öncesi EK 1’de adam asmaca sorularını kullanarak bir oyun tasarlar. Oyunun bir Web 2.0 aracı olan <https://learningapps.org> aracılığıyla oluşturulmuş versiyonuna materyallerden erişebilirsiniz (Resim 41).



Resim 41. Adam Asmaca oyunundan bir görüntü (<https://learningapps.org/watch?v=pots70zuj21>)

Uygulama: Eğitimci bu etkinlikte nesne yönelimli programlamanın temelini oluşturan kavramlarını inceleyecekleri bir oyun oynayacaklarını sunum üzerinden açıklar. Bunun için öğrenciler iki gruba ayrılır. Grup ayırma tekniği olarak eğitimci öğrenci isimlerinin alfabetik sıralamasını kullanır. Örneğin isminin baş harfi A-H arası olanlar birinci grup, diğerleri ise ikinci gruptadır. Daha sonra öğrencilere adam asmaca oyunu oynatılır. Eğitimci Web 2.0 araçlarını kullanarak ders öncesi hazırladığı oyunun linkini açar ve öğrencilere ekran paylaşımı yapar. Birinci gruptan, oyunda verilen ipuçlarıyla ilgili kelimeyi harf ya da rakam önererek tahmin etmeleri istenir. Eğitimci ipucunu sesli olarak okur ve birinci gruptan tahminlerine ilişkin bir harfi sohbet aracılığıyla iletmelerini ister. Eğitimci 'Hangi harfi almak istersiniz, sohbetten yazılan ilk harfi seçeceğim'; 'Sıradaki harfiniz ne olsun, sohbetten bana yazın' gibi ifadelerle öğrencileri yönlendirir. Bu aşamada tahmin etme sırası hangi grupta ise, o grup üyelerinin cevaplarını kabul etmeye dikkat edilmelidir. Web 2.0 aracıyla geliştirilen bu oyunda bilinmeyen her harf için çiçeğin bir yaprağı dökülür. Ancak tahminler doğru çıkarsa, birinci gruba 1 puan yazılır. Daha sonra ikinci ipucu için sıra ikinci gruba verilir ve aynı süreç tekrar edilir. Her sorunun ardından kelime ortaya çıktıkça, eğitimci ilgili kavram hakkında geri bildirimde bulunur. Veri Soyutlama, kapsülleme, kalıtım ve polimorfizm kavramları bu şekilde oyun temelli tahmin yolu ile özetlenir. Burada önemli olan nokta öğrencilerin kavramları doğru ya da yanlış tahmin etmesi değil, ipuçlarına odaklanmalarıdır. Böylece eğitimci konuyu özetlediğinde kavramı anlamlandırmaları kolaylaştırılmış olacaktır.

Eğitmene Öneriler: Eğitimci her kavramın ardından konuyu aşağıdaki gibi özetlemeye çalışır:

Konu İçeriği:

Nesne yönelimli programlamanın özü, kodda belirli özelliklere ve yöntemlere sahip bir nesne oluşturmaktır. Nesneleri kullanarak, kodun diğer bölümlerinin bu fonksiyon dışında bu verilere erişememesi için verileri ve üzerinde çalışan fonksiyonları birleştirmektir. C++ modülleri tasarlarlarken, 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 yönelimli programlamanın temelini oluşturan kavramlar şunlardır:



Resim 42. Nesne yönelimli programlamanın temelini oluşturan kavramlar

1. Veri Soyutlama

Veri soyutlama, dış dünyaya sadece gerekli bilgileri sağlama ve arka plan ayrıntılarını gizleme, yani ayrıntıları sunmadan programdaki gerekli bilgileri temsil etme anlamına gelir. Bir arabanın nasıl sürüleceğini ve nasıl yakıt ekleneceğini biliyor olabiliriz, fakat motorun nasıl çalıştığı hakkında bir fikrimiz var mı? Hayır ve ayrıca bilmemize gerek de yok çünkü bu detay soyutlanmış durumda. Araba örneğinden devam edersek, araba kullanan adam sadece gaza basmanın arabanın hızını artıracak olduğunu veya fren yapmanın arabayı durduracağını bilir, ancak gaza bastığında hızın gerçekte nasıl arttığını ya da arabanın iç mekanizması (gaz, fren vb. çalışma prensibini) bilmez. Başka bir örnek vermek gerekirse; mahalledeki pastaneden en sevdiğiniz kurabiyelerden satın alırken kurabiyenin yapım aşamasında kullanılan unun veya şekerin markasını ya da ne kadar kabartma tozu kullanıldığı gibi bilgileri düşünmüyorsunuz. Önemli olan kurabiyenin lezzetidir ki işin asıl amacı da budur yani beğenilmesidir.



Araba (veri soyutlamasız)



Araba (veri soyutlamalı)

Resim 43. Veri soyutlama araba örneği

Veri soyutlama, veri üyelerini gizleme ve sınıfın kullanıcısının verileri bozmaması için bir arayüzün arkasında bir sınıfın uygulanması sürecidir. Buradaki fikir, verilerin bir sınıf için uygulamanın içinde gizli olmasıdır. Veriler doğrudan değil, "public" fonksiyonlar aracılığıyla manipüle edilir.

Genel kural, tüm değişkenler ve üyeler için mümkün olan en küçük kapsamı kullanmaktır. Ayrıca, olabildiğince az global değişken kullanın. Bunu yaparken, yalnızca verilere erişimi olması gereken kod bu verileri değiştirebilir. Başka bir kodun özel bir nesne içindeki verileri kullanması veya değiştirmesi gerekiyorsa, o nesnenin üye fonksiyonları bu tür erişime izin veren yordamları sağlayabilir.

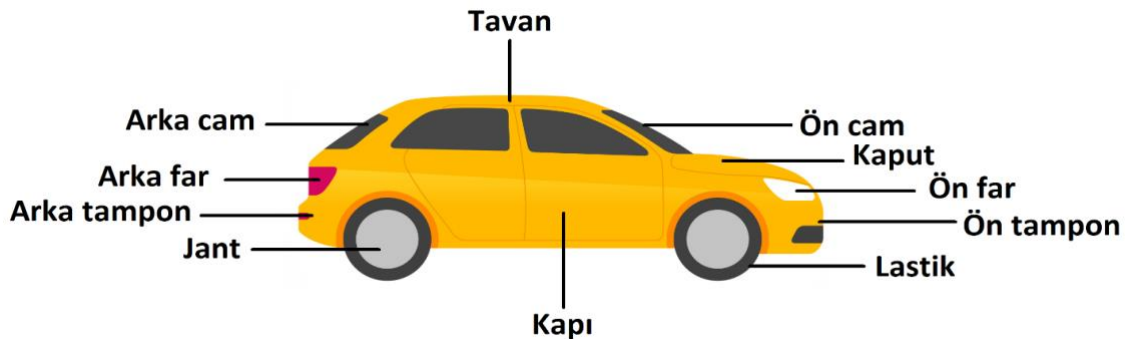
Örneğin, bir veritabanı sistemi, verilerin nasıl saklandığına, oluşturulduğuna ve muhafaza edildiğine ilişkin belirli ayrıntıları gizler. Benzer şekilde, C ++ sınıfları, bu yöntemler ve veriler hakkında dahili ayrıntılar vermeden dış dünyaya farklı yöntemler sağlar.

2. Kapsülleme

Verileri ve bu veriler üzerinde çalışan fonksiyonları aynı yere yerleştirmektir. Nesne yönelimli programlama, size verileri ve ilgili fonksiyonları aynı nesneye yerleştirmek için çerçeve sağlar. Bir sınıfın değişkenleri veya verileri başka herhangi bir sınıftan gizlenir ve yalnızca bildirildikleri kendi sınıfının herhangi bir üye fonksiyonu aracılığıyla erişilebilir. Kapsülleme fikri sınıfları birbirinden ayrı tutmak ve birbirleriyle sıkı sıkıya bağlı olmalarını önlemek için kullanılır. Oluşturduğunuz her sınıf, belirli bir şeyi veya kavramı temsil etmelidir. Birden çok sınıf, nesnelerin veya kavramların kombinasyonlarını temsil etmek için daha sonra bir araya gelir.

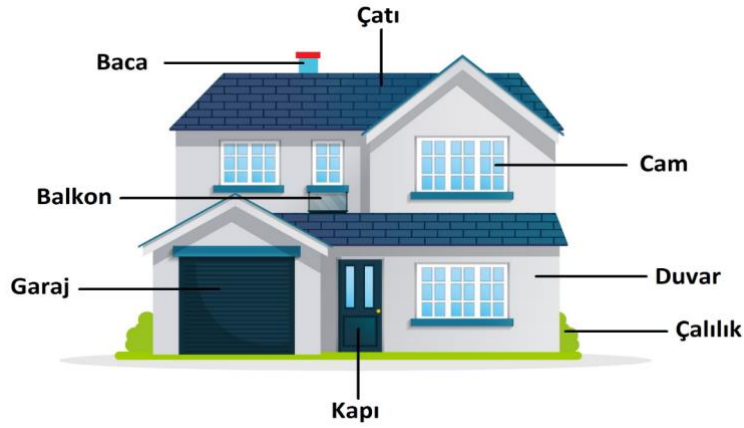
Araba örneğine tekrar bakalım. Bir arabanın hidrolik direksiyonu, dahili olarak birbirine bağlanmış çok sayıda bileşene sahip karmaşık bir sistemdir ve bu sistemdeki bileşenler aracı istenen yönde döndürmek için senkronize olarak çalışır. Motorun direksiyon simidine verdiği gücü bile kontrol eder. Ancak dış dünyaya yani bize ulaşan sadece bir arayüz mevcuttur ve karmaşıklığın geri kalanı gizlidir. Ayrıca, direksiyon ünitesi kendi içinde eksiksiz ve bağımsızdır. Başka bir mekanizmanın çalışmasını etkilemez.

Aşağıdaki gibi bir araba üretmek istediğinizi düşünün. Arabanın bitmiş hâlini oluşturmadan önce daha küçük alt parçaları oluşturmalsınız. Her küçük parça küçük bir fonksiyonu vardır, ancak hepsi bir araba üretmek için birleşir. Aşağıdaki görselde bir arabanın farklı parçaları gösterilmektedir. Her bölüm benzersiz bir amaca hizmet etmektedir.



Resim 44. Veri kapsülleme araba örneği

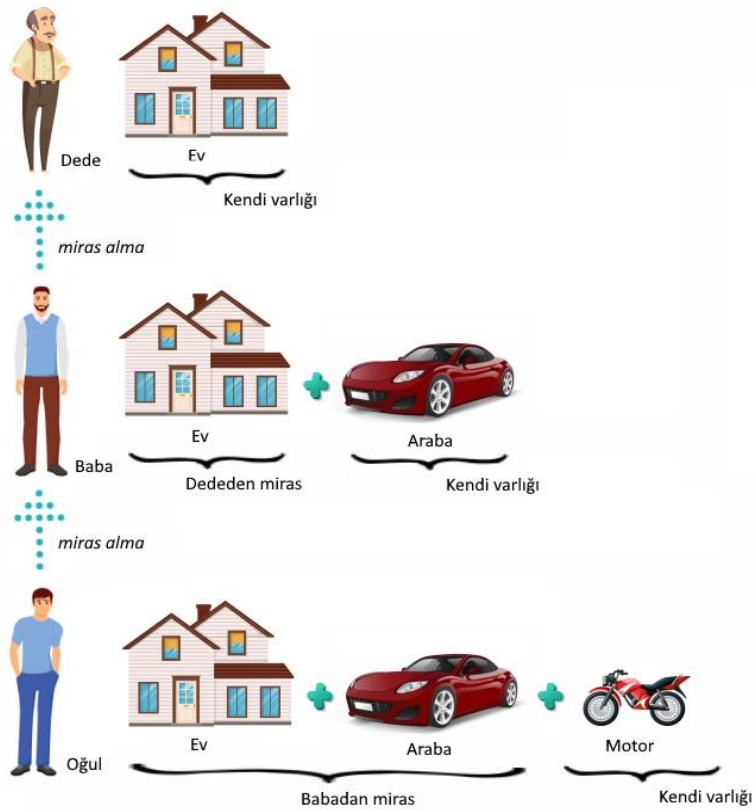
Aşağıdaki görselde ise bir evin farklı parçaları gösterilmektedir. Aynı şekilde burada da her bölüm benzersiz bir amaca hizmet etmektedir.



Resim 45. Veri kapsülleme ev örneği

3. Kalıtım

Kalıtım, bir nesnenin başka bir nesnenin belirli ya da tüm özelliklerini edinme mekanizmasıdır. Üst sınıfın özelliklerinin alt sınıfa aktarılmasını sağlayarak türetilmiş sınıflar oluşturur. Kodun yeniden kullanılabilirliğini sağladığı ve kod boyutunu azaltmaya yardımcı olduğu için nesne yönelimli programlamada çok önemli bir kavramdır.



Resim 46. Kalıtım örneği

Üyeleri temel bir sınıftan devralma yeteneği, temel sınıfta tanımlanan belirli ortak özellikleri paylaşan türetilmiş sınıfların oluşturulmasına izin verir.

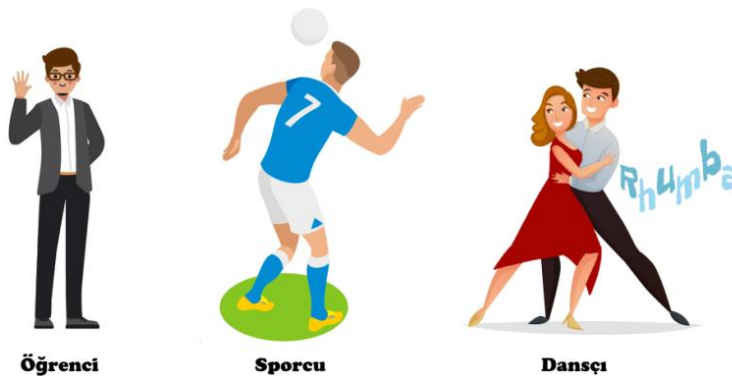
UYARI:

Sınıf örneklerini ve türetilmiş sınıfları birbiri ile karıştırmayın. Örnek, bir sınıfın kopyasıdır, türetilmiş sınıf ise türetildiği temel sınıfın özelliklerini miras alan yeni bir sınıftır.

Örneğin otomobiller, kamyonlar ve motosikletler dahil olmak üzere motorlu taşıtlarla ilgili bir uygulamanın gerektiğini varsayalım. Program, her türlü aracı temsil etmek için “taşıt” adlı bir sınıf kullanabilir. Arabalar, kamyonlar ve motosikletler taşıt türleri olduğundan, “taşıt” sınıfının alt sınıfları ile temsil edilecektir. “taşıt” sınıfı tüm taşıtlar için ortak olan plaka ve sahibi gibi genel bilgilere sahip olabilir. Bunlar tüm taşıtlarda ortak olan bilgilerdir. Ancak her bir taşıta özel bir değişken de eklenebilir. Araba sınıfı kaç kapılı bilgisini, kamyon sınıfı tekerlek sayısını ve motosiklet sınıfı da motosiklet sepeti olup olmadığını bilgisini alabilir.

4. Polimorfizm (Çok Biçimlilik)

Bir işleci veya fonksiyonu farklı şekillerde kullanma yeteneğine, başka bir deyişle işleçlere veya fonksiyonlara farklı anlam veya özellikler verme yeteneğine polimorfizm denir. Poli birçok anlamına gelirken morf ise form anlamındadır. Yani bir nesnenin aynı süreçte farklı şekillerde yanıt verebilme yeteneğidir. Tek bir fonksiyon veya kullanımdan farklı şekillerde fonksiyon gören bir operatöre polimorfizm denir. Daha iyi anlamak için aşağıdaki gerçek hayat örneğine bakalım. Bir kişi sınıfta kendini öğrenci, sahada bir sporcu ve dans kulübünde bir dansçı olarak temsil ediyor.



Resim 47. Polimorfizm kişi örneği

Aynı araba örneğine bakalım. Bir araçta vites iletim sistemi vardır. Dört ön vites ve bir arka vites vardır. Motor hızlandığında, hangi vites geçildiğine bağlı olarak, araca farklı miktarda güç ve hareket verilir. Eylem vites değiştirme olarak aynıdır, ancak vites tipine bağlı olarak eylem farklı davranır. İşte bu durumda polimorfizm mantığı devreye girmiş olur. Polimorfizm mantığını aşağıdaki örneği kullanılarak da açıklayabiliriz. Aşağıdaki verilen nesnelerin doğasında `ses_cikar()` yöntemi vardır, ancak her canlı bunu farklı şekilde gerçekleştirir.



Resim 48. Polimorfizm ses çıkarma örneği

Kalıtımı Sürdür

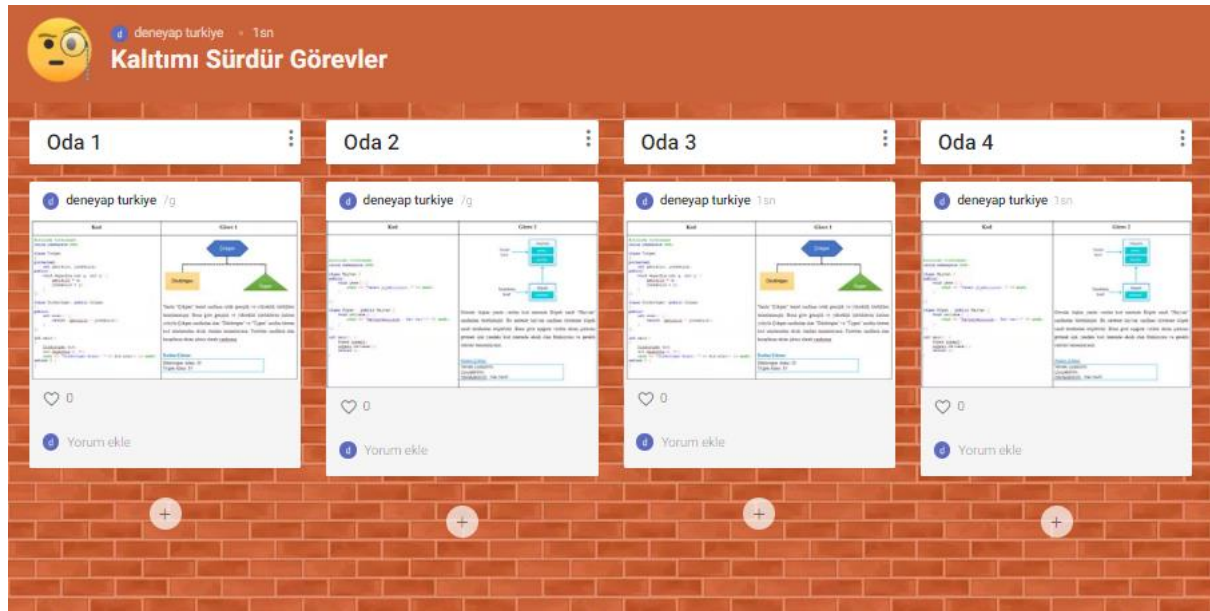
Süre: 30 dk

Kazanımlar: K2. Kalıtım tekniğini farklı problemler içinde uygular.

Materyaller: EK 2. Kalıtım Sürdür Görevleri

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 iletilecektir. EK 2'de yer alan görev 1 padlet ortamındaki ilk iki gruba, görev 2 ise diğer iki gruba yüklenecektir. Örnek bir tasarım Resim 49'da gösterilmiştir.



Resim 49. Kalıtımı Sürdür Örnek Padlet Tablosu

Uygulama: Eğitimci Hafta 10 sunumunda “Kalıtımı Sürdür” başlıklı slaytı kullanarak talimatları verir. Öğrenciler bu etkinlikte dört gruba ayrılarak çalışmaktadır. Dijital tartışma panosunda iki farklı gruba birinci görev, diğer iki gruba ise ikinci görev dağıtılır. Paylaşılan notlar kısmından padlet linki öğrencilere iletilir. 15 dk. süre ile öğrenciler rastgele çalışma odalarına gönderilir ve görevi tamamlamaya çalışırlar. Her grup üyesinin görev kodunu kendi bilgisayarında yazarak eksikleri tamamlamada grubuna katkı sağlaması istenir. Gruplardan süre bitmeden dijital panoya görev kodunun ekran görüntüsünü göndermeleri istenir. Burada ortak bir şekilde kodu yazarak tek bir ekran görüntüsü oluşturmaları beklenir. Bu şekilde gruplar performanslarını yansıtır. Süre sonunda eğitimci padleti ekran paylaşımı yapar ve ana odaya dönen gruplara yeni bir talimat verir.

Şimdi 5 dk’lık sözsüz bir müzik açacağım. Görev 1 üzerine çalışan grup üyeleri, diğer görev 1 üzerine çalışan grubun gönderilerini incelesin. Benzer şekilde görev 2’yi alanlar da birbirlerinin gönderilerini inceleyecekler. Arkadaşlarınızın yaptıklarında hata varsa, lütfen yorum olarak altta nasıl düzeltileceğini yazın. Müzik bitince yorum yazmayı bırakalım.

Bu şekilde gruplar birbirlerinin yaptıklarını düzenler ve ekran değerlendirmesi gerçekleştirilir. Öğrenciler yorumlarını yazdıktan sonra, her iki görev eğitimci geri bildirimleri eşliğinde dijital pano üzerinden tartışılır ve görevler özetlenir.

Eğitime Öneriler: Eğitimci ekran değerlendirmeleri sırasında öğrenciler tarafından hiç yorum bırakılmazsa, gruplara özetleme aşamasında söz hakkı vererek öğrencileri katılıma teşvik edebilir. Verilen görevlerden görev 1’i bilgisayarda kodlayanlara, görev 2 ödev olarak verilebilir. Benzer şekilde görev 2’yi bilgisayarında kodlayanlar ise, görev 1’i ödev olarak ders sonrası tamamlayabilirler. Eğitimci aşağıda verilen görevlerin doğru yanıtlarını etkinlik sonunda konuyu özetlerken dijital tartışma panosuna gönderebilir.

Görev 1: Çokgenler üzerinde kalıtım örneğindeki eksik kodun tamamlanmış hali aşağıdadır.

```
#include <iostream>

using namespace std;

class Cokgen
{
protected:
    int genislik, yukseklik;
public:
    void degerAta(int g, int y) {
        genislik = g;
        yukseklik = y;
    }
}
```

```

};

class Dikdortgen: public Cokgen
{
public:
    int alan() {
        return (genislik * yukseklik);
    }
};

class Ucgen: public Cokgen
{
public:
    int alan() {
        return ((genislik * yukseklik)/2);
    }
};

int main()
{
    Dikdortgen dik;
    Ucgen ucg;
    dik.degerAta(4, 5);
    ucg.degerAta(4, 5);
    cout << "Dikdorgen Alani: " << dik.alan() << endl;
    cout << "Ucgen Alani: " << ucg.alan() << endl;
    return 0 ;
}

```

Kodun Çıktısı:

```

Dikdorgen Alani: 20
Ucgen Alani: 10

```

Görev 2: Hayvan sınıfı üzerinde uygulanan kalıtım örneğindeki eksik kodun tamamlanmış hali aşağıdadır.

```
#include <iostream>

using namespace std;

class Hayvan {
public:
    void yeme() {
        cout << "Yemek yiyebilirim.." << endl;
    }

    void uyuma() {
        cout << "Uyuyabilirim.." << endl;
    }
};

class Kopek: public Hayvan {
public:
    void havlama() {
        cout << "Havlayabilirim.. Hav hav!!" << endl;
    }
};

int main() {
    Kopek kopek1;
    kopek1.yeme();
    kopek1.uyuma();
    kopek1.havlama();
    return 0;
}
```

Kodun Çıktısı:

Yemek yiyebilirim..

Uyuyabilirim..

Havlayabilirim.. Hav hav!!

Aşırı Yüklenenler

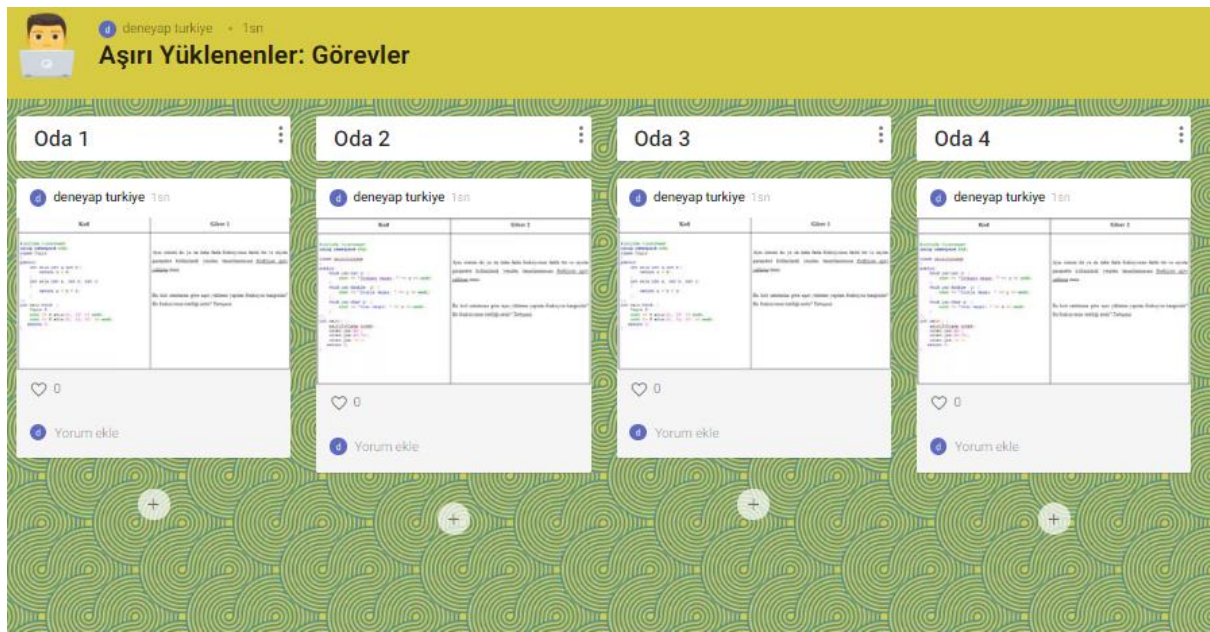
Süre: 30 dk

Kazanımlar: K3. Verilen programda kullanılan aşırı yükleme (overloading) tekniğini analiz eder.

Materyal: EK 3. Aşırı Yüklenenler

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. EK 3'te yer alan görev 1 padlet ortamındaki ilk iki gruba, görev 2 ise diğer iki gruba yüklenecektir. Örnek bir tasarım Resim 50'de gösterilmiştir.



Resim 50. Aşırı Yüklenenler Örnek Padlet Tablosu

Uygulama: Öğitmen uygulama öncesi Hafta 10 sunumunda “Polimorfizm” başlıklı slaytı polimorfizm kavramı hakkında hatırlatma yapar. İki tür polimorfizm olduğunu aşağıdaki gibi öğrencilere açıklar:

Polimorfizm, nesnenin farklı koşullarda farklı davranmasına izin veren poliformizm kavramını hatırlayınız. Bu ve bir sonraki uygulamada polimorfizmi daha yakından tanıyacağız. C++'da iki tür polimorfizm vardır:

1) Fonksiyon aşırı yükleme (Derleme zamanı polimorfizmi)

2) Fonksiyon geçersiz kılma (Çalışma Zamanı polimorfizmi)

Yukarıdaki gibi açıklama yapıldıktan sonra öğretmen sınıfı dörde ayırır. Paylaşılan notlar kısmından etkinliğin padlet linkini öğrencilere gönderir. “Kalıtımı Sürdür” etkinliğine benzer şekilde ilk iki gruba görev 1, diğer iki gruba görev 2 verilir ve kendi içlerinde görevleri tartışmaları beklenir. Öğitmen bunun için öğrencileri dört gruba rastgele dağıtarak 10 dk. süre verir. Öğitmen

süre sonunda ekran paylaşımında dijital tartışma panosunu açmıştır. İki görev arasında dikkat çekilmesi gereken bir nokta, birinci görevde farklı parametre sayısı kullanarak aşırı yükleme yapılırken, ikinci görevde farklı parametre türü kullanılarak aşırı yükleme yapılmaktadır. Ana odaya dönen öğrencilere iki görev arasındaki bu farkı görebilmeleri için aşağıdaki talimatı verir:

Şimdi 10 dk'lık sözsüz bir müzik açacağım. Görev 1 üzerine tartışan grup üyeleri, görev 2'yi, görev 2'yi tartışanlar ise görev 1'i incelesin. İncelediğiniz kodu bireysel olarak bilgisayarınızda yazıp, kod çıktılarını karşılaştırın. Sizce iki kod arasındaki farklılık nedir? Lütfen grup çalışmasında atandığınız oda numarası altında yorumunuzu bırakın. Müzik bitince yorum yazmayı bitirin.

Bu şekilde akran öğrenimini destekleyen 5 dk. daha geçer. Müzik sonunda yorumlar üzerinden eğitmen geri bildirimlerde bulunarak, görev 1'de parametre sayısına göre, görev 2'de ise parametre türüne göre fonksiyon aşırı yükleme tekniği uygulandığına dair aradaki farkı açıklar. Son olarak eğitmen dijital panoyu kapatır ve sunum üzerinden konuyu tekrar özetleyerek kalıcı öğrenme için pekiştirme yapar.

Eğitmene Öneriler: Eğitmen akran değerlendirmeleri sırasında öğrenciler tarafından hiç yorum bırakılmazsa, gruplara özetleme aşamasında söz hakkı vererek öğrencileri katılıma teşvik edebilir. Eğitmen aşağıda verilen görevlerin doğru yanıtlarını etkinlik sonunda konuyu özetlerken dijital tartışma panosuna gönderebilir.

Aşırı Yükleme (Overloading)

Aynı isimde iki ya da daha fazla fonksiyonun farklı tür ve sayıda parametre kullanılarak yeniden tanımlanmasına *fonksiyon aşırı yükleme* denir. Fonksiyon aşırı yüklemesinin avantajı, aynı fonksiyon için farklı isimler kullanmayı gerektirmediği için programın okunabilirliğinin artmasıdır.

Görev 1: Farklı parametre sayısı kullanarak fonksiyon aşırı yükleme.

```
#include <iostream>
using namespace std;
class Topla
{
public:
    int ekle(int a,int b){
        return a + b;
    }
    int ekle(int a, int b, int c)
    {
        return a + b + c;
    }
};
int main(void) {
    Topla f;
    cout << f.ekle(21, 13) << endl;
    cout << f.ekle(21, 13, 30) << endl;
    return 0;
}
```

Kodun Çıktısı:

34

64

Görev 2: Farklı parametre türü kullanarak fonksiyon aşırı yükleme.

```
#include <iostream>
using namespace std;

class asiriYukleme
{
public:
    void yaz(int x) {
        cout << "Integer deger: " << x << endl;
    }
    void yaz(double y) {
        cout << "Double deger: " << y << endl;
    }
    void yaz(char z) {
        cout << "Char deger: " << z << endl;
    }
};

int main() {
    asiriYukleme ornek;
    ornek.yaz(45);
    ornek.yaz(34.78);
    ornek.yaz('A');
    return 0;
}
```

Kodun Çıktısı:

Integer deger: 45

Double deger: 34.78

Char deger: A

Geçersiz Olanlar

Süre: 35 dk**Kazanımlar:** K4. Verilen programdaki geçersiz kılma/çiğneme (overriding) tekniğini yeniden tasarlar.**Materyal:** Hafta 10 Sunum

Dijital Tartışma Panosu: Bireysel Çalışma

Eğitmen derse girmeden önce bireysel çalışma için duvar temalı bir dijital panoyu padlet ortamında oluşturur. Linki ders sırasında öğrencilere paylaşılan notlar kısmından iletacaktır.

Uygulama: Eğitimci uygulama öncesi sunum üzerinde “Geçersiz Olanlar” başlıklı slaytı kullanarak aşağıdaki gibi öğrencilere açıklama yapar.

Bir önceki uygulamada Polimorfizmin türleri olduğunu gördük. Bunlardan ilki olan fonksiyon aşırı yükleme kavramını yakından inceledik. Şimdi ikinci poliformizm türü olan fonksiyon geçersiz kılma türünü bu uygulama ile inceleyeceğiz.

Yukarıdaki açıklamayı yaptıktan sonra, eğitimci sunumda ilerleyerek, fonksiyon geçersiz kılma üzerine bir kod örneği ve bu örneğe ilişkin geçersiz kılma tekniğini anlatan örneği gösterir. Eğitimci daha sonra öğrencilere örnekteki kod satırlarını kendi geçersiz kılma örnekleriyle revize etmelerini ister. Bunun için aşağıdaki talimat verilir:

Şimdi herkes bireysel olarak kendi bilgisayarında çalışsın. Örnekteki koda benzer şekilde “brans” adında bir temel sınıf oluşturun. Bu sınıftan türetilen başka bir sınıf altında (Ör: MatematikOgretmeni ya da SınıfOgretmeni gibi.) geçersiz kılınacak bir fonksiyon oluşturun. Böylece örnekteki kod satırlarını kendi geçersiz kılma tekniğini uygulamak için revize edin. Yazdığınız kodun ekran görüntüsünü isminizle birlikte dijital panoya gönderin.

Talimatı verdikten sonra eğitimci 10 dk. süre vererek öğrencileri serbest bırakır. Bu şekilde öğrenciler geçersiz kılma tekniğini uyguladıkları yeni bir örnek tasarlamış olacaktır. Daha sonra eğitimci 3 dk.’lık sözsüz bir müzik başlatacağını, müzik sırasında tüm gönderileri inceleyip doğru yazan kod satırlarına beğeni bırakmalarını ister. Müzik sonunda eğitimci öğrenci gönderileri için dijital panoyu ekran paylaşımı yapar. Eğitimci en çok beğeni alan kod satırını dijital pano üzerinden açarak, öğrencilere geri bildirimlerde bulunur. Böylece öğrenci ürünleri üzerinden geçersiz kılma tekniği tekrar özetlenerek pekiştirilir.

Eğitime Öneriler: Eğitimci dijital panoya tüm öğrencilerin gönderilerini atmaları için teşvikte bulunmalıdır. Örneğin aramızda gönderi yapmayan arkadaşlar var, 1 dk. daha süre ister misiniz? diyerek geri bildirimlerde bulunur. Gerekirse gönderisi eksik olan öğrenci ismi ile seslenilir ve biraz daha beklememizi ister misin? gibi ifadelerle gönderisini tamamlaması için öğrenci cesaretlendirilir. Konu içeriği ve verilen talimata göre öğrenciler tarafından oluşturulması beklenen kod satırları aşağıda verilmektedir:

Geçersiz Kılma/Çiğneme (Overriding)

Temel sınıftaki bir fonksiyonu geçersiz kılmak için o fonksiyonu türetilmiş bir sınıfta tekrar aynı isim, parametre ve dönüş türüne sahip olacak şekilde kullanırız. Bu şekilde, örnekte Temel sınıfı altındaki yaz() fonksiyonunu etkili bir şekilde gizlenmiş ya da geçersiz kılınmıştır. Bununla birlikte Temel sınıfı altındaki yaz() fonksiyonunu çağırabilmek için :: kapsam çözümleme operatörü kullanılabilir. Temel sınıf fonksiyonlarını geçersiz kılma tekniği, aşırı yüklenmiş fonksiyonları istemeden gizlemekten kaçınmak için dikkatle kullanılmalıdır. Türetilmiş bir sınıftaki tek bir geçersiz kılma fonksiyonu ile temel sınıfta aynı isme sahip tüm aşırı yüklenmiş fonksiyonlar gizlenecektir.

```

#include <iostream>
using namespace std;
class Temel
{
public:
    void yaz() {
        cout << "Temel Fonksiyon" << endl;
    }
};
class Turetilmis: public Temel
{
public:
    void yaz() {
        cout << "Turetilmis Fonksiyon" << endl;
    }
};
int main()
{
    Turetilmis turet1, turet2;
    turet1.yaz();
    turet2.Temel::yaz();
    return 0;
}

```

Kodun Çıktısı:

Turetilmis Fonksiyon

Temel Fonksiyon

Ogretmen temel sınıfından türetilen Matematik_Ogretmeni sınıfı altında yer alan brans() fonksiyonu ile oluşturulmuş geçersiz kılma tekniği:

```

#include <iostream>
using namespace std;
class Ogretmen
{
public:
    void brans(){
        cout << "Sinif Ogretmeni" << endl;
    }
};
class Ogretmen: public MatematikOgretmeni
{
public:
    void brans(){
        cout << "Matematik Ogretmeni" << endl;
    }
};
int main(void)
{
    MatematikOgretmeni o1, o2;
    o1.brans();
    o2.Ogretmen::brans();
    return 0;
}

```

Kodun Çıktısı:

Matematik Öğretmeni

Sınıf Öğretmeni

C. Sonuç

Süre: 10 dk.

Materyal: EK 4. Hafta 10 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 "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 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.

Sürekli Ödev Yanıtlar

Tasarlayıcı: Bir meyve sınıfı oluşturup içerisinde ekrana “Ben meyve sinifiyim” diyen bir meyve sınıfından türetilen Elma ve Muz sınıflarını oluşturunuz. Türetilmiş bu sınıfların içerisinde meyvelerin renklerini ekrana yazdırıp, tüm sınıflar için birer nesne tanımlayınız.

```
#include <iostream>

using namespace std;

class Meyve {
public:
    Meyve() {
        cout<<"Ben meyveyim."<<endl;
    }
};

class Elma: public Meyve {
public:
    Elma() {
        cout<<"Ben kirmizi renkliyim!!"<<endl;
    }
};

class Muz: public Meyve {
public:
    Muz() {
        cout<<"Ben sari renkliyim!!"<<endl;
    }
};

int main() {
    Elma e;
    Muz m;
}
```

Ekran Çıktısı:

Ben meyve sinifiyim.
Ben kirmizi renkliyim!!
Ben meyve sinifiyim.
Ben sari renkliyim!!

Denetleyici: Verilen kod satırlarını inceleyerek aşırı yüklenen fonksiyonun hangisi olduğunu ve kodun ekran çıktısını tahmin ediniz.

```
#include <iostream>

using namespace std;

class Dikdortgen{
public:
    void alanYaz1(int x, int y){
        cout << x * y << endl;
    }

    void alanYaz2(int x){
        cout << x * x << endl;
    }

    void alanYaz1(int x, double y){
        cout << x * y << endl;
    }

    void alanYaz3(double x){
        cout << x * x << endl;
    }
};

int main(){
    Dikdortgen dikt;
    dikt.alanYaz1(3,5);
    dikt.alanYaz1(4,2.1);
    dikt.alanYaz2(8);
    dikt.alanYaz3(6.7);
    return 0;
}
```

CEVAP: Farklı parametre türünde aşırı yüklenen fonksiyon alanYaz1'dir. Kodun ekran çıktısı ise aşağıdaki gibidir:

```
15
8.4
64
44.89
```

Kodlayıcı: Hayvan sınıfından türetilen bir Kopek sınıfı oluşturunuz. Bu sınıflara ait iki farklı nesne tanımlayarak, yer() isimli fonksiyonu geçersiz kılan (overriding) kod satırlarını oluşturunuz.

```
#include <iostream>

using namespace std;

class Hayvan
{
public:
    void yer() {
        cout << "Yer..." << endl;
    }
};

class Kopek: public Hayvan
{
public:
    void yer() {
        cout << "Ekmek yer..." << endl;
    }
};

int main(void)
{
    Kopek k1, k2;

    k1.yer();

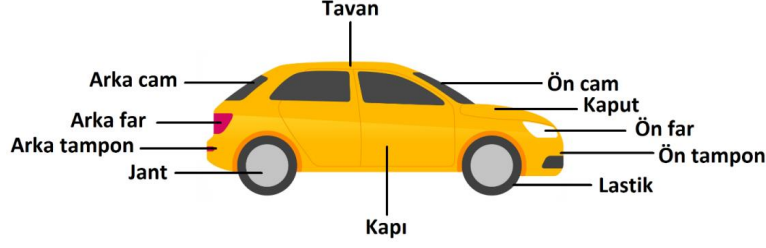
    k2.Hayvan::yer();

    return 0;
}
```

Hafta 10. Ders Materyalleri

EK 1. Adam Asmaca Oyunu

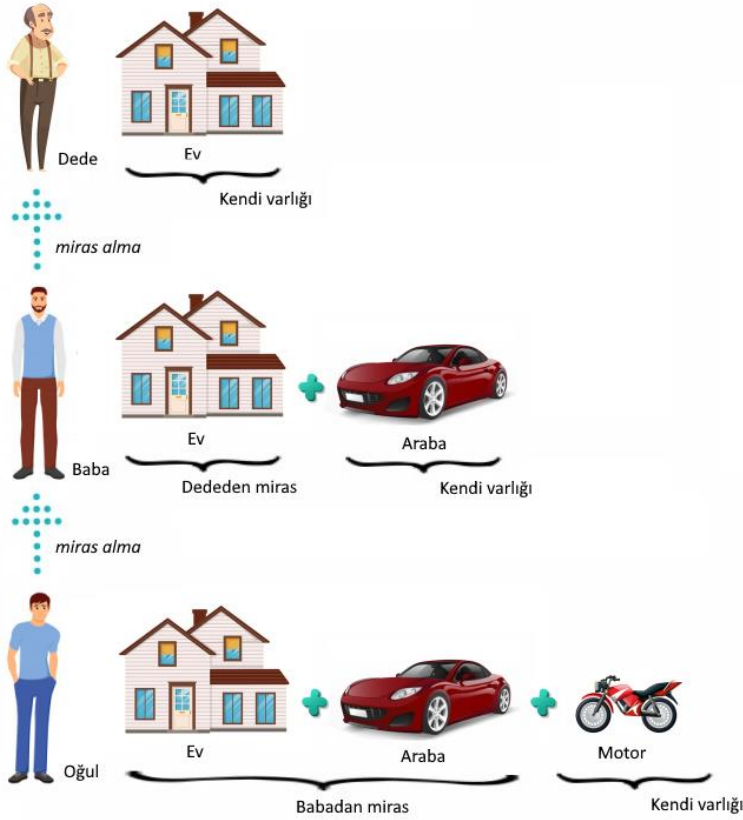
Kelime	
  Dış dünyaya sadece gerekli bilgileri sağlama ve arka plan ayrıntılarını gizleme, yani ayrıntıları sunmadan programdaki gerekli bilgileri temsil etme anlamına gelir. Buradaki fikir, verilerin bir sınıf için uygulamanın içinde gizli olmasıdır. Örneğin bir arabanın nasıl sürüleceğini ve nasıl yakıt ekleneceğini biliyor olabiliriz, fakat bu bilgiler için motorun nasıl çalıştığını bilmek zorunda değiliz.	



Sınıfları birbirinden ayrı tutmak ve birbirleriyle sıkı sıkıya bağlı olmalarını önlemek için kullanılır. Oluşturduğunuz her sınıf, belirli bir şeyi veya kavramı temsil etmelidir. Birden çok sınıf, nesnelerin veya kavramların kombinasyonlarını temsil etmek için daha sonra bir araya gelir.

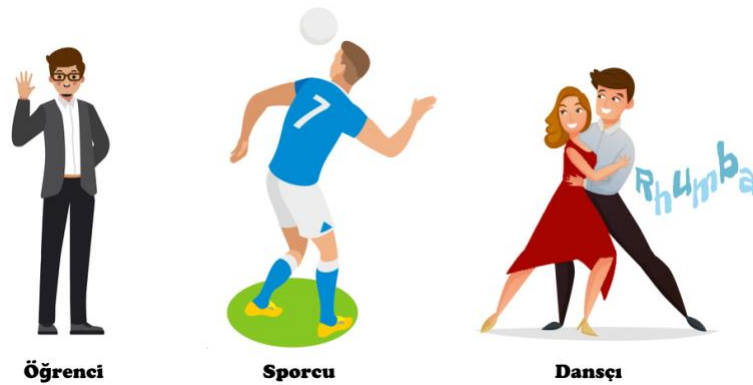
Örneğin, yukarıdaki gibi bir arabanın bitmiş hâlini oluşturmadan önce daha küçük alt parçaları oluşturmalısınız. Her küçük parça küçük bir işlevi vardır, ancak hepsi bir araba üretmek için birleşir. Aşağıdaki görselde bir arabanın farklı parçaları gösterilmektedir. Her bölüm benzersiz bir amaca hizmet etmektedir.

... ..



... ..

Bir nesnenin, başka bir nesnenin belirli ya da tüm özelliklerini edinme mekanizmasıdır. Başka bir ifadeyle veri üyelerini temel bir sınıftan devralma yeteneğidir. Üst sınıfın özelliklerinin alt sınıfa aktarılmasını sağlayarak türetilmiş sınıflar oluşturur. Kodun yeniden kullanılabilirliğini sağlar ve kod boyutunu azaltmaya yardımcıdır.



Bir nesnenin farklı koşullarda farklı davranmasına izin verme durumudur. Başka bir ifadeyle tek bir işlev veya kullanımdan farklı şekillerde işlev gören bir operatördür. Örneğin bir kişi sınıfta kendini öğrenci, sahada bir sporcu ve dans kulübünde bir dansçı olarak temsil edebilir ya da bir adam sınıfta öğretmen, evde baba, markette müşteri gibi davranır. Burada tek bir kişi duruma göre farklı davranmaktadır.

[illegible]

EK 2. Kalıtımı Sürdür Görevleri

Kod	Görev 1
<pre> #include <iostream> using namespace std; class Cokgen { protected: int genislik, yukseklik; public: void degerAta(int g, int y) { genislik = g; yukseklik = y; } }; class Dikdortgen: public Cokgen { public: int alan() { return (genislik * yukseklik); } }; int main() { Dikdortgen dik; dik.degerAta(4, 5); cout << "Dikdortgen Alani: " << dik.alan() << endl; return 0 ; } </pre>	<pre> classDiagram class Çokgen class Dikdortgen class Ucgen Çokgen < -- Dikdortgen Çokgen < -- Ucgen </pre> Yanda "Çokgen" temel sınıfının ortak genişlik ve yükseklik özellikleri tanımlanmıştır. Buna göre genişlik ve yükseklik özelliklerini kalıtım yoluyla Çokgen sınıfından alan "Dikdörtgen" ve "Üçgen" sınıfını türeten kod satırlarından eksik olanları tamamlayınız. Türetilen sınıfların alan hesaplarını ekran çıktısı olarak yazdırınız. Kodun Çıktısı: Dikdortgen Alani: 20 Ucgen Alani: 10

Kod	Görev 2
<pre> #include <iostream> using namespace std; class Hayvan { public: void yeme () { cout << "Yemek yiyebilirim.." << endl; } }; class Kopek: public Hayvan { void havlama () { cout << "Havlayabilirim.. Hav hav!!" << endl; } }; int main () { Kopek kopek1; kopek1.havlama (); return 0; } </pre>	<pre> classDiagram class Hayvan { yeme() uyuma() } class Kopek { havlama() } Hayvan < -- Kopek </pre> Görsele ilişkin yanda verilen kod üzerinde Köpek sınıfı "Hayvan" sınıfından türetilmiştir. Bu nedenle hayvan sınıfının üyelerine köpek sınıfı tarafından erişilebilir. Buna göre aşağıda verilen ekran çıktısını görmek için yandaki kod üzerinde eksik olan fonksiyonu ve gerekli satırları tamamlayınız. <u>Kodun Çıktısı:</u> <pre> Yemek yiyebilirim.. Uyuyabilirim.. Havlayabilirim.. Hav hav!! </pre>

EK 3. Aşırı Yüklenenler

Kod	Görev 1
<pre> #include <iostream> using namespace std; class Topla { public: int ekle(int a,int b){ return a + b; } int ekle(int a, int b, int c) { return a + b + c; } }; int main(void) { Topla f; cout << f.ekle(21, 13) << endl; cout << f.ekle(21, 13, 30) << endl; return 0; } </pre>	Aynı isimde iki ya da daha fazla fonksiyonun farklı tür ve sayıda parametre kullanılarak yeniden tanımlanmasına <u>fonksiyon aşırı yükleme</u> denir. Yukarıdaki tanıma göre yanda verilen kodu bilgisayarda yazarak ekran çıktısını bulunuz. Bu kod satırlarına göre aşırı yükleme yapılan fonksiyon hangisidir? Bu fonksiyonun özelliği nedir? Tartışınız.

Kod	Görev 2
<pre> #include <iostream> using namespace std; class asiriYukleme { public: void yaz(int x) { cout << "Integer deger: " << x << endl; } void yaz(double y) { cout << "Double deger: " << y << endl; } void yaz(char z) { cout << "Char deger: " << z << endl; } }; int main() { asiriYukleme ornek; ornek.yaz(45); ornek.yaz(34.78); ornek.yaz('A'); return 0; } </pre>	Aynı isimde iki ya da daha fazla fonksiyonun farklı tür ve sayıda parametre kullanılarak yeniden tanımlanmasına <u>fonksiyon aşırı yükleme</u> denir. Yukarıdaki tanıma göre yanda verilen kodu bilgisayarda yazarak ekran çıktısını bulunuz. Bu kod satırlarına göre aşırı yükleme yapılan fonksiyon hangisidir? Bu fonksiyonun özelliği nedir? Tartışınız.

EK 4. 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+ 60 + 40+ 60 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.)

Eksik Kodları Dolduruyorum (30 dk.)

EK 3. C++ Programında Fonksiyon Tanımlama Görevleri

Ders arası (5 dk.)

Kütüphanelerdeki Bazı Fonksiyonları Kullanarak Kodluyorum (30 dk.)

EK 4. Kütüphaneleri ve Fonksiyonları Kullanma: Görevler

Ders arası (5 dk.)

Neden Dosyalama İşlemleri Yaparız? (30 dk.)

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

Ders Arası (5 dk.)

C++ Dilinde Dosyalama İşlemleri Yapıyorum (30 dk.)

EK 9. Kodlar Arasında Farkı Bulma

Ders arası (10 dk.)

Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum (40 dk.)

EK 10. Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum

SONUÇ (5 dk.)

EK 11. Hafta 11 Süreli Ödev (60 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.

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

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: Eğitmen 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 kullanılabilecek 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 eğitmen sizce bu görevi yapacak bir fonksiyonun çıktısı ne olabilir şeklinde sorar. Verilen cevaplara göre geri bildirim verilerek bir sonraki fonksiyona geçilir.

Öğrenci gruplarının doldurdıkları görev kâğıtlarının aşağıdaki gibi olması beklenir. Cevaplar kırmızı renkte verilmiştir.

Tablo 26. 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

Tablo 26 (devamı). Karakter Fonksiyonları ve Örnek Kullanımları

Fonksiyon	İşlevi	Örnek Kullanım	Sonuç
strcpy (s1, s2)	s2 katarını s1 katarına kopyalar.	s1= "Merhaba" s2= "Dunya" strcpy (s1, s2)	Dunya
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= "Dunya" strcmp (s1, s2)	M harfi D harfinden sonra geldiği için 1 değerini döndürür.

Eksik Kodları Dolduruyorum

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 3. C++ Programa Dilinde Bulunan Kütüphaneleri ve Fonksiyonları Kullanma

Hazırlık: Hafta 11 Sunumunda "Eksik Kodları Dolduruyorum" adlı slaytlar sırası geldikçe açılır.

Uygulama: Sunuda verilen C++ programa dilinde bulunan kütüphaneleri ve fonksiyonları kullanımına yönelik iki görevin ekran çıktısını elde etmek için eksik olan yerlere hangi kodların geleceğini öğrencilerin bulması istenir. Her bir eksik kod için öğrencilerin 1dk. içerisinde chat kısmına yanıtları yazması beklenir. Her eksik kod için ayrı ayrı verilen cevaplara göre geri bildirim verilir.

Eğitmene Öneriler: Öğrencilere verilen görevlerin cevapları aşağıdaki gibidir. Öğrencinin verdiği yanıtların aşağıdaki gibi olması beklenir.

Görev 1: 'a' karakteri ile ilgili örnek kullanım ve ekran çıktısı aşağıdaki gibidir.

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
    char c='a';

    if(isalpha(c))
        cout << "Bu bir harftir."<<endl;
    else
        cout << "Bu bir harf degildir."<<endl;

    if(isdigit(c))
        cout << "Bu bir rakamdir."<<endl;
    else
        cout << "Bu bir rakam degildir."<<endl;

    if(islower(c))
        cout << "Bu bir kucuk harftir."<<endl;

    if(isupper(c))
        cout << "Bu bir buyuk harftir."<<endl;
}
```

Görev 2: Katarlar üzerinde farklı fonksiyonların kullanımı.

```

#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    char str1[] = "Bugun hava cok guzel.";
    char str2[] = "Piknige gidelim.";
    char str3[50];
    int uzunluk;

    cout << "str1 katari: " << str1 << endl;
    uzunluk = strlen(str1);
    cout << "str1 katari uzunlugu: " << uzunluk << endl << endl;

    cout << "str2 katari: " << str2 << endl;
    uzunluk = strlen(str2);
    cout << "str2 katari uzunlugu: " << uzunluk << endl << endl;

    strcpy(str3, str1);
    cout << "str3 katari: " << str3 << endl << endl;

    strcat(str1, str2);
    cout << "s1 katari: " << str1 << endl;

    uzunluk = strlen(str1);
    cout << "s1 katari uzunlugu: " << uzunluk << endl << endl;

    if(strcmp(str1, str3) == 0)
        cout << "Iki katar birbirine esittir" << endl;
    else
        cout << "Iki katar birbirine esit degildir." << endl;
}

```

Kütüphanelerdeki Bazı Fonksiyonları Kullanarak Kodluyorum

Süre: 20 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: Eğitmeden bu bölüm için çözülmesi gereken iki görevin bulunduğu slaytları sırası geldikçe açması beklenir. Ayrıca kodlama yapılacağı için Code::Blocks programı açılır.

Uygulama: Bu bölüm 2 görevden oluşmaktadır. Her bir görev için eğitmenin o görev esnasında neler yapacağı aşağıda sırasıyla verilmektedir.

Görev 1: Serkan yazdığı programda isim kısmına kullanıcının rakam girmesi durumunda programının “isminizde rakam olmaz” hatasını vermesini istiyor. Bunun için “isalpha” kodunda kullanan Serkan sizce nasıl bir kod yazmıştır?

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>
#include <cstring>

using namespace std;

int main()
{
    char ad[50];

    cout << "Adinizi giriniz:";
    cin >> ad;

    for(int i=0; i<strlen(ad);i++)
    {
        if(!isalpha(ad[i]))
        {
            cout << "Isminizde rakam olamaz!";
            return 0;
        }
    }
    cout << "Merhaba " << ad;
    return 0;
}
```

Görev 2: Ahmet yazacağı bir programın şifresinin otomatik olarak rastgele belirlenecek sekiz rakamdan oluşmasını istemektedir. Bunun için nasıl bir kod yazmalıdır?

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 eğitmen 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>
#include <ctime>
using namespace std;

int main() {
    srand(time(0));

    int karakterSayisi = 8;
    char sifre[9];

    for(int i=0; i<karakterSayisi; i++) {
        char karakter;
        do
        {
            karakter = rand() % 255;
        }
        while(!isalnum(karakter));
        sifre[i] = karakter;
    }
    sifre[karakterSayisi] = '\\0';
    cout << "Sifreniz: " << sifre;
    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

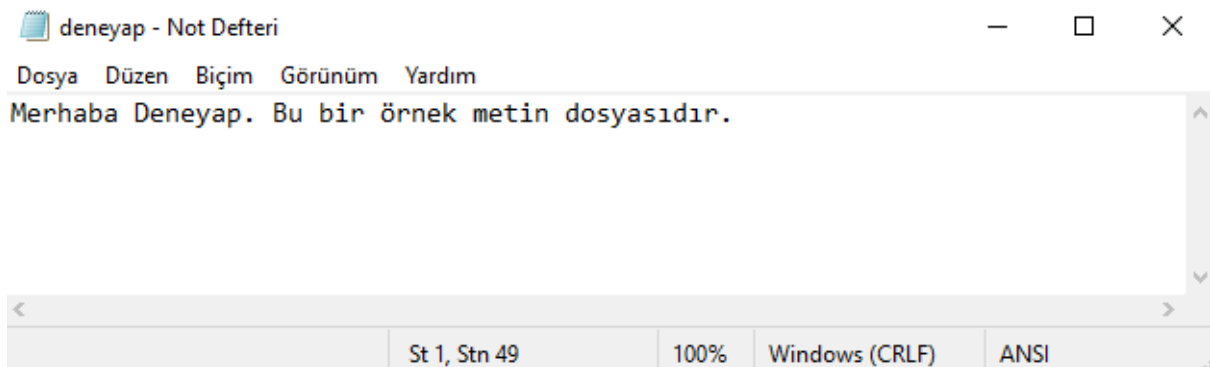
EK 8. Dosyalama İşlemleri

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 51. 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 27’de verilmiştir.

Tablo 27. 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 doldurdukları görev kâğıtlarının aşağıdaki gibi olması beklenir.

Tablo 28. 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. ofstream sınıfının varsayılan modu yazma için olan ios::out modudur. fstream ise varsayılan modu ios::in ve ios::out modlarıdır.	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.
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 işlemleri yapar.

K6. C++ programlama dilinde dosya yazma işlemleri yapar.

Materyaller: Hafta 11 Sunum

EK 9. Kodlar Arasında Farkı Bulma

Hazırlık: Eğitimci dersin bu bölümü için hazırlanan "Kodlar Arası Farkı Bulma" adlı slaytı 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 Bilgi:

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:

ilk olarak kütüphaneyi projemize dahil ettik. Ardından dosya isimli bir nesne oluşturduk. Parantez içerisine 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.

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: Eğitmeden 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.

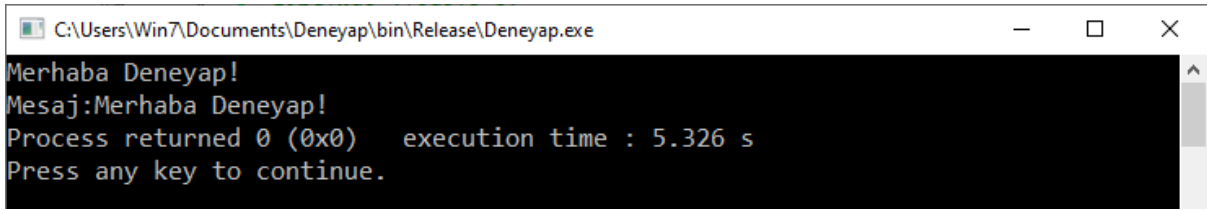
Görev 1: Klavyeden girilen “Merhaba Deneyap!” adlı cümleyi direkt string nesnesine aktarıp sonucu ekrana yazdıran kodu yazalım.

Yukarıdaki görev için “Görev 1” 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. Eğitmen bitirip el kaldıran öğrencilerden rastgele birini seçerek 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 1’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.

**Resim 52.** Ekran çıktısı

```
#include <iostream>
#include <fstream>
#include <string>
#include <locale.h>
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: C++ Programlama dili ile dosyalama kullanarak kendinize bir günlük oluşturun.

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 2:

```
#include <iostream>
#include <fstream>
#include <string>
#include <locale.h>
using namespace std;
int main()
{
    setlocale(LC_ALL, "");
    int islem;

    cout << "1- Günlük Oku." << endl;
    cout << "2- Günlük Yaz." << endl;
```

```

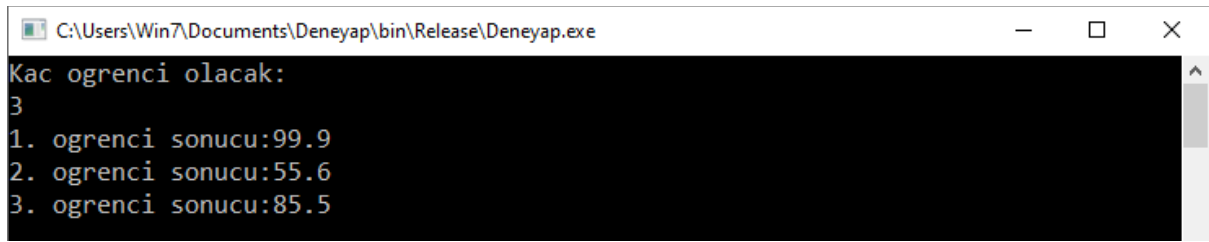
cin >> islem;

if(islem==1)
{
    ifstream dosya("gunluk.txt");
    if(!dosya.is_open())
    {
        cout << "Dosya Okunamadi!";
        return 0;
    }
    string satir;
    while(getline(dosya,satir))
        cout << satir <<endl;
    dosya.close();
}
else if(islem==2)
{
    ofstream dosya("gunluk.txt",ios::app);
    if(!dosya.is_open())
    {
        cout << "Dosya Okunamadi!";
        return 0;
    }
    string satir;
    cout << "Ne yazmak istersin:\n";
    cin.ignore();
    getline(cin, satir);
    cout << "Kaydediliyor...";
    dosya << satir <<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 eğitmenler öğrencilerin sıkıştığı noktalarda ipuçları vererek onlara destek olabilir. Süre bittikten sonra hangi öğrencilerin bu görevi tamamladığı sorulur ve bitirenlerin el kaldırma işlemi yapması istenir. Eğitmen 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 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

```

Resim 53. 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: 10 dk.

Materyal: Hafta 11 Süreli Ödev

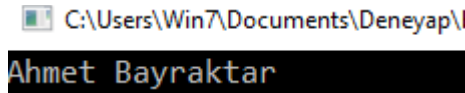
Hazırlık: EK 11 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır.

Ders dışı uygulama (60 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 dakika 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 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:

Kodlayıcı - "Ahmet BAYRaKtAR" şeklinde girilen ad soyad için, soyadının sadece ilk harfini büyük harfe diğer harflerini küçük harfe çeviren programı yazalım.



Resim 54. Ekran çıktısı

Cevap 1:

```
#include <iostream>
#include <cstring>

using namespace std;
int main()
{
    char mesaj[] = "Ahmet BAYRaKtAR";
    int kelimeSayisi = 0;
    bool bosluk = false;
    for(int i=0; i<strlen(mesaj);i++)
    {
        if(bosluk)
        {
```

```

        mesaj[i] = tolower(mesaj[i]);
    }
    if(mesaj[i] == ' ')
    {
        bosluk = true;
        mesaj[i+1] = toupper(mesaj[i+1]);
        i++;
    }
}

cout << mesaj ;
return 0;
}

```

Kodlayıcı - Parametre olarak gönderilen katar içerisinde parametre olarak gönderilen karakteri sayan fonksiyonu yazınız.

Cevap 2:

```

#include <iostream>
#include <cstring>

using namespace std;
int karakter_say(char metin[], char karakter)
{
    int sayac = 0;
    for(int i=0;i<strlen(metin);i++)
    {
        if(tolower(metin[i])==karakter)
            sayac++;
    }
    return sayac;
}

int main()
{
    char mesaj[] = "Bugun hava cok guzel. Azicik disari cikalim.";
    char karakter = tolower('B');

    cout << "Bu cumlede " << karakter_say(mesaj,karakter) <<" adet " << karakter
    <<" vardır.";
    return 0;
}

```

Kodlayıcı - Dosyadan okuduğunuz sayıların toplamını ekrana yazınız.

Cevap 3:

```

#include <iostream>
#include <stdlib.h>
#include <fstream>
#include <string>
using namespace std;

```

```
int main()
{
    int toplam = 0;
    ifstream dosya("sayilar.txt");
    string satir;
    if(dosya.is_open())
    {
        cout << "Sayilar:"<<endl;

        while(getline(dosya,satir))
        {
            toplam += atoi(satir.c_str());

            cout << satir << endl;
        }
        cout << "Toplam: " << toplam;
    }
    else
    {
        cout << "dosya Okunamadi!";
    }
}
```

Resim 55. Ekran çıktısı

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 29. 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 30. 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. C++ Programa Dilinde Bulunan Kütüphaneleri ve Fonksiyonları Kullanma

<u>Kod</u>	<u>Kodun Çıktısı:</u>
<pre> 1)..... 2)..... using namespace std; int main() { char c='a'; if(3.....(c)) cout << "Bu bir harftir."<<endl; else cout << "Bu bir harf degildir."<<endl; if(4.....(c)) cout << "Bu bir rakamdir."<<endl; else cout << "Bu bir rakam degildir."<<endl; if(5.....(c)) cout << "Bu bir kucuk harftir."<<endl; if(6.....(c)) cout << "Bu bir buyuk harftir."<<endl; return 0; } </pre>	Bu bir harftir. Bu bir rakam değildir. Bu bir küçük harftir.

```

#include <iostream>

1).....

using namespace std;

int main()
{
    char str1[] = "Bugun hava cok guzel.";
    char str2[] = "Piknige gidelim.";
    char str3[50];
    int uzunluk;

    cout << "str1 katari: " << str1 << endl;
    uzunluk = 2.....(str1);
    cout << "str1 katari uzunlugu: " << uzunluk << endl <<
endl;

    cout << "str2 katari: " << str2 << endl;
    uzunluk = strlen(str2);
    cout << "str2 katari uzunlugu: " << uzunluk << endl <<
endl;

    3.....(str3, str1);
    cout << "str3 katari: " << str3 << endl << endl;

    4.....(str1, str2);
    cout << "s1 katari: " << str1 << endl;

    uzunluk = 5.....(str1);
    cout << "s1 katari uzunlugu: " << uzunluk << endl <<
endl;

    if(6.....(str1, str3)==0)
        cout << "Iki katar birbirine esittir" << endl;
    else
        cout << "Iki katar birbirine esit degildir." <<
endl;
    return 0;
}

```

```

str1 katari: Bugun
hava cok guzel.

```

```

str1          katari
uzunlugu: 21

```

```

str2          katari:
Piknige gidelim.

```

```

str2          katari
uzunlugu: 16

```

```

str3 katari: Bugun
hava cok guzel.

```

```

s1 katari: Bugun
hava          cok
guzel.Piknige
gidelim.

```

```

s1          katari
uzunlugu: 37

```

```

Iki          katar
birbirine    esit
degildir.

```

EK 4. C++ Programa Dilinde Bulunan Kütüphaneleri ve Fonksiyonları Kullanma

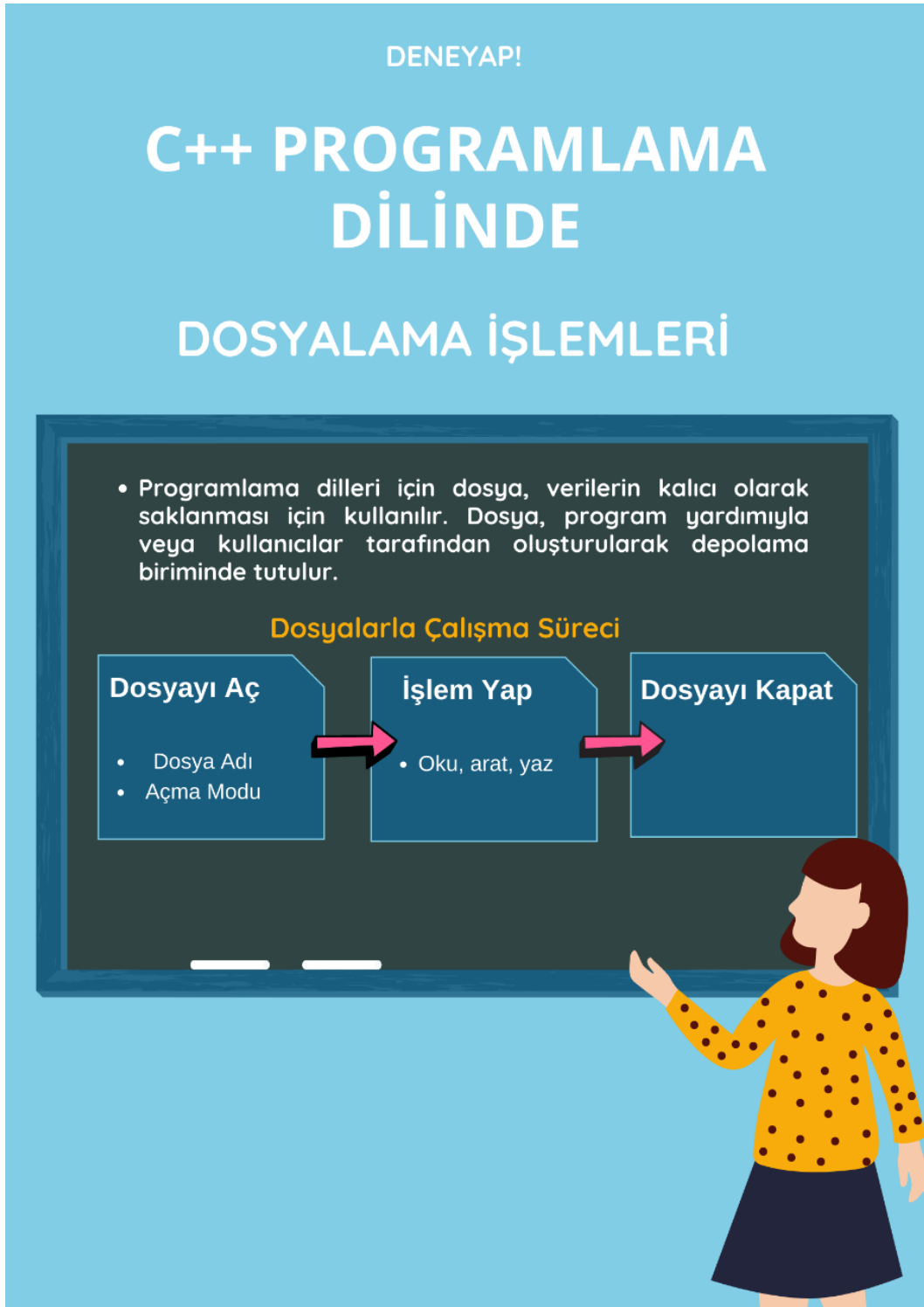
Görev 1

Serkan yazdığı programda isim kısmına kullanıcının rakam girmesi durumunda programının “isminizde rakam olmaz” hatasını vermesini istiyor. Bunun için “isalpha” kodunda kullanan Serkan sizce nasıl bir kod yazmıştır?

Görev 2

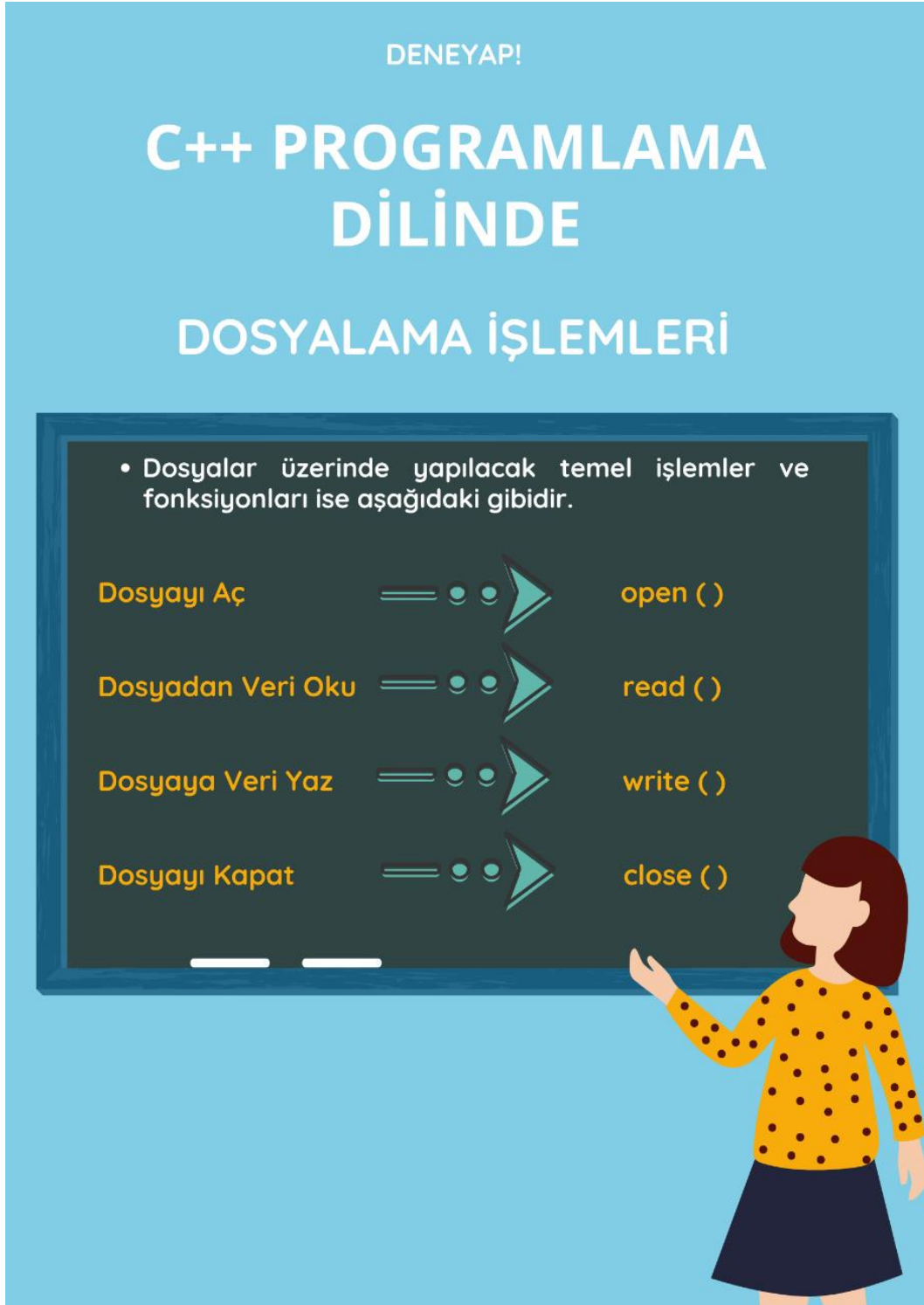
Ahmet yazacağı bir programın şifresinin otomatik olarak rastgele belirlenecek sekiz rakamdan oluşmasını istemektedir. Bunun için nasıl bir kod yazmalıdır?

EK 5. C++ Programlama Dilinde Dosyalama İşlemleri Yapıyorum Afişi 1



Resim 56. Dosyalama işlemleri afişi 1

EK 6. C++ Programlama Dilinde Dosyalama İşlemleri Yapıyorum Afişi 2



Resim 57. Dosyalama işlemleri afişi 2

EK 7. C++ Programlama Dilinde Dosyalama İşlemleri Yapıyorum Afişi 3

DENEYAP!

C++ PROGRAMLAMA DİLİNDE

DOSYALAMA İŞLEMLERİ

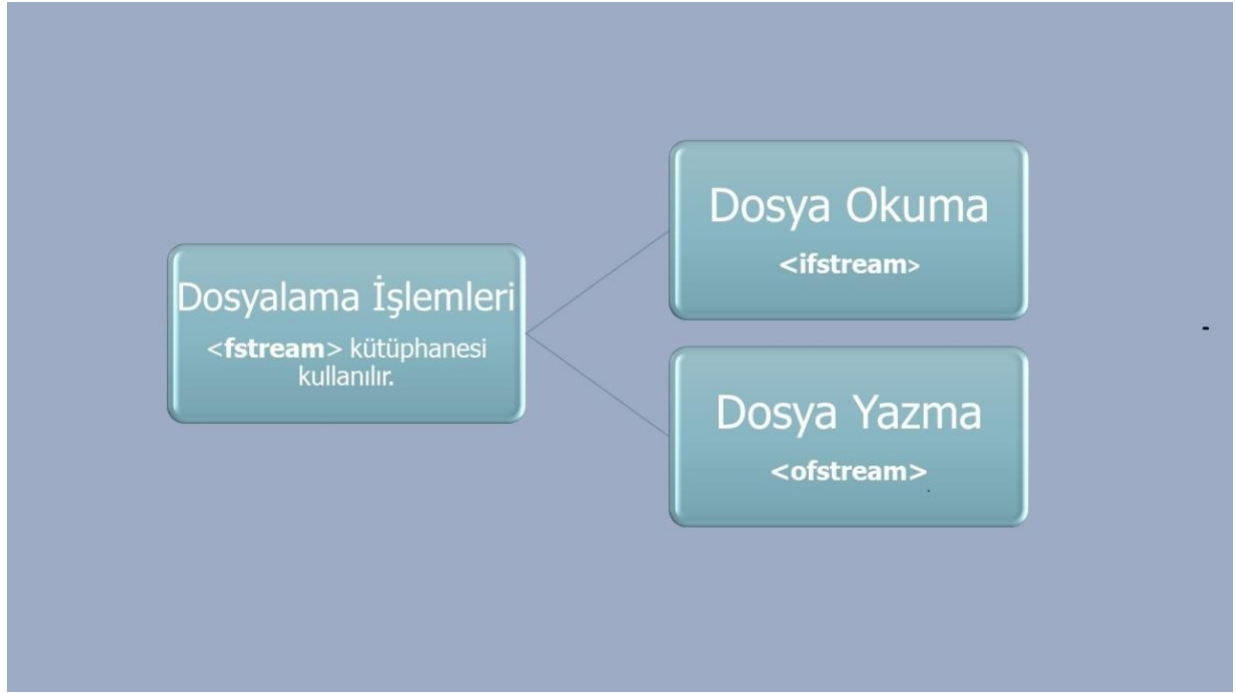
- Dosyayı açma sırasında eğer dosya mevcut değil ise, varsayılan olarak boş bir dosya oluşturulacaktır. Dosyayı farklı modlarda açabiliriz. Bu modlar;

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 58. Dosyalama işlemleri afişi 3

EK 8. Dosyalama İşlemleri



Resim 59. Dosyalama işlemleri

EK 9. Kodlar Arasındaki Farkı Bulma

1. İşlem	2. İşlem
<pre> #include <iostream> #include <fstream> using namespace std; int main() { ofstream dosya; dosya.open("deneyap.txt"); dosya << "Merhaba Deneyap!" << endl; dosya.close(); } </pre>	<pre> #include <iostream> #include <fstream> using namespace std; int main() { ofstream dosya; dosya.open("deneyap.txt",ios::app); dosya << "Merhaba Deneyap!" << endl; dosya.close(); } </pre>

!!! Yukarıda verilen kodları dikkatlice gözlemleyerek aralarında ne gibi bir fark olduğunu kodları bilgisayarda yazarak bulmaya çalışalım.

EK 10. Dosyalama İşlemleri ile İlgili Verilen Görevleri Kodluyorum

Görev 1

Klavyeden girilen
“Merhaba Deneyap!”
adlı cümleyi direk
string nesnesine
aktarıp sonucu ekrana
yazdıran kodu yazalım.

Görev 2

C++ Programlama dili
ile dosyalama kullana-
rak kendinize bir
günlük oluşturun.

Görev 3

Klavyeden girilen
öğrenci sayısı kadar
sınav notlarını
klavyeden okuyup
dosyaya yazdıran
programı oluşturunuz.

EK 11. 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 hataların 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 31. 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 karşılaşırlar. 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 küplerinin toplamından küçük ise bu sayıya “Deneyap sayısı” sayısı denilmektedir.

Örneğin, 22 sayısı $2^3+2^3= 16$ Deneyap sayısı değildir. Çünkü $22 > 16$.

Örneğin, 248 sayısı $2^3+4^3+8^3= 584$ dolayısıyla 248 Deneyap sayısıdır çünkü $248 < 584$.

Klavyeden girilen herhangi 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 main()
{
    char metin[] = "DeneYap";

    for(int i=0;i<strlen(metin);i++)
    {
        if(metin[i] == toupper(metin[i]))
            metin[i] = tolower(metin[i]);
        else
            metin[i] = toupper(metin[i]);
    }
    cout<< metin;
    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 edebilmek 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. Taban değerini atamak için *tabanAta()* ve yükseklik değerini atamak için *yukseklikAta()* fonksiyonlarını olmalıdır.
3. *Üçgen* sınıfının *alanHesapla()* isminde bir üye fonksiyonu olmalıdır ve bu fonksiyon 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, 6

Görev 2: Hafta 4, 5, 6, 7, 11

Görev 3: Hafta 4, 5, 6, 7, 8, 9

GÖREVLERİN DEĞERLENDİRİLMESİ

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/uVXC28HpHm1HwcRB8>

- 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)

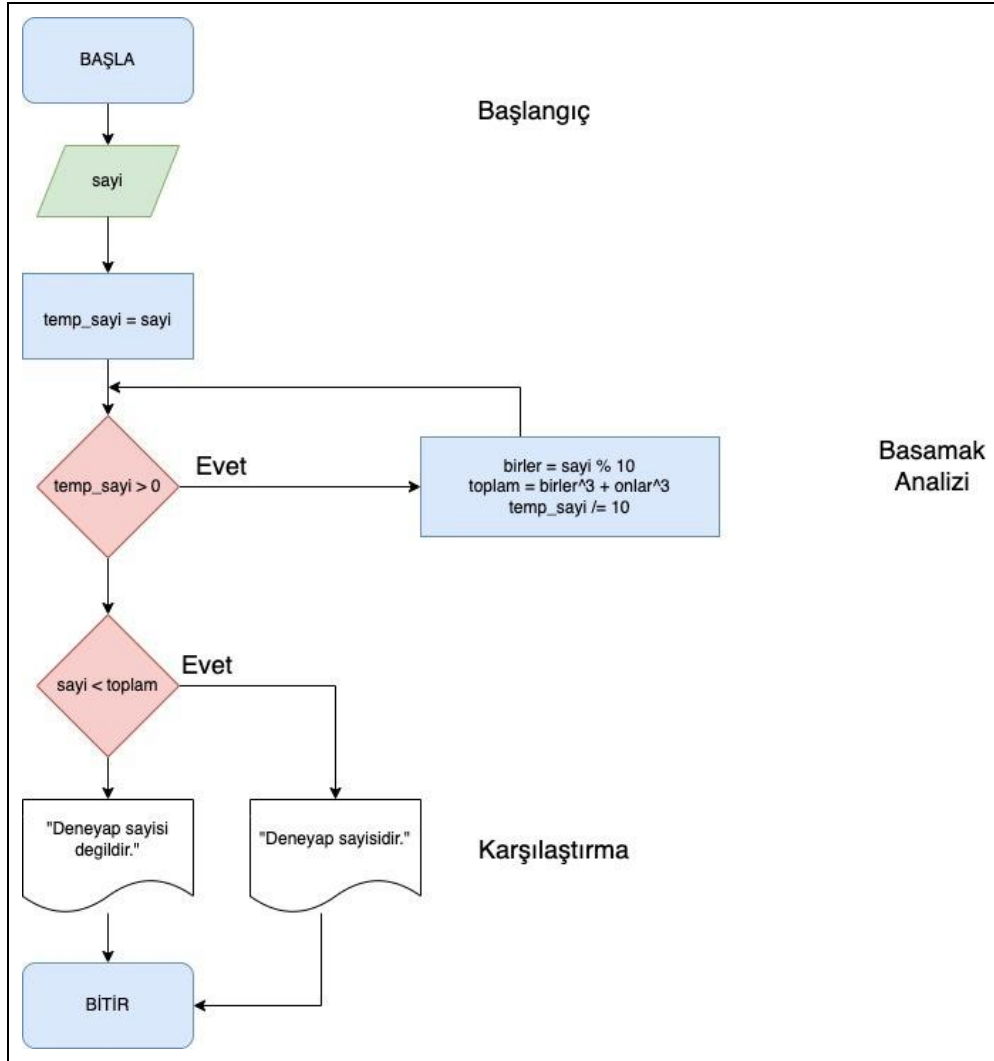
P3 = Problem 3 Çözümünden Alınan Puan (Min:0 Max:100)

GÖREVLERİN ÇÖZÜMLERİ

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, temp_sayi, toplam=0, rakam;

    cin >> sayi;
    temp_sayi = sayi;
    while(temp_sayi>0)
    {
        rakam = (temp_sayi % 10);
        toplam += rakam*rakam*rakam;
        temp_sayi = temp_sayi/10;
    }

    if(sayi < toplam)
        cout << "Deneyap sayisidir!";
    else
        cout << "Deneyap sayisi degildir!";

    return 0;
}
```

ÇÖZÜM 2:

Puanlama: Çıktı sonucu: 100 Puan

Çıktı: dENEyAP

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 = 15

Ucgenin yuksekligi = 13

Ucgenin alani=97.5

Kod:

```
// YARISMA GOREVI 3:

#include<iostream>
using namespace std;

// bir sinif tanimlayiniz

class Ucgen
{
    private:
        float taban, yukseklik;

    public:

        void tabanAta(float t)
        {
            taban = t;
        }
        void yukseklikAta(float y)
        {
            yukseklik = y;
        }

        float alanHesapla()
        {
            float alan;
            alan= 1.0 / 2.0 * taban * yukseklik;
            return alan;
        }
};
```

```
int main()
{
    // Ucgen sinifindan bir nesne tanımlayınız
    Ucgen ucgen1;
    float t,y;
    cout<<"Ucgenin tabani = ";
    cin>>t;
    cout<<"Ucgenin yuksekligi = ";
    cin>>y;
    //taban degerini atama
    ucgen1.tabanAta(t);
    // yukseklik degerini atama
    ucgen1.yukseklikAta(y);
    // Ucgenin alanini taban ve yukseklik bilgilerini kullanarak cagiriniz
    cout<<"Ucgenin alanı = "<<ucgen1.alanHesapla();

    return 0;
}
```

ÇEVİRİM İÇİ YAZILIM TEKNOLOJİLERİ

LİSE

