DERSİN ADI	**YAZILIM UYGULAMALARI**	
DERSİN SINIFI	**9. Sınıf**	
DERSİN SÜRESİ	**Haftalık 5 Ders Saati**	
DERSİN AMACI	Bu derste öğrenciye; yazılım uygulamaları geliştirme ile ilgili bilgi ve becerilerin kazandırılması amaçlanmaktadır.	
DERSE AİT ALAN BECERİSİ	Yazılım uygulamaları geliştirme	
EĞİLİMLER	E2.2. Sorumluluk E3.2. Odaklanma E3.3. Yaratıcılık E3.6. Analitiklik E3.7. Sistematiklik	
PROGRAMLAR ARASI BİLEŞENLER	**Sosyal-Duygusal Öğrenme Becerileri**	SDB2.1. İletişim Becerisi SDB2.2. İş Birliği Becerisi SDB3.3. Sorumlu Karar Verme Becerisi
	Değerler	D3. Çalışkanlık D4. Dostluk D5. Duyarlılık
	Okuryazarlık Beceriler	OB1. Bilgi Okuryazarlığı OB2. Dijital Okuryazarlık
DİSİPLİNLER ARASI İLİŞKİLER	Matematik, Mesleki Yabancı Dil	
BECERİLER ARASI İLİŞKİLER	KB2.3. Özetleme Becerisi KB2.4. Çözümleme Becerisi KB2.6. Bilgi Toplama Becerisi KB2.9. Genelleme Becerisi KB2.13. Yapılandırma Becerisi KB3.2. Problem Çözme Becerisi BTYAB2. Bilgi İşlemsel Düşünme Becerisi BTYAB3. Yazılım Geliştirme Becerisi	
ÖĞRENME ÇIKTILARI	Programlama yapılarını kullanabilme Nesneye yönelik programlama işlemlerini yapabilme Web ve mobil uygulamalar geliştirebilme Otomatik işlem uygulamaları gerçekleştirebilme	
ÖĞRENME KANITLARI (Ölçme ve Değerlendirme)	Bu derste öğrenci performansını belirlemeye yönelik çalışmalar değerlendirilirken gözlem formu, derecelendirme ölçeği, dereceli puanlama anahtarı gibi ölçme araçlarından uygun olanlar seçilerek kullanılabilir. Örneğin API oluşturma, bot geliştirme gibi konularda öğrenme kanıtı olarak performans görevi verilebilir. Yapılan çalışmalar, dereceli puanlama anahtarı, derecelendirme ölçeği, kontrol listesi gibi ölçme araçları ile değerlendirilebilir. Programlamanın temel yapılarına ilişkin ölçme aracı olarak zihin haritası, nesneye yönelik programlama işlemleri ile ilgili ise balık kılçığı kullanılabilir. Bunun yanında öz değerlendirme ve akran değerlendirme formları kullanılarak öğrencilerin öğretim sürecine aktif katılımları sağlanabilir. Web ve mobil uygulama geliştirmeye yönelik konularda portfolyo çalışması yapılabilir.	

<table>
<tr><td></td><td colspan="3">Portfolyo çalışması süreç sonunda dereceli puanlama anahtarı ile değerlendirilebilir. Ayrıca web kazıma ve bot oluşturma konularında senaryo tabanlı uygulama gerçekleştirilebilir. Bu ölçme yöntemleri sürecin her aşamasında geri bildirim sağlamak ve öğrencilerin gelişimlerini sistematik bir şekilde izlemek için kullanılır.</td></tr>
<tr><td>ÖĞRENME-ÖĞRETME YAŞANTILARI</td><td>Temel Kabuller

Öğrencilerin temel bilgisayar okuryazarlığına sahip olduğu, işletim sisteminde klasör ve dosya işlemleri yapabildiği, temel matematiksel işlemleri gerçekleştirebildiği ve algoritma mantığına dair ön bilgiye sahip olduğu kabul edilmektedir.</td><td>Ön Değerlendirme

Öğrencilere daha önce öğrendiği kavramları kullanarak yanıt verebilecekleri, merak uyandırıcı sorular yöneltilerek öğrencilerin konuya dikkatleri çekilebilir. Öğrencilerin ön öğrenmelerini kullanmalarını sağlayacak şekilde giriş-çıkış kartları kullanılabilir. Öğrencilere gündelik yaşamdan problem örnekleri verilip problem çözümüne ilişkin işlem adımlarını sıralaması istenebilir. Öğrencilere 3 satırlık hatalı bir algoritma/akış adımı verilerek ‘hata nerede olabilir?’ sorusu yöneltilebilir.</td><td>Köprü Kurma

Bir mobil uygulamada işlemlerin adım adım ilerlemesi algoritmalara, “giriş başarılı/başarısız” gibi karar noktaları koşullu ifadelere, bildirim kontrolü veya bir listede tekrar eden işlemler ise döngü yapılarına benzetilebilir. Ayrıca öğrencilerden günlük hayatta sık kullandıkları bir mobil uygulamayı seçerek “hangi adımlar sırayla gerçekleşiyor, hangi durumda karar veriliyor, hangi işlem tekrar ediyor?” sorularına kısa yanıtlar üretmesi ve sınıf içinde bu gözlemler üzerinden paylaşım yapması istenebilir.</td></tr>
<tr><td></td><td colspan="3">Öğrenme Öğretme Uygulamaları

Programlama Yapıları: Algoritma kavramı örneklerle açıklanır. Günlük yaşamdan basit problem senaryoları sunulur ve öğrencilerle birlikte çözüm yolları tartışılır. Öğrencilerden günlük yaşamda karşılaşılan basit sorunlara yönelik algoritmalar geliştirmesi ve bu algoritmaları akış diyagramına dönüştürmesi beklenir (SDB3.3, BTYAB2.1). Hazırlanan algoritmalar ve akış diyagramları, dereceli puanlama anahtarı veya kontrol listesi kullanılarak değerlendirilebilir. Algoritmaların kodlama ortamına nasıl aktarılacağı adım adım gösterilir ve temiz kod prensiplerinin önemi örneklerle vurgulanır. Öğrenciden hazırlanan algoritmaları daha sonra kodlama ortamında uygulaması ve temiz kod prensipleriyle test etmesi beklenir. Öğrencilerden farklı veri tiplerinde değişkenler tanımlaması, bu değişkenlerin dönüşümlerini uygulaması ve elde ettiği sonuçları özetleyerek sınıfta paylaşması beklenir (KB2.3). Bu uygulamalar, kontrol listesi, dereceli puanlama anahtarı vb. ile değerlendirilebilir. Kullanıcı girdisi-çıktısı örnekleri sunularak, veri tipi dönüşümleri, veri yapıları, karar yapıları (if-else) ve döngüler (for, while) ile ilgili konular sınıf ortamında uygulamalı olarak anlatılır. Öğrencilere farklı çözüm yolları gösterilir ve mini uygulamalar yaptırılır. Kullanıcıdan girdi alıp çıktı üreten uygulamalar geliştirmesi, karar yapıları (if-else) ve tekrarlı yapılar (for, while) kullanarak alternatif çözüm yolları üretmesi beklenir (E3.3). Fonksiyonlar konusuna geçildiğinde öğrencilerden hem temel fonksiyonları hem de özyinelemeli (recursive) fonksiyonları denemesi ve problem çözme becerilerini pekiştirmesi beklenir (E3.6, KB3.2). Lambda ve özel fonksiyonlar (filter, map, reduce ve zip) ile farklı veri işleme senaryoları uygulaması</td></tr>
</table>

	beklenir. Öğrencilerin hazır modüller (random, datetime, math vb.) ile uygulamalar yapılması ve dijital araçları etkin şekilde kullanması beklenir **(OB2)**. Bu uygulama kontrol listesi, dereceli puanlama anahtarı ve benzeri uygun olan ölçme değerlendirme aracıyla değerlendirilebilir. Ayrıca öğrencilerden uygulama sürecini öz değerlendirme formları ile değerlendirmeleri istenebilir. Hata yakalama (try-except-finally) yapıları uygulamalı olarak gösterilir, öğrencilerle birlikte sık karşılaşılan hata örnekleri sınıfta sunulur. Dosya işlemlerinin temeli örneklerle açıklanır ve farklı veri biçimleri üzerinde okuma, yazma ve düzenleme uygulamaları yaptırılır. Hata yakalama (try-except-finally) etkinlikleriyle öğrencilerden hataların kaynağını bulup çözüm önerileri geliştirmesi ve süreç boyunca azim göstermesi beklenir **(D3.1)**. Hazırlanan hata senaryoları grup değerlendirme ve akran değerlendirme formu ile değerlendirilebilir. Öğrencilerden dosya okuma, oluşturma ve yazma uygulamaları yapması ve verileri sistematik bir şekilde işlemesi beklenir **(E3.7)**. Tüm bu süreçte öğrencilerden özgün düşünme becerilerini kullanarak farklı çözümler üretmesi ve karşılaştıkları sorunları kendi yorumlarıyla aşmaya çalışması beklenir **(E3.3)**. Bu uygulamalar, kontrol listesi, dereceli puanlama anahtarı vb. ile değerlendirilebilir. **Nesneye Yönelik Programlama İşlemleri:** Nesne yönelimli programlamanın temel kavramları (sınıf, nesne, kapsülleme, kalıtım ve çok biçimlilik) örneklerle açıklanır ve kavramlar görsellerle desteklenir. Öğrencilerin önemli noktaları defterlerine not alması sağlanır. Öğrencilerden nesne yönelimli programlama kapsamında sınıf tanımlama ve örnekleme çalışmaları yapmaları beklenir. Bu süreçte metotlar ile veri üzerinde sistematik işlemler yapılabilmesi beklenir **(E3.7)**. Bu uygulamalar, kontrol listesi, dereceli puanlama anahtarı vb. ile değerlendirilebilir. Öğrencilerin örnek kodları defterlerine not almaları sağlanır ve yapılacak uygulama sınıfta açıklanır. Öğrencilerden metotlar ile yazılımın esnekliğini artırması, kapsülleme için setter ve getter metotları kullanması ve bilgi gizleme mantığı ile yazdığı kod yapısını pekiştirmesi beklenir **(KB2.4)**. Bu uygulamalar, kontrol listesi, dereceli puanlama anahtarı vb. ile değerlendirilebilir. Kalıtım (inheritance), çok biçimlilik (polymorphism) ve soyut sınıf (abstract class) kavramları, günlük yaşamdan örneklerle ilişkilendirerek anlatılır. Kavramlar görsel sınıf diyagramlarıyla açıklanır ve öğrencilerin diyagramları defterlerine çizmesi sağlanır. Öğrencilerden kalıtım örnekleri üzerinden nesneler arası ilişki kurabilmesi, çok biçimlilik (polymorphism) ve soyut sınıf uygulamaları ile soyutlama becerilerini geliştirebilmesi beklenir **(E3.7, KB2.9, KB2.13, BTYAB3.3)**. Hazırlanan sınıf ilişkileri, kavram haritası olarak düzenlenip kısa cevaplı sorulardan oluşan çalışma yapraklarıyla değerlendirilebilir. Arayüz geliştirmede PyQt kütüphanesinin kullanımı adım adım gösterilir. Görsel bir senaryo üzerinden pencere, buton ve menü oluşturma örnekleri sunulur. Uygulama sırasında öğrencilerin not almaları sağlanır ve yapılacak görevin kapsamı net biçimde açıklanır. Git ve GitHub kullanım süreci video veya görsel materyallerle tanıtılır. Ders kapsamında öğrencilerden PyQt kütüphanesini kullanılarak grafiksel arayüzler geliştirmesi beklenir. Ayrıca öğrencilerden versiyon kontrol sürecini öğrenebilmek için Git kurulumunu yapması, GitHub üzerinde depolar oluşturması ve hazırladığı uygulamaları sürüm kontrol sistemi aracılığıyla paylaşması **(OB2)**, ekip çalışması yaparak iş birliği içinde kod geliştirme deneyimi yaşaması **(SDB2.2)** ve arkadaşlarıyla dayanışma içinde olması **(D4.1)** beklenir. Tüm uygulamalar kontrol listesi, dereceli puanlama anahtarı ve benzeri uygun olan ölçme değerlendirme aracıyla değerlendirilebilir. Grup çalışmaları ise grup değerlendirme formu ile değerlendirilebilir. **Uygulama Geliştirme:** Web tarayıcı geliştirici araçlarını kullanarak HTTP request ve response süreci örneklerle gösterilir, ağ trafiği açıklanır. Öğrencilerden web tarayıcı geliştirici araçlarını kullanarak HTTP request ve response sürecini incelemesi, ağ trafiğini yorumlaması ve analiz sürecini raporlaması beklenir **(E3.2)**. Hazırlanan analiz raporları dereceli puanlama anahtarıyla değerlendirilebilir**.** HTML, CSS etiketleri açıklanır. Öğrencilerden

	HTML ve CSS kullanarak kişisel portföy web sitesi yapmaları beklenir**(KB2.13).** Bu uygulamalar kontrol listesi ve gözlem formu ile değerlendirilebilir. Javascript programlama yapıları örnek uygulamalar ile gösterilir. Öğrencilerden örnek tasarımlar üzerinde JavaScript programlama yapılarını kullanarak sayfalara etkileşim kazandırması beklenir **(KB3.2)**. Bu uygulamalar kontrol listesi ve gözlem formu ile değerlendirilebilir. Css ve javascript kütüphaneleri sınıfta örnek web siteleri gösterilerek sunulur. Daha modern ve ölçeklenebilir tasarımlar hazırlamak için öğrencilerden css ve javascript kütüphaneleri kullanarak uygulamalar geliştirip paylaşması beklenir **(OB2, BTYAB3.3)**. Hazırlanan uygulamalar kontrol listesi veya dereceli puanlama anahtarıyla değerlendirilebilir**.** Web uygulamaları geliştirmek için kullanılan arka uç programlama dilleri açıklanır. Farklı arka uç programlama dilleri ile hazırlanmış örnek web uygulamaları gösterilir, öğrencilerden web uygulamalarını incelemeleri beklenir. Öğrencilerden arka uç programlama dili ile form işlemlerinde GET ve POST metotlarını kullanması, HTTP başlıklarıyla ilgili örnekler hazırlaması beklenir. Oturum yönetimi ve hata yönetimi senaryoları geliştirmesi beklenir **(OB1, KB3.2, E3.7)**. Öğrencilerden web teknolojilerini kullanarak temiz kod yapısı ile kişisel portföy siteleri hazırlaması ve içeriklerini dijital ortamlarda paylaşması beklenir **(OB2, BTYAB3.3, KB3.2)**. CRUD uygulamaları ve REST API geliştirme çalışmaları yaparak bunları Postman ile test etmesi beklenir. Bu test süreci dereceli puanlama anahtarı, derecelendirme ölçeği vb. ile değerlendirilebilir**.** Hazırlanan uygulamalar dereceli puanlama anahtarıyla değerlendirilebilir. Mobil uygulama geliştirme ortamının kurulumu ve emülatör kullanımı adım adım gösterilir, görsellerle örnekler sunulur ve öğrencilerin not alması sağlanır. Öğrencilerden mobil uygulama geliştirme ortamını kurmaları beklenir. Kurulum sürecinde karşılaşılan hataların ve çözüm önerilerinin öğrenciler arasında paylaşılması beklenir **(KB3.2, SDB2.1).** Kurulum ve yapılandırma süreci kontrol listesiyle değerlendirilebilir. Emülatör kurulumu yaparak ve öğrencilerden farklı tasarım yapıları kullanarak yeni projeler oluşturması beklenir. Mobil uygulama programlama dili yapılarını kullanarak mobil uygulama geliştirme örneği gösterilir. Öğrencilerden farklı mobil uygulama örnekleri geliştirmeleri beklenir **(KB3.2)**. Hazırlanan uygulamalar kontrol listesi, dereceli puanlama anahtarı, derecelendirme ölçeği vb. ile değerlendirilebilir. Bu çalışmalar sırasında öğrencilerden sorumluluk bilinciyle hareket etmesi, yaptığı işlemlerin sonuçlarını öngörebilmesi ve yazılım geliştirme sürecini planlı şekilde yürütebilmesi beklenir **(E2.2, D3.2, SDB3.3, BTYAB3.3)**. Süreç, gözlem formu ile izlenebilir**.** **Otomatik İşlem Uygulamaları:** Selenium kütüphanesi ile kullanıcı parolalarının güvenlik düzeyini ölçen bot geliştirme süreci senaryo hâlinde sunulur, adım adım açıklanır ve gerekli kod parçaları örneklerle gösterilir. Öğrencilerin defterlerine not almaları sağlanır ve botun çalışma mantığı görsellerle desteklenir. Öğrencilerden Selenium kütüphanesini kullanarak kullanıcı parolalarının güvenlik düzeyini ölçen ve zayıf olanları raporlayan bir bot geliştirmesi beklenir. Bu sayede öğrencilerden hem dijital bilgilere ulaşmayı öğrenmesi **(OB2)** hem de yaratıcılıklarını yazılıma yansıtabilmesi beklenir **(E3.3)**. Hazırlanan raporlar dereceli puanlama anahtarıyla değerlendirilebilir. USOM web sitesinde (usom.gov.tr) yayımlanan zararlı bağlantıları tespit edip bu bağlantıları iletişim ve işbirliği araçlarıyla ağ güvenliği ekibine düzenli aralıklarla bildiren bot uygulamaları hazırlaması beklenir. Hazırlanan bot uygulamaları, kontrol listesiyle değerlendirilebilir. Öğrencilerden bot yazılımının sağlayabileceği faydaları arkadaşlarıyla paylaşması **(SDB2.1)** ve geliştirdiği projelerin topluma değer katabileceğini fark etmesi beklenir **(D5.1)**. Bu uygulama kontrol listesi, dereceli puanlama anahtarı ve benzeri uygun olan ölçme değerlendirme aracıyla değerlendirilebilir. Ayrıca öğrencilerden uygulama sürecini öz değerlendirme formları ile değerlendirmeleri istenebilir. Uygulama sürecinde öğrencilerden gerekli bilgileri toplaması **(KB2.6)** ve elde ettiği verileri botun işleyişinde kullanması beklenir. Ayrıca yazılımın düzenli çalışmasını sağlamak için geliştirme sürecini planlayabilmesi ve süreci yönetebilmesi beklenir

	(BTYAB3.3). Bu etkinliklerle öğrencilerden yalnızca teknik beceriler kazanmakla kalmayıp aynı zamanda sorumluluk bilinci ve toplumsal katkı farkındalığını da geliştirmesi beklenir.
ÖĞRENME ÖĞRETME ORTAM VE DONANIMI	**Ortam:** Siber Güvenlik Laboratuvarı **Güvenlik ve Altyapı:** Laboratuvar ortamında ağ trafiğini izleme, veri analizi ve yazılım geliştirme çalışmaları güvenli şekilde yapılmalıdır. 24 kişilik sınıf için kişisel bilgisayarlar, güvenlik duvarı (firewall) ve ağ anahtarlayıcıları (switch) laboratuvar ağına entegre edilmelidir. Ayrıca acil durum güç kesme butonu (acil stop), yangın söndürücü, uyarı levhaları, anti-statik bileklikler, izolasyon paspası ve ilk yardım seti **Bilgisayar ve Ağ Altyapısı:** Her öğrenciye bir bilgisayar sağlanmalıdır. Bilgisayarlarda yapılacak uygulamalara uygun bir işlemci, en az 8 GB RAM ve 256 GB SSD depolama kapasitesi bulunmalı; sanallaştırma destekli BIOS ayarları aktif edilmelidir. Sunucu olarak kullanılan sistemlerde sanal makine yönetimi (VMware, VirtualBox, Hyper-V vb.) kuruludur. Sanal ağ topolojileri oluşturmak için yönlendirici (router) ve anahtar (switch) cihazları **Yazılım ve Simülasyon Araçları:** Bilgisayarlarda yazılım geliştirme ortamları (VS Code, PyCharm, IntelliJ IDEA, Eclipse) ve web geliştirme araçları (XAMPP, Node.js, Postman) yüklü olmalıdır. Ayrıca sürüm kontrol sistemleri (Git, GitHub Desktop), konteyner altyapısı (Docker) **Sunum, Kayıt ve Düzen:** Etkileşimli tahta, projeksiyon cihazı ve öğretmen bilgisayarı, yazıcı, ekran kayıt yazılımları ve raporlama araçları, Laboratuvarda ders sürecine ait dokümantasyon (etkinlik raporları, kod örnekleri, değerlendirme formları)
ÖĞRENME BİRİMİ / MODÜL VE SÜRE TABLOSU	

ÖĞRENME BİRİMİ / MODÜL	**ÖĞRENME BİRİMİ / MODÜL SÜRESİ**
Programlama Yapıları	40/30
Nesneye Yönelik Programlama İşlemleri	80/60
Uygulama Geliştirme	80/65
Otomatik İşlem Uygulamaları	40/25
TOPLAM	**240/180**

DERS BİLGİ FORMU

ÖĞRENME BİRİMİ / MODÜL	**ÖĞRENME ÇIKTISI GRUBU VE AÇIKLAMALARI**	**İÇERİK ÇERÇEVESİ**

Programlama Yapıları	**Programlama yapılarını kullanabilme**	**Verilen problemi çözmek için algoritma hazırlar.** Algoritma ve akış diyagramı kavramları açıklanır. Farklı problemleri çözmek için algoritmalar hazırlatılır. Hazırlanan algoritmalar için akış diyagramları yaptırılır. **Programlamanın temel yapılarını kullanır.** Programlama kavramları (komut, program, yazılım, programlama dili, makine dili, derleme, yorumlama) açıklanır. Yazılım geliştirme yaşam döngüsü açıklanır. Şelale ve çevik yazılım geliştirme yaşam döngüsü modelleri karşılaştırılarak açıklanır. Temiz kod açıklanır. Temiz kod yazma prensipleri açıklanır. Değişken açıklanır. Değişken tanımlama kuralları açıklanır. Veri tipleri açıklanır. Farklı veri tiplerinde değişkenler tanımlanır. Operatör kavramı açıklanır. Operatör türleri (Atama operatörleri, aritmetiksel operatörler, ilişkisel operatörler ve mantıksal operatörler) açıklanır. Operatör önceliği açıklanır. Veri tipi dönüşümleri yaptırılır. Kullanıcıdan girdi alma ve kullanıcıdan aldığı girdiyi ekrana yazdırma uygulaması yaptırılır. Liste veri yapısı açıklanır. Liste veri yapısı ile ilgili uygulamalar yaptırılır. Demet veri yapısı açıklanır. Demet veri yapısı ile ilgili uygulamalar yaptırılır. Sözlük veri yapısı açıklanır.	**Algoritma** **Programlamanın Temel Yapıları**

		Sözlük veri yapısı ile ilgili uygulamalar yaptırılır. Küme veri yapısı açıklanır. Küme veri yapısı ile ilgili uygulamalar yaptırılır. Karar yapıları (if-else, if-elif-else, iç içe if, match-case) açıklanır. Karar yapıları ile ilgili uygulamalar yaptırılır. Tekrarlı yapılar (döngüler) açıklanır. Break ve continue komutları açıklanır. Tekrarlı yapılar ile ilgili uygulamalar yaptırılır. Fonksiyon kavramı açıklanır. Parametre kavramı açıklanır. Geri değer döndürme kavramı açıklanır. Fonksiyon uygulamaları yaptırılır. Yerel ve global değişkenler açıklanır. Özyinelemeli (recursive) fonksiyon açıklanır. Özyinelemeli fonksiyon uygulaması yaptırılır. Lambda fonksiyon açıklanır. Lambda fonksiyon uygulaması yaptırılır. Özel fonksiyonlar (filter, map, reduce, zip) açıklanır. Özel fonksiyonlar kullanılarak uygulama yaptırılır. Modüller ve sağladığı avantajlar açıklanır. Hazır modüller (random, time, datetime, math, sys, os vb.) ile uygulama yaptırılır. Hata yakalama (try-except-finally) uygulamaları yaptırılır. Dosya okuma, oluşturma ve yazma işlemleri yaptırılır.	
Nesneye Yönelik Programlama İşlemleri	**Nesneye yönelik programlama işlemlerini yapabilme**	**Programlamada nesneye yönelik işlemler yapar.** Nesneye yönelik programlama yaklaşımı açıklanır.	**Nesneye Yönelik Programlama** **Versiyon Kontrol Sistemleri**

		Nesneye yönelik programlama yaklaşımının sağladığı kolaylıklar açıklanır. Sınıf kavramı açıklanır. Sınıf üyeleri (özellikler ve metotlar) açıklanır. Erişim belirleyiciler (Access modifiers) açıklanır. Nesne kavramı açıklanır. Nesneye yönelik programlama yaklaşımının temel prensipleri (Soyutlama, Kapsülleme, Kalıtım, Çokbiçimlilik) açıklanır. Yapıcı metot (Constructors), yıkıcı metot (Destructors) ve self parametresi açıklanır. Sınıf tanımlama ve örnekleme (Instantiation) uygulamaları yaptırılır. Setter ve Getter metotları açıklanır. Setter ve Getter metotları ile kapsülleme ve bilgi gizleme uygulaması yaptırılır. Kalıtım işlemi açıklanır. Çoklu kalıtım işlemi açıklanır. Kalıtım uygulamaları yaptırılır. Çokbiçimlilik açıklanır. Çokbiçimlilik ile ilgili uygulama yaptırılır. Soyut sınıf açıklanır. Soyut sınıf ile ilgili uygulama yaptırılır. PyQt kütüphanesi açıklanır. Nesne yönelimli programlama yaklaşımı ile PyQt kütüphanesi kullandırılır. **Yapılan uygulamaları versiyon kontrol sistemlerine aktarır.** Versiyon kontrol sistemi açıklanır. Versiyon kontrol sisteminin sağladığı faydalar açıklanır. Versiyon kontrol sistemi türleri (yerel, merkezî ve dağıtık) açıklanır.	

		Git versiyon kontrol sistemi açıklanır. Git versiyon kontrol sistemini kullanan depolama hizmetleri (GitHub, GitLab) açıklanır. Git versiyon kontrol sisteminin kurulumu yaptırılır. Git versiyon kontrol sistemi kavramları (Repository, Branch, Master, Commit, Checkout, Fork, Pull Request, Merge vb.) açıklanır. Git versiyon kontrol sisteminin çalışma mantığı açıklanır. Git komutları açıklanır. GitHub depolama servisinde repo oluşturulur. Hazırlanan uygulamaların Git komutları kullanılarak versiyon kontrol sistemine gönderilmesi işlemi gerçekleştirilir.	
Uygulama Geliştirme	**Web ve mobil uygulamalar geliştirebilme**	**Web uygulamalarında ihtiyaç duyulan teknolojileri kullanır.** Web teknolojisinin temelleri açıklanır. Web tarihinin gelişim süreci (Web 1.0, Web 2.0, Web 3.0, Web 4.0) açıklanır. HTTP protokolü açıklanır. HTTP request ve HTTP response yapısı açıklanır. HTTP metotları (GET, POST, HEAD, PUT, DELETE) açıklanır. HTTP durum kodları (1xx, 2xx, 3xx, 4xx, 5xx) açıklanır. Web tarayıcı geliştirici araçları kullanılarak HTTP request ve HTTP response inceleme uygulaması yaptırılır. Çerez kavramı açıklanır. Oturum kavramı açıklanır. Uygulama programlama arayüzü [Application Programming Interface (API)] açıklanır. API mimarileri açıklanır. Veri değişim formatları (XML, JSON, YAML) açıklanır.	**Web Uygulamaları** **Mobil Uygulamalar**

		Web uygulamalarının yapı taşları (HTML5, CSS3, Javascript) açıklanır. Javascript programlama yapıları kullanılarak uygulamalar yaptırılır. Web uygulamaları geliştirmek için kullanılan arka uç programlama dilleri açıklanır. Temiz kod yapısı ile web uygulaması geliştirilir. REST API geliştirme uygulaması yaptırılır. Postman aracı kullanılarak geliştirilen API'nin testi yaptırılır. **Mobil uygulama geliştirir.** Mobil (Android) işletim sistemi mimari yapısı açıklanır. APK dosyası ve bileşenleri açıklanır. Mobil uygulamanın temel bileşenleri açıklanır. Mobil uygulama derleme süreci açıklanır. Mobil uygulama geliştirme ortamının kurulumu yaptırılır. Emülatör kurulumu yaptırılır. Farklı tasarım yapıları kullanılarak yeni proje oluşturma adımları yaptırılır. Activity yaşam döngüsü ve metotları açıklanır. Mobil programlama dili yapıları kullanılarak uygulamalar yaptırılır. Hazırlanan uygulamaların APK dosyaları oluşturulur.	
Otomatik İşlem Uygulamaları	**Otomatik işlem uygulamaları gerçekleştirebilme**	**Uygulamalarda web kazıma tekniklerini kullanır.** Web kazıma işlemi açıklanır. Web kazıma süreci ve teknikleri açıklanır. Web kazıma sürecinde kullanılan programlama dilleri ve kütüphaneler açıklanır. Veri toplamak için BeautifulSoup ve Requests kütüphaneleri ile web kazıma uygulaması yaptırılır.	**Web Kazıma** **Bot Geliştirme**

		Web kazıma işlemini önleme yöntemleri açıklanır. **Uygulamalarda otomatik işlemler gerçekleştirmek için bot oluşturur.** Bot yazılım açıklanır. Bot kullanım alanları açıklanır. Bot oluşturmak için kullanılan kütüphaneler açıklanır. Selenium kütüphanesi ile bot uygulamaları yaptırılır. Bot engelleme yöntemleri açıklanır.	

UYGULAMA FAALİYETLERİ/TEMRİNLER		
Uygulama faaliyeti / temrinler; ders çıktısına uygun olarak okulun fiziki kapasitesi ve donatımı, öğrenci sayısı göz önünde bulundurularak en fazla uygulama faaliyeti / temrini yaptıracak şekilde meslek alan zümre öğretmenler kurulu tarafından seçilir. Meslek alan zümre öğretmenleri tarafından aşağıda yer alan temrinlerden farklı temrinlerin uygulanmasına karar verilebilir.		
ÖĞRENME BİRİMİ / MODÜL	**UYGULAMA FAALİYETLERİ / TEMRİNLER**	**İLİŞKİLİ BİLEŞENLER**

Programlama Yapıları	Farklı problemleri çözmek için algoritmalar hazırlamak Hazırlanan algoritmalar için akış diyagramları oluşturma uygulaması yapmak Farklı veri tiplerinde değişken tanımlama uygulaması yapmak Veri tipi dönüştürme uygulamaları yapmak Kullanıcıdan girdi alma ve kullanıcıdan aldığı girdiyi ekrana yazdırma uygulaması yapmak Veri yapıları (liste, sözlük, demet vb.) ile ilgili uygulamalar yapmak Karar yapıları ile ilgili uygulamalar yapmak Tekrarlı yapılar (döngüler) ile ilgili uygulamalar yapmak Fonksiyon uygulamaları yapmak Özyinelemeli (Recursive) fonksiyon uygulaması yapmak Lambda fonksiyon uygulaması yapmak Özel fonksiyonlar (filter, map, reduce, zip) ile ilgili uygulamalar yapmak Hazır modüller (random, time, datetime, math, sys, os vb.) ile ilgili uygulamalar yapmak Hata yakalama (try-except-finally) uygulamaları yapmak Dosya okuma, oluşturma ve yazma işlemleri yapmak	E3.3. Yaratıcılık E3.6. Analitiklik E3.7. Sistematiklik SDB3.3. Sorumlu Karar Verme Becerisi D3.1. Azimli olmak OB2. Dijital Okuryazarlık KB2.3. Özetleme Becerisi BTYAB2.1. Algoritmik Düşünme
Nesneye Yönelik Programlama İşlemleri	Sınıf tanımlama ve örnekleme (Instantiation) uygulamaları yapmak Setter ve Getter metotları ile kapsülleme uygulaması yapmak Kalıtım uygulamaları yapmak Çok biçimlilik ile ilgili uygulama yapmak Soyut sınıf ile ilgili uygulama yapmak. Nesne yönelimli programlama yaklaşımı ile PyQt kütüphanesini kullanarak uygulama yapmak Git versiyon kontrol sisteminin kurulumunu yapmak. GitHub depolama servisinde repo oluşturmak Hazırlanan uygulamaları Git komutlarını kullanarak versiyon kontrol sistemine gönderme işlemini gerçekleştirmek	E3.7. Sistematiklik SDB2.2. İş Birliği Becerisi D4.1.Arkadaşlarına destek olmak OB2. Dijital Okuryazarlık KB2.4. Çözümleme Becerisi KB2.9. Genelleme Becerisi KB2.13. Yapılandırma Becerisi KB3.2. Problem Çözme Becerisi BTYAB3.3. Yazılım Geliştirme Sürecini Yönetme

Uygulama Geliştirme	Web tarayıcı geliştirici araçlarını kullanarak HTTP request ve HTTP response inceleme uygulaması yapmak Web teknolojilerini kullanarak kişisel portföy sitesi (hakkında, iletişim, proje sayfaları) geliştirme uygulaması yapmak Javascript programlama yapılarını kullanarak uygulamalar yapmak Esnek, ölçeklenebilir ve modern web tasarımları yapmak için css ve js kütüphaneleri kullanarak uygulamalar yapmak REST API geliştirme uygulaması yapmak Geliştirilen API'nin testini postman aracını kullanarak yapmak Mobil uygulama geliştirme ortamının kurulumunu yapmak Emülatör kurulumunu yapmak Farklı tasarım yapılarını kullanarak yeni proje oluşturma uygulamaları yapmak Mobil programlama dili yapılarını kullanarak uygulamalar yapmak	E2.2. Sorumluluk E3.2. Odaklanma E3.7. Sistematiklik SDB2.1. İletişim Becerisi SDB3.3. Sorumlu Karar Verme Becerisi D3.2. Planlı olmak OB1.Bilgi Okuryazarlığı OB2.Dijital Okuryazarlık KB3.2. Problem Çözme Becerisi BTYAB3.3. Yazılım Geliştirme Sürecini Yönetme
Otomatik İşlem Uygulamaları	Selenium kütüphanesi ile kullanıcıların parolalarını toplayan, parola gücünü analiz eden ve zayıf parolaları raporlayan bot uygulaması yapmak USOM (usom.gov.tr) web sitesinde paylaşılan zararlı bağlantıları ağ güvenliği ekibine (Telegram, Slack vb.) periyodik olarak gönderen bot uygulaması yapmak	E3.3. Yaratıcılık SDB2.1. İletişim Becerisi D5.1. İnsana ve topluma değer vermek OB2. Dijital Okuryazarlık KB2.6. Bilgi Toplama Becerisi BTYAB3.3. Yazılım Geliştirme Sürecini Yönetme

DERSİN UYGULANMASINA İLİŞKİN AÇIKLAMALAR

Her öğrencinin uygulama yapabilmesi için uygun ortam oluşturulur.

Uygulama faaliyetlerinde iş sağlığı ve güvenliğine ilişkin risk ve tehlike oluşturabilecek her türlü duruma karşı tedbirler alınır.

Ders işlenirken öğrencilerde birlikte iş yapabilme, kendini ifade edebilme, sorumluluk bilinci, öz güven vb. değer, tutum ve davranışları ön plana çıkaran etkinliklere yer verilmelidir. Bu etkinliklerde beyin fırtınası, grup tartışması, düz anlatım, soru cevap, örnek olay incelemesi gibi yöntem ve teknikler kullanılır.

Anlatım ve örnek çalışmalardan sonra dersin öğrenme çıktılarının öğrencide pekiştirilmesi amacıyla birden fazla uygulama faaliyeti yapılması istenir.

Ahilik kültürüne ilişkin değerler ile programlar arası bileşenler altındaki "Değerler" eşleştirilerek öğrenme ve öğretme faaliyetleri içinde öğrenciye verilmelidir.

Öğrencilerden kendi güvenlik gözlemlerini paylaşmaları istenir.