DERS BİLGİ FORMU

DERSİN ADI	**PROGRAMLAMA TEMELLERİ**	
DERSİN SINIFI	**9. Sınıf**	
DERSİN SÜRESİ	**Haftalık 4 Ders Saati**	
DERSİN AMACI	Bu derste öğrenciye; temel programlama prensiplerini ve algoritmik tasarım süreçlerini kullanarak gerçek dünya problemlerine modüler ve sürdürülebilir yazılım çözümleri uygulamalarını yapma ile ilgili bilgi ve becerilerin kazandırılması amaçlanmaktadır.	
DERSE AİT ALAN BECERİSİ	**Öğrencinin bir problemi analiz ederek algoritmik çözüm tasarlama ve bu tasarımı programlama dilleriyle standartlara uygun şekilde yazılıma dönüştürme becerisidir.**	
EĞİLİMLER	E1.1. Merak E3.6. Analitiklik E3.7. Sistematiklik E3.10. Eleştirel Bakma	
PROGRAMLAR ARASI BİLEŞENLER	**Sosyal-Duygusal Öğrenme Becerileri**	SDB1.2. Kendini Düzenleme (Öz Düzenleme Becerisi) SDB3.1. Uyum Becerisi
	Değerler	D3. Çalışkanlık D6. Dürüslük D19. Vatanseverlik
	Okuryazarlık Beceriler	OB1. Bilgi Okuryazarlığı OB2. Dijital Okuryazarlık OB7. Veri Okuryazarlığı
DİSİPLİNLER ARASI İLİŞKİLER	Matematik, Yabancı Dil	
BECERİLER ARASI İLİŞKİLER	KB2.4. Çözümleme Becerisi KB2.7. Karşılaştırma Becerisi KB2.8. Sorgulama Becerisi KB2.10. Çıkarım Yapma KB2.13. Yapılandırma Becerisi KB2.16. Muhakeme (Akıl Yürütme) Becerisi KB2.17. Değerlendirme Becerisi KB2.19. Mantıksal Denetleme Becerisi KB3.3 Eleştirel Düşünme Becerisi MAB2.1. Matematiksel Çözümler Geliştirme BTYAB2.1. Algoritmik Düşünme BTYAB6.2. Dijital Güvenlik Önlemlerini Uygulama	
ÖĞRENME ÇIKTILARI	Problemlere uygun algoritmik çözümler tasarlayabilme Temel programlama yapılarını kullanarak kullanıcı etkileşimli yazılımlar geliştirebilme Kontrol yapılarını kullanarak program akışını yönetebilme Tekrar eden kodları fonksiyonlar kullanarak modüler ve sürdürülebilir yapılara dönüştürebilme Gelişmiş veri yapılarını ve dosya sistemlerini kullanarak verileri organize edebilme İstisna yönetimi tekniklerini kullanarak program dayanıklılığını optimize eder	

<table>
<tr><td>ÖĞRENME KANITLARI
(Ölçme ve Değerlendirme)</td><td colspan="3">Öğrencilerin bir konunun öğrenme çıktılarına ulaştıklarını gösteren gözlemlenebilir ve somut ürünler bu bölümün temelini oluşturur. Süreç boyunca geliştirilen "Sarmal Bir Proje (Kod-Asistan vb.)", öğrenmenin gerçekleştiğinin ispatı olarak kabul edilir.
Bütüncül Performans Kanıtları (Ürün Odaklı):
● Sarmal Proje Kaynak Kodları (v3.0): Fonksiyonel, modüler yapıya sahip, istisna yönetimi içeren ve kalıcı veri yapısı (JSON) kullanan çalışan yazılım dosyalarıdır.
● Dijital Gelişim Günlüğü (Repo): Projenin v0.1'den v3.0'a kadar olan değişimini gösteren sarmal sürüm kontrol (commit) geçmişidir.
● Teknik Dokümantasyon (README.md): Proje mimarisini, kullanılan teknolojileri ve kurulum adımlarını içeren profesyonel doküman dosyasıdır.
● Görsel Kanıt (Grafik Çıktısı): Soyut algoritmik mantığın görselleştirildiğini gösteren grafik kütüphanesi çıktılarıdır.

Süreç Odaklı Ölçme Araçları:
● Kod Gözden Geçirme (Code Review) Formu: Kodun okunabilirliği, isimlendirme standartları ve modülerliği için kullanılan ölçektir.
● Yapay Zekâ Kullanım Yansıtma Raporu: Yapay zekâyı öğrenme odaklı ve etik kurallar çerçevesinde kullanma tutumuna dair rapordur.
● Hata Avcılığı (Debugging) Gözlem Formu: Hata mesajlarını analiz etme ve çözüm üretme becerisinin izlendiği formdur.</td></tr>
<tr><td rowspan="2">ÖĞRENME-ÖĞRETME YAŞANTILARI</td><td>Temel Kabuller

Öğrencilerin temel bilgisayar kullanım becerilerine sahip olduğu varsayılır.
Günlük yaşam rutinlerini mantıksal bir sıraya koyabildikleri ve aritmetiksel işlemleri yapabildikleri kabul edilir.
Bilişim teknolojileri araçlarını kullanırken temel etik kuralların farkında oldukları varsayılır.</td><td>Ön Değerlendirme

Öğrencilerin algoritma, değişken ve mantıksal işlem süreçlerine dair mevcut bilgi düzeylerini belirlemek amacıyla açık uçlu sorular ve hazır bulunuşluk testleri uygulanır.
Öğrencilerin karşılaştıkları teknik problemlere yaklaşımlarını ve olası kavram yanılgılarını tespit etmek için kelime ilişkilendirme çalışmaları yapılır.
İlgi ve ihtiyaçları belirlemek üzere öğrencilere yönelik anketler düzenlenerek eğitim içeriğinin bireyselleştirilmesi sağlanır.</td><td>Köprü Kurma

Günlük hayatta karşılaşılan karmaşık süreçlerin (yol tarifi, yemek tarifi, otomasyon sistemleri) programlama mantığı ile olan bağı tartışılır.
"Sarmal Proje" (Kod-Asistan vb.) yaklaşımıyla, öğrenilen her yeni teknik kavramın gerçek dünyadaki bir soruna nasıl çözüm üretebileceği örneklenir.
Sınıfta öğrenilen soyut kod yapılarının, dijital ürünlere ve profesyonel yazılım geliştirme süreçlerine nasıl dönüştüğü modellenerek somutlaştırılır.</td></tr>
<tr><td colspan="3">Algoritmik Düşünme ve Tasarım: Öğrenme süreci, dönem boyunca kademeli olarak geliştirilecek "Sarmal Bir Proje" (Kod-Asistan vb.) odağında yapılandırılır. Günlük hayattan bir problemin çözüm adımları üzerine ile sınıf içi tartışma yürütülür. Öğrenci cevapları doğrudan düzeltilmeyerek "Bu adımı atlamış olabilir miyiz?" gibi sorgulayıcı dönütlerle algoritmik düşünme becerisi desteklenir (E1.1, KB2.8, KB2.16).
Algoritma kavramının tarihsel gelişimi kapsamında El-Harezmi'nin "adım adım ilerleme" yöntemi ve günümüz teknolojilerindeki yeri görsel materyallerle açıklanır. Öğrencilerin milli teknoloji tarihine dair farkındalık kazanmaları sağlanır (D19.2).
Seçilen günlük problemler girdi–işlem–çıktı basamaklarına ayrılarak analiz edilirken, yapay zekâ araçları çözümü sorgulama ve karşılaştırma amacıyla kullanılır (D3.3, E3.10, KB2.4).
Akış şeması sembolleri tanıtılarak her sembolün günlük hayattaki karşılığı (yol ayrımı, eylem vb.) üzerinden ilişkilendirme yapılır. Tahtaya çizilen hatalı şemalar üzerinden öğrencilerin mantık hatalarını bulmaları sağlanarak hata ayıklama süreci modellenir</td></tr>
</table>

	(KB2.19, BTYAB2.1). Oluşturulan sözde kodların (pseudo-code) standart sembollerle akış şemasına dönüştürülmesi sağlanır. Sarmal projenin ilk aşaması (v0.1) olarak kod içermeyen tasarım sürümü planlanır. Öğrencilerin proje dokümantasyonu için dijital bir depo (repository) oluşturmaları ve projenin amacını anlatan README dosyalarını hazırlamaları sağlanır **(OB2, D16.3).** Yapılan çalışmalar; girdi-işlem-çıktı tanımları ve sembol kullanımı kriterlerini içeren dereceli puanlama anahtarı ile değerlendirilir. **Temel Programlama Yapıları:** Derse bilgisayarlar kapalıyken başlanarak değişkenlerin bellekteki (RAM) geçici varlığı üzerine bir tartışma yürütülür. Temel veri türleri ve isimlendirme standartları tahtada modellenerek, input() fonksiyonundan gelen verinin varsayılan olarak metinsel (str) olduğu vurgulanır **(KB2.4, OB7).** Bilerek oluşturulan tür hataları (ValueError) üzerinden çözüm yolları sınıfça analiz edilir. Öğrencilerin sarmal projelerine (Kod-Asistan vb.) kullanıcıyı ismiyle karşılayan bir "Selamlama Modülü" ve yaş hesaplayan bir "Hesaplama Katmanı" eklemeleri sağlanır **(D3.2, OB2, E3.7).** İşlem önceliği ve parantez kuralları, vücut kitle indeksi veya döviz çevirici gibi özgün hesaplayıcılar üzerinden uygulanır. Geliştirilen kodların dijital depoya yüklenme ve sürüm notu yazılma süreçleri modellenir. Yapay zekâ destekli kod üretiminde "dikkatsiz stajyer" benzetmesi yapılarak, LLM araçlarının kısıtlılıkları açıklanır. Hatalı yazdırılmış kod blokları üzerinden öğrencilerin hatayı çalıştırmadan önce tahmin etmeleri ve düzeltmeleri istenir **(D3.3, KB3.3).** Yapay zekâ kodları ile öğrenci kodları; okunaklılık ve amaca uygunluk açısından karşılaştırılarak savunulur. Süreç boyunca öğrencilerin sistemli olma ve eleştirel düşünme tutumları gözlemlenir **(D16.3, KB2.17).** Değerlendirme için dereceli puanlama anahtarı (rubrik) kullanılabilir. **Karar Yapıları ve Döngüler:** Derse otonom araçların karar verme süreçleri tartışılarak başlanır. Banka PIN kontrolü gibi örnekler üzerinden if-elif-else yapıları modellenerek "Eğer... Değilse..." mantığının hayati önemi vurgulanır. Öğrencilerin sarmal projelerine (Kod-Asistan vb.) menü numarasına göre farklı modülleri tetikleyen bir "Akıllı Menü Sistemi" eklemeleri sağlanır **(E3.7)**. Döngülerin verimlilikteki rolü açıklanarak for ve while döngüleri arasındaki farklar örneklenir **(KB2.7)**. Sayı tahmin oyunu veya şifre giriş ekranı gibi uygulamalar üzerinden döngüyü sonlandıran break ve belirli bir adımı (örneğin hatalı girişi) atlayarak döngüyü sürdüren continue komutlarının işlevleri modellenir. Öğrencilerin asistanlarına, "ÇIKIŞ" komutuyla programı kapatan veya hatalı menü seçimlerini continue ile atlayıp kullanıcıyı tekrar menüye yönlendiren bir mekanizma eklemeleri desteklenir **(D3.2, KB2.4).** Değerlendirme için senaryo tabanlı değerlendirme araçları kullanılabilir. Arı peteği veya kar tanesi gibi doğadaki geometrik tekrarlar üzerinden grafiksel görselleştirmenin önemi tartışılır. Öğretmen tarafından belirlenen bir grafik kütüphanesi (örn. Turtle vb.) ile döngü sayısı ve dönüş açısı arasındaki matematiksel ilişki modellenir. İç içe döngülerle karmaşık geometrik desenlerin üretim süreci canlı kodlama ile gösterilir. Öğrencilerin asistan açılışı için döngüsel bir görsel bileşen tasarlamaları sağlanır **(OB7, KB2.16, MAB2.1).** Yazılım geliştirme sürecinde sürüm kontrolünün önemi vurgulanarak depoda "Release" oluşturma ve anlamlı commit mesajları yazma süreçleri modellenir. Yapay zekâ, tasarlanan kontrol yapılarındaki mantık hatalarını denetleyen bir kalite kontrolörü olarak konumlandırılır. Öğrencilerin v1.0 sürümlerini dijital depolara yüklemeleri sağlanır **(OB2, SDB1.2, D16.3).** Değerlendirme için performans görevi kullanılabilir. **Modüler Programlama ve Fonksiyonlar:** Derse, önceki modüllerde yazılan ve karmaşıklaşan sarmal projenin (Kod-Asistan vb.) kodları yansıtılarak başlanır. Fonksiyonların, kodun okunabilirliğini artıran "bağımsız makineler" olduğu vurgulanır. Parametresiz ve değer döndürmeyen basit fonksiyon tasarımları üzerinden Python

	standartları (def, girintileme) modellenir **(E3.7, KB2.4).** Değerlendirme için kontrol listesi kullanılabilir. Fonksiyonların esnekliğini anlatmak için kahve makinesi benzetmesi (Parametre: Seçim, Return: Kahve) kullanılır. Genel bir "Birim Dönüştürücü" uygulaması üzerinden fonksiyon tanımlama ve return ifadesinin veri aktarma rolü modellenir. Öğrencilerin sarmal projelerindeki selamlama, hesaplama ve menü süreçlerini bağımsız fonksiyonlara dönüştürmeleri sağlanır **(D3.2, KB2.13, SDB3.1).** Değerlendirme için performans görevi kullanılabilir. Koda ait değişkenlerin etki alanı (local/global) ve veri güvenliği uygulamalı olarak gösterilir. Tek parça halindeki yazılım projesini, modülerlik prensiplerine uygun olarak bağımsız modüllere dönüştürür **(E3.7, OB2).** Dijital depo (repository) üzerinde "Issue" açma ve yapay zekâ desteğiyle karmaşık kodları fonksiyonlara ayırma (refactoring) işlemleri modellenir. Öğrencilerin yapay zekâ önerilerini eleştirel bir süzgeçten geçirerek gerekçelendirmeleri ve profesyonel belgelendirme (Docstring) oluşturmaları desteklenir **(D3.3, KB2.17, SDB3.1, E3.6).** **Veri Yönetimi ve Kalıcı Program Yapıları:** Derse kütüphane fihristi görseli sunularak başlanır. Verilerin sistemli depolanmasının sürdürülebilirlik üzerindeki etkisi tartışılır **(E3.7).** Genel bir "Envanter" örneği üzerinden sözlük (dict) yapısı ve anahtar-değer erişim yöntemleri modellenir **(KB2.4, OB7).** Öğrencilerin sarmal projelerinde (Kod-Asistan vb.) kullanıcı tercihlerini tutan bir "Profil Sözlüğü" oluşturmaları sağlanır **(D3.2, OB2).** Basit bir not alma uygulaması üzerinden metin dosyasına veri yazma ve okuma işlemlerini gerçekleştirir. Dosya bulunamadığında programın çökmemesi için hata yönetim yöntemleri gösterilir. Öğrencilerin asistanlarına geçmiş komutları kaydeden bir "Hafıza Dosyası" eklemeleri desteklenir **(D16.3, E3.10, KB2.10).** Verilerin JSON formatında saklanmasının avantajları hava durumu simülasyonu örneğiyle modellenir. Yapay zekâ, veri formatı doğruluğunu denetleyen bir danışman olarak kullanılır. .gitignore dosyası kullanılarak veri bütünlüğü ve sürüm kontrolü dokümantasyon süreçleri açıklanır **(OB2, BTYAB6.2, KB2.13).** Değerlendirme için proje tabanlı değerlendirme kullanılabilir. **Dayanıklılık, Etik ve Sunum:** Derse etik değerler çerçevesinde hatasız ve güvenli kodun toplumsal faydası tartışılarak başlanır **(E1.1, SDB1.2).** Hatalı kullanıcı girişleri karşısında programın çöküşü ve try-except yapısı ile bu durumun yönetilmesi modellenir. Öğrencilerin sarmal projelerini (Kod-Asistan vb.) her türlü kullanıcı hatasına karşı dayanıklı hale getirmeleri sağlanır **(D16.3, KB2.16, E3.10).** Veri sızıntısı riskleri ve yapay zekâ modellerinin üretebileceği yanıltıcı bilgilerin denetlenme süreçleri açıklanır. Kullanıcı verilerinin (JSON dosyaları) korunması ve etik dürüstlük ilkeleri vurgulanır **(D6.2, OB2).** Profesyonel bir README dosyasının teknik şema ve ekran görüntülerini içermesi gerektiği örneklerle gösterilir. Öğrencilerin projelerini dijital depolarda final sürümüyle (v3.0) yayına almaları desteklenir. Sınıf içinde düzenlenen "Yazılım Lansmanı" ile projelerin sunumu ve savunması gerçekleştirilir **(SDB1.2, OB1).** Diğer öğrencilerin "Kod Denetçisi" rolüyle asistanın hata yönetimi ve etik duruşunu sorgulamaları sağlanır. Süreç boyunca sistemli olma ve eleştirel düşünme tutumları gözlemlenir **(D3.2, E3.7, E3.10, KB2.17).** Değerlendirme için dijital portfolyo kullanılabilir.

ÖĞRENME ÖĞRETME ORTAM VE DONANIMI	**Ortam:** Bilişim Teknolojileri Atölyesi **Donanım:** Etkileşimli tahta / projeksiyon, bilgisayar, akıllı tahta, yazılım programları vb. araç gereç sağlanmalıdır.

ÖĞRENME BİRİMİ / MODÜL VE SÜRE TABLOSU	ÖĞRENME BİRİMİ/MODÜL	ÖĞRENME BİRİMİ / MODÜL SÜRESİ
	Algoritmik Düşünme ve Tasarım	40/16
	Temel Programlama Yapıları	40/24
	Karar Yapıları ve Döngüler	40/32
	Modüler Programlama ve Fonksiyonlar	40/24
	Veri Yönetimi ve Kalıcı Program Yapıları	40/24
	Dayanıklılık, Etik ve Sunum	40/24
	TOPLAM	**240/144**

ÖĞRENME BİRİMİ / MODÜL	ÖĞRENME ÇIKTISI GRUBU VE AÇIKLAMALARI		İÇERİK ÇERÇEVESİ
Algoritmik Düşünme ve Tasarım	**Problemlere uygun algoritmik çözümler tasarlayabilme**	**Algoritmik düşünme sürecini problem senaryoları üzerinde yapılandırır** Günlük yaşam problemleri girdi–işlem–çıktı basamaklarına ayrılarak analiz edilir. Karmaşık problemler daha küçük ve yönetilebilir alt problemlere bölünür. **Çözüm algoritmaları için akış şemaları hazırlar.** Algoritmalar sözde kod (pseudo-code) ve standart akış şeması sembolleri ile modellenir. Kodlamaya geçmeden önce çözüm mantığı görsel olarak ifade edilir.	**Problem analizi ve bileşenleri** **Akış şeması (Flowchart)**
Temel Programlama Yapıları	**Temel programlama yapılarını kullanarak kullanıcı etkileşimli yazılımlar geliştirebilme**	**Değişkenleri ve sabitleri programlama standartlarına göre tanımlar** Değişken kavramı ve RAM üzerindeki çalışma mantığı incelenir. Temel veri türleri (int, float, str, bool) arasındaki farklar açıklanır. İsimlendirme kuralları (snake_case) ve sabit ayrımı uygulanır. **Giriş-çıkış fonksiyonlarını kullanarak kullanıcı etkileşimi sağlar** Programın kullanıcıyla olan etkileşim döngüsünü, input() ve print() fonksiyonları aracılığıyla veri standartlarına uygun şekilde yapılandırır.	**Değişkenler, Sabitler ve Veri Türleri** **Kullanıcı Etkileşimi ve Tür Dönüşümü** **Operatörler ve İşlem Önceliği** **Yapay Zekâ Kod Denetimi**

		type() fonksiyonu ile veri türü incelemeleri yaptırılır. Tür dönüşümleri (casting) ve olası hataların (ValueError) çözümü sağlanır. **Operatörleri ve işlem önceliğini kullanarak verileri işler** Aritmetik operatörler ve parantez kullanımı ile işlem önceliği kuralları uygulatılır. Değişkenler üzerinden yapılan hesaplamaların doğruluğu denetlenir. **Yapay zekâ tarafından üretilen kodları insan denetimi prensibiyle doğrular.** Yapay zekâ destekli kod üretiminin kısıtlılıkları açıklanır. Önerilen çözümler "insan denetimi" (human-in-the-loop) prensibiyle doğrulanır.	
Karar Yapıları ve Döngüler	**Kontrol yapılarını kullanarak program akışını yönetebilme**	**Koşul yapılarını kullanarak program akışını yönetir** if – elif – else yapıları kullanılarak farklı senaryolara göre çıktılar üretilir. Karşılaştırma ve mantıksal operatörlerin birleşik kullanımı açıklanır. Karar yapılarının mantıksal doğruluk kontrolleri sağlanır. **Tekrarlı işlemleri yönetmek için döngü yapılarını ve liste ilişkisini kullanır** for ve while döngüleriyle range() fonksiyonu yönetilir. break ve continue ifadeleri ile sonsuz döngü riskleri çözümletilir. Liste elemanlarına döngü ile erişim süreçleri gerçekleştirilir. **Algoritma süreçlerini grafik kütüphaneleri ile görselleştirir** Seçilen bir grafik kütüphanesi (Turtle, Pygame vb.) komutları ile döngüler arasındaki ilişki kurulur. Tekrar eden geometrik desenlerin döngüsel mantıkla üretilmesi sağlanır.	**Koşul Yapıları ve Mantıksal Operatörler** **Döngü Yapıları ve Liste İşlemleri** **Grafiksel Görselleştirme**
Modüler Programlama ve Fonksiyonlar	**Tekrar eden kodları fonksiyonlar kullanarak modüler ve sürdürülebilir yapılara dönüştürebilme**	**Fonksiyon yapılarını ve veri akışını standartlara göre tasarlar** def anahtar kelimesi, isimlendirme standartları ve girintileme kuralları gösterilir. Parametre yönetimi ve return ifadesinin veri aktarma rolü çözümlenir.	**Fonksiyon Tanımlama ve Veri Akışı** **Modüler Mimari ve Değişken Etki Alanı** **Kod Kalitesi ve Denetimi**

		Modüler mimariyi ve değişken etki alanını yönetir Yerel (local) ve küresel (global) değişken farkları ile veri güvenliği sağlanır. Farklı dosyaların içe aktarılması (import) ile kodun modüler parçalara ayrılması işlemleri yaptırılır. **Yapay zekâ ve dijital araçlarla kod kalitesini denetler** Dijital depo üzerinden "Issue" takibiyle hata yönetimi yaptırılır. Yapay zekâ ile kod iyileştirme (refactoring) ve teknik dokümantasyon (Docstring) işlemleri yaptırılır.	
Veri Yönetimi ve Kalıcı Program Yapıları	**Gelişmiş veri yapılarını ve dosya sistemlerini kullanarak verileri organize edebilme**	**Veri yapılarını (liste, sözlük) verileri gruplandırmak için kullanır** Sözlük (dict) yapısı, anahtar-değer (key-value) ilişkisi ve veri organizasyonu inceletilir. Listeler ve sözlüklerin iç içe kullanımı ile veri modelleme işlemleri yaptırılır. **Dosya sistemlerini kullanarak verileri kalıcı olarak saklar** Metin (txt) ve yapılandırılmış (json) dosyalar üzerinden okuma ve yazma işlemleri yaptırılır. with\ open() yapısı ve dosya yolu (path) yönetimi ile veri kalıcılığı sağlanır. **Yapay zekâ desteğiyle veri yönetimini tasarlayarak iyileştirir** Verilerin JSON formatına dönüştürülmesi ve dosya hatalarının önlenmesinde yapay zekâ danışman olarak kullanılır. Veri bütünlüğü ve sürüm kontrolü için dokümantasyon işlemleri yaptırılır.	**Gelişmiş Veri Yapıları (Dictionary)** **Dosya İşlemleri (I/O)** **Veri Formatları ve Denetim**
Dayanıklılık, Etik ve Sunum	**İstisna yönetimi tekniklerini kullanarak program dayanıklılığını optimize edebilme**	**İstisna yönetimi ve hata ayıklama teknikleriyle dayanıklı kod geliştirir** try-except-finally blokları ile çalışma zamanı hataları (Runtime Errors) yönetir. Traceback mesajları analiz edilerek mantıksal hatalar giderilir ve programın çökmemesi sağlanır. **Yazılım etiği, veri güvenliği ve AI güvenliği prensiplerini uygular** Kullanıcı girdilerinin doğrulanması (sanitization) ve veri gizliliği sağlanır.	**İstisna Yönetimi ve Hata Ayıklama (Debugging)** **Bilişim Etiği ve Veri Güvenliği** **Dijital Portfolyo ve Proje Sunumu**

		Yapay zekâ çıktılarında oluşabilecek telif, önyargı (bias) ve etik dürüstlük ihlallerini çözümler. **Yazılım ürününü teknik ve etik boyutlarıyla sunar** Dijital depo üzerinde final sürüm (Release v3.0) oluşturulur. Proje mimarisi ve çözüm aşamaları teknik bir sunumla açıklanır.	

UYGULAMA FAALİYETLERİ/TEMRİNLER		
Uygulama faaliyeti/temrinler; ders öğrenme 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**
Algoritmik Düşünme ve Tasarım	Bir otomat makinesinin çalışma mantığını girdi-işlem-çıktı tablosuyla analiz etmek. Analiz edilen otomat sürecini standart akış şeması sembolleriyle kağıt üzerinde çizmek.	E1.1. Merak E3.10. Eleştirel Bakma OB2. Dijital Okuryazarlık D3.3. Araştırmacı ve sorgulayıcı olmak D16.3. Görev Bilincine Sahip Olmak D19.2. Millî kimliğini tanımak KB2.4. Çözümleme Becerisi KB2.8. Sorgulama Becerisi KB2.16. Muhakeme (Akıl Yürütme) Becerisi KB2.19. Mantıksal Denetleme Becerisi BTYAB2.1. Algoritmik Düşünme
Temel Programlama Yapıları	Kod-Asistan için "Kullanıcı Kayıt Etiketleri" (isim, yaş, boy) ve isimlendirme standartlarını oluşturmak. input() verilerini int/float tipine dönüştürerek "Yaş Hesaplayıcı" modülünü kodlamak. İşlem önceliğine dikkat ederek "Vücut Kitle İndeksi" hesaplama fonksiyonunu yazmak. Yapay zekâ tarafından üretilen hatalı bir kod bloğundaki "tür hatasını" bulup düzeltmek.	KB2.4. Çözümleme Becerisi OB2. Dijital Okuryazarlık E3.7. Sistematiklik OB7. Veri Okuryazarlığı D3.2 Planlı Olmak D3.3. Araştırmacı ve sorgulayıcı olmak

		KB3.3 Eleştirel Düşünme Becerisi D16.3. Görev Bilincine Sahip Olmak KB2.17. Değerlendirme Becerisi
Karar Yapıları ve Döngüler	Kullanıcı girişine göre farklı tepkiler veren "Akıllı Menü Karar Yapısı"nı oluşturmak. "Günlük Görev Listesi"ndeki elemanları for döngüsüyle numaralandırarak ekrana yazdırmak. Grafik kütüphanesi (Turtle vb.) ve döngüler kullanarak sarmal bir logo tasarımı yapmak.	E3.7. Sistematiklik D3.2 Planlı Olmak D16.3. Görev Bilincine Sahip Olmak OB2. Dijital Okuryazarlık OB7. Veri Okuryazarlığı KB2.4. Çözümleme Becerisi KB2.7. Karşılaştırma Becerisi KB2.16. Muhakeme (Akıl Yürütme) Becerisi MAB2.1. Matematiksel Çözümler Geliştirme SDB1.2. Kendini Düzenleme (Öz Düzenleme Becerisi)
Modüler Programlama ve Fonksiyonlar	Projedeki selamlama ve hesaplama adımlarını parametre alan ve return dönen fonksiyonlara taşımak. Kodları "arayuz.py" ve "hesaplamalar.py" şeklinde ayırıp import komutuyla birleştirmek. Fonksiyonlar için YZ desteğiyle Docstring dökümantasyonu yazma ve Issue takibi yapmak.	E3.7. Sistematiklik KB2.4. Çözümleme Becerisi D3.2 Planlı Olmak KB2.13. Yapılandırma Becerisi SDB3.1. Uyum Becerisi E3.7. Sistematiklik OB2. Dijital Okuryazarlık D3.3. Araştırmacı ve sorgulayıcı olmak KB2.17. Değerlendirme Becerisi E3.6. Analitiklik
Veri Yönetimi ve Kalıcı Program Yapıları	Kullanıcı tercihlerini (tema, dil, ad) tutan iç içe geçmiş bir Profil Sözlüğü (dict) kurgulamak. Profil verilerini program kapandığında kaybolmaması için JSON dosyasına yazma ve okumak. YZ yardımıyla karmaşık veri setlerini JSON formatına dönüştürme ve .gitignore yapılandırmak.	E3.7. Sistematiklik KB2.4. Çözümleme Becerisi OB7. Veri Okuryazarlığı D3.2 Planlı Olmak

		OB2. Dijital Okuryazarlık D16.3. Görev Bilincine Sahip Olmak E3.10. Eleştirel Bakma KB2.10. Çıkarım Yapma KB2.13. Yapılandırma Becerisi BTYAB6.2. Dijital Güvenlik Önlemlerini Uygulama
Dayanıklılık, Etik ve Sunum	Sayı beklendiği yere metin girilmesi gibi durumlara karşı try-except blokları eklemek. Kullanıcı verilerinin gizliliğini (GDPR/KVKK farkındalığı) sağlayan "Veri Güvenliği Kontrolü" yapmak. Tüm gelişim sürecini kapsayan v3.0 final sürümünü dijital portfolyo (GitHub) üzerinden savunmak.	E1.1. Merak SDB1.2. Kendini Düzenleme (Öz Düzenleme Becerisi) D16.3. Görev Bilincine Sahip Olmak KB2.16. Muhakeme (Akıl Yürütme) Becerisi E3.10. Eleştirel Bakma D6.2. Doğru ve Güvenilir Olmak OB2. Dijital Okuryazarlık OB1. Bilgi Okuryazarlığı D3.2 Planlı Olmak E3.7. Sistematiklik KB2.17. Değerlendirme Becerisi

DERSİN UYGULANMASINA İLİŞKİN AÇIKLAMALAR

Her öğrencinin uygulama yapması için ortam oluşturulmalıdır.
Uygulama faaliyetlerinde iş sağlığı ve güvenliğine ilişkin risk ve tehlike oluşturacak her türlü duruma karşı tedbirler alınmalıdır.
Bu dersin işlenişi sırasında (sorumluluk) israf etmeme, okuluna ve çevresine karşı sorumlu olma, emanete sahip çıkma, emeğe saygı duyma, birlikte iş yapabilme, paylaşma 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 kullanılabilir.
Anlatımdan ve örnek çalışmalardan sonra, dersin öğrenme çıktılarının öğrencide pekiştirilmesi amacıyla birden fazla uygulama faaliyeti yapılmalıdır.
Ahilik kültürüne ilişkin değerler ile programlar arası bileşenler altındaki “Değerler” eşleştirilerek öğrenme ve öğretme faaliyetleri içinde öğrenciye verilmelidir.
Öğrencilerden güvenlik gözlemlerini paylaşmaları istenir.