

DERS BİLGİ FORMU

DERSİN ADI	PROGRAMLAMA TEMELLERİ			
DERSİN SINIFI	9. Sınıf			
DERSİN SÜRESİ	Haftalık 4 Ders Saati			
DERSİN AMACI	Bu derste öğrenciye; temel algoritma, programlama işlemlerini yapma ile ilgili bilgi ve becerilerin kazandırılması amaçlanmaktadır.			
DERSİN ÖĞRENME KAZANIMLARI	<ol style="list-style-type: none">1. Problem ve problem çözme ile ilgili kavramları açıklar ve bir problemi çözebilmek için gerekli becerileri kazanarak algoritma ve akış şemaları hazırlar.2. Çeşitli alanlarda uygulama geliştirebilmek amacıyla kullanılan programlama dili yazılımını kurar ve açıklar.3. Değişken, sabit ve operatörleri kullanarak farklı veri tipleriyle listeler oluşturup basit kod parçaları yazar.4. Karar yapılarını açıklayıp, karar-kontrol yapılarını kullanarak basit kodları yazar. Döngü kavramını açıklayarak döngü türlerini karşılaştırır ve döngü işlemlerini yapar.5. Fonksiyon kavramını ve programlama dili içerisinde nasıl tanımlanıp kullanılacağını öğrenir. Kendi tanımladığı fonksiyonlara parametre gönderip geri dönüşleri işler.6. Programlama dili içerisinde tarih ve metin veri tipleri ile çalışmayı öğrenir. Metin veri tipi değişkenler içinde değişiklik yapmayı öğrenir.7. Hata ve hata yakalama kavramlarını bilir, programda karşılaşılabileceği hatalara uygun çözümler üretir.8. Programlama dili ile dosya oluşturma, var olan dosyaya erişme ve okuma becerileri kazanır. Dosya silme ve yedekleme konularında bilgi sahibi olur.			
EĞİTİM-ÖĞRETİM ORTAM VE DONANIMI	Ortam: Bilişim Teknolojileri laboratuvarı, Donanım: Akıllı tahta/projeksiyon, bilgisayar, yazıcı/tarayıcı			
ÖLÇME VE DEĞERLENDİRME	Bu derste; öğrenci performansı belirlemeye yönelik çalışmalar değerlendirilirken gözlem formu, derecelendirme ölçeği ve dereceli puanlama anahtarı gibi ölçme araçlarından uygun olanlar seçilerek kullanılabilir. Bunun yanında öz değerlendirme ve akran değerlendirme formları kullanılarak öğrencilerin, öğretimin süreç boyutuna katılmaları sağlanabilir.			
KAZANIM SAYISI VE SÜRE TABLOSU	ÖĞRENME BİRİMİ	KAZANIM SAYISI	DERS SAATİ	ORAN (%)
	Problem Çözme ve Algoritmalar	6	16	11,1
	Programlama Dili Temelleri	3	8	5,7

	Veri Yapıları	4	28	19,4
	Karar Ve Döngü Yapıları	2	32	22,2
	Fonksiyonlar	2	28	19,4
	Tarih Ve String İşlemleri	3	12	8,3
	Hata Yakalama İşlemleri	3	8	5,6
	Dosya İşlemleri	5	12	8,3
TOPLAM		28	144	100

ÖĞRENME BİRİMİ	KONULAR	ÖĞRENME BİRİMİ KAZANIMLARI ve KAZANIM AÇIKLAMALARI
Problem Çözme ve Algoritmalar	<ul style="list-style-type: none"> • Problem çözme ve temel kavramlar • Problem Türleri • Problem Çözmede Temel İşlemler • Algoritmalar • Algoritma Hataları • Akış Diyagramları 	<ol style="list-style-type: none"> 1. Problem çözme sürecindeki temel kavramları açıklar. <ul style="list-style-type: none"> • Problem kavramı açıklanır. • Problem çözme süreci açıklanır. 2. Problem türlerini açıklar. <ul style="list-style-type: none"> • Günlük hayatta karşılaşılabilecek problemler açıklanır. • Bilgisayarlar aracılığıyla çözülebilecek problemler açıklanır. 3. Verilen problem için uygun teknikleri kullanarak çözüm bulur. <ul style="list-style-type: none"> • Problem çözmek için kullanılabilecek bazı yöntemler açıklanır. • Problem çözme adımlarını kullanarak bir problemin nasıl çözülebileceği açıklanır. 4. Verilen problemi çözmek üzere farklı algoritmalar tasarlar. <ul style="list-style-type: none"> • Operatörler açıklanır. • İşlem önceliği açıklanır. • Algoritma kavramı açıklanır. • Farklı algoritma örnekleri verilerek çözümün nasıl yapılacağı açıklanır. 5. Algoritmanın hatalarını giderir. <ul style="list-style-type: none"> • Problem çözmede algoritma hataları ve hata giderme anlatılır. 6. Verilen problemin çözümüne uygun akış şemaları oluşturur. <ul style="list-style-type: none"> • Flowchart(akış diyagramı) programının masaüstü kurulumu gösterilir. • Flowchart programının çevrimiçi kullanımı anlatılır. • Flowchart programı kullanarak akış şemaları hazırlatılır. • Doğrusal akış şemalarının nasıl hazırlanacağı örneklerle gösterilir. • Karar ifadeleri içeren akış şemalarının nasıl hazırlanacağı örneklerle gösterilir. • Döngü içeren akış şemalarının nasıl hazırlanacağı örneklerle gösterilir.

Programlama Dili Temelleri	<ul style="list-style-type: none">• Program ve Yazılım• Python Kurulumu• Python İçin Gerekli Araçlar	<ol style="list-style-type: none">1. Programlama dilinin özelliklerini ve diğer programlama dillerinden farklarını açıklar.<ul style="list-style-type: none">• Popüler programlama dilleri listelenir.• Popüler programlama dillerinin benzerlikleri/farklılıkları açıklanır.• Kullanılacak olan programlama dilinin diğer dillerden farkları açıklanır.2. Programlama dilini bilgisayara kurar.<ul style="list-style-type: none">• Programlama dilinin programının işletim sistemine uygun versiyonu bilgisayara kurulmasını açıklar.3. Programlama dili kullanmak için gerekli araçları kurar.<ul style="list-style-type: none">• Uygun bir editör seçilir ve bilgisayara kurulur.• Gerekli editör ayarlarını yaptırır.• Kütüphane kavramını açıklar ve gerekli kütüphanelerin kurulumunu yaptırır.
-----------------------------------	--	--

<p>Veri Yapıları</p>	<ul style="list-style-type: none"> • Değişken, Sabit ve Operatörler • Veri Tipleri • Veri Tipi Dönüşümleri • Farklı veri tipleri 	<ol style="list-style-type: none"> 1. Programlama dilinde, değişken, sabit ve operatörleri kullanır. <ul style="list-style-type: none"> • Değişken ve sabit açıklanır. • Değişken isimlendirme kuralları açıklanır. • Operatörler açıklanır. 2. Programlama dilinde veri tiplerini amacına uygun kullanır. <ul style="list-style-type: none"> • Karakter türündeki veriler açıklanır. • Sayısal türündeki veriler açıklanır. 3. Programlama dilinde tanımladığı veriye ait temel fonksiyonların yer aldığı programları geliştirir. <ul style="list-style-type: none"> • Input() fonksiyonu ile kullanıcıdan veri aldırır. • Tür dönüşümü yaptırır. • Basit uygulamalar yaptırır. 4. Farklı veri tiplerini (listeler, sözlükler vb.) kullanarak programlar geliştirir. <ul style="list-style-type: none"> • Liste veri tipi açıklanır. • Tuples veri tipi açıklanır. • Dictionary veri tipi açıklanır. • Sets veri tipi açıklanır.
<p>Karar Ve Döngü Yapıları</p>	<ul style="list-style-type: none"> • Karar Yapıları • Döngüler 	<ol style="list-style-type: none"> 1. Kontrol yapılarını kullanarak programlar geliştirir. <ul style="list-style-type: none"> • Karar kontrol deyimi açıklanır. • Karar kontrol deyimlerini kullanır. 2. Tekrarlı yapıları kullanarak programlar geliştirir. <ul style="list-style-type: none"> • Döngü kavramı açıklanır. • Döngüleri kullanır.
<p>Fonksiyonlar</p>	<ul style="list-style-type: none"> • Fonksiyonlar ve fonksiyon tanımlama işlemleri • Fonksiyon türleri 	<ol style="list-style-type: none"> 1. Programlama dilinde fonksiyonları kullanır. <ul style="list-style-type: none"> • Fonksiyon kavramı açıklanır. • Fonksiyon kullanır. • Fonksiyon kavramı açıklanır. • Fonksiyon kullanır. • Kendi fonksiyonunu tanımla işlemi açıklanır. • Tanımladığı fonksiyonu tanımlar ve düzenletir. • Parametre kavramı açıklanır. • Fonksiyonlarda parametreleri kullanır. • Değer döndürme kavramı açıklanır. • Değer döndüren fonksiyon yaptırır. 2. Fonksiyon türlerine göre programlar geliştirir. <ul style="list-style-type: none"> • Lambda fonksiyon kavramı açıklanır. • Lambda fonksiyonu kullanır. • Özyinelemeli fonksiyon açıklanır. • Özyinelemeli fonksiyon yazdırılır. • Fonksiyonların yerel ve genel değişkenler ile etkileşimi hakkında bilgi verir.

Tarih Ve String İşlemleri	<ul style="list-style-type: none"> Tarih Nesnesi Tarih Bilgisinin Biçimlendirilmesi String (Metin) İşlemleri 	<ol style="list-style-type: none"> Tarih nesnesi oluşturur. <ul style="list-style-type: none"> Tarih nesnesi kullanılır. Tarih bilgisini biçimlendirir. <ul style="list-style-type: none"> Tarih bilgisi biçimlendirme kriterleri açıklanır. Metin bilgisini biçimlendirir. <ul style="list-style-type: none"> Metin bilgisi biçimlendirme kriterleri açıklanır
Hata Yakalama İşlemleri	<ul style="list-style-type: none"> Hata kavramı ve Hata Türleri Hata Yakalama Python Hata Türleri 	<ol style="list-style-type: none"> İstisna işlemlerini açıklar. <ul style="list-style-type: none"> İstisna kavramı açıklanır. Try-except bloklarını kullanır. <ul style="list-style-type: none"> Try-except blokları kullanılır. Finally bloğunu kullanır. <ul style="list-style-type: none"> Finally bloğu kullanılır.
Dosya İşlemleri	<ul style="list-style-type: none"> Çalışma Dizini Ayarları ve Klasör Oluşturma Dosya Oluşturma ve Yazma Dosya Silme ve Yedekleme 	<ol style="list-style-type: none"> Programlama dilinde dosya okuma işlemlerini yapar. <ul style="list-style-type: none"> Var olan bir metin dosyası açılır. Var olan bir metin dosyasını okuyup içeriğini ekrana yazdırır. Dosya oluşturma ve yazma işlemlerini yapar. <ul style="list-style-type: none"> Yeni bir metin dosyası oluşturma işlemi yaptırır. Dosyaya veri yazma işlemi yaptırılır. Dosyadaki verilerin okunup değiştirilmesi sağlanır. Dosya silme ve yedekleme işlemlerini yapar. <ul style="list-style-type: none"> Dosya silme işlemlerini gerçekleştirmesi sağlanır. Dosya yedekleme işlemi açıklanır.
UYGULAMA FAALİYETLERİ/TEMRİNLER		
<p>Uygulama faaliyeti/temrinler; ders kazanımına 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.</p>		

Problem Çözme ve Algoritmalar

1. Sabahları okula geç kalma problemini, problem çözme adımlarını kullanarak çözmek.
2. Bir problemin çözümü sırasında planlamanın önemini kavratmak.
3. Farklı çözüm yolları olan, deneme yanılma yöntemi kullanan örnek çözmek.
4. Operatörlere yönelik işlemleri yapmak.
5. İşlem önceliğine göre verilen işlemleri sonuçlandırmak.
6. Algoritma yoluyla problem çözmek.
7. Sade Türk kahvesi algoritmasını hazırlamak.
8. Çay demleme algoritmasını hazırlamak.
9. Girilen sayının karesini bulan algoritmayı metinsel ve sözde kod kullanarak hazırlamak.
10. İki sayıyı çarpıp sonucu ekrana yazdıran algoritmayı metinsel ve sözde kod kullanarak hazırlamak.
11. Bir öğrencinin matematik dersinden aldığı iki notun ortalamasını hesaplayan algoritmayı metinsel ve sözde kod kullanarak yazmak.
12. Dik üçgenin alanını hesaplayan metinsel algoritmayı ve sözde kodu yazmak.
13. Klavyeden girilen iki sayının büyük olanından küçük olanını çıkaran algoritmayı yazmak.
14. Suyun sıcaklığına göre maddenin katı, sıvı veya gaz olma durumunu gösteren algoritmayı hazırlamak.
15. Klavyeden girilen yaş değeri 18'den büyük ve eşitse **"Reşittir"**, aksi halde **"reşit değildir"** yazan algoritmayı hazırlamak.
16. Ekrana 5 defa **"merhaba"** yazdıran algoritmayı yazmak.
17. 0'dan 100'e kadar olan çift sayıları ekrana yazdıran algoritmayı hazırlamak.
18. Klavyeden girilen 5 adet sayıdan 20'den küçük olanların sayısını gösteren algoritmayı hazırlamak.
19. Problem çözmeye algoritma hatalarını gidermek.
20. Hatalı verilmiş olan "el yıkama algoritmasını" hatalarını gidererek doğrusunu yazmak.
21. Sınıfınızda üçer kişilik gruplar oluşturup, herkes ayrı ayrı ATM'den para çekme algoritmasını yazdıktan sonra, her grupta algoritmaları birlikte inceleyerek hatalarını gidermeye çalışmak.
22. Dikdörtgenin alanını hesaplayan algoritmayı ve akış şemasını hazırlamak.
23. Klavyeden girilecek 3 sayıyı toplayıp sonucu ekrana yazdıran programın algoritmasını ve akış şemasını hazırlamak.
24. Dikdörtgenin çevresini hesaplayan algoritmayı ve akış şemasını hazırlamak.
25. Klavyeden girilen bir sayının negatif mi, pozitif mi yoksa sıfır mı olduğunu yazdıran programın algoritmasını ve akış şemasını hazırlamak.
26. Klavyeden girilen iki sayıdan birincisi büyük ise toplama, ikincisi büyük ise çarpma işlemi yapan algoritmayı ve akış şemasını hazırlamak.
27. Klavyeden girilen iki sayıdan büyük olanı ekrana yazdıran algoritmayı ve akış şemasını hazırlamak.
28. 20 Öğrencinin Programlama temelleri dersi birinci sınav notları giriliyor. Geçme notu 60 olan sistemde, kalan öğrenci sayısını bulan algoritmayı ve akış şemasını hazırlamak.
29. Klavyeden girilen 10 sayının toplamını hesaplayan algoritmayı ve akış diyagramını hazırlamak.
30. Sıfır girilinceye kadar girilmiş olan sayıların karesini hesaplayan algoritmayı ve akış şemasını hazırlamak.

Programlama Dili Temelleri	<ol style="list-style-type: none"> 1. Program, yazılım ve programlama dili kavramlarını anlatan bir sunum hazırlamak. 2. Yorumlama ve derleme kavramlarını karşılaştıran bir sunum hazırlamak. 3. Python dili özelliklerini anlatan bir sunum hazırlamak. 4. Python dilini bilgisayara kurmak. 5. IDLE / Shell ekranında aritmetik işlemler gerçekleştirmek. 6. IDLE / Shell ekranı üzerinden kod dosyası oluşturmak ve buraya yazılan kodları çalıştırmak. 7. PyCharm editörünü kurmak ve bu editörde kod çalıştırmak. 8. Visual Studio Code editörünü kurmak ve bu editörde kod çalıştırmak. 9. PIP ile bilgisayara paket yüklemek ve bu paketleri program içinde kullanmak.
Veri Yapıları	<ol style="list-style-type: none"> 1. Değişken tanımlamalarını doğru ya da yanlış olarak değerlendirmek. 2. Tanımlanan değişkeni ekrana yazdırmak. 3. Dikdörtgenin alanını hesaplamak. 4. İşlem önceliğine göre verilen işlemleri sonuçlandırmak. 5. Operatörlere yönelik işlemleri yapmak. 6. Değişken veri tiplerini bulmak. 7. Metinsel ifadeleri birbirine bağlamak. 8. Metinsel ifadeleri tekrarlayarak yazdırmak. 9. Int ve float değişkenleri ekrana yazdırmak. 10. Girilen üç notun ortalaması hesaplamak. 11. Verilen ifadeleri bool veri tipine uygun olarak doğru ve yanlış olarak değerlendirmek. 12. Kullanıcının girdiği yaşı ekrana yazdırmak 13. Kullanıcının girdiği üç notun ortalamasını hesaplamak. 14. Kullanıcının girdiği kenar uzunluklarına göre dikdörtgenin alanı ve çevresini bulmak. 15. Kullanıcının girdiği doğum tarihine göre yaş hesaplamak. 16. Kullanıcının girdiği adı ve okuduğu kitap sayısını ekrana yazdırmak. 17. Veri tiplerini dönüştürmek. 18. Liste oluşturmak ve liste fonksiyonlarını kullanmak. 19. İç içe liste oluşturmak. 20. Demet oluşturmak ve fonksiyonlarını kullanmak. 21. Sözlük oluşturmak ve fonksiyonlarını kullanmak. 22. Set oluşturmak ve fonksiyonlarını kullanmak.
	<ol style="list-style-type: none"> 1. Girilen yaşa göre kişinin ehliyet alıp alamayacağını belirlemek. 2. Kullanıcının girdiği sayının tek mi çift mi olduğunu bulmak. 3. Girilen üç notun ortalamasına göre başarı durumunu bulmak. 4. Girilen üç açıya göre şeklin üçgen olup olmadığını değerlendirmek.

Karar ve Döngü Yapıları	<ol style="list-style-type: none"> 5. Belirlenen bagaj hakkı kurallarına göre kişinin ödeyeceği ücreti belirlemek. 6. Kullanıcının girdiği ürün fiyatlarına ve belirlenen kurallara göre indirim miktarını bulmak. 7. Liste içerisindeki elemanı bulmak. 8. Belirlenen işe alma kurallarına göre kişinin iş başvurusu sonucunu değerlendirmek. 9. Girilen kullanıcı adı ve şifrenin doğru olup olmadığını kontrol etmek. 10. Girilen sayının hem 3 hem de 5'e kalansız bölünüp bölünmeyeceğini belirlemek. 11. Girilen sayının pozitif, negatif ya da sıfır olma durumunu belirlemek. 12. Girilen iki sayı ve operatöre göre aritmetiksel işlemleri yapmak. 13. Girilen yaşa göre kişinin yaş grubunu belirlemek. 14. Girilen hava sıcaklığına göre hava durumunu ekrana yazdırmak. 15. Girilen kenar uzunluğuna göre üçgen türünü belirlemek. 16. Vücut kitle indeksi hesaplayarak kişinin obez olma durumunu bulmak. 17. Girilen kıdem yılına göre maaş zammı uygulamak. 18. Girilen üç sayının en büyüğünü bulmak. 19. Belirlenen aralıktayı sayıları döngü kullanarak ekrana yazdırmak. 20. Belirlenen aralıktayı sayıları artırarak ya da azaltarak ekrana yazdırmak. 21. Belirlenen aralıktayı tek ve çift sayıları döngü kullanarak ekrana yazdırmak. 22. Belirlenen aralıktaki sayıların toplamını bulmak. 23. Girilen sayının faktoriyelini hesaplamak. 24. Listedeki elemanları döngü kullanarak toplamak. 25. Metinsel ifadeyi harf harf ekrana yazmak. 26. Liste içerisinde 3'e tam bölünen sayıları bulmak. 27. Girilen metin içinde aranan harfin sayısını bulmak. 28. Belirlenen metni döngü ile ekrana yazdırmak. 29. Sonsuz döngüyü kullanmak ve sonlandırmak. 30. Girilen sayı 0 olana kadar girilen tüm sayıları toplamak. 31. İç içe döngü kurarak çarpım tablosu uygulaması yapmak. 32. Break ve continue deyimlerini kullanmak.
Fonksiyonlar	<ol style="list-style-type: none"> 1. Gömülü fonksiyonları çağırıp kullanmak. 2. Programdan modül içeriğindeki gömülü fonksiyonları çağırıp kullanmak. 3. Bir modüldeki tüm fonksiyonlara erişmek. 4. Kendi fonksiyonumuzu tanımlamak. 5. Tanımladığımız fonksiyonu çağırarak ve kullanmak. 6. Tanımladığımız fonksiyonu birden fazla çağırarak. 7. Tanımladığımız fonksiyonu kullanıcıdan bir isim bilgisi isteyip ekrana yazdıracak şekilde geliştirmek 8. While döngüsü kullanarak ekrana 10'a kadar olan sayıları yazan fonksiyonu yazmak ve kullanmak. 9. For döngüsü kullanarak ekrana 10 kere "Merhaba Arkadaşlar!" metnini yazdırmak. 10. Döngü kullanarak ekrana, " * " karakterinden oluşan yıldız şeklini çizen fonksiyonu yazıp çalıştırmak. 11. Önceki örnekte çizilen yıldız tersten çizerek şekilde fonksiyonu düzenlemek. 12. Verilen bir ad değerini parametre olarak alıp ekrana parametreyi yazdıran fonksiyonu tanımlayıp kullanmak. 13. Fonksiyonlarda varsayılan parametre kullanmak. 14. Fonksiyona parametre gönderirken parametre adı ile atama yapmak. 15. Fonksiyona parametre olarak liste göndermek. 16. Daha önce yazılan yıldız üçgen çizme fonksiyonunu satır sayısını dışarıdan parametre olarak alacak şekilde düzenlemek. 17. Kenar uzunluklarını parametre olarak alıp döngü alan hesabı yapan fonksiyonu tanımlayıp kullanmak. 18. Parametre olarak gönderilen sayının tek mi çift mi olduğunu bulan fonksiyonu yazıp kullanmak.

	<ol style="list-style-type: none"> 19. Şekilde verilen yıldız üçgen şeklini çizen fonksiyonu satır sayısı parametre olarak alınacak şekilde yazmak. 20. Aldığı parametrelerle sınav ortalaması hesaplayıp ekrana yazan fonksiyonu yazıp kullanmak. 21. Ortalama bulan fonksiyonu sonucu ekrana yazmak yerine değer olarak geri döndürecek şekilde düzenlemek. 22. Verilen sayının tam bölenlerini bulan fonksiyonu yazıp kullanmak. 23. Verilen sayıları seçilen işleme göre toplayan, çıkaran, çarpan yada bölen hesap makinesi programını yazmak. 24. Parametre olarak gönderilen sayının asal olup olmadığını bulan fonksiyonu yazıp kullanmak. 25. Bir ürünün kdv eklenmiş fiyatını hesaplayan lambda fonksiyonu yazmak ve kullanmak. 26. Lambda fonksiyonları birden fazla argümanla kullanmak. 27. Lambda fonksiyonu bir başka fonksiyon içinde kullanmak 28. Tümevarım hesabı yapan fonksiyonu özyinelemeli olarak yazıp kullanmak. 29. Fibonacci dizisinin ilk 20 elemanını veren fonksiyonu özyinelemeli olarak yazmak. 30. Parametre olarak verilen sayının faktoriyelini hesaplayan özyinelemeli fonksiyonu yazmak ve kullanmak. 31. Yerel ve genel değişkenlerin kapsamını örneklerle incelemek. 32. Fonksiyon içerisinde genel değişken ataması yapmak. 33. Verilen özyinelemeli fonksiyonda bulunan hatayı çözerek hata ayıklamak. 34. Öğrendiğimiz teknikleri kullanarak adam asmacı oyunu yazmak.
Tarih Ve String İşlemleri	<ol style="list-style-type: none"> 1. Datetime modülünü kullanarak ekrana tarih ve saati yazdırmak. 2. today() fonksiyonu ile kрана sadece günün tarihini yazdırmak. 3. today() fonksiyonu ile alınan tarih bilgisinin bileşenlerini ekrana yazdırmak. 4. Tarih bilgisi vererek tarih nesnesi tanımlamak. 5. Tarih ve saat bilgisi vererek tarih nesnesi tanımlamak. 6. Tarih nesnesini kullanıcının klavyeden girdiği nesnelerle tanımlamak. 7. Tarih nesnesine zaman ekleyip çıkararak tarih hesaplaması yaptırmak. 8. Tarih nesnesini biçimlendirerek ekrana yazdırmak. 9. Tarih biçimlendirme ayarlarını dillere göre ayarlamak. 10. Tarih nesnesini Türkçe dil kurallarına göre biçimlendirmek ve ekrana yazdırmak. 11. Metin olarak girilen tarih bilgisini strptime() fonksiyonu ile tarih nesnesine çevirmek. 12. Verilen metni ABD formatında tarih olarak tarih nesnesine aktarmak. 13. String verileri birleştirmek. 14. Bir string veri içerisinde istenen karaktere erişmek. 15. String verinin uzunluğunu bularak, bu değeri döngü ile kullanmak. 16. slice() fonksiyonu ile string veriden parça almak. 17. split() fonksiyonu ile string veriyi parçalamak. 18. String veri içindeki bir karakteri replace() fonksiyonu ile değiştirmek. 19. String veri içinden strip() fonksiyonu ile karakter çıkarmak. 20. join() fonksiyonu ile string veriye karakter eklemek. 21. find() fonksiyonu ile string veri içinde bir karakterin konumunu bulmak ve veri içinde arama yapmak. 22. in operatörü ile string veri içerisinde bir karakterin olup olmadığını kontrol etmek. 23. String veri içinde büyük – küçük harf değişimi yapmak.
Hata Yakalama İşlemleri	<p>.Aşağıda istenen programları her türlü hatanın oluşabileceğini düşünerek hazırlayınız.</p> <ol style="list-style-type: none"> 1. Hata türlerini örneklerle anlatan bir sunum hazırlamak. 2. Bir diktörtgenin kenar uzunluklarını kullanıcıdan alıp alan hesabını yapan programı yazmak. 3. Bir diktörtgenin kenar uzunluklarını kullanıcıdan alıp alan hesabını yapan programı -eğer oluşursa- orijinal hata mesajını da gösterecek şekilde yazmak. 4. Kullanıcıdan 1-100 arası sayı girmesi istenip bu sayının asal sayı olduğunu sınavan programı yazmak.

	<p>5. İki sayıyı bölme işlemine tabi tutan programı yazmak. (assert ifadesi kullanılacaktır.)</p> <p>6. Listeler üzerinde işlem yaparken, aralık dışındaki bir elemanı ekranda gösteren programı yazmak.</p> <p>7. ['a', 0, 2] elemanlarından oluşan listenin her bir elemanının çarpmaya göre tersini hesaplayan programı yazmak.</p> <p>8. sys.platform modülünü kullanarak sadece "Windows" işletim sisteminde çalışan program yazmak.</p>
Dosya İşlemleri	<p>1. Çalışma dizinini değiştirmek.</p> <p>2. Çalışma dizini ve klasör oluşturmak için yol tanımlamak.</p> <p>3. Tanımlanan yolun var olup olmadığını bulmak.</p> <p>4. Tanımlanan yolda klasör oluşturmak.</p> <p>5. Çalışma dizininde yer alan bir dosyaya erişmek.</p> <p>6. Erişilen dosyayı okuyup, ekrana yazdırmak.</p> <p>7. Erişilen dosyayı satır satır okumak.</p> <p>8. Çalışma dizininde bir metin belgesi oluşturmak.</p> <p>9. Oluşturulan dosyaya „Gençliğe Hitabe“yi yazdırmak.</p> <p>10. Çalışma dizininde „tarih.txt“ dosyası oluşturup içeriğine günün tarihini yazdırmak.</p> <p>11. "tarih.txt" dosyası içerirğine ekleme yapmak.</p> <p>12. Belirtilen yoldaki dosyayı silmek.</p> <p>13. Belirtilen yoldaki dosyayı başka konuma kopyalamak.</p>

DERSİN UYGULANMASINA İLİŞKİN AÇIKLAMALAR

1. Bu derste, verilen görevi yapma (algoritma oluşturma, akış diyagramı çizme, kodlama yapma, programa ait kod yazma) değer, tutum ve davranışları ön plana çıkaran etkinliklere yer verilmelidir.
2. Gerekli malzemeler kullanılarak (tahta, kalem, bilgisayar) uygulama yaptırılmalıdır.
3. Gerekli donanım malzemeleri ders öncesinde hazırlanmalı ve gerekli bakım onarım ve temizlik yapılmalıdır.
4. Kullanılacak açık kaynak kodlu programlama dili, bilgisayarlarda olup olmadığı ve sorunsuz çalıştığı kontrol edilmeli ve gerekirse yedek bilgisayarlar devreye alınıp uygulamalar yapılmalıdır.
5. Öğrencinin kendi kodlarını oluşturabilmesi ve yaratıcılığının geliştirilebilmesi için farklı yöntemlerle yeniden yazması teşvik edilmelidir.
6. Anlatımdan ve örnek çalışmalardan sonra, dersin öğrenme kazanımlarının öğrencide pekiştirilmesi amacıyla birden fazla uygulama faaliyeti yapılmalıdır.