



DERS BİLGİ FORMU

FAKÜLTE / ENSTİTÜ ADI	Fen Edebiyat Fakültesi
BÖLÜM / PROGRAM / ANABİLİM DALI ADI	Matematik
DERSİN ADI	Bilgisayar Programlama 1
DERSİN KODU	MAT4520
YEREL KREDİSİ	3
AKTS KREDİSİ	5
HAFTALIK DERS SAATİ	3
HAFTALIK UYGULAMA SAATİ	0
HAFTALIK LABORATUVAR SAATİ	0
ÖNKOŞULLAR	Yok
YARIYIL	Güz, Bahar
DERSİN DİLİ	İngilizce, Türkçe
DERSİN SEVİYESİ	Lisans
DERSİN TÜRÜ	Seçmeli @Matematik Lisans Programı Seçmeli @Fen Bilgisi Eğitimi Lisans Programı
DERSİN KATEGORİSİ	Temel Meslek Dersleri
DERSİN VERİLİŞ ŞEKLİ	Yüz Yüze
DERSİ SUNAN AKADEMİK BİRİM	Matematik Bölümü
DERSİN KOORDİNATÖRÜ	Nuran GÜZEL
ASİSTAN(LAR)	
DERSİN AMACI	Bu dersin amacı, öğrencilere C programlama dilini kullanarak programlama problemlerini analiz etme ve çözme, kodlarını test etme ve optimize etme yöntemlerini öğretmektir. Ayrıca, programlama dillerinde yeterlilik, problem çözme ve hata ayıklama gibi temel yazılım geliştirme becerilerinin geliştirilmesine de odaklanmaktadır.
DERSİN İÇERİĞİ	C programlama diline giriş; değişkenler ve veri tipleri; program yazımı ve giriş-çıkış deyimleri; operatörler; kontrol deyimleri; kütüphaneler; fonksiyonlar; özyinelemeli fonksiyonlar; diziler; iki boyutlu ve çok boyutlu diziler; karakter dizileri; göstericiler.
DERS KİTABI / MALZEMESİ / ÖNERİLEN KAYNAKLAR	Ders Kitabı: Dmitrović, Slobodan. <i>Modern C for Absolute Beginners: A Friendly Introduction to the C Programming Language</i> . Springer Nature, 2024. Zorunlu Kaynaklar: [1] Hwang, Andrew D. <i>Sets, Groups, and Mappings: An Introduction to Abstract Mathematics</i> . American Mathematical Society, 2019. [2] Landi, Giovanni, Zampini, Alessandro. <i>Linear Algebra and Analytic Geometry for Physical Sciences</i> . Springer, 2018. [3] Stewart, James. <i>Single Variable Calculus: Early Transcendentals</i> . 8. baskı, Cengage Learning, 2015. Önerilen Kaynaklar:



	Deitel, Paul J., Deitel, Harvey M. <i>How to Program with an Introduction to C++</i>. 8.baskı, Pearson Education Limited, 2016. Karaçay, Timur. <i>Çözümlü C Programları ve Laboratuvar Uygulamaları</i>. 1. baskı, Abaküs Kitap Yayın Pazarlama, 2015. King, Kim N. <i>C Programming: A Modern Approach</i>. 2. baskı, New York, US-NY: W.W. Norton & Company, 2008.
Ders Öğrenim Çıktıları	Bu dersi başarıyla tamamlayan öğrenciler, 1. C Programlama dilinin temel yapısını tanıyarak değişken tanımlama, veri tiplerini ayırt etme ve bu tipleri uygun durumlarda kullanabilme becerisi kazanabileceklerdir. 2. Aritmetik, karşılaştırma ve mantıksal operatörleri programlama bağlamında etkili bir şekilde kullanarak ifade ve işlem kontrolü gerçekleştirebileceklerdir. 3. Karar yapıları (if, switch) ve döngüler (for, while, do-while) gibi kontrol deyimlerini kullanarak algoritmaları farklı durumlara göre yönlendirebileceklerdir. 4. Karmaşık işlemleri fonksiyonlar aracılığıyla yönetebilmek için kullanıcı tanımlı fonksiyonlar yazarak programın modülerliğini artıracaklardır. 5. Özyinelemeli (recursive) fonksiyonları kullanarak kendini yineleyen problemlere algoritmik çözümler geliştirebileceklerdir. 6. Dizileri tanımlayarak matris tabanlı veri yapıları üzerinde çalışabileceklerdir. 7. Gösterici (pointer) kavramını kullanarak ve bellek adresleriyle çalışarak oluşturacakları dinamik veri yapıları ile programlarını geliştirebileceklerdir.

DEĞERLENDİRME SİSTEMİ

Etkinlikler	Sayı	Katkı Payı
Devam/Katılım		
Laboratuvar		
Uygulama		
Arazi Çalışması		
Derse Özgü Staj		
Kısa Sınavlar/Stüdyo Kritiği (Zorunlu): • İçerik: Sınav haftasına kadar işlenen konuların tümünü kapsayan kapsamlı soruların sorulması • Format: Yüz yüze kısa sınav (25-30 dakika) • Detaylı Değerlendirme Kriterleri: - Derste işlenen programlama tekniklerinin verilen bir problemin çözümüne yönelik algoritmasının kurulması ve programın yazılması - Verilen programın çıktısını yazabilme - Mevcut programda hata ayıklama - Mantıksal bütünlük oluşturma becerisine yönelik programdaki eksik parçaları tamamlama	4	%20
Ödev		
Sunum/Jüri		
Proje		



Seminer/Workshop		
Ara Sınavlar: • İçerik: Sınav haftasına kadar işlenen konuların tümünü kapsayan kapsamlı sorular • Format: Yüz yüze yazılı sınav (90 dakika) • Detaylı Değerlendirme Kriterleri: - Derste işlenen konulara uygun, anlamlı ve çalışabilir kod yazma - Verilen çalışır haldeki kodun çıktısını yazma - Verilen kod parçalarında bulunan yazım (syntax) ya da mantık hatalarını tespit etme - Eksik bırakılan kod bölümlerini doğru yapılarla tamamlayarak programı işler hale getirebilme - Kodun amaca uygun çalışıp çalışmadığını değerlendirebilme ve gerektiğinde düzenlemeler yapabilme	1	%40
Final: • İçerik: Dersin tüm içeriğini kapsayan kapsamlı sorular • Format: Yüz yüze yazılı sınav (100 dakika) • Detaylı Değerlendirme Kriterleri: - Verilen algoritmayı anlayarak doğru ve işlevsel biçimde programlayabilme - Programdaki sözdizimi (syntax) ve mantık hatalarını fark edip düzeltebilme - Verilen bir kodun çıktısını doğru şekilde analiz edebilme - Kod parçalarını birleştirerek tutarlı ve çalışabilir bir yapı oluşturabilme	1	%40
Dönem İçi Çalışmaların Başarı Notuna Katkısı		%60
Final Sınavının Başarı Notuna Katkısı		%40
TOPLAM		%100

HAFTALIK KONULAR VE İLGİLİ ÖN HAZIRLIK ÇALIŞMALARI

HAFTALAR	KONULAR	Ön Hazırlık
1	Konu Anlatımı: C programlama diline giriş (programlamanın genel yapısı) Sınıf-içi Uygulama (80 dk.): Programlamanın genel yapısında bulunan parçaların açıklanması ve kodlama dilinde gösterimlerinin tanıtılması Sınıf-içi Tartışma (35 dk.): Genellikle yapılan kodlama hataları üzerine tartışma	1. C programlama diline giriş ve C derleyicileri konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 3-8.
2	Konu Anlatımı: Değişkenler ve veri tipleri (değişkenleri veri tiplerine göre tanımlama) Sınıf-içi Uygulama (75 dk.): Farklı veri tipleriyle değişkenlerin tanımlanması ve değişkenlerin yerelliği Sınıf-içi Tartışma (35 dk.): Değişken türlerinin tanımlanmasında yapılan hataların incelenmesi ve yerel ve yerel olmayan değişkenin farklılıkları üzerine tartışma	1. C programlama dilinde değişkenler ve veri tiplerine göre tanımlama konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 15-30.
3	Konu Anlatımı: Program yazımı ve giriş-çıkış deyimleri (temel programlama ve içerik, basit anlamda girdi verme ve çıktı alma) Sınıf-içi Uygulama (80 dk.): Giriş-çıkış deyimlerinin doğru kullanımına yönelik program yazılması ve farklı girdilerle elde edilecek çıktıların incelenmesi Sınıf-içi Tartışma (30 dk.): Giriş-çıkış deyimlerinde yapılan hataların değerlendirilmesi üzerine tartışma	1. C programlama dilinde program yazımında temel kavramlar ve giriş-çıkış deyimleri konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 9-15; 31-34.



4	Konu Anlatımı: Operatörler (aritmetik, mantık, eşitlik ve bağıntılar, atama, artış ve azalış, koşulsal) Sınıf-İçi Uygulama (60 dk.): Aritmetik ve karşılaştırma operatörlerini içeren kısa algoritmaların verilmesi ve çıktıların tahmin edilmesi Sınıf-İçi Tartışma (30 dk.): Operatörlerin matematiksel anlamları ile programlamadaki işlevleri arasındaki farkların tartışılması ve özellikle “=” ve “==” operatörlerinin anlamlarının değerlendirilmesi, mantıksal bağlaçların doğruluk tablolarının ve işlem önceliğinin önemine yönelik alıştırmaların yapılması	1. Mantık ve bağıntılar kavramlarına ilişkin ön bilgilerin hatırlanması ve etkinleştirilmesi. Kaynak: [1], 1-8; 58-62. 2. C programlama dilinde operatörler konusunu içeren bölümlerin okunması. Kaynak: Ders Kitabı, 35-48.
5	Konu Anlatımı: Kontrol deyimleri (seçim ve karar deyimleri) Sınıf-İçi Uygulama (80 dk.): Seçim ve karar yapılarında kullanılan if, else if ve else deyimlerinin temel kullanım biçimlerini içeren kısa kod örnekleri ile çalışma yapılması, farklı giriş verilerine bağlı olarak farklı çıktılar üretmesini sağlayan küçük uygulamalar yapılması ve koşul sıralaması, iç içe yapılar ve mantıksal ifadelerin program kontrol akışı üzerindeki etkilerinin gözlemlenmesi Sınıf-İçi Tartışma (30 dk.): Karar yapılarının matematiksel akıl yürütme süreçleriyle olan benzerliği üzerine tartışma yapılması, birden fazla koşulun bir arada değerlendirilmesi durumunun incelenmesi ve kodun okunabilirliği, kontrol yapılarının sadeleştirilmesi ve alternatif ifade biçimlerinin kullanımı ile ilgili tartışma yapılması üzerine tartışma Kısa Sınav 1 (25 dk.): Ters yüz edilmiş öğrenme (flipped learning) yöntemi çerçevesinde, ders sonunda, derste önceki haftalarda işlenmiş konulara yönelik kısa sınavın yapılması	1. Seçim ve karar deyimleri konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 49-60. 2. Kısa Sınav 1: (C programlama diline giriş; değişkenler ve veri tipleri; program yazımı ve giriş-çıkış deyimleri; operatörler) Kaynak: Ders Kitabı, 3-48.
6	Konu Anlatımı: Kontrol deyimleri (iteratif deyimler ve atlama deyimleri) Sınıf-İçi Uygulama (70 dk.): İteratif kontrol yapılarının program akışını nasıl yönlendirdiğini gösteren basit örneklerin verilmesi, döngülerin farklı koşullar altında nasıl çalıştığı ve döngüden çıkma (break) veya iterasyonun belirli bir kısmını atlama (continue) gibi deyimlerin bu yapılar üzerindeki etkisi gözlemlenmesi ve iteratif deyimlerle seçim ve karar deyimlerinin bir arada kullanımına yönelik alıştırmalar yapılması Sınıf-İçi Tartışma (60 dk.): Döngülerin matematiksel örüntüler ve işlem tekrarlarıyla ilişkisinin tartışılması, break, continue gibi deyimlerin hangi durumlarda tercih edilmesi gerektiği, kodun akışına etkileri ve okunabilirliği üzerindeki rolü üzerine fikir alışverişi yapılması ve gereksiz döngü kullanımı ya da döngü içi kontrol deyimlerinin kötüye kullanımı gibi yaygın hataların gözlemlenmesi üzerine tartışma	1. Lise düzeyindeki matematikte iteratif işlemlerin (ardışık toplama-çarpma, faktöriyel, vb.) çalışma yapılarına ilişkin ön bilgilerin hatırlanması ve etkinleştirilmesi. 2. İteratif ve atlama deyimleri konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 61-64.
7	Konu Anlatımı: Kütüphaneler (bazı standart kütüphaneler ve matematik kütüphanesi) Sınıf-İçi Uygulama (40 dk.): Standart kütüphaneler ve özellikle matematik kütüphanesi kullanılarak erişilebilen matematiksel fonksiyonların temel kullanımına dair örneklerin verilmesi Sınıf-İçi Tartışma (60 dk.): Matematiksel işlemlerin elle kodlanması yerine uygun kütüphane fonksiyonlarının tercih edilmesi ve doğru sözdizimi ile çağırılmasına yönelik uygulama ve tartışma yapılması Kısa Sınav 2 (30 dk.): Ters yüz edilmiş öğrenme (flipped learning) yöntemi çerçevesinde, ders sonunda, derste önceki haftalarda işlenmiş konulara yönelik kısa sınavın yapılması	1. C dilinde yaygın olarak kullanılan kütüphaneler ve matematik kütüphanesi konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 250-253. 2. Kısa Sınav 2: (kütüphaneler konusuna kadar tüm konular) Kaynak: Ders Kitabı, 3-64.
8	Ara Sınav 1	Sınav haftasına kadar işlenen konuların tümünün tekrar edilmesi
9	Konu Anlatımı: Fonksiyonlar (modüler programlama, fonksiyon tanımlama, parametre türleri, fonksiyon kategorileri) Sınıf-İçi Uygulama (40 dk.): Parametrelili ve parametresiz, farklı çıktı türlerine yönelik fonksiyon örneklerinin verilmesi ve modüler yapının kod okunabilirliği ve yeniden kullanılabilirlik açısından katkıları uygulamalı olarak incelenmesi	1. Fonksiyonlar (kalkülüs) kavramına ilişkin ön bilgilerin hatırlanması ve etkinleştirilmesi. Kaynak: [3], 9-35. 2. Modüler programlama, fonksiyon tanımlama, parametre türleri, fonksiyon kategorileri konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 103-122.



	Sınıf-İçi Tartışma (80 dk.): Programlamada fonksiyon kullanımının, matematikteki fonksiyon kavramıyla nasıl benzerlik ve farklar taşıdığı üzerine tartışılması bir programın modüllere ayrılarak yazılmasının hata ayıklama, güncelleme ve yeniden kullanım açısından sağladığı avantajların değerlendirilmesi ve fonksiyonların türleri arasındaki farkların ve tercih gerekçelerinin değerlendirilmesi	
10	Konu Anlatımı: Özyinelemeli fonksiyonlar (iteratif döngülerle kıyaslama ve kuyruk özyinelemesi) Sınıf-İçi Uygulama (30 dk.): Özyinelemeli olarak tanımlanmış algoritmaların analiz edilmesi ve fonksiyonun çağrı sırasının izlenerek çağrı yığını üzerindeki etkisi gözlemlenmesi Sınıf-İçi Tartışma (80 dk.): Özyinelemeli verilmiş fonksiyonların döngü yapısıyla nasıl çözülebileceğinin tartışılması ve her iki yaklaşımın karşılaştırmalı uygulanması	1. Özyineleme kavramına ilişkin ön bilgilerin hatırlanması ve etkinleştirilmesi. Kaynak: [1], 75-77. 2. Özyinelemeli fonksiyonlar ve kuyruk özyinelemesi konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 133-138.
11	Konu Anlatımı: Diziler (dizilerin tanımlanması ve temel diziler) Sınıf-İçi Uygulama (50 dk.): Sabit uzunlukta çeşitli dizilerin tanımlanarak elemanlara indis yoluyla erişilmesi ve dizi elemanlarının toplamını veya ortalamasını hesaplayan basit işlemler uygulanması Sınıf-İçi Tartışma (40 dk.): Tekrarlayan verilerle çalışırken programı nasıl sadeleştirdiğinin incelenmesi ve tek tek değişken tanımlayarak yapmak ile dizi kullanarak yapmak arasındaki farkların değerlendirilmesi Kısa Sınav 3 (30 dk.): Ters yüz edilmiş öğrenme (flipped learning) yöntemi çerçevesinde, ders sonunda, derste önceki haftalarda işlenmiş konulara yönelik kısa sınavın yapılması	1. Vektörler kavramına ilişkin ön bilgilerin hatırlanması ve etkinleştirilmesi. Kaynak: [2], 1-5. 2. Diziler konusunu içeren bölümlerin okunması. Kaynak: Ders Kitabı, 71-75. 3. Kısa Sınav 3: (diziler konusuna kadar tüm konular) Kaynak: Ders Kitabı, 3-64; 103-138; 250-253.
12	Konu Anlatımı: İki boyutlu ve çok boyutlu diziler (matris yapısı ve dizilerin fonksiyonlarla kullanımı) Sınıf-İçi Uygulama (50 dk.): İki boyutlu ve çok boyutlu dizilerin tanımlanmasına, bu diziler üzerinde temel işlemler yapılmasına ve söz konusu dizilerin fonksiyonlar içerisinde kullanılmasına yönelik uygulamalar yapılması Sınıf-İçi Tartışma (60 dk.): İki boyutlu dizilerin matris temsiliyle örtüşen yapıları ile çok boyutlu dizilerin karmaşık veri düzenlemelerine olanak sağlaması üzerine tartışılması ve bu yapıların okunabilirlik, fonksiyonel kullanım ve belleğe erişim açısından avantajları ve sınırlılıkları üzerine değerlendirme yapılması	1. Matris kavramına ilişkin ön bilgilerin hatırlanması ve etkinleştirilmesi. Kaynak: [2], 47-52. 2. İki boyutlu ve çok boyutlu diziler konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 77-80.
13	Konu Anlatımı: Karakter dizileri (karakter dizilerinin sayı dizilerinden farkı, karakter dizileri kütüphanesi ve karakter dizisi manipülasyonu, karakter kütüphanesi) Sınıf-İçi Uygulama (50 dk.): Standart karakter kütüphanelerinden yararlanılarak karakter dizilerinin tanımlanması, karakter dizileri üzerinde temel işlemler yapılması ve karakter düzeyinde veri işleme, metin karşılaştırma veya dönüştürme gibi işlemlerin gerçekleştirilmesi Sınıf-İçi Tartışma (60 dk.): Karakter dizileri ile sayı dizileri arasındaki farklar; veri tipi, kullanım amacı ve işlem yapıları bakımından karşılaştırılması ve giriş-çıkış deyimleri ile karakter dizilerinin farklarının incelenmesi üzerine tartışma	1. Karakter dizilerinin sayı dizilerinden farkı, karakter dizileri kütüphanesi ve karakter dizisi manipülasyonu, karakter kütüphanesi konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 76-77; 213-218; 239-243.
14	Konu Anlatımı: Göstericiler (gösterici operatörü, adres tanımlaması ve değer atamaları) Sınıf-İçi Uygulama (35 dk.): Adreslerin okunması ve göstericiler aracılığıyla değişken değerlerine erişilmesi yönünde temel uygulamalar gerçekleştirilmesi Sınıf-İçi Tartışma (45 dk.): Göstericilere değer atama işlemlerinin ve bu işlemlerin bellek üzerindeki etkilerinin gözlemlenmesi üzerine tartışma Kısa Sınav 4 (30 dk.): Ters yüz edilmiş öğrenme (flipped learning)	1. Gösterici operatörü, adres tanımlaması ve değer atamaları konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 81-84. 2. Kısa Sınav 4: (göstericiler konusuna kadar tüm konular) Kaynak: Ders Kitabı, 3-80; 103-138; 250-253.



	yöntemi çerçevesinde, ders sonunda, derste önceki haftalarda işlenmiş konulara yönelik kısa sınavın yapılması	
15	Konu Anlatımı: Göstericiler (göstericilerin fonksiyonlarla ve dizilerle kullanımı, adres aritmetiği, bellek atama) Sınıf-içi Uygulama (70 dk.): Göstericilerin dizi elemanlarına erişimde ve fonksiyonlara parametre aktarımında nasıl kullanılabildiğine yönelik uygulamaların yapılması, adres aritmetiği kullanılarak ardışık bellek alanlarına erişim sağlanması ve dinamik bellek tahsisine dair temel işlemlerin gerçekleştirilmesi Sınıf-içi Tartışma (30 dk.): Özellikle bellek sızıntısı gibi kavramlara dikkat çekilerek bilinçli gösterici kullanımının tartışılması	1. Göstericilerin fonksiyonlarla ve dizilerle kullanımı, adres aritmetiği, bellek atama konularını içeren bölümlerin okunması. Kaynak: Ders Kitabı, 85-94; 157-160.
16	Final	İşlenen konuların tümünün tekrar edilmesi

AKTS İŞYÜKÜ TABLOSU

Etkinlikler	Sayı	Süresi (Saat)	Toplam İş yükü
Ders Saati	14	3	42
Laboratuvar			
Uygulama			
Arazi Çalışması			
Sınıf Dışı Ders Çalışması	14	4	56
Derse Özgü Staj			
Ödev			
Küçük Sınavlar / Stüdyo Kritiği	4	3	12
Projeler			
Sunum / Seminer			
Ara Sınavlar (Sınav Süresi + Sınav Hazırlık Süresi)	1	20	20
Final (Sınav Süresi + Sınav Hazırlık Süresi)	1	25	25
Toplam İş yükü:			155
Toplam İş yükü / 30(s):			5.16
AKTS Kredisi:			5



COURSE INFORMATION FORM

FACULTY / GRADUATE SCHOOL	Faculty of Arts and Sciences
DEPARTMENT / PROGRAMME	Mathematics
TITLE OF COURSE	Computer Programming 1
CODE	MAT4520
LOCAL CREDIT	3
ECTS	5
LECTURE HOUR / WEEK	3
PRACTICAL HOUR / WEEK	0
LABORATORY HOUR / WEEK	0
PREREQUISITE	None
SEMESTER	Fall, Spring
COURSE LANGUAGE	English, Turkish
LEVEL OF COURSE	First Cycle
COURSE TYPE	Elective @Bachelor Programme in Mathematics Elective @Bachelor Programme in Science Education
COURSE CATEGORY	Core Courses
MODE OF DELIVERY	Face-to-Face
OWNER ACADEMIC UNIT	Department of Mathematics
COURSE COORDINATOR	Nuran GÜZEL
ASSISTANT(S)	
COURSE OBJECTIVES	This course aims to help students analyze and solve programming problems, test and optimize their code using the C programming language. It also focuses on developing essential software development skills, including proficiency in programming languages, problem-solving, and debugging.
COURSE CONTENT	Introduction to C; variables and data types; program writing and input-output statements; operators; control statements; libraries; functions; recursive functions; arrays; two-dimensional and multidimensional arrays; character strings; pointers.
RECOMMENDED OR REQUIRED READINGS	Coursebook: Dmitrović, Slobodan. <i>Modern C for Absolute Beginners: A Friendly Introduction to the C Programming Language</i> . Springer Nature, 2024. Required Readings: [1] Hwang, Andrew D. <i>Sets, Groups, and Mappings: An Introduction to Abstract Mathematics</i> . American Mathematical Society, 2019. [2] Landi, Giovanni, Zampini, Alessandro. <i>Linear Algebra and Analytic Geometry for Physical Sciences</i> . Springer, 2018. [3] Stewart, James. <i>Single Variable Calculus: Early Transcendentals</i> . 8th ed., Cengage Learning, 2015. Recommended Readings: Deitel, Paul J., Deitel, Harvey M. <i>How to Program with an Introduction to C++</i> . 8th ed., Pearson Education Limited, 2016.



Karaçay, Timur. *Çözümlü C Programları ve Laboratuvar Uygulamaları*. 1st ed., Abaküs Kitap Yayın Pazarlama, 2015.

King, Kim N. *C Programming: A Modern Approach*. 2nd ed., W.W. Norton & Company, New York, 2008.

Course Learning Outcomes

Upon successful completion of the course, students will be able to

1. Comprehend the basic structure of the C programming language by acquiring the skills of defining variables, distinguishing data types, and using these types appropriately.
2. Perform expression and operation control by effectively applying arithmetic, comparison, and logical operators in the context of programming.
3. Use control statements such as decision structures (if, switch) and loops (for, while, do-while) to direct algorithms according to different situations.
4. Enhance program modularity by writing user-defined functions to manage complex operations.
5. Develop algorithmic solutions to recursive problems using recursive functions.
6. Work on matrix-based data structures by defining arrays.
7. Develop their programs with dynamic data structures created using the concept of pointers and memory addresses.

EVALUATION SYSTEM

Activities	Number	Percentage of Grade
Attendance/Participation		
Laboratory		
Application		
Field Work		
Special Course Internship (Work Placement)		
Quizzes/Studio Critics: • Content: Comprehensive questions covering all topics up to the exam week • Format: Face-to-face quiz (25-30 minutes) • Detailed Assessment Criteria: - Establishing an algorithm for solving a given problem using the programming techniques covered in class, and writing the program - Ability to write the output of the given program - Debugging the current program - Completing missing parts in the program related to logical integrity skills	4	%20
Homework Assignments		
Presentations/Jury		
Project		
Seminar/Workshop		
Midterms: • Content: Comprehensive questions covering all topics up to exam week • Format: Face-to-face written exam (90 minutes)	1	%40



Detailed Assessment Criteria: - Writing meaningful and workable code appropriate to the topics covered in class - Writing the output of the given working code - Identifying syntax or logic errors in the given code snippets - Ability to complete missing code sections with the correct structures to make the program work - Ability to evaluate whether the code works as intended and make adjustments when necessary				
Final: Content: Comprehensive questions covering all topics Format: Face-to-face written exam (100 minutes) Detailed Assessment Criteria: - Ability to understand the given algorithm and program it correctly and functionally - Ability to identify and correct syntax and logic errors in the program - Ability to correctly analyze the output of a given code - Ability to combine code fragments to create a consistent and functional structure			1	%40
Percentage of In-Term Studies				%60
Percentage of Final Examination				%40
TOTAL				%100
WEEKLY SUBJECTS AND RELATED PREPARATION STUDIES				
WEEKS	COURSE OUTLINE		Related Preparation	
1	Lecture: Introduction to C (general structure of programming) Practice (80 minutes): Explanation of the components found in the general structure of programming and introduction to their representation in coding languages In-Class Discussion (35 minutes): Discussion of demonstration of commonly made coding errors		1. Reading the sections on introduction to the C and C compilers. Source: Coursebook, 3-8.	
2	Lecture: Variables and data types (defining variables according to data types) Practice (75 minutes): Definition of variables with different data types and the locality of variables In-Class Discussion (35 minutes): Discussion of examination of errors made in the definition of variable types and differences between local and global variables		1. Reading the sections on variables and declarations according to data types in C. Source: Coursebook, 15-30.	
3	Lecture: Program writing and input-output statements (basic programming and content, inputting and outputting in simple terms) Practice (80 minutes): Writing a program for the correct use of input-output expressions and examining the outputs obtained with different inputs In-Class Discussion (30 minutes): Discussion of evaluation of errors made in input-output expressions		1. Reading the sections on basic concepts in writing programs in C and input-output statements. Source: Coursebook, 9-15; 31-34.	
4	Lecture: Operators (arithmetic, logical, equality and relations, assignment, increment and decrement, conditional) Practice (60 minutes): Providing short algorithms containing arithmetic and comparison operators and predicting their outputs		1. Recollection and activation of prior knowledge on the concepts of logic and relations. Source: [1], 1-8; 58-62. 2. Reading the sections on operators	



	In-Class Discussion (30 minutes): Discussion of the differences between the mathematical meanings of operators and their functions in programming, and evaluation of the meanings of the “==” and “=” operators, as well as conducting exercises on the importance of truth tables and operator precedence for logical operators	in C. Source: Coursebook, 35-48.
5	Lecture: Control statements (selection and decision statements) Practice (80 minutes): Working with short code examples containing basic usage forms of if, else if, and else statements used in selection and decision structures, creating small applications that produce different outputs depending on various input data, and observing the effects of condition ordering, nested structures, and logical expressions on program control flow In-Class Discussion (30 minutes): Discussion of the similarity between decision structures and mathematical reasoning processes, examination of the situations where multiple conditions are evaluated simultaneously, and discussion on code readability, simplification of control structures, and the use of alternative expressions Quiz 1 (25 minutes): A quiz on topics covered in previous weeks at the end of the lesson, within the framework of the flipped learning method	1. Reading the sections on selection and decision statements. Source: Coursebook, 49-60. 2. Quiz 1: (introduction to the C; variables and data types; program writing and input-output statements; operators) Source: Coursebook, 3-48.
6	Lecture: Control statements (iterative statements and jump statements) Practice (70 minutes): Providing simple examples that show how iterative control structures direct program flow, observing how loops work under different conditions and the effect of statements such as breaking out of a loop (break) or skipping a certain part of an iteration (continue) on these structures, and doing exercises on the combined use of iterative statements and selection and decision statements In-Class Discussion (60 minutes): Discussion of the relationship between loops and mathematical patterns and operation repetitions, exchanging ideas on when to use expressions such as break and continue, their effects on code flow, and their role in readability, and observing common mistakes such as unnecessary loop usage or misuse of loop control expressions	1. Recollection and activation of prior knowledge of high school-level concepts of operational structures of iterative operations (sequential addition-multiplication, factorial, etc.). 2. Reading the sections on iterative and jump statements. Source: Coursebook, 61-64.
7	Lecture: Libraries (some standard libraries and math library) Practice (40 minutes): Providing examples of the basic use of mathematical functions accessible using standard libraries, particularly the math library In-Class Discussion (60 minutes): Practice and discussion on preferring appropriate library functions over manually coding mathematical operations and calling them with the correct syntax Quiz 2 (30 min.): A quiz on topics covered in previous weeks at the end of the lesson, within the framework of the flipped learning method	1. Reading the sections on libraries commonly used in C and the math library. Source: Coursebook, 250-253. 2. Quiz 2: (all topics up to the topic of libraries) Source: Coursebook, 3-64.
8	Midterm 1	Review of all topics covered up to the exam week
9	Lecture: Functions (modular programming, function definition, parameter types, function categories) Practice (40 minutes): Providing examples of parameterized and non-parameterized functions for different output types and examining the contributions of modular structure to code readability and reusability in practice In-Class Discussion (80 minutes): Discussion on how the use of functions in programming is similar to and different from the concept of functions in mathematics, evaluating the advantages of writing a program in modules in terms of debugging, updating, and reusing, and assessing the differences between the types of functions and the reasons for preference	1. Recollection and activation of prior knowledge on the concepts of functions (Calculus). Source: [3], 9-35. 2. Reading the sections on modular programming, function definition, parameter types, and function categories. Source: Coursebook, 103-122.



10	Lecture: Recursive functions (comparison with iterative loops and tail recursion) Practice (30 minutes): Analyzing recursively defined algorithms and observing the effect on the call stack by monitoring the call order of the function In-Class Discussion (80 minutes): Discussion on how to solve recursively given functions with loop structure and comparative application of both approaches	1. Recollection and activation of prior knowledge on the concepts of recursion. Source: [1], 75-77. 2. Reading the sections on recursive functions and tail recursion. Source: Coursebook, 133-138.
11	Lecture: Arrays (definition of arrays and basic arrays) Practice (50 minutes): Defining various arrays of fixed length, accessing the elements through indices, and applying simple operations that calculate the sum or average of the array elements In-Class Discussion (40 minutes): Examination of how it simplifies the program when working with repetitive data and evaluation of the differences between defining individual variables and using arrays Quiz 3 (30 min.): A quiz on topics covered in previous weeks at the end of the lesson, within the framework of the flipped learning method	1. Recollection and activation of prior knowledge on the concepts of vectors. Source: [2], 1-5. 2. Reading the sections on arrays. Source: Coursebook, 71-75. 3. Quiz 3: (all topics up to sequences) Source: Coursebook, 3-64; 103-138; 250-253.
12	Lecture: Two-dimensional and multidimensional arrays (matrix structure and use of arrays with functions) Practice (50 minutes): Defining two-dimensional and multidimensional arrays, performing basic operations on these arrays, and making applications for using these arrays in functions In-Class Discussion (60 minutes): Discussion on the overlapping structures of two-dimensional arrays with matrix representation and multidimensional arrays that allow complex data organization and evaluation of the advantages and limitations of these structures in terms of readability, functional use, and memory access	1. Recollection and activation of prior knowledge on the concepts of matrices. Source: [2], 47-52. 2. Reading the sections on two-dimensional and multidimensional arrays. Source: Coursebook, 77-80.
13	Lecture: Character strings (the difference between character strings and number strings, character string library, character string manipulation, and character library) Practice (50 minutes): Defining character strings using standard character libraries, performing basic operations on character strings, and performing character-level data processing, text comparison, or conversion In-Class Discussion (60 minutes): Differences between character strings and number arrays; comparing them in terms of data type, purpose of use, and operation structures, and examining the differences between input-output statements and character strings	1. Reading the sections on the difference between character strings and number strings, character string library, character string manipulation, and character library. Source: Coursebook, 76-77; 213-218; 239-243.
14	Lecture: Pointers (pointer operator, address identification, and value assignments) Practice (35 minutes): Performing basic applications for reading addresses and accessing variable values through pointers In-Class Discussion (45 minutes): Discussion on observation of operations for assigning values to pointers and their effects on memory Quiz 4 (30 min.): A quiz on topics covered in previous weeks at the end of the lesson, within the framework of the flipped learning method	1. Reading the sections on pointer operator, address declaration, and value assignments. Source: Coursebook, 81-84. 2. Quiz 4: (all topics up to the topic of pointers) Source: Coursebook, 3-80; 103-138; 250-253.
15	Lecture: Pointers (use of pointers with functions and arrays, address arithmetic, memory allocation) Practice (70 minutes): Practicing how pointers can be used to access array elements and pass parameters to functions, accessing sequential memory areas using address arithmetic, and performing basic operations on dynamic memory allocation	1. Reading the sections on the use of pointers with functions and arrays, address arithmetic, and memory allocation. Source: Coursebook, 85-94; 157-160.



	In-Class Discussion (30 minutes): Discussion of the use of conscious pointers, with particular attention to concepts such as memory leakage	
16	Final	Review of all topics covered

ECTS WORKLOAD TABLE

Activities	Number	Duration (Hour)	Total Workload
Course Hours	14	3	42
Laboratory			
Application			
Field Work			
Study Hours Out of Class	14	4	56
Special Course Internship (Work Placement)			
Homework Assignments			
Quizzes / Studio Critics	4	3	12
Project			
Presentations / Seminar			
Mid-Terms (Examination Duration + Examination Prep. Duration)	1	20	20
Final (Examination Duration + Examination Prep. Duration)	1	25	25
Total Workload:			155
Total Workload / 30(h):			5.16
ECTS Credit:			5



Ders Öğrenim Çıktısı & Program Çıktısı Matrisi

	DÖC-1	DÖC-2	DÖC-3	DÖC-4	DÖC-5	DÖC-6	DÖC-7
PC-1 Geometri, analiz, lineer cebir, soyut matematik, cebir ve diferansiyel denklemler gibi matematiğin temel alanlarındaki tanımları ve teoremleri ifade edebilecek, bunlar arasında ilişki kurabilecek ve teoremlerin uygulamalarını açıklayabileceklerdir. / State the definitions and theorems in fundamental areas of mathematics such as geometry, analysis, linear algebra, abstract mathematics, and differential equations, establish connections between them and explain their applications.	=	=	=	=	=	=	=
PC-2 Matematik alanındaki teorik ve uygulamalı bilgilerini ve matematiksel düşünme becerilerini kullanarak standart ve karmaşık matematiksel problemleri tanımlayabilecek ve analitik ve/veya sayısal yöntemlerle çözebilecek ve matematiksel bir önermenin geçerliliğini farklı ispat yöntemleri ile ispatlayabileceklerdir. / Identify standard and complex mathematical problems, solve these problems with analytical and/or numerical methods and prove the validity of a mathematical proposition through various proof techniques using their theoretical and applied knowledge of mathematics and mathematical thinking skills.	=	=	=	=	=	=	=
PC-3 Matematik alanındaki teorik ve uygulamalı bilgilerini ve matematiksel düşünme becerilerini kullanarak gerçek hayattaki problemlerin matematiksel modellerini geliştirebilecek ve bu modelleri kullanarak problemlere geçerli çözümler üretebileceklerdir. / Develop mathematical models for real-life problems and generate valid solutions based on these models using their theoretical and applied knowledge of mathematics and mathematical thinking skills.	<u>3</u>	<u>4</u>	<u>4</u>	<u>5</u>	<u>4</u>	<u>4</u>	<u>3</u>
PC-4 Disiplinlerarası bir yaklaşımla, farklı alanlarda edinmiş oldukları bilgileri sentezleyebileceklerdir. / Synthesis knowledge acquired from different disciplines through an interdisciplinary approach.	<u>3</u>	<u>2</u>	<u>3</u>	<u>3</u>	<u>2</u>	<u>3</u>	=
PC-5 Matematik alanında edindikleri bilgi birikimlerini ve matematiksel düşünme becerilerini, teorik matematik, yazılım, finans ve yönetim matematiği, biyomatematik, veri bilimi ve finansal istatistik yöntemleri gibi disiplin-içi ve disiplinlerarası uzmanlık alanlarında geliştirebileceklerdir. / Advance their acquired knowledge in mathematics and mathematical thinking skills in both disciplinary and interdisciplinary areas of specialisation such as pure mathematics, software, financial and managerial mathematics, biomathematics, data science, and financial statistical methods.	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>	<u>4</u>
PC-6 Matematik alanında yaygın olarak kullanılan en az bir programlama dili ile bilgisayar ve yapay zekâ teknolojilerini, problemleri çözmek, veri analizi yapmak ve simülasyonlar gerçekleştirmek için kullanabileceklerdir. / Use at least one programming language and computer and artificial intelligence technologies widely employed in mathematics for problem-solving, data analysis, and simulations.	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>
PC-7 Matematik ve ilgili alanlardaki bilimsel ve teknolojik gelişmeleri izleyebilecek, kariyer fırsatlarını değerlendirerek kişisel ve mesleki gelişim hedeflerini belirleyebilecek ve bu hedeflere ulaşmak için hayat boyu öğrenme stratejilerini kullanabileceklerdir. / Follow scientific and technological developments in mathematics and related fields, assess career	=	=	=	=	=	=	=



opportunities, identify personal and professional development goals, and adopt lifelong learning strategies to achieve these goals.							
PC-8 Bilimsel arařtırmalarını ve mesleki faaliyetlerini yürütürken doğabilecek hukuksal sonuçları ve toplumsal etkileri dikkate alarak mesleki etik ilkeler, kalite standartları ile evrensel değerler doğrultusunda ve sosyal sorumluluk bilinci ve adalet duygusuyla hareket edebileceklerdir. / Act with a sense of social responsibility and justice and in accordance with professional ethical principles, quality standards, and universal values by taking into account potential legal and societal consequences of their scientific research and professional activities.	=	=	=	=	=	=	=
PC-9 Bireysel olarak ya da takımlarda etkin biçimde çalışabileceklerdir. / Work effectively both independently and as part of a team.	=	=	=	=	=	=	=
PC-10 Matematik alanında güvenilir bilgi kaynaklarına ulaşarak literatür taraması yapabilecek ve akademik araştırma tasarlayıp yürütebileceklerdir. / Access reliable sources of information, conduct literature reviews, and design and carry out academic research in the field of mathematics.	=	=	=	=	=	=	=
PC-11 Matematiksel konuları, teorileri, ispatları, arařtırmaları ve problem çözümlerini, matematik terminolojisi kullanarak tüm paydařlara Türkçe ve İngilizcede sözlü ve yazılı olarak etkili biçimde aktarabileceklerdir. / Effectively communicate mathematical topics, theories, proofs, research, and problem solutions to all relevant stakeholders using appropriate mathematical terminology, both orally and in writing, in Turkish and in English.	=	=	=	=	=	=	=