



DERS BİLGİ FORMU

FAKÜLTE / ENSTİTÜ ADI	Fen Edebiyat Fakültesi
BÖLÜM / PROGRAM / ANABİLİM DALI ADI	İstatistik
DERSİN ADI	Bilgisayar Programlamaya Giriş
DERSİN KODU	IST1081
YEREL KREDİSİ	3
AKTS KREDİSİ	5
HAFTALIK DERS SAATİ	3
HAFTALIK UYGULAMA SAATİ	0
HAFTALIK LABORATUVAR SAATİ	0
ÖNKOŞULLAR	Yok
YARIYIL	Güz
DERSİN DİLİ	İngilizce, Türkçe
DERSİN SEVİYESİ	Lisans
DERSİN TÜRÜ	Zorunlu @İstatistik Lisans Programı
DERSİN KATEGORİSİ	Temel Meslek Dersleri
DERSİN VERİLİŞ ŞEKLİ	Yüz Yüze
DERSİ SUNAN AKADEMİK BİRİM	İstatistik Bölümü
DERSİN KOORDİNATÖRÜ	Ersoy ÖZ
ASİSTAN(LAR)	Zuhal Fatma CELLAT
DERSİN AMACI	Bu dersin amacı, öğrencilerin bilgisayar programlamanın temel prensiplerini anlamalarını ve uygulamalarını sağlamaktır. Öğrenciler, programlama mantığını kavramak için akış şemalarını tasarlayıp kullanmayı, giriş-çıkış operatörleri, değişkenler, koşul ifadeleri ve döngü yapılarını etkin bir şekilde uygulamayı, ayrıca dizi, matris işlemleri ve fonksiyonlarla problem çözme becerilerini geliştirmeyi öğrenecektir. Ders aynı zamanda, öğrencilerin algoritmik düşünme yeteneklerini güçlendirerek, temel programlama becerilerini gerçek dünya problemlerine uygulayabilecek düzeye getirmeyi hedefler.
DERSİN İÇERİĞİ	Algoritmalar; Akış şemaları; Operatörler; Giriş-Çıkış Operatörleri; Koşul İfadeleri; Döngü Yapıları; Dizi işlemleri; Matris işlemleri ve Alt programlar.
DERS KİTABI / MALZEMESİ / ÖNERİLEN KAYNAKLAR	Ders Kitabı: ALP, Selçuk; KİLİTÇİ, Arzu. <i>C Programlama Dili</i> . 1. Baskı, Umuttepe Yayınları, 2015. Zorunlu Kaynak: ÇÖLKESEN, Rıfat. <i>Veri Yapıları ve Algoritmalar</i> . 12. Baskı. Papatya Yayıncılık, 2002.

**Ders Öğrenim Çıktıları**

Bu dersi başarıyla tamamlayan öğrenciler,

1. Bilgisayar programlama temel prensiplerine ve program kodlarının nasıl yazılacağına sahip olabileceklerdir.
2. Akış şemalarını okuyup tasarlayarak, problem çözme süreçlerinde uygulayabileceklerdir.
3. Giriş çıkış operatörlerini, değişkenleri ve koşul ifadelerini kullanabileceklerdir.
4. Döngü yapılarını anlayıp, programlama problemlerinde etkin bir şekilde kullanabileceklerdir.
5. Dizi ve matris işlemlerini yapıp, fonksiyon kullanımına hakim olabileceklerdir.

DEĞERLENDİRME SİSTEMİ

Etkinlikler	Sayı	Katkı Payı
Devam/Katılım		
Laboratuvar		
Uygulama (Sözlü Sınav)		
Arazi Çalışması		
Derse Özgü Staj		
Kısa Sınavlar/Stüdyo Kritiği (Zorunlu): • İçerik: Kısa sınavın yapılacağı haftaya kadar işlenen konuların tümünü kapsayan sorular. • Format: Yüz yüze. Yazılı (60 dakika) veya çoktan seçmeli kısa sınav (30 dakika). • Detaylı Değerlendirme Kriterleri: - Derste işlenen uygulamalar ile ilgili problemleri çözebilme. - 1 tane kısa sınav %20 - 3 tane kısa sınav %15 (%5 + %5 + %5)	4	%35
Ödev		
Sunum/Jüri		
Proje		
Seminer/Workshop		
Ara Sınavlar: • İçerik: Sınav haftasına kadar işlenen konuların tümünü kapsayan kapsamlı sorular. • Format: Yüz yüze. Yazılı (90 dakika) veya çoktan seçmeli sınav (60 dakika). • Detaylı Değerlendirme Kriterleri: - Dersin temel kavramlarının anlaşıldığının gösterilmesi. - Teorik konularla ilgili problemlerin çözülebilmesi. - Teorik düşünme süreçlerinin yürütülmesi.	1	%25



Final: • İçerik: Dersin tüm içeriğini kapsayan sorular. • Format: Yüz yüze. Yazılı (90 dakika) veya çoktan seçmeli sınav (60 dakika). • Detaylı Değerlendirme Kriterleri: - Derste işlenen tüm konuların derinlemesine kavranmış olduğunun gösterilmesi. - İleri düzey problem çözme becerilerinin kullanılabilmesi.		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: Algoritmalar. Sınıf-içi Uygulama (20 dk): Basit bir sıralama algoritması üzerinden adım adım bir dizi sayıyı sıralama pratiği yaptırılması. Sınıf-içi Tartışma (10 dk): Algoritmaların günlük hayatta ve farklı disiplinlerde (bilgisayar bilimi, mühendislik, veri analizi) kullanım alanlarının tartışılması.	1. Temel algoritma kavramlarını ve sıralama algoritmalarına ilişkin ön bilgilerin gözden geçirilmesi ve algoritmaların pratik uygulamalarına dair örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 1.	
2	Konu Anlatımı: Akış Diyagramı çizimi. Sınıf-içi Uygulama (20 dk): Basit bir karar verme sürecini (örneğin, bir sayının çift veya tek olduğunu kontrol etme) akış diyagramı ile çizdirilmesi. Sınıf-içi Tartışma (10 dk): Akış diyagramlarının algoritma tasarımında ve problem çözme süreçlerinde nasıl bir kolaylık sağladığının tartışılması.	1. Akış diyagramı sembollerinin ve temel yapısının öğrenilmesi ve basit bir karar verme sürecinin akış diyagramıyla nasıl temsil edileceğine dair örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 1.	
3	Konu Anlatımı: Programlama Araçları, Değişkenler ve Sabitler. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde basit bir değişken ve sabit tanımlama alıştırması yaptırılması. Sınıf-içi Tartışma (10 dk): Değişkenlerin ve sabitlerin programlama araçlarındaki rolü ve farklı programlama dillerindeki kullanım örneklerinin tartışılması. Kısa Sınav 1 (30 dk): Kısa sınavın yapılacağı haftaya kadar işlenen konuların tümünü kapsayan sorular.	1. Dev-C++ ortamında temel programlama araçlarının, değişken ve sabit kavramların öğrenilmesi ve farklı programlama dillerindeki değişken/sabit kullanımlarına dair örneklerin gözden geçirilmesi. Kaynak: Ders Kitabı Konu 2.	
4	Konu Anlatımı: Giriş-Çıkış İşlemleri, Operatörler. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde kullanıcıdan veri alan ve temel aritmetik operatörlerle işlem yapan basit bir program yazdırılması. Sınıf-içi Tartışma (10 dk): Giriş-çıkış işlemlerinin ve operatörlerin programlama süreçlerinde veri işleme ve kullanıcı etkileşimi açısından öneminin tartışılması.	1. Dev-C++ ortamında giriş-çıkış işlemlerinin ve temel aritmetik operatörlerin öğrenilmesi ve bu işlemlerin nasıl kullanıldığına dair örnek programların incelenmesi. Kaynak: Ders Kitabı Konu 2.	
5	Konu Anlatımı: Karar Yapıları. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, basit bir karar yapısı (if-else) kodlatılması. Sınıf-içi Tartışma (10 dk): Karar yapılarının program akışını nasıl yönlendirdiği ve gerçek dünya problemlerinde kullanımının tartışılması.	1. Dev-C++ ortamında if-else karar yapılarının sözdiziminin ve mantığının öğrenilmesi ve gerçek dünya problemlerinde karar yapılarının kullanımına dair örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 3.	



6	Konu Anlatımı: Döngü Kontrolleri. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, bir for veya while döngüsü yazdırılması. Sınıf-içi Tartışma (10 dk): Döngü kontrollerinin tekrarlayan görevleri otomatikleştirmedeki rolü ve farklı senaryolardaki (örneğin, veri işleme) kullanımının tartışılması. Kısa Sınav 2 (30 dk): Kısa sınavın yapılacağı haftaya kadar işlenen konuların tümünü kapsayan sorular.	1. Dev-C++ ortamında for-while döndülerinin sözdiziminin ve kullanımının öğrenilmesi ve döngülerin veri işleme gibi senaryolarda nasıl kullanıldığına dair örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 4.
7	Konu Anlatımı: Döngü Kontrolleri üzerine uygulamalar. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, belirli bir koşula uyan elemanları filtreleyen bir döngü uygulaması yazdırılması. Sınıf-içi Tartışma (10 dk): Döngü kontrollerinin karmaşık problemlerde (örneğin, veri analizi veya oyun programlama) nasıl optimize edilebileceği ve performans etkilerinin tartışılması.	1. Dev-C++ ortamında döngü yapılarının (for, while) kullanılarak veri filtreleme örneklerinin incelenmesi ve döngü optimizasyonu ile performans konularına dair temel bilgilerin gözden geçirilmesi. Kaynak: Ders Kitabı Konu 4.
8	Ara Sınav 1	Sınav haftasına kadar işlenen konuların tümünün tekrar edilmesi.
9	Konu Anlatımı: Tek Boyutlu Diziler. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, tek boyutlu bir dizi oluşturarak elemanlarını kullanıcıdan alan ve diziyi kullanarak hesaplama yapan bir program yazdırılması. Sınıf-içi Tartışma (10 dk): Tek boyutlu dizilerin veri saklama ve işleme süreçlerinde (örneğin, liste yönetimi veya istatistiksel analiz) nasıl kullanıldığının tartışılması.	1. Dev-C++ ortamında tek boyutlu dizi tanımlama ve kullanma syntax'ının öğrenilmesi ve dizilerin liste yönetimi veya istatistiksel analiz gibi alanlarda nasıl kullanıldığına dair örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 5.
10	Konu Anlatımı: Çok Boyutlu Diziler. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, bir çok boyutlu dizi (matris) oluşturarak elemanlarını kullanıcıdan alan ve matrisi kullanarak hesaplama yapan bir program yazdırılması. Sınıf-içi Tartışma (10 dk): Çok boyutlu dizilerin farklı alanlarda nasıl kullanıldığının tartışılması.	1. Dev-C++ ortamında çok boyutlu dizi (matris) tanımlama ve kullanma syntax'ının öğrenilmesi ve matrislerin görüntü işleme, bilimsel hesaplamalar veya oyun geliştirme gibi alanlarda kullanımına dair örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 5.
11	Konu Anlatımı: Matris İşlemleri. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, matris oluşturarak toplama veya çarpma işlemini gerçekleştiren bir program yazdırılması. Sınıf-içi Tartışma (10 dk): Matris işlemlerinin lineer cebir, veri bilimi ve yapay zeka uygulamalarındaki öneminin ve Pratik kullanım alanlarının tartışılması.	1. Dev-C++ ortamında matris tanımlama ve temel matris işlemleri (toplama, çarpma) için gerekli syntax'ın öğrenilmesi ve matrislerin lineer cebir, veri bilimi veya yapay zeka alanlarındaki uygulamalarına dair örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 8.
12	Konu Anlatımı: Değer Döndürmeyen Alt Programlar. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde belirli bir mesajı ekrana yazdıran veya basit bir hesaplama yapıp sonucu gösteren bir değer döndürmeyen alt program (fonksiyon) yazdırılması. Sınıf-içi Tartışma (10 dk): Değer döndürmeyen alt programların kod modülerliğini artırmadaki rolü ve farklı programlama senaryolarındaki kullanımının tartışılması.	1. Dev-C++ ortamında değer döndürmeyen fonksiyonların (void fonksiyonlar) syntax'ının ve kullanımının öğrenilmesi ve kod modülerliği kavramına dair temel örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 6.



13	Konu Anlatımı: Değer Döndürmeyen Alt Programlar üzerine uygulamalar. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, bir dizinin elamanlarını ekrana yazdıran veya belirli bir koşula göre elemanları işleyen bir değer döndürmeyen alt program (fonksiyon) yazdırılması. Sınıf-içi Tartışma (10 dk): Değer döndürmeyen alt programların kod organizasyonunda ve tekrar eden görevlerin sadeleştirilmesinde sağladığı avantajların tartışılması.	1. Dev-C++ ortamında değer döndürmeyen fonksiyonların dizilerle kullanımına dair syntax'ın öğrenilmesi ve kod organizasyonu ile tekrar eden görevlerin sadeleştirilmesi için örnek uygulamaların incelenmesi. Kaynak: Ders Kitabı Konu 6.
14	Konu Anlatımı: Değer Döndüren Alt Programlar. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, bir sayının karesini hesaplayan veya bir dizinin en büyük elemanını döndüren bir değer döndüren alt program (fonksiyon) yazdırılması. Sınıf-içi Tartışma (10 dk): Değer döndüren alt programların program akışında veri işleme ve modülerlik açısından nasıl bir kolaylık sağladığının tartışılması. Kısa Sınav 4 (60 dk): Kısa sınavın yapılacağı haftaya kadar işlenen konuların tümünü kapsayan sorular.	1. Dev-C++ ortamında değer döndüren fonksiyonların (örneğin, return ifadesi) syntax'ının ve kullanımını öğrenilmesi ve veri işleme ile modülerlik açısından bu fonksiyonların sağladığı avantajlara dair örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 6.
15	Konu Anlatımı: Değer Döndüren Alt Programlar üzerine uygulamalar. Sınıf-içi Uygulama (20 dk): Dev-C++ programlama dilinde, bir dizinin ortalamasını hesaplayan veya bir stringin tersini döndüren bir değer döndüren alt program (fonksiyon) yazdırılması. Sınıf-içi Tartışma (10 dk): Değer döndüren alt programların karmaşık problemlerde (örneğin, veri analizi veya algoritma tasarımı) sonuçları yeniden kullanma ve kod verimliliği açısından sağladığı faydaların tartışılması.	1. Dev-C++ ortamında değer döndüren fonksiyonların diziler ve stringlerle kullanımına dair syntax'ın öğrenilmesi ve veri analizi ile algoritma tasarımında bu fonksiyonların kod verimliliğine katkısını gösteren örneklerin incelenmesi. Kaynak: Ders Kitabı Konu 7.
16	Final	İşlenen konuların tümünün tekrar edilmesi.

AKTS İŞYÜKÜ TABLOSU

Etkinlikler	Sayı	Süresi (Saat)	Toplam İşyükü
Ders Saati	14	3	42
Laboratuvar			
Uygulama (sözlü sınav)			
Arazi Çalışmaları			
Sınıf Dışı Ders Çalışması	14	2	28
Derse Özgü Staj			
Ödev			
Küçük Sınavlar/Stüdyo Kritiği	4	8	32
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:			147
Toplam İş yükü / 30(s):			4.9
AKTS Kredisi:			5



COURSE INFORMATION FORM

FACULTY / GRADUATE SCHOOL	Faculty of Arts and Sciences
DEPARTMENT / PROGRAMME	Statistics
TITLE OF COURSE	Introduction to Computer Programming
CODE	IST1081
LOCAL CREDIT	3
ECTS	5
LECTURE HOUR / WEEK	3
PRACTICAL HOUR / WEEK	0
LABORATORY HOUR / WEEK	0
PREREQUISITE	None
SEMESTER	Fall
COURSE LANGUAGE	English, Turkish
LEVEL OF COURSE	First Cycle
COURSE TYPE	Required @ Bachelor Programme in Statistics
COURSE CATEGORY	Core Courses
MODE OF DELIVERY	Face-to-Face
OWNER ACADEMIC UNIT	Department of Statistics
COURSE COORDINATOR	Ersoy ÖZ
ASSISTANT(S)	Zuhal Fatma CELLAT
COURSE OBJECTIVES	The objective of this course is to enable students to comprehend and apply the fundamental principles of computer programming. Students will learn to design and utilize flowcharts to grasp programming logic, effectively implement input-output operators, variables, conditional statements, and loop structures, and develop problem-solving skills through arrays, matrix operations, and function. Additionally, the course aims to enhance student's algorithmic thinking abilities, equipping them with the foundational programming skills necessary to address real-world problems.
COURSE CONTENT	Algorithms; Flowcharts; Operators; Input-Output Operators; Conditional Statements; Loop Structures; Array Operations; Matrix Operations; and Subroutines.
RECOMMENDED OR REQUIRED READINGS	Coursebook: ALP, Selçuk; KİLİTCİ, Arzu. <i>C Programlama Dili</i> . 1st edition, Umuttepe Yayınları, 2015. Required Reading: ÇÖLKESEN, Rıfat. <i>Veri Yapıları ve Algoritmalar</i> . 12th edition. Papatya Yayıncılık, 2002.



Course Learning Outcomes

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

1. Explain the fundamental principles of computer programming and develop program code.
2. Read, explain and design flowcharts and apply them in problem-solving processes.
3. Utilize input-output operators, variables, and conditional statements effectively.
4. Explain loop structures and apply them efficiently in programming problems.
5. Demonstrate proficiency in using functions and perform array and matrix operations.

EVALUATION SYSTEM

Activities	Number	Percentage of Grade
Attendance/Participation		
Laboratory		
Application (Oral Examination)		
Field Work		
Special Course Internship (Work Placement)		
Quizzes/Studio Critics: • Content: Questions covering all topics addressed up to the week of the quiz. • Format: Face-to-face. Written (60 minutes) or multiple-choice quiz (30 minutes). • Detailed Assessment Criteria: - Ability to solve problems related to applications covered in class. - 1 quiz: 20% - 3 quizzes: 15% (5% + 5% + 5%)	4	35%
Homework Assignments		
Presentations/Jury		
Project		
Seminar/Workshop		
Midterms: • Content: Comprehensive questions covering all topics addressed up to the exam week. • Format: Face-to-face. Written (90 minutes) or multiple-choice quiz (60 minutes). • Detailed Assessment Criteria: - Demonstration of understanding of the fundamental concepts of the course. - Ability to solve problems related to theoretical topics. - Ability to carry out theoretical reasoning processes.	1	25%
Final: • Content: Comprehensive questions covering the entire content of the course • Format: Face-to-face. Written (90 minutes) or multiple-choice quiz (60 minutes). • Detailed Assessment Criteria: - Ability to apply advanced problem-solving skills. - Demonstration of a thorough understanding of all topics covered in the course.	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: Algorithms. Quick Practice (20 min): Conducting a step-by-step practice of sorting a sequence of numbers using a simple sorting algorithm. In-Class Discussion (10 min): Exploring the applications of algorithms in daily life and various disciplines, such as computer science, engineering, and data analysis.	1. Reviewing fundamental algorithm concepts and preliminary knowledge of sorting algorithms, as well as examining examples of their practical applications. Source: Coursebook Chapter 1.
2	Lecture: Flow charts. Quick Practice (20 min): Having students draw a flowchart for a simple decision-making process (e.g., checking whether a number is even or odd). In-Class Discussion (10 min): Discussing how flowcharts facilitate algorithm design and problem-solving processes.	1. Learning the symbols and basic structure of flowcharts and examining examples of how a simple decision-making process is represented using a flowchart. Source: Coursebook Chapter 1.
3	Lecture: Programming tools, variables, and contents. Quick Practice (20 min): Conducting an exercise in the Dev-C++ to define simple variables and constants. In-Class Discussion (10 min): Discussing the role of variables and constants in programming tools and their usage examples across different programming languages. Quiz 1 (30 min): Questions covering all topics addressed up to the week of the quiz.	1. Learning the fundamental programming tools, variable, and constant concepts in the Dev-C++ environment, and reviewing examples of variables and constant usage in different programming languages. Source: Coursebook Chapter 2.
4	Lecture: Input Output operations, operators. Quick Practice (20 min): Writing a simple program in the Dev-C++ that takes input from the user and perform operations using basic arithmetic operators. In-Class Discussion (10 min): Discussing the significance of input-output operations and operators in programming processes, particularly in terms of data processing and user interaction.	1. Learning input-output operations and basic arithmetic operators in the Dev-C++ environment and reviewing example programs demonstrating their usage. Source: Coursebook Chapter 2.
5	Lecture: Decision structures. Quick Practice (20 min): Writing a simple decision structure (if-else) in the Dev-C++. In-Class Discussion (10 min): Discussing how decision structures direct program flow and their applications in real-world problem-solving.	1. Learning the syntax and logic of if-else decision structures in the Dev-C++ environment and reviewing examples of their use in real-world problem-solving scenarios. Source: Coursebook Chapter 3.
6	Lecture: Loop Controls. Quick Practice (20 min): Writing a for of while loop in the Dev-C++. In-Class Discussion (10 min): Discussing the role of loop controls in automating repetitive tasks and their applications in various scenarios, such as data processing. Quiz 2 (30 min): Questions covering all topics addressed up to the week of the quiz.	1. Learning the syntax and usage of for and while loops in the Dev-C++ environment and reviewing examples of how loops are applied in scenarios such as data processing. Source: Coursebook Chapter 4.



7	Lecture: Loop Controls applications. Quick Practice (20 min): Writing a loop application in the Dev-C++ that filters elements based on a specific condition. In-Class Discussion (10 min): Discussing how loop controls can be optimized in complex problems (e.g., data analysis or game programming) and their impact on performance.	Examining examples of data filtering using loop structures (for, while) in the Dev-C++ environment and reviewing fundamental concepts related to loop optimization and performance. Source: Coursebook Chapter 4.
8	Midterm 1	Review of all topics covered up to the exam week.
9	Lecture: Sequences. Quick Practice (20 min): Writing a program in the Dev-C++ that creates a one-dimensional array, takes its elements as input from the user, and performs a calculation using the array. In-Class Discussion (10 min): Discussing the role of one-dimensional arrays in data storage and processing, such as in list management or statistical analysis.	Learning the syntax for defining and using one-dimensional arrays in the Dev-C++ environment and reviewing examples of their applications in areas such as list management or statistical analysis. Source: Coursebook Chapter 5.
10	Lecture: Arrays. Quick Practice (20 min): Writing a program in the Dev-C++ that creates a multi-dimensional array (matrix), takes its elements as input from the user, and performs a calculation using the matrix. In-Class Discussion (10 min): Discussing the applications of multi-dimensional arrays in various fields, such as image processing, scientific computations, or game development.	Learning the syntax for defining and using multi-dimensional arrays (matrices) in the Dev-C++ environment and reviewing examples of their applications in fields such as image processing, scientific computations, or game development. Source: Coursebook Chapter 5.
11	Lecture: Matrix. Quick Practice (20 min): Writing a program in the Dev-C++ that creates matrices and performs addition or multiplication operations. In-Class Discussion (10 min): Discussing the significance of matrix operations in linear algebra, data science, and artificial intelligence applications, along with their practical uses. Quiz 3 (30 min): Questions covering all topics addressed up to the week of the quiz.	Learning the syntax for defining matrices and performing basic matrix operations (addition, multiplication) in the Dev-C++ environment, and reviewing examples of matrix applications in fields such as linear algebra, data science, or artificial intelligence. Source: Coursebook Chapter 8.
12	Lecture: Non-Value Returning Subprograms. Quick Practice (20 min): Writing a non-value returning subprogram (function) in the Dev-C++ that displays a specific message on the screen or performs a simple calculation and shows the result. In-Class Discussion (10 min): Discussing the role of non-value returning subprograms in enhancing code modularity and their applications in various programming scenarios.	Learning the syntax and usage of non-value returning functions in the Dev-C++ environment and reviewing fundamental examples related to the concept of code modularity. Source: Coursebook Chapter 6.
13	Lecture: Non-Value Returning Subprograms applications. Quick Practice (20 min): Writing a non-value returning subprogram (function) in the Dev-C++ that prints the elements of an array to the screen or processes elements based on a specific condition. In-Class Discussion (10 min): Discussing the advantages of non-value returning subprograms in code organization and simplifying repetitive tasks.	Learning the syntax for using non-value returning subprograms with arrays in the Dev-C++ environment and reviewing example applications for code organization and simplifying repetitive tasks. Source: Coursebook Chapter 6.



14	Lecture: Value Returning Subprograms. Quick Practice (20 min): Writing a value-returning subprogram (function) in the Dev-C++ that calculates the square of a number or returns the largest element of an array. In-Class Discussion (10 min): Discussing how value-returning subprograms facilitate data processing and modularity in program flow. Quiz 4 (60 min): Questions covering all topics addressed up to the week of the quiz.	1. Learning the syntax and usage of value-returning functions (e.g., using the return statement) in the Dev-C++ environment and reviewing examples of their advantages in data processing and modularity. Source: Coursebook Chapter 8.
15	Lecture: Value Returning Subprograms applications. Quick Practice (20 min): Writing a value-returning subprogram (function) in the Dev-C++ that calculates the average of an array or returns the reverse of a string. In-Class Discussion (10 min): Discussing the benefits of value-returning subprograms in reusing and enhancing code efficiency in complex problems, such as data analysis or algorithm design.	1. Learning the syntax for using value-returning functions with arrays and string in the Dev-C++ environment and reviewing examples that demonstrate their contribution to code efficiency in data analysis and algorithm design. Source: Coursebook Chapter 7.
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	2	28
Special Course Internship (Work Placement)			
Homework Assignments			
Quizzes/Studio Critics	4	8	32
Project			
Presentations / Seminar			
Mid-Terms (Examination Duration + Examination Prep. Duration)	1	20	20
Final (Examination Duration + Examination Prep. Duration)	1	25	25
Total Workload:			147
Total Workload / 30(h):			4.9
ECTS Credit:			5



Ders Öğrenim Çıktısı & Program Çıktısı Matrisi

	<u>DÖC-1</u>	<u>DÖC-2</u>	<u>DÖC-3</u>	<u>DÖC-4</u>	<u>DÖC-5</u>
<u>PC-1</u> İstatistik biliminin temelini oluşturan teorik ve uygulamalı alanlardaki bilgilerini, alanla ilgili problemleri tanımlama, modelleme ve çözüm üretmede kullanabileceklerdir. / Use their theoretical and applied knowledge in the science of statistics to identify, model, and solve problems related to the field.					
<u>PC-2</u> İstatistiksel problemlerin belirlenmesi, uygun veri toplama yöntemlerinin seçilmesi, verinin düzenlenmesi ve yorumlanması süreçlerinde yeterlilik gösterecek ve bu problemlerin çözümünde gerekli analiz ve modelleme yöntemlerini seçip uygulayabileceklerdir. / Demonstrate competence in identifying statistical problems, selecting appropriate data collection methods, organizing and interpreting data, and select and apply necessary analysis and modelling methods to solve these problems.					
<u>PC-3</u> Herhangi bir olgu, süreç ya da ürünü istatistiksel bakış açısıyla analiz edip yorumlayabilecek ve karşılaştıkları sorunlara uygun modern istatistiksel yöntemlerle çözüm geliştirebileceklerdir. / Analyse and interpret a phenomenon, process, or product from a statistical perspective and develop solutions to encountered problems using appropriate modern statistical methods.	<u>4</u>	<u>4</u>	<u>4</u>	<u>5</u>	<u>5</u>
<u>PC-4</u> Disiplinlerarası bir yaklaşımla, farklı alanlarda edinmiş oldukları bilgileri sentezleyebileceklerdir. / Synthesise knowledge acquired from different disciplines through an interdisciplinary approach.					
<u>PC-5</u> İstatistik alanında edindikleri bilgi birikimlerini ve istatistiksel düşünme becerilerini, teorik istatistik, veri bilimi, finansal istatistik, sağlık bilimlerinde istatistik gibi disiplin-İçi ve disiplinlerarası uzmanlık alanlarında geliştirebileceklerdir. / Advance their acquired knowledge in statistics and statistical thinking skills in both disciplinary and interdisciplinary areas of specialisation such as theoretical statistics, data science, financial statistics, and statistics in health sciences.					
<u>PC-6</u> İstatistik 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 statistics for problem-solving, data analysis, and simulations.	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>	<u>5</u>
<u>PC-7</u> İstatistik ve ilgili alanlardaki bilimsel ve teknolojik gelişmeleri izleyebilecek, kariyer fırsatlarını değerlendirerek kişisel ve mesleki gelişim hedeflerini belirleyebilecek ve bu hedeflere	<u>4</u>	<u>4</u>	<u>4</u>	<u>4</u>	<u>4</u>



ulaşmak için hayat boyu öğrenme stratejilerini kullanabileceklerdir. / Follow scientific and technological developments in statistics and related fields, assess career opportunities, identify personal and professional development goals, and adopt lifelong learning strategies to achieve these goals.					
PC-8 Bilimsel araştırmalarını 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 İstatistik 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 statistics.					
PC-11 İstatistiksel konuları, teorileri, ispatları, araştırmaları ve problem çözümlerini, istatistiksel terminoloji kullanarak tüm paydaşlara Türkçe ve İngilizcede sözlü ve yazılı olarak etkili biçimde aktarabileceklerdir. / Effectively communicate statistical 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.					