

## GENEL TANIM / GENERAL DESCRIPTION

Ders Adı / Course Name	DATA STRUCTURES / DATA STRUCTURES	
Ders Kodu / Course Code	501002042010	
Ders Türü / Course Type		
Ders Seviyesi / Course Level	First Cycle / First Cycle	
Ders Akts Kredi / ECTS	5.00	
Haftalık Ders Saati (Kuramsal) / Course Hours For Week (Theoretical)	3.00	
Haftalık Uygulama Saati / Course Hours For Week (Objected)	1.00	
Haftalık Laboratuvar Saati / Course Hours For Week (Laboratory)	0.00	
Dersin Verildiği Yıl / Year	2	
Öğretim Sistemi / Teaching System	Face to Face / Face to Face	
Eğitim Dili / Education Language	English / English	
Ön Koşulu Olan Ders(ler) / Precondition Courses	Yok	None
Amacı / Purpose	Veri Yapıları kavramlarının anlaşılmasını ve öğrenilmesini sağlamak ve bu kavramları problemlerin çözümünde doğru ve etkin bir şekilde kullanabilme yeteneğini kazandırmak. Veri düzenleme ve işleminin problem çözümedeki yararlılıklarını göstermek.	The objective of the course is to give detailed information about Data Structures concepts and to give ability to use suitable data structures efficiently for solving problems. To show data organization and manipulation concepts are useful in problem solving.
İçeriği / Content	Yığıt, Kuyruklar, Listeler, Özyineleme, Ağaçlar (İkili Ağaçlar, Dengeli Ağaçlar, Yığınlar, Huffman Kodlama Ağacı), Algoritmaların Karşılaştırılması, Sıralama ve Arama, Çizgeler (Representation, Traversals, Shortest Path, Minimum Spanning Tree, ...)	Stack, Queues, Lists, Recursion, Trees (Binary Trees, Balanced Trees, Heaps, Huffman Encoding Tree), Comparison of Algorithms, Sorting and Search, Graphs (Representation, Traversals, Shortest Path, Minimum Spanning Tree, ...)
Önerilen Diğer Hususlar / Recommended Other Considerations	Yok	None
Staj Durumu / Internship Status	Yok	None

Kitabı / Malzemesi / Önerilen Kaynaklar / Books / Materials / Recommended Reading	Lafore, R., "Data Structures and Algorithms in Java", Second Edition, Sams Publishing, (2003). (Ders Kitabı)  Goodrich, M.T., Tamassia, R., "Data Structures and Algorithms in Java 6th Edition", Wiley, 2014. Drozdek, A., "Data Structures and Algorithms in Java 4th Edition", Cengage Learning Asia, 2013. Weiss, M.A., "Data Structures and Problem Solving Using Java (Fourth Edition)", Addison-Wesley, 2010.  C && Java ile Veri Yapılarına Giriş, Olcay Taner YILDIZ, Boğaziçi Üniversitesi Yayınevi, 2013. Algoritma Geliştirme ve Veri Yapıları, Dr. Rifat ÇÖLKESEN, Papatya Yayıncılık, 2011.	Lafore, R., "Data Structures and Algorithms in Java", Second Edition, Sams Publishing, (2003). (Textbook)  Goodrich, M.T., Tamassia, R., "Data Structures and Algorithms in Java 6th Edition", Wiley, 2014. Drozdek, A., "Data Structures and Algorithms in Java 4th Edition", Cengage Learning Asia, 2013. Weiss, M.A., "Data Structures and Problem Solving Using Java (Fourth Edition)", Addison-Wesley, 2010.  C && Java ile Veri Yapılarına Giriş, Olcay Taner YILDIZ, Boğaziçi Üniversitesi Yayınevi, 2013. Algoritma Geliştirme ve Veri Yapıları, Dr. Rifat ÇÖLKESEN, Papatya Yayıncılık, 2011.
Öğretim Üyesi (Üyeleri) / Faculty Member (Members)	Prof. Dr. Aybars UĞUR	

### ÖĞRENME ÇIKTILARI / LEARNING OUTCOMES

1	Nesne yönelimli ve yeni bir programlama dilindeki temel kavramları, veri tiplerini, yapıları ve komutları kavrayabilme ve kullanabilme. Çözümlerini, nesne yönelimli bir programlama dilinde ifade edebilme.	To be able to comprehend and use basic concepts, data types, structures and commands in an object-oriented and new programming language. Ability to express their solutions in an object-oriented programming language.
2	Problemlere veri yapıları kullanarak farklı, etkin ve hızlı çözümler üretebilme.	Ability to produce different, effective and quick solutions to problems using data structures.
3	Program yazarken gereken verileri, değişik biçimlere ve yapılaraya göre düzenleyebilme. Yazılım geliştirirken gereken veri yapılarını etkin olarak tasarlayabilme. Uygun veri yapılarını seçerek daha etkin algoritmalar yazabilme.	Organizing data in computer programs for different forms and structures. Designing data structures efficiently in software development. Writing efficient algorithms by choosing suitable data structures.
4	Programlama ortamlarındaki hazır veri yapılarını kullanabilme.	To use data structures provided by the software development environments.
5	Veri Yapıları içeren programlar yazabilme. Bu kapsamda proje hazırlayabilme ve rapor hazırlayarak yazılı olarak sunabilme.	Being able to write programs containing Data Structures. In this context, to be able to prepare projects and to present as project reports in written form.
6	Grup çalışmalarında, problem çözümlerine katkı verebilme.	To contribute the problem solutions in group studies.
7	Veri Yapıları konusunda öğrendiklerini, güncel problemlerin çözümünde kullanabilme.	To use data structures concepts in state-of-the-art problems.
8	Doğrusal veri yapılarını temsil edebilme, eleman ekleme, silme, arama, sıralama ve dolaşma gibi işlemleri yapabilme.	Being able to represent linear data structures, and make add, delete, search, sort and traverse operations.
9	Ağaç ve Çizge veri yapılarını temsil edebilme, eleman ekleme, silme ve üzerlerinde arama ve dolaşma gibi işlemleri yapabilme. Temel ağaç ve çizge algoritmalarını anlayabilme ve kodlayabilme.	To be able to represent Tree and Graph data structures, and perform operations such as adding and deleting, searching and traversing on them. To be able to understand and code basic tree and graph algorithms.

### HAFTALIK DERS İÇERİĞİ / DETAILED COURSE OUTLINE

Hafta / Week					
1	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Veri Yapılarına Giriş: Veri, Yapı ve Algoritma Kavramlarının önemi. C, C++, Java, Python, C# Dillerinin Karşılaştırılması ve Bilişim Sektöründeki Konular.	C# Ortamı Tanıtımı ve İlk Program	C# Programlama Platformunun Kurulumu ve Tanıtımı		
	Introduction to Data Structures: The importance of Data, Structure and Algorithm concepts. Comparison of C, C++, Java, Python, C# languages and places in software development area.	Introduction to C# Environment and First C# Program	Introducing C# Programming Environment		
2	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	C#/Java Programlama Dili'ne Giriş: C# Yazılım Geliştirme Araçlarının Tanıtımı. C# Dilindeki Temel Kavramlar. C# ile Uygulama yazma. Örnekler.	Metotlar, Diziler ve Hata Ayıklama	C# Dilinde Bilgisayar Başında Örnekler Yapılması		
	Introduction to C#/Java Programming Language: Introducing C# Software Development Tools. Basic Concepts in C#. Writing application in C#. Examples.	Methods, Arrays and Debugging	Writing and Debugging C# Programs and examples.		
3	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	C# Programlama Dilinde Nesne Yönelimli Programlama. Vektörlere İlişkin Örnekler. Grafik Kullanıcı Arayüzü (GUI) Oluşturma.	Sınıflar, Veri Yapıları ve Metotlar içeren Uygulama Geliştirme	C# Dilinde Sınıf, Vektör ve GUI Oluşturmaya İlişkin Örnekler Yapılması		
	Object-oriented programming in C#. Vector Examples. Simple Graphical User Interface (GUI) Design.	Application Development including Classes, Data Structures and Methods	Examples about Class, Vector and GUI in C#.		
4	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Nesne Yönelimli Bakış Açısı ile Yığın, Kuyruk ve Bağlılı Listeler: Yığın Tanımları ve İşlemleri, Kuyruk Tanımları ve İşlemleri, Kuyruk Tasarımı ve Kullanımı, Listeler Üzerindeki Bazı İşlemler ve Tanımları, Bağlılı Listeler, Öncelik Kuyrukları, Yığın ve Kuyrukların Bağlılı Liste Gerçekleştirmeleri.	Kuyruk ve ArrayList Uygulaması Geliştirme	Konu ile ilgili örnekler ve Proje Kontrol		
	Stack, Queues and Linked Lists in Object-Oriented View: Stack Definitions and Operations. Queue Definitions and Operations; Designing and using a Queue. Some Operations on Lists and Definitions; Linked Lists; Priority Queues; Implementation of Stack and Queues based on Linked Lists.	Queue and ArrayList Application Development	Examples about Stack, Queue and Linked Lists. Project evaluation.		
5	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	C#'ta Özyineleme: Yineleme ve Özyineleme, Faktöryel Fonksiyonu, Fibonacci Dizisi, İkili Arama, Hanoi Kuleleri, Özyineleme Zinciri.	Özyineleme Örneği	Konu ile ilgili örnekler ve Proje Kontrol		
	Recursion in C#: Iteration and Recursion, Factorial Function, Fibonacci numbers, Binary Search, Hanoi Towers, Recursion chain.	Recursion Example	Examples about Recursion Project Evaluation.		

	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
6	Ağaçlar: Tanımlar, İkili Ağaçlar ve İkili Ağaçlar Üzerindeki Dolaşma İşlemleri, İkili Arama Ağaçları, Ağaçlara Eleman Ekleme ve Silme İşlemleri, Ağaç Örnekleri.	İkili Arama Ağacı Uygulaması	Örnekler		
	Trees: Definitions, Binary Trees and Traversals, Binary Search Trees, Insertion and Delete Operations. Example Questions and Solutions.	Binary Search Tree Application	Examples		
7	Ağaçlar (Devam): Tanımlar, İkili Ağaçlar ve İkili Ağaçlar Üzerindeki Dolaşma İşlemleri, İkili Arama Ağaçları, Ağaçlara Eleman Ekleme ve Silme İşlemleri, Ağaç Örnekleri.	İkili Arama Ağacı Uygulaması	Örnekler		
	Trees (continued): Definitions, Binary Trees and Traversals, Binary Search Trees, Insertion and Delete Operations. Example Questions and Solutions.	Binary Search Tree Application	Examples		
8	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Ara sınav				
	Midterm exam				
9	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Algoritmaların Karşılaştırılması: Algoritmalarda Karmaşıklık ve Zaman Karmaşıklığı Analizi, Sıralama Tekniklerinin Etkinlikleri ve Analizi.	Sıralama Algoritması Uygulaması	Proje Kontrol		
	Comparison of Algorithms: Algorithm Complexity and Analysis of Time Complexity, Efficiencies of Sorting Algorithms.	Sorting Algorithm Application	Project Evaluation.		
10	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Çizgeler 1: Çizge Tanımları ve Uygulamaları, Bilgisayarda Gösterimleri, Komşuluk Matrisi, Çizgelerde Dolaşma, Derinlik Öncelikli Dolaşma, Genişlik Öncelikli Dolaşma.	Bir program içerisinde Çizge temsil etme	Konu ile ilgili örnekler		
	Graphs 1: Graph Definitions and Applications. Representation, Adjacency Matrix, Graph Traversals, Depth-First Traversal, Breadth-First Traversal.	Representing a Graph in a Program	Examples		
11	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Çizgeler 2: En Kısa Yol Algoritmaları (Dijkstra's, Floyd-Warshall) En Küçük Kapsayan Ağaç (MST) Algoritmaları (Prim's, Kruskal's).	Çizge Algoritmaları Programlama	Konu ile ilgili örnekler		
	Graphs 2: Shortest Path Algorithms (Dijkstra's, Floyd-Warshall), Minimum Spanning Tree (MST) Algorithms (Prim's, Kruskal's).	Programming Graph Algorithms	Examples		

	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
12	Yığın Ağacı, Dengeli Ağaçlar (AVL Ağacı, B-Tree, B+ Tree, ...), Trie	AVL Ağacına Eleman Ekleme	Konu ile ilgili örnekler		
	Heap Balanced Trees (AVL Trees, B-Tree, B+ Tree, ...) Trie	AVL Tree - Insertion	Examples		
13	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Hash Tablosu C#'ta Hazır Veri Yapıları ve Koleksiyonlar : "Stack", "ArrayList", "Hashtable", "sort", "binarySearch", ... Dinamik Programlama	Topluluklar	Konu ile ilgili örnekler		
	Hash Table C# Collections : "Stack", "ArrayList", "Hashtable", "sort", "binarySearch", ... Dynamic Programming	Collections	Examples		
14	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Güncel ve ileri düzeyli problem örnekleri, uygulamaları, NP-Tam Problemler	Proje Kontrolleri	Konu ile ilgili örnekler		
	Advanced and state-of-the-art problem examples and applications, NP-Complete Problems	Project Evaluation	Examples		
15	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Topolojik Sıralama, Huffman Kodlama Ağacı, Dönem Değerlendirmesi.	Proje Kontrolleri	Konu ile ilgili örnekler ve Proje Kontrol		
	Topological Sorting, Huffman Encoding Tree, Evaluation of the Course.	Project Evaluation	Examples and Project Evaluation		
16	Teorik Dersler / Theoretical	Uygulama	Lab	Öğretim Yöntem ve Teknikleri/Teaching Methods Techniques	Ön Hazırlık / Preliminary
	Final Sınavı				
	Final Exam				

## DEĞERLENDİRME / EVALUATION

Yarıyıl (Yıl) İçi Etkinlikleri / Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Ara Sınav / Midterm Examination	1	50
Ev Ödevi / Homework	4	50
Toplam / Total:	5	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		40
Yarıyıl (Yıl) Sonu Etkinlikleri / End Of Term (or Year) Learning Activities	Sayı / Number	Katkı Yüzdesi / Percentage of Contribution (%)
Final Sınavı / Final Examination	1	100
Toplam / Total:	1	100
Başarı Notuna Katkı Yüzdesi / Contribution to Success Grade(%):		60
Etkinliklerinin Başarı Notuna Katkı Yüzdesi(%) Toplamı / Total Percentage of Contribution (%) to Success Grade:		100
Değerlendirme Tipi / Evaluation Type:		

## İŞ YÜKÜ / WORKLOADS

Etkinlikler / Workloads	Sayı / Number	Süresi (Saat) / Duration (Hours)	Toplam İş Yüğü (Saat) / Total Work Load (Hour)
Ara Sınav / Midterm Examination	1	2.00	2.00
Final Sınavı / Final Examination	1	2.00	2.00
Derse Katılım / Attending Lectures	14	3.00	42.00
Laboratuvar / Laboratory	14	1.00	14.00
Rapor Hazırlama / Report Preparation	4	4.00	16.00
Proje Hazırlama / Project Preparation	4	6.00	24.00
Ara Sınav İçin Bireysel Çalışma / Individual Study for Mid term Examination	1	15.00	15.00
Final Sınavı için Bireysel Çalışma / Individual Study for Final Examination	1	21.00	21.00
Okuma / Reading	14	1.00	14.00
<b>Toplam / Total:</b>	<b>54</b>	<b>55.00</b>	<b>150.00</b>
Dersin AKTS Kredisi = Toplam İş Yüğü (Saat) / 30.00 (Saat/AKTS) = 150.00/30.00 = 5.00 ~ 5.00 / Course ECTS Credit = Total Workload (Hour) / 30.00 (Hour / ECTS) = 150.00 / 30.00 = 5.00 ~ 5.00			

## PROGRAM VE ÖĞRENME ÇIKTISI / PROGRAM LEARNING OUTCOMES

Öğrenme Çıktıları / Learning Outcomes	Program Çıktıları / Program Outcomes										
	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.1.6	1.1.7	1.1.8	1.1.9	1.1.10	1.1.11
1.Nesne yönelimli ve yeni bir programlama dilindeki temel kavramları, veri tiplerini, yapıları ve komutları kavrayabilme ve kullanabilme. Çözümlerini, nesne yönelimli bir programlama dilinde ifade edebilme. / To be able to comprehend and use basic concepts, data types, structures and commands in an object-oriented and new programming language. Ability to express their solutions in an object-oriented programming language.	5	3	3	3	2						

2.Problemlere veri yapıları kullanarak farklı, etkin ve hızlı çözümler üretebilme. / Ability to produce different, effective and quick solutions to problems using data structures.	5	5	3	4	4					2	
3.Program yazarken gereken verileri, değişik biçimlere ve yapılaraya göre düzenleyebilme. Yazılım geliştirirken gereken veri yapılarını etkin olarak tasarlayabilme. Uygun veri yapılarını seçerek daha etkin algoritmalar yazabilme. / Organizing data in computer programs for different forms and structures. Designing data structures efficiently in software development. Writing efficient algorithms by choosing suitable data structures.	5	4	4	4	3					2	
4.Programlama ortamlarındaki hazır veri yapılarını kullanabilme. / To use data structures provided by the software development environments.	3	4	2	3	2						
5.Verit Yapıları içeren programlar yazabilme. Bu kapsamda proje hazırlayabilme ve rapor hazırlayarak yazılı olarak sunabilme. / Being able to write programs containing Data Structures. In this context, to be able to prepare projects and to present as project reports in written form.	3	3	3	3	3	2	2			2	
6.Grup çalışmalarında, problem çözümlerine katkı verebilme. / To contribute the problem solutions in group studies.	3	3	3	3	3					2	
7.Verit Yapıları konusunda öğrendiklerini, güncel problemlerin çözümünde kullanabilme. / To use data structures concepts in state-of-the-art problems.	5	5	4	4	4			2		3	

8.Doğrusal veri yapılarını temsil edebilme, eleman ekleme, silme, arama, sıralama ve dolaşma gibi işlemleri yapabilme. / Being able to represent linear data structures, and make add, delete, search, sort and traverse operations.	3	3	3	3							
9.Ağaç ve Çizge veri yapılarını temsil edebilme, eleman ekleme, silme ve üzerlerinde arama ve dolaşma gibi işlemleri yapabilme. Temel ağaç ve çizge algoritmalarını anlayabilme ve kodlayabilme. / To be able to represent Tree and Graph data structures, and perform operations such as adding and deleting, searching and traversing on them. To be able to understand and code basic tree and graph algorithms.	3	4	4	4							

Katkı Düzeyi / Contribution Level : 1-Çok Düşük / Very low, 2-Düşük / Low, 3-Orta / Moderate, 4-Yüksek / High, 5-Çok Yüksek / Very high