



YAKIN DOĐU ÜNİVERSİTESİ DIŐA AÇIK DERSLER KOORDİNATÖRLÜĐÜ

Okul/Fakülte: MÜHENDİSLİK FAKÜLTESİ

Bölüm/Program: BİLİŐİM SİSTEMLERİ MÜHENDİSLİĐİ - İNGİLİZCE

Ders Dili:	English	Ders Kodu:	CS221
DersTürkçe İsmi:	VERİ YAPILARI VE ALGORİTMALAR		
Ders İngilizce İsmi:	DATA STRUCTURES AND ALGORITHMS		
Dersi Verecek:	Yard .Doç. Dr. Hüseyin SEVAY		
Dersin Türü:	ZORUNLU	Dersin Seviyesi:	
Yıl	2	Semester	3
Ders Kredisi:	3	AKTS Kredisi:	6
Teori(saat/hafta):	4,00	Uygulama(saat/hafta):	0,00
		Laboratuar(saat/hafta):	0,00
Dersin İçeriĐi:			

This course comprises an introductory exploration into the design and implementation of Abstract Data Types (ADTs) along with the study of algorithm design and complexity analysis. Even though the discussions during lectures about ADTs are language independent, this course uses Python, a very high-level general programming language, to implement these ideas using object-oriented programming. This class starts with a brief introduction to object-oriented programming.

Course Objectives

1. Teach the fundamental principles of computer programming via the Python programming language
2. Teach a very basic introduction to object-oriented programming using Python
3. Teach how to write algorithms, and implement and test those algorithms
4. Convey that using fundamental programming-related tools well is essential to programming efficiently. These tools include the syntax of the Python language, a capable text editor suited for programming such as VIM/GVIM, the Python debugger, and the GNU/Linux operating system as a productive learning environment
5. Teach problem solving techniques and teach how to solve problems using programming
6. Teach that testing is an essential part of programming
7. Encourage students to incorporate all tools mentioned into their learning of programming.

Course Outcomes

Upon the completion of this course, a student will be able to:

1. Learn basic programming concepts and importance of testing software
2. Develop an understanding of how real-life problems can/may be solved using programming
3. Understand the important role programming plays in our lives
4. Write basic Python programs mainly using built-in data structures to solve problems
5. Develop basic algorithms in pseudo-code format for given problems
6. Write test code to test functions and methods in Python
7. Use input files and write output files in binary/text format
8. Recognize the breadth of Python libraries in solving many real-life problems
9. Understand basic object-oriented programming concepts
10. Understand the difference between mutable and immutable data types
11. How data structures are used to represent objects in memory
12. Understand how the stack is used in executing functions and methods
13. Understand scoping rules
14. Recognize the trade-off between space and time
15. Understand the importance of abstraction
16. Understand the concept of refactoring and code reuse
17. Understand the importance of right data structure selection for implementation

Öğrenme Kazanımları:

İlgili kavramları/kuramları anlayabilecek
İlgili kavram/kuramların geçerliliğini tartışabilecek
İlgili kavram/kuramların, gerçek hayattaki muhtemel uygulamalarını tartışabilecek ve öneriler sunabilecek
İlgili kavram/kuramları gerçek hayata/verilen diğer durumlara/vakalara uygulayabilecek
İlgili kavram/kuramların gerçek hayatta var olan uygulamalarını eleştirel olarak analiz edebilecek
Farklı kavram ve kuramları kendi özgün yaklaşımlarını yaratılmak için sentezleyebilecek
İlgili kavramlarla ilgili özgün bir yaklaşım geliştirebilecek
Sunum(lara)a hazırlık
Verilen ölçütlere göre kendi çalışmalarını değerlendirebilecek
Verilen ölçütlere göre arkadaşlarının çalışmalarını değerlendirebilecek
Yeni yaklaşım geliştirebilecek/yaratabilecek
Verilen parametreler çerçevesinde yeni bir ürün geliştirebilecek/yaratabilecek
Verilen çalışmayı bağımsızca yürütebilecek
Verilen bir çalışma üzerinde grup halinde çalışabilecek
İlgili kavramları sayabilecek ve açıklayabilecek
Öğrenmenin değerini takdir edecek
Akademik bir makale üretmek için seçilen kaynak gösterme biçiminin ilkelerini uygulayabilecek
Hedeflenen becerileri geliştirebilecek
.

Dersin Amaçları:

Belirlenen kavram(ları) açıklamak/anlatmak
İlgili kavram(lar)la alakalı farkındalık yaratmak ve bunu geliştirmek.
Belirlenen kavram(lar)ın geçerliliğini tartışmak.
Seçilen/belirlenen becerileri geliştirmek
Seçilen konuların derinlemesine/detaylı bir şekilde incelemek
Belirlenen kavram/kuram/konularla ilgili öğrencilerin var olan bilgilerini geliştirmek
Seçilen kavramlar bağlamında öğrencilerin fikirlerini/bilgilerini/kavrayışlarını geliştirmek
Belirlenen kavram/kuram/konularla ilgili öğrencilerle var olan bilgilerini yenilemek
Yeniliği teşvik etmek
Eleştirel düşünceyi geliştirmek

Öğrenci İş Yüğü:

Ara sınav
Final sınavı
Sınıf içi tartışma(lar)
Ödev(ler)
Ders planlama
Materyal uyarlama
Materyal geliştirme

AKTS Formülü:

Activities
Quantity
Duration (hour)
Total Workload
Course duration in class (including Exam weeks)
16
4
64
Labs and Tutorials
Homework
5
3
15
Project/Presentation/Report
E-learning activities
Quizzes
Midterm Examination Study
1
5
5
Final Examination Study
1
5
5
Self Study
40
2.5
100
Total Workload (hours)
189
Total Workload / 30 (hours)
6.3
ECTS Credit of the Course
6

Kaynaklar:	Data Structures and Algorithms Using Python, Rance D. Necaise, 2011, John Wiley & Sons, other textbooks and written material
Değerlendirme:	This course comprises an introductory exploration into the design and implementation of Abstract Data Types (ADTs) along with the study of algorithm design and complexity analysis. Even though the discussions during lectures about ADTs are language independent, this course uses Python, a very high-level general programming language, to implement these ideas using object-oriented programming. This class starts with a brief introduction to object-oriented programming.
İşe Yerleştirme(Staj):	Introduction to OOP - 1
Ön Koşul Ders Kodları:	CS122
1. Hafta (19 – 23 Eylül)	Introduction to OOP - 2

2. Hafta (26 – 30 Eylül)	ADTs
3. Hafta (3 – 7 Ekim)	Arrays
4. Hafta (10 – 14 Ekim)	Sets & Maps
5. Hafta (17 – 21 Ekim)	Algorithm Analysis
6. Hafta (24 – 28 Ekim)	Searching & Sorting
7. Hafta (31 - 4 Kasım)	Linked Structures
8. Hafta (7 - 11 Kasım)	Stacks
9. Hafta (14 – 18 Kasım)	Queues
10. Hafta (21 – 25 Kasım)	Advanced Linked Lists
11. Hafta (28 - 2 Aralık)	Recursion
12. Hafta (5 – 9 Aralık)	Hash tables
13. Hafta (12 -16 Aralık)	Advanced Sorting
14. Hafta (19 - 23 Aralık)	Binary Trees
15. Hafta (24 – 30 Aralık)	FİNAL SINAVLARI HAFTASI
16. Hafta	Search Trees
17. Hafta	
18. Hafta	
19. Hafta	
20. Hafta	
21. Hafta	
22. Hafta	
23. Hafta	
24. Hafta	
25. Hafta	
26. Hafta	
27. Hafta	
28. Hafta	
