TED UNIVERSITY, COURSE SYLLABUS

Faculty	Engineering	**Department**	Computer Engineering

Course Code & Number	CS 501	**Course Title**	Design&Analysis of Algorithms
Type of Course	☑ Compulsory ☐ Elective	**Semester**	☑Fall ☐ Spring ☐ Summer
Course Credit Hours	(3+0+0) 3	**Number of ECTS Credits**	7.5
Pre-requisite		**Co-requisite**	
Mode of Delivery	☑ Face-to-face ☑ Distance learning	**Language of Instruction**	☑ English ☐ Turkish
Course Coordinator	Dr. Ulaş GÜLEÇ	**Course Lecturer(s)**	Dr. Ulaş GÜLEÇ
Required Reading	- Algorithm Design by Jon Kleinberg and Éva Tardos. Addison-Wesley, 2005. - Algorithms 4/e by Robert Sedgewick and Kevin Wayne. Addison-Wesley Professional, 2011.	**Recommended Reading**	- Introduction to Algorithms, Third Edition by Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. MIT Press, 2009. - M. A. Weiss, Data Structures and Algorithm Analysis in Java, 3rd Edition, Pearson, 2012.

Course Catalog Description	Algorithm Design and Analysis, Sorting and Searching, Search Trees, Graphs, Shortest Paths, Maximum Flow and Minimum Cut, Tries, Reductions, Intractability, Algorithm Design Paradigms, Divide and Conquer, Greedy Algorithms, Backtracking, Genetic Algorithms
Course Objectives	The general objective of this course is to analyze the asymptotic performance of algorithms and understand algorithm design techniques. This course introduces familiarity with major algorithms and data structures make students apply important algorithmic design paradigms and methods of analysis. This course introduces the use of graphs in problem solving and algorithm development and describes how to develop algorithms using advanced graph data structures. Another objective of the course is to introduce different algorithm-design techniques, such as greedy, divide-and-conquer, and dynamic programming techniques, to solve particular problems. Synthesize efficient algorithms in common engineering design situations.

Course Learning Outcomes	Upon successful completion of this course, students will be able to: 1. Analyze worst-case running times of algorithms using asymptotic analysis. 2. Understand graph-processing concepts to employ the major graph algorithms and their analyses in engineering problems. 3. Recognize network flow problems, modeling and solving related problems. 4. Understand data structures used in mathematical modelling of complex problems. 5. Understand divide-and-conquer, dynamic-programming, and greedy algorithm paradigms to recite algorithms that employ these paradigms. 6. Analyze algorithms and data structure to measure their space complexity. 7. Recognize evolutionary algorithms and their applications.

Learning Activities & Teaching Methods[1]	☒ Brainstorming ☒ Case Study/Scenario Analysis ☒ Collaborating ☐ Concept Mapping ☒ Demonstrating ☐ Discussions / Debates ☐ Drama / Role Playing ☒ Experiments ☐ Field Trips ☒ Guest Speakers	☒ Hands-on Activities ☐ Inquiry ☐ Microteaching ☒ Oral Presentations / Reports ☐ Peer Teaching ☐ Predict-Observe-Explain ☒ Problem Solving ☐ Questioning ☒ Reading	☐ Scaffolding / Coaching ☐ Seminars ☐ Service Learning ☐ Simulations & Games ☒ Telling / Explaining ☐ Think-Pair-Share ☐ Video Presentations ☒ Web Searching ☐ Other(s):……

Assessment Methods & Criteria[2]	☒ Case Studies / Homework	(30%)	☒ Presentation (Oral, Poster)	(10%)
	☐ Lab Assignment	(...%)	☒ Project	(20%)
	☐ Observation	(...%)	☐ Quiz	(...%)
	☐ Oral Questioning	(...%)	☐ Self-evaluation	(...%)
	☐ Peer Evaluation	(...%)	☒ Test/Exam	(40%)
	☐ Performance Project (Written, Oral)	(...%)	☐ Other(s):……	(...%)
	☐ Portfolio	(...%)		

Student Workload[3]	☒ Case Study Analysis	(26 hrs)	☐ Online Discussion	(... hrs)
	☒ Course Readings	(10 hrs)	☒ Oral Presentation	(10 hrs)
	☐ Debate	(... hrs)	☐ Poster Presentation	(... hrs)
	☐ Demonstration	(... hrs)	☐ Report on a Topic	(... hrs)
	☒ Exams/Quizzes	(20 hrs)	☒ Research Review	(30 hrs)
	☐ Field Trips/Visits	(... hrs)	☐ Resource Review	(... hrs)
	☒ Hands-on Work	(70 hrs)	☒ Team Meetings	(10 hrs)
	☐ Lab Applications	(... hrs)	☐ Web Designs	(... hrs)
	☒ Lectures	(42 hrs)	☐ Work Placement	(... hrs)

[1] Multiple options possible.
[2] Multiple options possible. A percentage must be stated for the selected assessment method & criteria.
[3] Multiple options possible. The student workload is found by multiplying the number and duration (hour) of the activity involved.

	☐ Mock Designs	(... hrs)	☐ Workshop	(... hrs)
	☐ Observation	(... hrs)	☐ Other(s):……	(... hrs)
			Total Workload[4]	218
Prepared By[5]	**Dr. Haydar ÇUKURTEPE**		**Date**	04/24/2021
Revised By[6]			**Rev. Date**	9/20/2022

GRADING
A. Midterm [30%]
One midterm exam that is worth 30% of the overall course grade.
B. Project [30%]
You will be given a project that includes designing and analyzing an algorithm. The projects will be presented in the class. Presentations will be evaluated separately (%10 of 30)
C. Final Exam [40%]
One Final exam that is worth 40% of the overall course grade.

[4] Computing the ECTS credits of a course: Total workload / 25 or 30 hours = ECTS credit and1 ECTS credit = 25-30 hours
[5] It is the first person to prepare the course profile form and the first preparation date.
[6] It is the person who revised the course profile form and the date of revision. It will be used if the course profile form is revised. In the new course proposal, this field will be left blank.

COURSE POLICIES

Attendance

Attending at least **70%** of all lectures is **mandatory**.

Missed Work

Makeups for midterm exam will be provided if the student can provide a legal document confirming a life threatening health issue at the time of the exam, or with the consensus of the faculty.

Late Assignment Submission Policy

Late submissions will be graded with penalty.

Extra Credit

Extra credits will not be offered.

Assignment Rules

All assignment works must be done individually. A student can submit only one work. In case of multiple submissions, only the latest submission will be considered. Students cannot submit work on other students' behalf.

Plagiarism

All of the following are considered plagiarism:

- turning in someone else's work as your own
- copying words or ideas from someone else without giving credit
- failing to put a quotation in quotation marks
- giving incorrect information about the source of a quotation
- changing words but copying the sentence structure of a source without giving credit
- copying so many words or ideas from a source that it makes up the majority of your work, whether you give credit or not"

Plagiarism is a very serious offense and will be penalized accordingly by the university disciplinary committee. The best way to avoid accidentally plagiarizing is to work on your own before you ask for the help of other resources.

Cheating

Cheating has a very broad description which can be summarized as "acting dishonestly". Some of the things that can be considered as cheating are the following:

- Copying answers on examinations, homework and laboratory works,
- Using prohibited material on examinations,
- Lying to gain any type of advantage in class
- Providing false, modified or forged data in a report
- Plagiarizing.
- Modifying graded material to be regraded.
- Causing harm to colleagues by distributing false information about an examination, homework or laboratory

Cheating is a very serious offense and will be penalized accordingly by the university disciplinary committee.

Class Readings

Class readings are necessary but not mandatory. The material covered in class by your instructor will only provide a fundamental understanding of the general context.
The reading materials will be provided by the instructor, at the relevant week.

COURSE OUTLINE			
Week	**Topics**	**Readings**	**Assignments, quizzes, and exams**
1	Introduction Algorithmic Thinking Karatsuba Multiplication	Lecture notes	Introduction assignment
2	Basics of Algorithm Analysis - Computational Tractability - Scientific AoA Approach - Asymptotic Order of Growth - Memory	Kleinberg – Ch 2 Lecture Notes	Problem set 1
3	Sorting - QuickSort Analysis	Sedgewick – Ch 2	**Projects out** Problem set 2
4	Searching - Search Trees - Hash Tables	Sedgewick – Ch 3	Problem set 3
5	Graphs Strings - Tries	Kleinberg – Ch 3 Sedgewick – Ch 5	**Project proposal due.** Problem set 4
6	Greedy Algorithms I - Interval scheduling - Interval partitioning	Kleinberg – Ch 4 (4.1-3)	Problem set 5
7	Greedy Algorithms II - Shortest paths and MSTs - Clustering	Kleinberg – Ch 4 (4.4-8) Sedgewick – Ch 4(4.3, 4.4)	Problem set 6
8	Divide and Conquer I - Basics	Kleinberg – Ch 5 (5.1)	**Midterm**

	- Sorting and selection		
9	Divide and Conquer II - Recurrence Relations - Master Theorem	Kleinberg – Ch 5 (5.2)	Problem set 7
10	Dynamic Programming - The Bellman-Ford Algorithm - Negative Cycles	Kleinberg – Ch 6 (6.2, 6.8) Sedgewick – Ch 4(4.4)	**Project Progress due** Problem set 8
11	Network Flow - Ford-Fulkerson - Max-flow Min-cut - Bipartite Matching Problem	Kleinberg – Ch 7 (7.1, 7.2, 7.5,7.6) Sedgewick – Ch 6	Problem set 9
12	Intractability I - P, NP, and NP-complete	Kleinberg – Ch 8	
13	Intractability II - Sequencing Problems - Graph Coloring	Kleinberg – Ch 8	**Projects Due**
14	Further Topics in Algorithms - Backtracking - Genetic Algorithms	Lecture Notes	Project presentations

Prepared By & Date	Dr. Haydar CUKURTEPE 29.04.2021	**Revision Date**	Dr. Ulaş GÜLEÇ 20/09/2022