School of Engineering and Natural Sciences / Biomedical Engineering (English)

2024 - 2025 Academic Year

DATA STRUCTURES

Syllabus

Course Description					
Name	**Code**	**Semester**	**T+A Hour**	**Credit**	**ECTS**
DATA STRUCTURES	BME2233850	Spring Semester	3+2	4	8

Prerequisites Courses	PROGRAMLAMAYA GİRİŞ
Recommended Elective Courses	Data Communication and Networking, Object Oriented Programming and Discrete Math
Language of Instruction	English
Course Level	First Cycle (Bachelor's Degree)
Course Type	Elective
Course Coordinator	Assist.Prof. Ahmet KAPLAN
Name of Lecturer(s)	Assist.Prof. Ahmet KAPLAN
Assistant(s)	Teaching assistant for the lab sessions.
Aim	This course aims to teach how to organize data in a computer so that it can be used for designing efficient algorithms to solve various types of problems. Topics covered include arrays, lists, stacks, queues, trees, heaps, graphs and the use of these data structures for searching, sorting, selection and other related applications. Python Programming language will be used for the implementation of data-structures.
Course Content	This course contains; Introduction to data structures and algorithms; Introduction to Basics of PythonSetting up AI IDE extensions. Using AI to explain complexity classes.,Functions, Arrays, and Pointers.AI prompt engineering to generate node classes and basic functions.,Strings, Structs, and Memory AllocationUsing AI to simulate real-world scenarios (e.g., CPU scheduling) using these structures.,Algorithm analysis and complexity notationsAsking AI to analyze the algorithms and identify the problems when hallucinations occur.,Fundamental data structures: Linked Lists, Stacks and QueuesWorking with AI for visualizing Linked Lists, Stacks and Queues and identify differences,Recursion & SearchAsking AI to visualize recursive calls and identify base cases to prevent stack overflow.,Sorting AlgorithmsAI-assisted comparative analysis of quicksort vs. mergesort.,Trees and Binary Search TreesAI-assisted implementation of insertion, deletion, and balancing logic of trees,Priority Queues and HeapsUsing AI to implement priority queues for engineering optimization problems.,Graph Algorithms (BFS/DFS)Working with AI to generate traversal paths for adjacency matrices and adjacency lists.,Hashing and CollisionUsing AI to suggest hash functions, analyze collision rates and large datasets using hash maps and trees.,Refactoring & OptimizationFeeding working, inefficient code to AI and asking for optimized alternatives.,Debugging AI CodeIntentionally using flawed AI code and teaching students how to debug it..

Course Learning Outcomes	**Teaching Methods**	**Assessment Methods**
Explain basic principles of algorithm analysis.	12, 21, 6, 9	A
Apply basic data structures, such as arrays, lists, stacks and queues, to algorithmic design	12, 17, 2, 21, 6, 9	A, F
Applys the tree, binary tree, heap, hash tables, and graph data structures in problem solutions	12, 17, 6, 9	A
Choose the right data type for efficient solution of a problem.	12, 17, 2, 6, 9	F
Analyze the accuracy, complexity and efficiency of an algorithmic solution.	17, 2, 21, 6, 9	A, F
Use AI tools to generate boilerplate code, explain complex algorithms, and debug code efficiently.	5	E
Evaluate AI-generated code for correctness, efficiency, and security vulnerabilities.	19, 37	F

Teaching Methods	12: Problem Solving Method, 17: Experimental Technique, 19: Brainstorming Technique, 2: Project Based Learning Model, 21: Simulation Technique, 37: Computer-Internet Supported Instruction, 5: Cooperative Learning, 6: Experiential Learning, 9: Lecture Method
Assessment Methods	A: Traditional Written Exam, E: Homework, F: Project Task

Lecture Schedule

Sequence	**Topics**	**Preliminary Preparation**
1	Introduction to data structures and algorithms; Introduction to Basics of PythonSetting up AI IDE extensions. Using AI to explain complexity classes.	Book Chapter 1, Lecture Slides 1
2	Functions, Arrays, and Pointers.AI prompt engineering to generate node classes and basic functions.	Book Chapter 1 and 3, Lecture Slides 2
3	Strings, Structs, and Memory AllocationUsing AI to simulate real-world scenarios (e.g., CPU scheduling) using these structures.	Book Chapter 4 ve 5, Lecture Slides 3
4	Algorithm analysis and complexity notationsAsking AI to analyze the algorithms and identify the problems when hallucinations occur.	Book Chapter 2, Lecture Slides 4
5	Fundamental data structures: Linked Lists, Stacks and QueuesWorking with AI for visualizing Linked Lists, Stacks and Queues and identify differences	Book Chapter 6, Lecture Slides 4
6	Recursion & SearchAsking AI to visualize recursive calls and identify base cases to prevent stack overflow.	Book Chapter 7, Lecture Slides 6
7	Sorting AlgorithmsAI-assisted comparative analysis of quicksort vs. mergesort.	Book Chapter 8, Lecture Slides 7
8	Trees and Binary Search TreesAI-assisted implementation of insertion, deletion, and balancing logic of trees	Book Chapter 9,10,11 Lecture Slides 8
9	Priority Queues and HeapsUsing AI to implement priority queues for engineering optimization problems.	Book Chapter 12, Lecture Slides 10
10	Graph Algorithms (BFS/DFS)Working with AI to generate traversal paths for adjacency matrices and adjacency lists.	Book Chapter 13, Lecture Slides 11
11	Hashing and CollisionUsing AI to suggest hash functions, analyze collision rates and large datasets using hash maps and trees.	Book Chapter 14, 15, Lecture Slides 13
12	Refactoring & OptimizationFeeding working, inefficient code to AI and asking for optimized alternatives.	Book Chapter 16, Lecture Slides 14
13	Debugging AI CodeIntentionally using flawed AI code and teaching students how to debug it.	

Evaluation Methods	**Weight(%)**
Midterm Exam	30
General Exam	70

Resources
Course Textbook: Problem Solving with Algorithms and Data Structures using Python By Brad Miller and David Ranum, Luther CollegeSupplementary Material: https://runestone.academy/ns/books/published/medipol_datastructures_spring2026/index.html Lecture presentations and notes