Óbuda University Bánki Donát Faculty of Mechanical and Safety Engineering	Institute of Mechatronics and Vehicle Engineering

Subject title and code: **Complex Data Structures and Programming BMXKAE3MNF** **Credits:** *4*

Full-time study 2026/2027 *ac. year* 1 *semester*

The course is available at: mechatronics engineer MSc

Supervised by: Prof. Dr. Ludányi-Laufer Edit

Instructors: Dr. habil Zsolt Csaba Johanyák

Prerequisite (neptun code): -

Weekly number of lessons

Lecture: 1	Group seminar:	Lab: 3	Consultation:

Way of assessment: midterm mark (Written and oral)

Online consultation *(in case it's required):* *... (BBB link)*

Educational goal: The purpose of the lecture is to deepen students' algorithmic thinking through a review of basic algorithms and data structures, an introduction to dynamic data structures and their algorithms, and the study of graph algorithms. In the laboratory classes, students consolidate their knowledge through practical tasks and develop the ability to solve more complex problems by the end of the semester.

Schedule

Education week	**Topics**
1	*Lecture*: Python fundamentals, including language basics, numeric and complex types, operations, type conversion, the print function, and text data handling. *Lab*: Students practicing the print function, managing variables in the Python environment, performing type conversions, working with text variables and string operations, and then applying these basics in simple control structures.
2	*Lecture*: NumPy basics, including arrays and matrix operations, random numbers, and Pandas fundamentals such as DataFrames and data loading and querying. *Lab*: working with NumPy to perform matrix operations and explore linear algebra concepts, then using Pandas for reading, querying, and processing CSV data while also practicing logical and bitwise operators, branching, loops, dictionaries, and sets.
3	*Lecture*: List manipulation, Matplotlib basics, and JSON serialization, introducing more advanced use of Python data structures. *Lab*: practice lists, stacks, dictionaries, and sets, handle exceptions, and create visualizations with Matplotlib using NumPy and Pandas data.
4	*Lecture*: advanced data structures, functions, and modular programming in Python, integrating NumPy, Pandas, and Matplotlib for complex data analysis tasks. *Lab*: comprehensive project that combines data structures, exception handling, NumPy matrix operations, Pandas data processing, and Matplotlib visualizations into a cohesive Python program.
5	*Lecture*: linked lists and binary search trees, covering their conceptual structure, advantages over arrays, and fundamental operations like insertion, search, and deletion. *Lab*: Students implement a linked list in Python, practice insertion and search operations, and begin implementing deletion while testing their code with various input cases.
6	*Lecture*: binary search tree implementation details, including insertion, traversal methods (inorder, preorder, postorder), and search operations within the tree structure. *Lab*: Students implement a binary search tree, perform all three traversals, and implement search functionality, then validate correctness using diverse test datasets.
7	*Lecture*: graph algorithms with emphasis on shortest path finding from a single node, covering Dijkstra, A*, and Bellman-Ford algorithms and their use cases. *Lab*: Students implement graph structures and code the Dijkstra algorithm to find shortest paths from a starting node, testing on sample graphs with weighted edges.
8	*Lecture*: graph implementation details and graph traversal techniques, including inorder, preorder, and postorder traversals adapted for graph structures. *Lab*: Students implement graphs using adjacency lists or matrices, perform graph traversals, and begin implementing shortest path algorithms including A* and Bellman-Ford.

9	*Lecture*: algorithms for finding minimum cost paths between all node pairs, focusing on the Floyd and Warshall algorithms and their differences. *Lab*: Students implement the Floyd algorithm to compute all-pairs shortest paths, then implement Warshall's algorithm and compare results on the same graph datasets.
10	*Lecture*: practical exercises on finding minimum cost paths for all node pairs, emphasizing algorithm optimization, edge case handling, and performance analysis. *Lab*: Students work on comprehensive practice problems implementing Floyd and Warshall algorithms on larger graphs, analyzing runtime and memory usage.
11	*Lecture*: minimum spanning tree algorithms, covering the red-blue algorithm, Prim's algorithm, and Kruskal's algorithm with their theoretical foundations. *Lab*: Students implement Prim's and Kruskal's algorithms to find minimum spanning trees, test on multiple graphs, and compare their efficiency and output.
12	Rector’s break
13	Midterm exam
14	Make-up exam

Mid-semester requirements

Test		Assignment to be submitted		Lab measurement	
amount	week	amount	deadlines	amount	dates
1	13	1	week 9		

According to the Study and Examination regulations of Óbuda University attendance of group seminars and lab exercises are mandatory.

Other requirements for participation in sessions not covered by the regulations and restrictions on substitutions:

Test		Presentation - Assignment to be submitted		Lab measurement	
maximum points available	minimum score required to pass /test	maximum points available	minimum score required to pass / assignment	maximum points available	minimum score required to pass /lab
70 points	28 points	30 points	12 points	…points	…points

Total number of points achievable in semester: 100 points				
Grading thresholds	**satisfactory** 40 % and above	**average** 55 % and above	**good** 70 % and above	**excellent** 85 % and above

Other evaluation criteria:
Each student must deliver a presentation during Week 10 on an assigned topic. This presentation accounts for 30% of the final grade (30 points total) and is passed with a minimum score of 12 points.

A compulsory midterm test (written format) contributes 70% to the final grade (70 points total) and requires at least 28 points to pass.

Either assessment may be retaken during the final week's class if the student attended the original but failed or was absent with a valid medical certificate.

The course signature is granted upon successfully completing both the presentation and midterm quiz.

Students eligible for a retake who fail it and receive a signature denial may attempt a second retake in the first two weeks of the exam period.

The final exam accounts for the remaining 30% of the grade.

Receive a signature denied entry:	The signature is denied to students who fail to meet the minimum requirements after exhausting the retake opportunity.
Required references:	B. Miller and D. Ranum, Problem Solving with Algorithms and Data Structures Using Python, 3rd ed. Franklin, Beedle & Associates, 2013. [Online]. Available: https://runestone.academy/ns/books/published/pythonds/index.html
Recommended references:	W3Schools, "Python Tutorial," W3Schools, 2026. [Online]. Available: https://www.w3schools.com/python/ (accessed Jun. 8, 2026)
Quality assurance methods of the subject:	

Things, that are not included, can be found within the regulations of Óbuda University.