

Software Engineering Institute			Semester 1. of the curriculum 2026-27-1			
Name of the subject:	Code of the subject:	Credits:	Weekly hours:			
				lec	sem	lab
Programming paradigms and data structures	NSXPA1EMNF	5	full-time	3	0	2
Responsible person for the subject: Prof. Dr. Sándor Szénási						
Subject lecturer(s): Kaziwa Saleh						
Prerequisites:	none	none				
Way of the assessment:	written exam					
Course description						
Goal:	The aim of the course is to introduce the basic data structures, their implementation and basic use cases. In addition, students will be introduced to the basic strategies and programming paradigms used in general problem solving and optimization.					
Course description:	The course introduces the basic operations of data structures (list, queue, stack, set, dictionary) and their use cases. It then discusses the commonly used possible implementations (arrays, sorted arrays, linked lists, binary search trees, hash tables). It then introduces special-purpose data structures (graphs, B-trees, heaps) and additional graph algorithms (shortest path search, spanning tree search, topological ordering). Students will be introduced to the basic strategies that can be used to solve general and optimization problems (brute force method, divide and conquer, memoization method, dynamic programming, greedy algorithms, backtracking, branch and bound). Finally, they gain insights into the world of functional and logic programming.					

Lecture schedule	
Education week	Topic
1.	Generic types. List, queue, stack, set, dictionary operations. Implementation using array and sorted array.
2.	Linked list structure and operations. Implementation of queue, stack.
3.	Binary search tree structure and operations.
4.	Construction and operations of heap. Priority queue implementation. Heap-sort.
5.	B-tree structure and operations.
6.	Hash functions. Structure and operations of hash table. Dictionary implementation.
7.	Graph structure and basic operations (Breadth-First Search, Depth-First Search, topological ordering).
8.	Operations with weighted graphs (finding shortest paths, finding minimum spanning tree).
9.	Brute force method. Divide and conquer strategy. Memoization method. Dynamic programming.
10.	Design and use of greedy algorithms. Backtracking. Branch and bound method.
11.	Basics of functional programming.
12.	<i>Rector's holiday</i>
13.	Fundamentals of logic programming.
14.	Consultation, pre-examination.
Mid-term requirements	
Conditions for obtaining a mid-term grade/signature	Attendance at lectures and laboratory sessions at the scheduled times is mandatory. Students are required to complete two midterm exams, both of which must be passed with a minimum score of 50% . A signature will be granted to students who successfully pass both midterm exams during the semester.

Assessment schedule													
Education week	Topic												
6	Midterm test from the topics presented in the lecture and lab												
13	Midterm test from the topics presented in the lecture and lab												
14	Replacement of either the first or the second midterm test (optional)												
Type of the replacement													
Type of the replacement of written test/mid-term grade/signature	One of the two midterm exams may be retaken during week 14. If a student has failed to attempt both exams or has not achieved a satisfactory score on either, they may only secure a signature by passing the signature replacement exam, which will be scheduled during the examination period. To obtain a signature on this replacement exam, a minimum score of 50% is required												
Type of the exam													
The course can be completed by successfully passing an exam. The exam is written.													
Calculation of the exam mark													
A student who has obtained a signature and passes the exam will receive a final mark computed as an arithmetic mean of the midterm exam grade and the final exam grade. The midterm grade is calculated based on the student's total score as follows:													
<table border="1"> <thead> <tr> <th>average of midterm grade and written test</th> <th>final mark</th> </tr> </thead> <tbody> <tr> <td>0-49%</td> <td>1</td> </tr> <tr> <td>50-61%</td> <td>2</td> </tr> <tr> <td>62-73%</td> <td>3</td> </tr> <tr> <td>74-85%</td> <td>4</td> </tr> <tr> <td>86-100%</td> <td>5</td> </tr> </tbody> </table>		average of midterm grade and written test	final mark	0-49%	1	50-61%	2	62-73%	3	74-85%	4	86-100%	5
average of midterm grade and written test	final mark												
0-49%	1												
50-61%	2												
62-73%	3												
74-85%	4												
86-100%	5												
References													
Obligatory:	<ul style="list-style-type: none"> • Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms (The MIT Press, 2009) • Practical introductory videos and notes available in Moodle • Bradley L. Jones: Teach Yourself C# in 21 Days (Sams Publishing, 2001) 												
Recommended:													
Other references:													