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
Problemsolving using programming	**NSXPP1EBNF**	6	full-time	1	0	3
Responsible person for the subject: László Csink			Classification: associate professor			
Subject lecturer(s): László Csink						
Prerequisites:	none	none				
Way of the assessment:	written exam					

Course description	
Goal:	The purpose of the subject is to present the basics of computer programming and to develop the skills of algorithmic thinking and computer problem solving.
Course description:	The subject material covers the most important elements of general-purpose programming languages, such as the use of variables, control structures and functions, as well as the methodology of structured programming. Students will also learn the basics of the object-oriented programming paradigm, the process of program code development with objects, the use of complex data structures, strings and files. Within the scope of the subject, students learn the basic use of a specific programming language by implementing some well-known and commonly used algorithms, and by solving practical problems with a computer program.

Lecture schedule	
Education week	Topic
1.	Basics of programming languages, instructions, keywords
2.	Using variables, data types and operators
3.	Program control with branches and loops
4.	Arrays
5.	Characters and character strings
6.	File handling, reading and writing data
7.	Procedures, functions, parameter handling
8.	Introduction to object-oriented programming, structure of classes
9.	Basic query operations with arrays
10.	Program development in an object-oriented approach
11.	Basics of recursive algorithms
12.	Sorting algorithms
13.	Debugging approaches in practice
14.	Summary

Mid-term requirements	
Conditions for obtaining a mid-term grade/signature	Attendance of laboratory sessions is mandatory. Attendance of lectures is strongly recommended. By fulfilling the course's mid-term requirements, a signature (the right to register for the exam) can be obtained. During the semester, two programming assignments must be completed and submitted by the specified deadlines, with no possibility for late submission. A maximum of 10 points can be earned from the assignments. Students will take two mid-term tests during the semester. The first test is worth up to 20 points and primarily serves to provide feedback on the level of knowledge acquired up to that point. The second test is worth up to 70 points. To obtain the signature, students must score at least 50% on the second test, earn a minimum of 50 points out of the total 100 points available throughout the semester. Absences from laboratory sessions must not exceed 30 %.

Assessment schedule	
Education week	Topic
week 6	Midterm test from the topics presented in the lecture and lab
week 13	Midterm test from the topics presented in the lecture and lab
last week	Retake of the second midterm test (optional)

Method used to calculate the *mid-term grade* (to be filled out only for subjects with mid-term grades)

Type of the replacement	
Type of the replacement of written test/mid-term grade/signature	The second midterm test can be retaken in week 14. If the student did not complete the retake test at least 50%, or could not achieve at least 50 points out of the 100 points that can be obtained during the semester, the signature can be obtained on the signature retake exam in the examination period. If the student obtains the signature at the signature retake, the midterm points will be 50 % (no more).

Type of the exam (to be filled out only for subjects with exams)

written exam

Calculation of the exam mark (to be filled only for subjects with exams)

The written exam is evaluated in percentage (0-100 %). The written exam is unsuccessful below 50 %.

The final evaluation is based on (total midterm scores + written exam) /2.

average of midterm result and written exam	final mark
0-49%	1
50-61%	2
62-73%	3
74-85%	4
86-100%	5

References

Obligatory:	• Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms (The MIT Press, 2009) • Moodle • Bradley L. Jones: Teach Yourself C# in 21 Days (Sams Publishing, 2001)
Recommended:	
Other references:	none