Óbuda University John von Neumann Faculty of Informatics	Software Engineering Institute

Name and code: Problem solving using programming (NSXPP1EBNF) **Credits:** 6

Computer Science BSc Full-time, 2024/25 year, 1st semester

Subject lecturers: Dr. László Csink, Dr. Zoltán Király, Kaziwa Hassan Saleh, Zsolt Krutilla

Prerequisites:	-

Weekly hours:	Lecture: 1	Seminar: 0	Lab hours: 3	Consultation: 0
Way of assessment:	Colloquium (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.	Basic 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.	Programming tasks

Mid-term requirements

Attendance at lectures and laboratory sessions is mandatory. By fulfilling the course's mid-term requirements, a *signature* 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 exams during the semester. The first exam is worth up to 20 points and primarily serves to provide feedback on the level of knowledge acquired up to that point. The second exam is worth up to 70 points. To obtain the signature, students must score at least 50% on the second exam, earn a minimum of 50 points out of the total 100 points available throughout the semester, and ensure that absences do not exceed 30% of the total sessions.

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 the second midterm test (optional)

Type of the replacement

The second midterm exam can be retaken during the final week. If, after the retake, the student still fails to achieve at least 50% on the second midterm exam, or does not score at least 50 points out of the total 100 available during the semester, the signature can only be obtained through a special *signature-replacement exam* offered during the exam period. Absences cannot be made up by attending another practical session.

Type of the exam

The course ends with an *exam*. The exam will be a written test based on the material of the lectures.

Calculation of the exam mark

The final grade is the average of the mid-term grade and the result of the written test, provided that both parts separately reach the minimum level. The midterm grade is calculated as follows from the total score obtained by the student:

0 – 49:	Fail (1)
50 – 61:	Pass (2)
62 – 73:	Satisfactory (3)
74 – 85:	Good (4)
86 – 100:	Excellent (5)

References

Obligatory:

Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms (The MIT Press, 2022)

Practical introductory notes available in the e-learning system

Recommended:

Bradley L. Jones: Teach Yourself C# in 21 Days (Sams Publishing, 2001)

Other references:

Additional material (including notes) available in the e-learning system