POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

COURSE DESCRIPTION CARD - SYLLABUS

Course name
Discrete mathematics [S1SI1E>MATD]

Course

Field of study
Artificial Intelligence

Year/Semester
1/1

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
English

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture	Laboratory classes	Other
30	0	0
Tutorials	**Projects/seminars**	
30	0	

Number of credit points

5,00

Coordinators

dr Juan Carrascoza Mayén
juan.carrascozamayen@put.poznan.pl

Lecturers

Prerequisites

Students starting this course should have a basic knowledge of mathematics and computer science adequate for the admission requirements. In addition, in the field of social competence, students must exhibit attitudes such as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, and respect for others.

Course objective

The course presents the fundamentals—basic ideas and methods—of discrete mathematics, particularly from the fields of logic, set theory, graph theory, transversal theory, and combinatorics. Its general objective is to develop the student's capability to express ideas arising in the field of computing in terms of functions and relations, applying logical principles, various proof techniques, recursion, and approaches from graph theory to solve problems related to computer science. Moreover, the course develops the students' ability to reason logically and to express their concepts and ideas in an understandable way.

Course-related learning outcomes

Knowledge:
Upon completion of the course, the student:

1. has a general knowledge of discrete mathematics sufficient for formulating and solving complex problems arising in the field of computing using formal logic and set theory; knows the fundamental laws of logic and set theory, as well as the properties of relations and functions, enabling the detection and analysis of relations and dependencies in computing problems;
2. has the knowledge necessary for formulating computing problems in terms of graph theory and for solving them with approaches used in this field;
3. has the knowledge necessary for formulating computing problems using various combinatorial objects and is aware of the relationship between the number of these objects and the number of potential solutions; knows counting principles that enable determining the number of combinatorial objects and understands their importance in estimating the time complexity of algorithms;
4. knows the principle of mathematical induction and is able to use inductive reasoning and recursion to formulate and solve computing problems;
5. knows the basic rules for determining the limiting behaviour of functions, which are necessary for estimating the computational complexity of algorithms.

Skills:
Upon completion of the course, the student:
1. can use the terminology of discrete mathematics to formulate problems arising in the field of computing;
2. is able to apply methods of formal logic, set theory, and graph theory to formulate and solve computing problems;
3. can use Latin squares and rook polynomials to formulate and solve computing problems, particularly assignment problems;
4. is able to use methods for determining the limiting behaviour of functions—particularly asymptotic notations—to establish the computational complexity of algorithms.

Social competences:
Upon completion of the course, the student:
1. is able to express themselves precisely and logically in terms of discrete mathematics.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The learning outcomes presented above are verified as follows:

Lectures:
Learning outcomes from lectures are assessed through at least one 90-minute written or oral exam consisting of single-choice, multiple-choice, and direct-answer questions, evaluating both theoretical and practical aspects of the course. Exams may be conducted in traditional format (pen and paper) or electronically via the university's official website platform. Exams require in-person attendance.
A passing grade requires at least 50% of the maximum points on each exam. The final grade is calculated as a weighted sum of all evaluations, subject to the current grading rules set by the university's central administration.
The use of auxiliary materials during exams is prohibited.

Exercises:
The learning outcomes achieved during the exercises are verified based on the results of at least one written test consisting of several open-ended questions (exercises). The final grade is determined by the number of points obtained on all the tests. A passing grade is awarded after obtaining at least 50% of the maximum number of points. The use of auxiliary materials (including calculators) during the tests is not permitted.

Programme content

Logic, set theory, asymptotics, relations and functions, combinatorics, advanced principles of counting, mathematical induction, recursion, introduction to graph theory, fundamentals of transversal theory, latin squares, rook polynomials, generating Functions.

Course topics

Lectures and exercises will cover the following topics:

1. Fundamentals of logic
2. Set theory
3. Asymptotics
4. Relations and functions
5. Combinatorics
6. Advanced principles of counting: inclusion-exclusion principle, Dirichlet's principle (the pigeonhole principle).
7. Mathematical induction: the well-ordering principle, the first and second principles of mathematical induction.
8. Recursion: recurrence definitions, relations, problems and algorithms, mathematical induction for recursively defined sets.
9. Introduction to graph theory: directed and undirected graphs, paths, walks, cycles, Eulerian and Hamiltonian graphs, graph colouring, trees, etc.
10. Fundamentals of transversal theory: Hall's theorem, tournaments.
11. Latin squares: Latin square design, orthogonal Latin squares.
12. Rook polynomials.
13. Generating functions.
14. Advanced graph problems.

Teaching methods

1. Lectures: multimedia presentations of programme contents with numerous examples.
2. Tutorials illustrating programme contents presented during lectures with exercises solved by teachers using a blackboard or by students eager to present their ideas. Discussions in a student group on the ideas proposed by particular students.

Bibliography

Basic
1. Discrete and combinatorial mathematics. An applied introduction, R.P. Grimaldi, Addison Wesley Publishing Company, New York, 1999.
2. Discrete Mathematics, Kenneth A. Ross, Charles R.B. Wright, Prentice Hall, 1992
3. Concrete Mathematics, Donald E. Knuth, Ronald L. Graham, Oren Patashnik, Addison-Wesley, 1994
4. Matematyka dyskretna, K.A. Ross, Ch.R.B. Wright, PWN, Warszawa, 2012.
5. Matematyka konkretna, R.L. Graham, D.E. Knuth, O. Patashnik, PWN, Warszawa, 2012.
Additional
1. Discrete Mathematics and its Applications, Kenneth H.Rosen, McGraw-Hill, 2007
2. Discrete Mathematics, Swapan K. Chakraborty, Oxford University Press, 2011

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,00
Classes requiring direct contact with the teacher	62	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	63	2,50