COURSE DESCRIPTION CARD - SYLLABUS

Course name
Discrete mathematics

Course

Field of study
Mathematics in Technology
Area of study (specialization)

Year/Semester
1/2
Profile of study

Level of study
First-cycle studies
Form of study
full-time

Course offered in
Polish
Requirements
compulsory

Number of hours

Lecture
15
Laboratory classes
15
Other (e.g. online)

Tutorials
15
Projects/seminars

Number of credit points

3

Lecturers

Responsible for the course/lecturer:
Prof. dr hab. Ryszard Płuciennik

ryszard.pluciennik@put.poznan.pl

Responsible for the course/lecturer:

Prerequisites

Basic knowledge in domain of calculus, algebra and programing on the level of studies of the first semester.

Course objective

Abbility of creating of mathematical models of concrete situations. Using of IT tools to solving of mathematical problems in discrete mathematics. Mastering of advanced combinatorial models.

Course-related learning outcomes

Knowledge

Basic knowledge of discrete and applied mathematics. Knowledge of basic algorithms and their analysis, algorithm design techniques and their implementation to solve computationally difficult problems. Knowledge of advanced programming tools and software packages for data processing and analysis - K_W07.

Advanced knowledge and understanding of the theory and applications of mathematical models - K_W10.

Skills

The ability to use recursion in the approach to combinatorial problems. Ability to create mathematical models to describe a specific situation in reality K_U03.

Ability to select and use appropriate programming tools and software packages for data processing and analysis - K_U09.

Social competences

Readiness to acquire knowledge based on available solutions to cognitive and practical problems, for example in literature, also in English - K_K05.

Readiness to apply current knowledge and acquired mathematical skills, including logical thinking, to solve cognitive and practical problems - K_K06.

Awareness of responsibility for own work and readiness to comply with the rules of teamwork and take responsibility for jointly implemented tasks -K_K07.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lecture

Evaluation of the student's knowledge and skills in an oral exam.

Tutorials

Control of the ability to use the knowledge provided during the lectures to solve tasks in the form of two tests (the student can then use prepared notes and lecture materials).

Systematic control of acquired theoretical knowledge in the form of short tests.

Evaluation of the student's answers during the classes.

Evaluation of activity in classes.

Laboratory classes

Presentation of the project, the basic element of which is a self-written script in MatLab, as well as the student's activity during classes is evaluateded.

Programme content

Lecture: Theorem proving techniques and the principle of mathematical induction. Recursive equations and methods of solving them. Recursive problems arising from practice. Combinatorics. The principle of inclusions and exclusions. Advanced counting techniques. Various applications of Dirichlet's pigeonhole principle. Binomial methods. Problems involving Latin squares and chess polynomials. Stirling numbers.

Tutorials: Program content consistent with the content of the lecture.

Laboratory classes: Computer representations of combinatorial objects (characteristic vector, inversion vector, etc.). Counting combinatorial objects. Algorithms for generating permutations and combinations of n-element sets. The problem of eight queens on a chessboard. Recursion in programming on the example of generating certain numbers known in mathematics, e.g. Stirling numbers of the first and second kind. Laboratory classes are conducted using the MatLab environment.

Teaching methods

Lecture: A traditional lecture conducted in an interactive way consisting in formulating questions addressed to a group of students and guiding them to the right line of reasoning. The content is richly illustrated with examples and counterexamples.

Tutorials: Blackboard exercises involving the analysis and solving of sample tasks. Posing problems requiring the creation (individually or in a team) of algorithms for solving complex problems in discrete mathematics. Creating a mathematical model for specific real situations.

Laboratory classes: Practical classes in the computer laboratory.

Bibliography

Basic

1. R. L. Graham, D. E. Knuth, O. Patashnik, Matematyka konkretna, PWN, Warszawa 2020.

2. K.A. Ross, C.R.B. Wright, Matematyka dyskretna, PWN, Warszawa 2012.

Additional

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest, Wprowadzenie do algorytmów, PWN, Warszawa 2012.

Breakdown of average student's workload

	Hours	ECTS
Total workload	90	3,0
Classes requiring direct contact with the teacher	45	1,5
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	45	1,5

[1] delete or add other activities as appropriate