COURSE DESCRIPTION CARD - SYLLABUS

Course name

Introduction to mathematics for computer science

Course

Field of study	Year/Semester
Artificial intelligence	1/1
Area of study (specialization)	Profile of study
-	general academic
Level of study	Course offered in
First-cycle studies	English
Form of study	Requirements
full-time	compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
30	-	-
Tutorials	Projects/seminars	
30	-	

Number of credit points

5

Lecturers

Responsible for the course/lecturer:	Responsible for the course/lecturer:
dr Agnieszka Ziemkowska-Siwek	-

Prerequisites

Basic mathematical knowledge from secondary school.

Course objective

The aim of this course is to acquaint students with selected topics in mathematics, which are useful in formulation and solving complex IT problems.

Course-related learning outcomes

Knowledge

Knowledge of propositional calculus, predicates calculus, set theory, basic algebraic structures and the basics of numetical methods.

Skills

Ability to uderstand the structure of mathematical theories.

Ability to perform correctly logical reasoning.

Ability to use logical formalism to built and analyse the models of artificial intelligence.

Social competences

Understanding the need of systematic learning and developing of skills.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Lectures: written test at the end of the semester, pass treshold: 50% of points.

Tuttorials: two test (in the middle and at the end of the semester), additional point for activity, pass treshold: 50% of points

Programme content

Lectures:

Propositional calculus (propostion, truth value, logical operators, converse, contrapositive, logical equivalences, disjunctive normal form, conjunctive normal form, tautologies, rules of inference, application examples).

Predicate calculus (predicate, n-ary predicate, universal quantifier, existential quantifier, truth value of quantified statements, negation of quantification, boud and free variables, scope of the quantifier, quantifier laws, rules of inference, application examples).

Mathematical induction. Proofs (direct proof, proof by contradiction, proof by contrapositive, proof by cases).

Set theory (union, internection, complement, difference, simmetric difference, laws of set theory, Venn diagram, Cartesian product, n-fold Cartesian product, indexed family of sets, finite sets, infinite sets, cardinality, equipotent sets, countable/uncoutable sets).

Relations (reflexive, symmetric, antisymmetric, transitive, equivalence relation, equivalence classes).

Basic algebraic structures (binary operation, properties of binary operation, semigroup, monoid, group, subgroup, abelian group, isomorphism, homomorphism, ring, commutative ring, subring, zero divisor, entire ring, field, permutation groups, cycles, transposition)

Elements of numerical methods (error analysis, floating-point arithmetic, numerical methods of solving nonlinear equations: bisection method, regula falsi method, secant method, Newton-Raphson method)

Tutorials:

Propositional calculus (determining the logical value of proposition, writing propositions using logical connectives, writing the converse and the contrapositive of the sentence, showing that the statement is a tautology, simplifying propostions, converting statements to conjunctive/disjunctive normal form, applying propositional laws to the list of premises, giving counterexamples)

Predicate calculus (determining the logical value of formulas containing quantifiers, expressing statements using quantifiers, logical connectives and predicates, showing the logical equivalency between statements, writitng the negation of the sentence, proving the laws of predicate calculus)

Proofs (proving theorems using: proof by induction, direct proof, proof by contradiction, proof by contrapositive, proof by cases)

Set theory (proving the laws of set theory, drawing Venn diagrams, giving conterexamples, proving the inclusion of sets, showing that the sets are equipotent)

Relations (examples, properties of relations, equivalence relations, identification of equivalence classes)

Algebraic structures (checking the properties of binary operations, constructing a Cayley table, finding identity alements and inverses, proving that something is a group/an abelian group, finding subgroups, finding homomorphism between groups, finding zero divisors, composition of permutations, solving permutation group equations, finding the cycle decomposition of a permutation, expressing a permutation as a product of transpositions)

Elements of numerical methods (solving nonlinear equations with selected numerical methods, implementation of selected numerical methods).

Teaching methods

Lectures: multimedia presentation, traditional lecture on the board

Tutorials : solving examples, discussions

Bibliography

Basic

G. J. Janacek, M. Lemmon Close - Mathematics for computer scientists

E. Lehman, F. T. Leighton, A.R. Meyer - Mathematics for computer science

C. Newstead - An infinite descent into pure mathematics

J. Stoer, R. Bulirsch - Introduction to Numerical Analysis

(all available online)

Additional

R. Murawski, K. Świrydowicz - Podstawy logiki i teorii mnogości

H. Rasiowa - Wstęp do matematyki współczesnej

K. A. Ross, CH. R. B. Wright - Matematyka dyskretna

Breakdown of average student's workload

	Hours	ECTS
Total workload	125	5,0
Classes requiring direct contact with the teacher	60	2,5
Student's own work (literature studies, preparation for tutorials, preparation for tests) [1]	65	2,5

[1] delete or add other activities as appropriate