



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Introduction to mathematics for computer science

Course

Field of study

Artificial intelligence

Area of study (specialization)

-

Level of study

First-cycle studies

Form of study

full-time

Year/Semester

1/1

Profile of study

general academic

Course offered in

English

Requirements

compulsory

Number of hours

Lecture

30

Laboratory classes

-

Other (e.g. online)

-

Tutorials

30

Projects/seminars

-

Number of credit points

5

Lecturers

Responsible for the course/lecturer:

dr Agnieszka Ziemkowska-Siwiek

Responsible for the course/lecturer:

-

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

Deep knowledge of propositional calculus, predicates calculus, set theory and basic algebraic structures.

Skills

Ability to understand the structure of mathematical theories.

Ability to perform correctly logical reasoning.

Ability to use logical formalism to build 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

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

Programme content

Lectures:

Propositional calculus (proposition, truth value, logical operators, converse, contrapositive, logical equivalences, disjunctive normal form, conjunctive normal form, tautologies, rules of inference, valid arguments)

Predicate calculus (predicate, n-ary predicate, universal quantifier, existential quantifier, truth value of quantified statements, negation of quantification, bound and free variables, scope of the quantifier, quantifier laws, translating statements into mathematical logic, rules of inference)

Proofs (direct proof, proof by contradiction, proof by contrapositive, proof by cases)

Set theory (union, intersection, complement, difference, symmetric difference, laws of set theory (proofs), Venn diagram, Cartesian product, indexed families of sets, partition of set, n-fold cartesian product, finite sets, infinite sets, equipotent sets, countable/uncountable sets, cardinality)

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)

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 propositions, converting statements to conjunctive/disjunctive normal form, applying propositional laws to the list of premises, proving the valid arguments, 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, writing the negation of the sentence, proving the laws of predicate calculus)

Proofs (proving theorems using: 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)

Algebraic structures (checking the properties of binary operations, constructing a Cayley table, finding identity elements 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)

Teaching methods

Lectures: traditional lecture on the board

Tutorials : solving examples on the board, 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

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