



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Introduction to mathematics for computer science [S1S1E>MAT]

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

30

Laboratory classes

0

Other (e.g. online)

0

Tutorials

30

Projects/seminars

0

Number of credit points

5,00

Coordinators

dr Daniel Ballesteros Chavez

daniel.ballesteroschavez@put.poznan.pl

Lecturers

dr Daniel Ballesteros Chavez

daniel.ballesteroschavez@put.poznan.pl

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:

basics of numerical methods.

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:

Learning outcomes presented above are verified as follows:

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

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, application examples).

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, rules of inference, application examples).

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

Set theory (union, intersection, complement, difference, symmetric 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/uncountable 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 (numerical methods of solving

nonlinear equations: bisection method, regula falsi method, secant method, Newton-Raphson method; polynomial interpolation: Lagrange interpolating polynomial, Hermite interpolating polynomial)

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, 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: 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 counterexamples, 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 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)

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

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 | 5,00 |
| Classes requiring direct contact with the teacher | 60 | 2,50 |
| Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation) | 65 | 2,50 |