COURSE DESCRIPTION CARD - SYLLABUS

Course name
Declarative programming [N1Inf1>PDEK]

Course

Field of study
Computing

Year/Semester
2/3

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
polish

Form of study
part-time

Requirements
compulsory

Number of hours

Lecture
12

Laboratory classes
12

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

2,00

Coordinators

dr inż. Grażyna Brzykcy
grazyna.brzykcy@put.poznan.pl

Lecturers

Prerequisites

Student has basic knowledge of mathematics, especially in such fields as algebra, analysis and logic, basic knowledge of program constructs, implementation of algorithms, formal languages and programming platforms. Student is able to use basic techniques to create algorithms, to analyze their complexity, and to use software platforms and environments for simple programs encoding, running and testing.

Course objective

Presentation of declarative programming styles and rules of choosing the adequate style and language to a class of problems. Development of declarative programming skills in logic programming environments.

Course-related learning outcomes

Knowledge:
1. Student has organized and theoretically founded knowledge of computation models and basic declarative program constructions.[K1st_W4]
2. Student has organized and theoretically founded knowledge of creation, implementation and applicability of recursive data structures.[K1st_W7]
3. Student is familiarized with state of the art and current trends in programming paradigms.[K1st_W7]

Skills:
1. Student is able to create engineer work documentation and declaratively present the work result. [K1st_U10]
2. Student is able to use declarative software platforms and environments for simple programs encoding, running and testing. [K1st_U11]
3. Student can use techniques of logic programming to create algorithms. [K1st_U11]

Social competences:
1. Student understands and is aware of the importance of issues related to computer engineer activity. Student understands the responsibility for his engineering decisions. [K1st_K1]
2. Student understands the importance of stringent accomplishment of a given project with proper notation standards. [K1st_K3]

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
Lecture
Written test based on lecture (basic concepts and techniques used in declarative programming).
Laboratory
Student mark is based on continuous assessment of their programming activity and result of written tests (creation of simple programs).

Programme content

Lecture
1. Introduction to Prolog (program syntax, Prolog rules, recursive definitions, goals and answers).
2. Prolog program interpretations (data representation, terms unification, declarative and procedural interpretation of Prolog program).
3. Lists (representation, list manipulations,arithmetic operators in Prolog).
4. Backpropagation in Prolog (cut procedure, negation as failure).
5. Meta-predicates (term processing, in/out predicates, rule base manipulation, set of results).
6. Style and programming techniques in Prolog (good practices in Prolog).
Laboratory
1. Introduction to Prolog (program syntax, execution, buit-in predicates, the first program).
2. Recursion and iteration (tail recursion with accumulators).
3. Lists (predicates member/2, append/3, insertion and deletion of elements).
4. Meta-predicates (cut, fail, negation by failure, in/out predicates, set of results).
5. Term processing (term recognition).
6. Test.

Teaching methods

Lecture: multimodal presentation, examples of Prolog programs.
Laboratory classes: Creation of algorithms and their implementation in declarative programming language Prolog.

Bibliography

Basic
1. Nilsen U., Małuszyński J.: Logic, Programming, and PROLOG, John Wiley & Sons, 2000.
2. Van Roy P., Haridi S.: Concepts, Techniques, and Models of Computer Programming, The MIT Press, 2004.
Additional
1. Kowalski R.: Logic for problem solving, North-Holland, 1979.
2. Sterling L., Shapiro E.: The Art of Prolog. Advanced Programming Techniques, MIT Press, 1986.

Breakdown of average student's workload

	Hours	ECTS
Total workload	50	2,00
Classes requiring direct contact with the teacher	24	1,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	26	1,00