# MODULE DESCRIPTION

| Module title | Module code |
|---|---|
| Logic Programming | |

| Lecturer(s) | Department where the module is delivered |
|---|---|
| **Coordinator:** prof. dr. Rimantas Vaicekauskas<br>**Other lecturers:** | Department of Computer Science<br>Faculty of Mathematics and Informatics<br>Vilnius University |

| Cycle | Type of the module |
|---|---|
| First | Optional |

| Mode of delivery | Semester or period when the module is delivered | Language of instruction |
|---|---|---|
| Face-to-face | 5th and 7th semester | Lithuanian and English |

| Prerequisites |
|---|
| **Prerequisites:** Data structures and algorithms, Mathematical logic. |

| Number of credits allocated | Student‘s workload | Contact hours | Self-study hours |
|---|---|---|---|
| 5 | 130 | 68 | 62 |

## Purpose of the module: programme competences to be developed

Purpose of the module: to introduce the key concepts and principles of the Logic programming, to present Prolog problem-solving techniques.

***Generic competences:***

- Communication and collaboration *(GK1).*
- Life-long learning *(GK2).*

***Specific competences:***

- Knowledge and skills of underlying conceptual basis *(SK4).*
- Software development knowledge and skills *(SK5).*

| Learning outcomes of the module: students will be able to | Teaching and learning methods | Assessment methods |
|---|---|---|
| • Understand the principles of declarative specification using Horn clause logic. | Lectures, problem-oriented teaching, case studies, literary reading, individual work, tutorials, laboratory work. | Laboratory works and results presentation, written exam (open, semi-open and close-ended questions and tasks). |
| • Explain procedural behavior of the Prolog program. | | |
| • Use Prolog statements and structures for knowledge representation and reasoning. | | |
| • Create, debug, and evaluate small-to-moderate size Prolog program as software prototype. | | |
| • Use advanced Logic Programming techniques for solving problems related to the Artificial Intelligence (AI) applications. | | |

| Content: breakdown of the topics | Contact hours | | | | | | | Self-study work: time and assignments | |
|---|---|---|---|---|---|---|---|---|---|
| | Lectures | Tutorials | Seminars | Practice | Laboratory work | Tutorial during LW | **Contact hours** | **Self-study hours** | **Assignments** |
| Introduction to logic programming (LP) paradigm. Declarative vs. procedural programming. Representation of knowledge with Horn clauses. The meaning of the logic program. Relation between LP and Artificial Intelligence. | 4 | | | | 5 | *8* | **9** | **9** | Individual reading. Self-preparation for laboratory works. Self-control tasks. |
| Prolog as a logic programming language. Procedural semantics of Prolog: execution model in term of goals and sub-goals, unification (pattern matching) and backtracking | 4 | | | | 5 | | **9** | **9** | |
| Recursive programming with Prolog: representation and processing of simple data structures– lists and trees. Prolog arithmetic | 2 | | | | 4 | | **6** | **6** | |
| Advanced search techniques: generate –and –test search pattern, searching space–state graphs. | 4 | | | | 4 | | **8** | **8** | |
| Manipulating symbolic expressions. Symbolic differentiation example | 2 | | | | 2 | | **4** | **4** | |
| Interpreters and rule-based systems. | 4 | | | | 4 | | **8** | **8** | |
| Grammars and applications to natural language understanding. | 3 | | | | 2 | | **5** | **5** | |
| Meta-predicates *assert, retract, findall*. Breadth–first search applications. Searching games tree. | 3 | | | | 2 | | **5** | **5** | |
| Incomplete data structures: difference lists, dictionaries. | 3 | | | | 2 | | **5** | **5** | |
| Prolog implementations. Parallel Prolog. | 3 | | | | 2 | | **5** | **3** | |
| Tutorials during the semester | | 2 | | | | | **2** | | |
| Final exam (written) | | | | | | | **2** | | |
| **Total** | **32** | | | | **32** | *8* | **68** | **62** | |

| **Assessment strategy** | **Weight %** | **Deadline** | **Assessment criteria** |
|---|---|---|---|
| Laboratory works | 40 | There are 4 assignments with deadlines 3th, 7th, 11th, 16th week of the semester. | During the semester, a student is required to carry out 4 laboratory works (individual Prolog programming assignments). The 1st laboratory work is to describe logical relations between well-known entities. The 2nd assignment requires using recursive rules. The third one is about list processing. The 4th is to write a Prolog program that solves an AI-related problem.<br>Each laboratory work is evaluated from 0 to 1 points and total (4 points) is equivalent to 40% of the final score.<br>Assessment criteria are: soundness and correctness of the program, relevance to the Prolog programming paradigm, ability to defend solutions and answer to the questions. |

| | | | |
|---|---|---|---|
| Exam (written) | 60 | During exam session | During the exam, it is possible to get at most 6 points, which are equivalent to 60% of the final score. The exam is divided into 3 parts. For the first part, the student must answer various questions (open, semi-open and close-ended questions and tasks) of diverse complexity (0-3 points). For the second part, the student must practically solve a given problem, which involves writing the code in Prolog (0-2 points). For the third part, the student must demonstrate an understanding of the given topic by writing a thorough summary and providing explanatory examples (0-1 points). |

| **Author** | **Publishing year** | **Title** | **Number or volume** | **Publisher or URL** |
|---|---|---|---|---|
| **Required reading** | | | | |
| Clocksin W.F., Mellish C.S. | 2003 | Programming in Prolog: Using the ISO Standard | 5th ed. | Springer |
| Sterling, L. and Shapiro, E. | 1994 | The Art of Prolog | 2nd ed. | MIT Press |
| **Recommended reading** | | | | |
| Ulf Nilsson, Jan Maluszynski | 2012 | Logic, Programming and Prolog | 2nd ed. | http://www.ida.liu.se/~ulfni/lpp/ |
| Ivan Bratko | 2000 | Prolog Programming for Artificial Intelligence | 3rd ed. | Addison Wesley |