1. Course title: CONSTRAINT LOGIC PROGRAMMING
2. Course code

3. Validity of course description: 2016/2017

4. Level of studies: BSc programme

5. Mode of studies: intramural studies

6. Field of study: MACROCURSE RAU

7. Profile of studies: general

8. Programme: Automatic Control

9. Semester: 6

10. Faculty teaching the course: Automatic Control, Electronics and Computer Science

11. Course instructor: dr inż. Szymon Ogonowski

12. Course classification: common courses

13. Course status: elective

14. Language of instruction: English


16. Course objectives: The aim of this lecture is to present Constraint Logic Programming techniques for solving Constraint Satisfaction Problems and Constraint Optimisation Problems. Applications of those techniques are demonstrated by many examples taken from real-world combinatorial problems, such as scheduling, planning or job-shop problems. The theory presented in the lecture is supported by problems solving using Java language (JaCoP library).

17. Description of learning outcomes:

<table>
<thead>
<tr>
<th>Nr</th>
<th>Learning outcomes description</th>
<th>Method of assessment</th>
<th>Teaching methods</th>
<th>Learning outcomes reference code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Knows and understands meaning of: constraint, variable domain, distribution, search method, constraints propagation.</td>
<td>laboratory exercise</td>
<td>lecture</td>
<td>K_W16</td>
</tr>
<tr>
<td>2.</td>
<td>Understands functionality of: arithemtic, reified, alldifferent, cumulative, circuit, element, count constraints and knows how to model soft constraints.</td>
<td>laboratory exercise</td>
<td>lecture</td>
<td>K_W19</td>
</tr>
<tr>
<td>3.</td>
<td>Understands functionality of search methods: depth first search, limited discrepancy search, credit search, branch-and-bound.</td>
<td>laboratory exercise</td>
<td>lecture</td>
<td>K_W16</td>
</tr>
<tr>
<td>4.</td>
<td>Can describe the given problem with constraints and choose proper search method.</td>
<td>laboratory exercise</td>
<td>laboratory</td>
<td>K_W16</td>
</tr>
<tr>
<td>5.</td>
<td>Can resolve scheduling and job-shop problems.</td>
<td>laboratory exercise</td>
<td>laboratory</td>
<td>K_W19,K_U24</td>
</tr>
<tr>
<td>6.</td>
<td>Can develop software with module structure.</td>
<td>laboratory exercise</td>
<td>laboratory</td>
<td>K_W05</td>
</tr>
<tr>
<td>7.</td>
<td>Knows the range of problems that can be solved with CLP.</td>
<td>laboratory exercise</td>
<td>laboratory</td>
<td>K_U20</td>
</tr>
<tr>
<td>8.</td>
<td>Can prioritise the tasks required to solve the given problem.</td>
<td>laboratory exercise</td>
<td>laboratory</td>
<td>K_U02, K_K04</td>
</tr>
</tbody>
</table>

18. Teaching modes and hours
Lecture: 15 h / Laboratory: 15 h

19. Syllabus description:

Course
Constraint Logic Programming is a methodology having backgrounds in Operational Research and belongs to wide concept of Artificial Intelligence Methods. Its main concept is based on declarative programming – the programmer needs only to describe what computation should
be performed and not how to compute it (this part is already implemented in solver). Lecture is based on a multimedia presentation (available for the course students), supported with example code testing. The course presents concepts of:

- constraint propagation, global constraints, arc-consistency,
- different domains of variables - boolean domains, finite domains, sets and real intervals,
- global constraints such as: alldifferent, count, element, cumulative, diff2, circuit, knapsack, among, regular,
- search distributions – first-fail, most-constrained, max-regret,
- tree-based search such as - branch-and-bound, depth first search, limited discrepancy search, special search techniques,
- methods of default search algorithms modification – creating own search techniques,
- modelling and solving constraint optimisation problems and soft constraints - Weighted CSP, Fuzzy/Possibilistic CSP, Probabilistic CSP, over-constrained problems,
- scheduling/planning problems,
- incorporating local search techniques into constraint programming methodology.

Details of concepts mentioned above are presented while solving examples of simple academic and more complex, real-world combinatorial problems with usage of Java language and dedicated JaCoP (Java Constraint Programming) library. Advantages of using well known and well documented language allows the students to focus on getting to know the idea behind constraint programming. During laboratory exercises students constructs Java desktop applications and incorporates in it different modules, designed to solve different combinatorial problems.

**Laboratories**

All laboratory exercises are focusing on different CP problems, that are solved as a separate Java modules. Designed modules are incorporated in main Java desktop application, creating compact and scalable project.

1. Design of CLP application base
2. Simple combinatorial problems modules
3. Soft constraint problem module - reified constraint
4. Scheduling/planning problem module – building bridge example
5. Search strategy module – queens problem
6. Optimisation problem module

20. Examination: no

21. Primary sources:
Antoni Niederliński “Programowanie w logice z ograniczeniami”,
Website www.jacop.eu (JaCoP API and documentation).

22. Secondary sources:
Francesca Rossi, Peter Van Beek, Toby Walsh “Handbook of constraint programming”,
Peter van Hentenryck “Constraint-based Local Search”.

23. Total workload required to achieve learning outcomes

<table>
<thead>
<tr>
<th>Lp.</th>
<th>Teaching mode</th>
<th>Contact hours / Student workload hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lecture</td>
<td>15/5</td>
</tr>
<tr>
<td>2</td>
<td>Classes</td>
<td>0/0</td>
</tr>
<tr>
<td>3</td>
<td>Laboratory</td>
<td>15/10</td>
</tr>
<tr>
<td>4</td>
<td>Project</td>
<td>0/0</td>
</tr>
<tr>
<td>5</td>
<td>BA/ MA Seminar</td>
<td>0/0</td>
</tr>
<tr>
<td>6</td>
<td>Other</td>
<td>7/20</td>
</tr>
<tr>
<td></td>
<td>Total number of hours</td>
<td>37/35</td>
</tr>
</tbody>
</table>

24. Total hours: 72

25. Number of ECTS credits: 3

26. Number of ECTS credits allocated for contact hours: 2

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 1

26. Comments:

Approved:

(date, Instructor’s signature) (date, the Director of the Faculty Unit signature)