SYLLABUS

Intelligent methods and their applications in software engineering

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babeş-Bolyai University
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Field of study	Mathematics
1.5. Study cycle	Bachelor
1.6. Study programme/Qualification	Mathematics and Computer Science (in English)
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline	**Intelligent methods and their applications in software engineering**					Discipline code	**MLE7040**
2.2. Course coordinator				Lect. PhD. Oneț-Marian Zsuzsanna			
2.3. Seminar coordinator				Lect. PhD. Oneț-Marian Zsuzsanna			
2.4. Year of study	3	2.5. Semester	6	2.6. Type of evaluation	E	2.7. Discipline regime	Optional

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	**4**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**1S + 1L**
3.4. Total hours in the curriculum	48	of which: 3.5 course	24	3.6 seminar/laboratory/project	**24**
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					22
Additional documentation (in libraries, on electronic platforms, field documentation)					30
Preparation for seminars/labs, homework, papers, portfolios and essays					30
Tutorship					15
Evaluations					5
Other activities:					
3.7. Total individual study hours	**102**				
3.8. Total hours per semester	**150**				
3.9. Number of ECTS credits	**6**				

4. Prerequisites (if necessary)

4.1. curriculum	Algorithms and Programming, Object oriented programming basics, Advanced methods of programming software applications
4.2. competencies	Good programming skills in Python

5. Conditions (if necessary)

5.1. for the course	Lecture room with projector
5.2. for the seminar /lab activities	

6. Specific competencies acquired [1]

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/essential competencies	• development and analysis of algorithms for solving problems
Transversal competencies	• application of rigorous and efficient work rules, manifestation of responsible attitudes towards the didactic-scientific field, to bring optimal and creative values to own potential in specific situations, with respect to professional ethics principles and norms • efficient and effective development of organized activities of teamworks • use of efficient information resources and techniques to learn and develop the professional abilities in Romanian language and in an international language

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• The goal of this course is to familiarize the students with intelligent search methods and clustering algorithms and how these can be used to solve different software engineering related problems.
7.2 Specific objective of the discipline	• Understand and recognize the components of a search/optimization problem. • Learn about different software testing types and understand how to describe software testing activities as search problems. • Recognize some of the most well-known bad smells in source code. • Understand clustering algorithms and how they can work with software engineering data.

8. Content

8.1 Course	Teaching methods	Remarks
1. Course organization. Search based software engineering. Components of an optimization problem.	• Interactive exposure • Explanation • Conversation • Didactical demonstration • Case studies	
2. Hill climbing, Simulated Annealing, Tabu Search		
3. Genetic Algorithms		
4. Applications of search algorithms for mutation testing		
5. Application of search algorithms for unit testing		
6. Applications of search algorithms for fuzz testing		
7. Applications of search algorithms for integration testing		
8. Applications of search algorithms for regression testing.		
9. Clustering algorithms and their		

applications for software testing		
10. Bad smells and refactoring.		
11. Intelligent methods for software refactoring		
12. Software code embedding based approaches		

Bibliography
1. Mark Harman, Bryan F. Jones: Search-based software engineering, Information and software Technology, Nr. 43, pp. 833-839, 2001
2. Mark Harman, S. Afshin Mansouri, Yuanyuan Zhang, Search-based Software Engineering: Trends, Techniques and Applications, ACM Cmoputing Surveys, Vol. 45, Nr. 1, Article No. 11, pp. 1- 61, 2012
3. David Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Professional, 1989
4. Martin Fowler: Refactoring. Improving the design of Existing Code, Addison-Wesley Professional, 2018

8.2 Seminar	Teaching methods	Remarks
1. Seminar organization	• Conversation • Dialog • Case studies	Seminar will be organizaed as 2 hours every two weeks.
2. Report I topic selection		
3. Report I presentations		
4. Report I presentations + report II topic selection		
5. Report II presentations		
6. Report II presentations		

Bibliography
1. Mark Harman, Bryan F. Jones: Search-based software engineering, Information and software Technology, Nr. 43, pp. 833-839, 2001
2. Mark Harman, S. Afshin Mansouri, Yuanyuan Zhang, Search-based Software Engineering: Trends, Techniques and Applications, ACM Cmoputing Surveys, Vol. 45, Nr. 1, Article No. 11, pp. 1- 61, 2012
3. David Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Professional, 1989
4. Martin Fowler: Refactoring. Improving the design of Existing Code, Addison-Wesley Professional, 2018

8.3 Laborator	Teaching methods	Remarks
1. Python libraries and implementing Hill climbing	• Conversation • Dialog • Case studies	Lab will be organized as 2 hours every two weeks. During every lab, students will work in 2-3 person teams and will have to solve simple problems in Python related to the topics discussed at the lecture.
2. Mutation testing in Python		
3. Unit / fuzz testing in Python		
4. Clustering algorithms		
5. Software refactoring		
6. Intelligent methods for software refactoring		

Bibliography
1. Mark Harman, Bryan F. Jones: Search-based software engineering, Information and software Technology, Nr. 43, pp. 833-839, 2001
2. Mark Harman, S. Afshin Mansouri, Yuanyuan Zhang, Search-based Software Engineering: Trends, Techniques and Applications, ACM Cmoputing Surveys, Vol. 45, Nr. 1, Article No. 11, pp. 1- 61, 2012
3. David Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Professional, 1989
4. Martin Fowler: Refactoring. Improving the design of Existing Code, Addison-Wesley Professional, 2018

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

The content of the discipline is consistent with the similar disciplines from other Romanian universities and universities from abroad, as well as with the requirements that potential employers would have in the software engineering field.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	Knowledge of basic notions about search and clustering algorithms and their applications in software engineering.	Written exam in the exam session	40%
10.5 Seminar	Capability of recognizing and understanding the discussed topics in a research paper. Capability of understanding a new search algorithm	Presentation of a report about a search algorithm and the content of one research paper.	30%
10.6 Seminar/laboratory	Correctness and completeness of the lab projects solved in 2-3 person teams.	Continous observation during the labs. Average grade of the 6 lab projects.	30%
10.7 Minimum standard of performance			
• Each students needs to demonstrate that he/she acquired an acceptable level of knowledge and understanding of the domain and that he/she is capable of coherently expressing this knowledge. • Written exam grade should be greater than 5 and final grade should be greater than 5. • At least 4 attendances are mandatory at the labs and 3 at the seminars, otherwise the student is not allowed to participate at the written exam. • At least one (out of six) lab project should be solved in order to be able to participate at the written exam.			

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*”.

Date:
15.04.2025

Signature of course coordinator

Lect. PhD. Zsuzsanna ONEȚ-MARIAN

Signature of seminar coordinator

Lect. PhD. Zsuzsanna ONEȚ-MARIAN

Date of approval:
...

Signature of the head of department

Assoc. prof. phd. Adrian STERCA