SYLLABUS

Computational Intelligence applications in Software Engineering

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babeş-Bolyai University of Cluj-Napoca
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	Department of Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Master
1.6. Study programme/Qualification	Software Engineering
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline	**Computational Intelligence applications in Software Engineering**					Discipline code	**MME8063**
2.2. Course coordinator						Prof. PhD Czibula Istvan	
2.3. Seminar coordinator						Prof. PhD Czibula Istvan	
2.4. Year of study	2	2.5. Semester	3	2.6. Type of evaluation	E	2.7. Discipline regime	Compulsory

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	**1 sem+ 1 pr**
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	**28**
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					26
Additional documentation (in libraries, on electronic platforms, field documentation)					36
Preparation for seminars/labs, homework, papers, portfolios and essays					35
Tutorship					12
Evaluations					10
Other activities:					
3.7. Total individual study hours	**119**				
3.8. Total hours per semester	**175**				
3.9. Number of ECTS credits	**7**				

4. Prerequisites (if necessary)

4.1. curriculum	Artificial Intelligence
4.2. competencies	Programming skills

5. Conditions (if necessary)

5.1. for the course	Classroom with a projector
5.2. for the seminar /lab activities	Laboratory with computers; high level programming language environment (.NET or any Java environement a.s.o.)

6.1. Specific competencies acquired [1]

Professional/essential competencies	• understanding and working with basic concepts in software engineering; • capability of analysis and synthesis; • modeling and solving real-life problems;
Transversal competencies	• ethic and fair behavior, committment to professional deontology; • team work capabilities; able to fulfill different roles.

6.2. Learning outcomes

Knowledge	The student knows: • the ethical and legal principles and rules in scientific research; • the requirements of research activities in the domain of computer science in general and software engineering in particular and he/she understands the role of research in promoting progress; • the fundamental knowledge for modelling, being able to analyse real life problems and to translate them in concrete requirements and to design a corresponding software model.
Skills	The student is able to: • have an interdisciplinary vision between computer science subdomains in order to combine them in a software system; • devise, model and design of complex software application.
Responsibility and autonomy:	The student has the ability to work independently to: • obtain knowledge necessary for designing, managing and evaluating research activities in software engineering; • devise, model and design of complex software application.

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

7.1 General objective of the discipline	To present the field of Search Based Software Engineering as a new research and application domain of software engineering.
7.2 Specific objective of the discipline	• To introduce the student a new field of Software Engineering- Search Based Software Engineering. • To induce the necessity and importance of using computational intelligence techniques for solving software engineering problems. • To present some important activities within software engineering and how are they solved using computational intelligence techniques.

[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.

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction - Search Based Software Engineering Main concepts and approached problems	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
2. Machine learning in Software Engineering ▪ Machine learning techniques Applications	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
3. CI techniques for Program Comprehension	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
4. CI techniques for Refactoring	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
5. CI techniques for Defect Detection and prediction	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
6. CI techniques for Software Testing	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
7. CI techniques for Software Vizualization	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
8. CI techniques for Effort prediction and Cost estimation	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
9. CI techniques for Software Reuse	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
10. CI techniques for Design Patterns identification	- Interactive exposure - Explanation - Conversation - Didactical demonstration	
CISE research reports presentation	- Interactive exposure - Conversation - Oral assessment	

Bibliography

1. Czibula, I., G., Use of search techniques to software development, Editura Risoprint, ISBN 978-973-53- 0119-4, 2009 (248 pagini)

2. Mark Harman and Bryan F. Jones. Search-based software engineering. Information & Software Technology, 43(14):833-839, 2001.

3. Olaf Seng, Johannes Stammel, and David Burkhart. Search-based determination of refactorings for improving the class structure of object-oriented systems. In GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pages 1909{1916, New York, NY, USA, 2006. ACM Press.

4. Frank Simon, Frank Steinbruckner, and Claus Lewerentz. Metrics based refactoring. In CSMR '01: Proceedings of the Fifth European Conference on Software Maintenance and Reengineering, pages 30- 38, Washington, DC, USA, 2001. IEEE Computer Society.

8.2 Seminar / laboratory	Teaching methods	Remarks

		The activity is structured as 2 hours classes every
1. Administration of seminars. Survey of the sources of information available on Internet and Intranet	- Interactive exposure - Explanation - Conversation	
2. Survey of the sources of information available on Internet and Intranet; chosing the paper topic and scheduling the presentation.	- Documentation - Explanation - Conversation	
A software project on a SBSE topic (Project 1) will be developed using an open source ML development environment. The second project (Project 2) will be realized from scratch and documented. The software will have to demonstrate the use of CI techniques for some specific SE task.		
3. Problem definition and specification for Project 2	- Lab assignment - Explanation - Conversation	
4. Comments about the solution (problem analysis) and search based modeling of the problem (Project 2). Demonstration of Project 1	- Lab assignment - Explanation - Conversation	
5. Design documentation for Project 2	- Lab assignment - Explanation - Conversation	
6. The electronic version of the source code, test files and any other files required to test Project 2. Project 2 demonstration	- Lab assignment - Explanation Conversation	
Bibliography 1. Czibula, I., G., Use of search techniques to software development, Editura Risoprint, ISBN 978-973-53- 0119-4, 2009 (248 pagini) 2. Mark Harman and Bryan F. Jones. Search-based software engineering. Information & Software Technology, 43(14):833-839, 2001. 3. Olaf Seng, Johannes Stammel, and David Burkhart. Search-based determination of refactorings for improving the class structure of object-oriented systems. In GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pages 1909{1916, New York, NY, USA, 2006. ACM Press. 4. Frank Simon, Frank Steinbruckner, and Claus Lewerentz. Metrics based refactoring. In CSMR '01: Proceedings of the Fifth European Conference on Software Maintenance and Reengineering, pages 30- 38, Washington, DC, USA, 2001. IEEE Computer Society.		

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	A theoretical research report on a SBSE topic, based on some recent research papers should be prepared and presented	Evaluation of the research report (a written paper of about 10 pages and an oral presentation)	20%
	The correctness and	Written exam (in the	40%

	completeness of the accumulated knowledge.	regular session)	
	Class attendance	4 unmotivated absences are accepted, but each unmotivated absence other than those specified above are penalised	10%
10.5 Seminar/laboratory	A software project developed using an open source ML software	Evaluation of the project (documentation and demonstration)	15%
	A software project on a SBSE topic will be fully implemented, without using existing libraries.	Evaluation of the project (software implementation, documentation and demonstration)	15%
10.6 Minimum standard of performance			
• Each student has to prove that (s)he acquired an acceptable level of knowledge and understanding of the SBSE field, that (s)he is capable of stating this knowledge in a coherent form, that (s)he has the ability to establish certain connections and to use the knowledge in solving different problems. • Successful passing of the exam is conditioned by the final grade that has to be at least 5.			

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

Not applicable.

Date: Signature of course coordinator Signature of seminar coordinator

10.04.2025 Prof. PhD Czibula Istvan Prof. PhD Czibula Istvan

Date of approval: Signature of the head of department

Assoc. prof. PhD. Adrian STERCA

[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.*".