SYLLABUS

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	**Departament of Computer Science**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**Master**
1.6 Study programme / Qualification	**Applied Computational Intelligence**

2. Information regarding the discipline

2.1 Name of the discipline	**Computational Intelligence applications in Software Engineering**						
2.2 Course coordinator	**Assoc. prof. PhD Czibula Istvan**						
2.3 Seminar coordinator	**Assoc. prof. PhD Czibula Istvan**						
2.4. Year of study	**2**	2.5 Semester	**4**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Compulsory**

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

3.1 Hours per week	3	Of which: 3.2 course	2	3.3 seminar/laboratory	1 sem
3.4 Total hours in the curriculum	36	Of which: 3.5 course	24	3.6 seminar/laboratory	12
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					32
Additional documentation (in libraries, on electronic platforms, field documentation)					43
Preparation for seminars/labs, homework, papers, portfolios and essays					42
Tutorship					10
Evaluations					12
Other activities:					-

3.7 Total individual study hours	139
3.8 Total hours per semester	175
3.9 Number of ECTS credits	7

4. Prerequisites (if necessary)

4.1. curriculum	
4.2. competencies	

5. Conditions (if necessary)

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

6. Specific competencies acquired

Professional competencies	• Advanced ability to approach, model and solve phenomena and problems from nature and economy using fundamental knowledge from mathematics and computer science. • Ability to approach and solve complex problems using various techniques of computational intelligence. • Proficient use of methodologies and tools specific to programming languages and software systems.
Transversal competencies	• Ethic and fair behaviour, commitment to professional deontology • Team work capabilities; able to fulfill different roles • Professional communication skills; concise and precise description, both oral and written, of professional results , negotiation abilities; • Entrepreneurial skills; working with economical knowledge; continuous learning • Good English communication skills

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.

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. SBSE 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	
11. CISE research reports presentation	• Interactive exposure • Conversation	
12. CISE research reports presentation	• Interactive exposure • 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.

8.2 Seminar / laboratory	Teaching methods	Remarks
		The seminar is structured as 2 hours

		classes every second week
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. Design documentation for Project 2	• Lab assignment • Explanation • Conversation	
7. 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

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the 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 completeness of the accumulated knowledge.	Written exam (in the regular session)	40%
	• Class attendance	4 unmotivated absences are accepted, but each unmotivated absence other than those specified above are penalised	10%
10.5 Seminar/lab activities	• 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 ML libraries.	Evaluation of the project (software implementation, documentation and demonstration)	15%
10.6 Minimum performance standards			
• 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 these 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.			

Date	Signature of course coordinator	Signature of seminar coordinator
20.04.2015	Assoc. prof. Istvan Gergely Czibula	Assoc. prof. Istvan Gergely Czibula

Date of approval

Signature of the head of department

Prof. dr. Bazil Pârv