

SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	Babeş Bolyai University
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

2. Information regarding the discipline

2.1 Name of the discipline	Rule based Programming						
2.2 Course coordinator	Assoc.Prof.PhD. Simona Motogna						
2.3 Seminar coordinator	Assoc.Prof.PhD. Simona Motogna						
2.4. Year of study	2	2.5 Semester	3	2.6. Type of evaluation	E	2.7 Type of discipline	Optional

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	
3.4 Total hours in the curriculum	42	Of which: 3.5 course	28	3.6 seminar/laboratory	14	
Time allotment:						hours
Learning using manual, course support, bibliography, course notes						30
Additional documentation (in libraries, on electronic platforms, field documentation)						30
Preparation for seminars/labs, homework, papers, portfolios and essays						70
Tutorship						14
Evaluations						14
Other activities:						-
3.7 Total individual study hours			158			
3.8 Total hours per semester			200			
3.9 Number of ECTS credits			8			

4. Prerequisites (if necessary)

4.1. curriculum	<ul style="list-style-type: none"> None
4.2. competencies	<ul style="list-style-type: none"> Average Java programming skills

5. Conditions (if necessary)

5.1. for the course	<ul style="list-style-type: none"> None
5.2. for the seminar /lab activities	<ul style="list-style-type: none"> Computers, Eclipse framework (free license), Jess (free academic license)

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> • Advanced knowledge of methodological and practical developments in software engineering; • Abilities of using software tools for software development activities • Practical skills referring to advanced knowledge of component-based technologies and languages
Transversal competencies	<ul style="list-style-type: none"> • Project development • Project presentation • Using different programming paradigms in software development

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

7.1 General objective of the discipline	The course will introduce students to a completely different way of programming, in which you specify rules of behavior. It will discuss paradigms that allow rule constructions, or addition of rules, and the application areas for which they are suited.
7.2 Specific objective of the discipline	<ul style="list-style-type: none"> • to demonstrate medium to large scale rule-based program design, • to survey the application areas for which rule based techniques are best suited, and • to provide an introduction to the implementation and semantics of rules.

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction. A review of fundamental data types, rules, and definitions; discussion of various programming paradigms and differences between them [5]	Exposure,description, explanation, debate and dialogue, discussion of case studies	
2. Principles of Rule-based programming: Review of declarative programming and Prolog languages. Using relations as building blocks in program design. Special features of declarative languages. [2]	explanation, debate and dialogue, discussion of case studies	
3. Java Rule Engine: [jsr]	Exposure,description, explanation	
4. Introduction to Jess: structure, basic constructs [7]	Exposure,description, explanation	
5. Facts in Jess: [7]	Exposure,description, explanation	
6. Rules in Jess: writing rules. Firing and execution; Rete algorithm[7]	Exposure,description, explanation	
7. Java and Jess [7]	Exposure,description, explanation, discussion of case studies	
8. Application development using Jess [7]	Exposure,description, explanation, discussion of case studies	
9. XML Transformation Languages [4]	Exposure,description,	

	explanation, discussion of case studies	
10. Rule based systems in Model Transformations [7]	Exposure,description, explanation, discussion of case studies	
11. Rule based systems for .NET framework	Exposure,description, explanation, discussion of case studies	
12. Case study: Junit test framework [8]	Exposure,description, explanation, discussion of case studies	
13. Rule based systems used in industrial applications	Exposure,description, explanation, discussion of case studies	
14. Reserved topic		Usually dedicated to an invited guest from a software company

Bibliography

1. R Bird and P Wadler. An Introduction to Functional Programming (2nd Edition if available). Prentice-Hall. 1996
2. I Bratko. Prolog Programming for Artificial Intelligence. Addison-Wesley
3. Friedman-Hill, Ernest, JESS in Action, Manning, Greenwich, CT, 2003.
4. Kowalski, T., Levy, L.; Rule-Based Programming, Springer, 1996
5. Mitchell, J. Concepts in Programming Languages, Cambridge Univ. Press, 2003
6. S Thomson. The Craft of Functional Programming. Addison-Wesley. 1996.
7. Jess Homepage - <http://www.jessrules.com/jess/index.shtml>
8. JUnit homepage www.junit.org

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Download, install and get used to Jess		Seminar is organized as a total of 7 hours – 2 hours every second week
2. Write small programs in Jess	Dialogue, debate, case studies, examples, proofs	
3. Establish project theme and project architecture	Dialogue, debate, case studies, examples, proofs	
4. Project milestone: facts	Dialogue, debate, case studies, examples, proofs	
5. Project milestone: rules	Dialogue, debate, case studies, examples, proofs	
6. Integration, testing	Dialogue, debate, case studies, examples, proofs	
7. Project presentation	Evaluation	

Bibliography

Same as course

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 course respects the IEEE and ACM Curricula Recommendations for Computer Science studies;

- The content of the course is considered by the software companies as important for average programming skills

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	- know the basic principle of the domain; - apply the course concepts - problem solving	Written exam	60%
10.5 Seminar/lab activities	- be able to implement course concepts - apply techniques for different classes of problems	-Project -documentation -portofolio -continous observations	40%
10.6 Minimum performance standards			
➤ At least grade 5 (from a scale of 1 to 10) at both written exam and laboratory work.			

Date

Signature of course coordinator

Signature of seminar coordinator

.....

Assoc.Prof.PhD. Simona MOTOGNA

Assoc.Prof.PhD. Simona MOTOGNA

Date of approval

Signature of the head of department

.....

.....