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	**Bachelor**
1.6 Study programme / Qualification	**Computer Science**

2. Information regarding the discipline

2.1 Name of the discipline	**Aspect Oriented Programming**						
2.2 Course coordinator	**Assoc. Prof. PhD. Grigoreta Cojocar**						
2.3 Seminar coordinator	**Assoc. Prof. PhD. Grigoreta Cojocar**						
2.4. Year of study	**3**	2.5 Semester	**6**	2.6. Type of evaluation	**C**	2.7 Type of discipline	**Optional**

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

3.1 Hours per week	5	Of which: 3.2 course	2	3.3 seminar/ laboratory	1 lab +2 pr
3.4 Total hours in the curriculum	60	Of which: 3.5 course	24	3.6 seminar/ laboratory	36
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					20
Additional documentation (in libraries, on electronic platforms, field documentation)					40
Preparation for seminars/labs, homework, papers, portfolios and essays					38
Tutorship					8
Evaluations					9
Other activities:					-

3.7 Total individual study hours	115
3.8 Total hours per semester	175
3.9 Number of ECTS credits	7

4. Prerequisites (if necessary)

4.1. curriculum	· Advanced Programming Methods
4.2. competencies	· Average programming skills in Java programming language

5. Conditions (if necessary)

5.1. for the course	· projector
5.2. for the seminar /lab activities	· Laboratory with computers; Java programming language, Eclipse IDE

6. Specific competencies acquired

Professional competencies	· C1.1 Description of programming paradigms and language specific mechanims, and identification of semantics and syntactic aspects differences. · C1.2 Existing software systems explanation based on abstraction levels (architecture, packages, classes, methods) using appropriate basic concepts. · C1.3 Source code elaboration and unit testing of modules in a well-known programming language based on given specification and design data.
Transversal competencies	· CT1 Application of rules for organized and efficient work, of responsible attitudes towards education-scientific domain for creative revaluation of self-potential, respecting the professional ethics principles and norms. · CT3 Usage of efficient learning, information, research and development methods and techniques for knowledge revaluation abilities, for adaptation to the requirements of a dynamic society, and for communication in romanian language and another foreign language.

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

7.1 General objective of the discipline	· Be able to understand AOP and crosscutting concerns · Improved object oriented programming skills · Average aspect oriented programming skills
7.2 Specific objective of the discipline	· To know the concepts of the aspect oriented paradigm · To develop software systems using aspect oriented programming · To be familiar with AspectJ, Spring AOP

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction to AOP. Logging concepts	Exposure: description, explanation, examples, discussion of case studies	

2. AspectJ Language: The join point model, pointcuts syntax	Exposure: description, explanation, examples, discussion of case studies	
3. AspectJ Language: Dynamic behaviour: advice syntax	Exposure: description, explanation, examples, debate, dialogue	
4. AspectJ Language: Static crosscutting	Exposure: description, explanation, examples, discussion of case studies	
5. AspectJ Language: Aspects	Exposure: description, explanation, examples, proofs	
6. AspectJ Language: @AspectJ syntax	Exposure: description, explanation, examples, proofs, debate, dialogue	
7. AspectJ Weaving Models	Exposure: description, explanation, examples, discussion of case studies	
8. Spring AOP	Exposure: description, explanation, examples, discussion of case studies	
9.Design and implementation of security using (Spring) AOP	Exposure: description, explanation, examples, debate	
10. AOP Design Patterns	Exposure: description, explanation, examples, discussion of case studies	
11.Projects presentation	Exposure: description, explanation, examples, discussion of case studies	
12. Reports presentation	Exposure: description, explanation, examples, discussion of case studies	

Bibliography 1. AspectJ Project homepage: http://www.eclipse.org/aspectj/ 2. Ivar Jacobson and Pan-Wei Ng. Aspect-Oriented Software Development with Use Cases. Addison-Wesley, 2004 3. Ramnivas Laddad. AspectJ in Action. Enterprise AOP With Spring Applications, Second Edition, Manning Publications, 2009. 4. Ramnivas Laddad. AspectJ in Action. Practical Aspect-Oriented Programming, Manning Publications, 2003. 5. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015. 6. Spring Documentation http://www.springsource.org 7. Slides: http://www.cs.ubbcluj.ro/~grigo/aop/		
8.2 Laboratory	Teaching methods	Remarks
1. Eclipse and AJDT IDE	Explanation	The lab is structured as 2 hours classes every second week
2. Tracing using Log4J/Logging API	Dialogue, case studies, evaluation	
3. Tracing with AOP	Dialogue, case studies, evaluation	
4. Observer with AOP	Dialogue, case studies, evaluation	
5. Spring AOP for performance monitoring and caching	Dialogue, case studies, evaluation	
6. Spring Security	Dialogue, case studies, evaluation	
Bibliography 8. AspectJ Project homepage: http://www.eclipse.org/aspectj/ 9. Ivar Jacobson and Pan-Wei Ng. Aspect-Oriented Software Development with Use Cases. Addison-Wesley, 2004 10. Ramnivas Laddad. AspectJ in Action. Enterprise AOP With Spring Applications, Second Edition, Manning Publications, 2009. 11. Walls, Craig, Spring in Action, Third Edition, Ed. O'Reilley, 2011. 1. Spring Documentation http://www.springsource.org		

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 course exists in the studying program of all major universities from abroad;
- The content of the course is considered by software companies as important for advanced programming skills

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	· To know the basic concepts of aspect oriented programming	Project	30%
	· To describe another Aspect Oriented language	Report	20%
10.5 Lab activities	- To be able to use aspect oriented concepts to design and implement different crosscutting concerns	Practical examination -observation, running tests	50%
10.6 Minimum performance standards			
At least grade 5 (from a scale of 1 to 10) at project and report. At least grade 5 for the final mark.			

Date	Signature of course coordinator	Signature of seminar coordinator
	Assoc. Prof. PhD. Grigoreta Cojocar	Assoc. Prof. PhD. Grigoreta Cojocar

Date of approval Signature of the head of department

.... ……………………..