SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babes-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	**Software Component Models**						
2.2 Course coordinator	**PhD Lecturer Andreea Vescan**						
2.3 Seminar coordinator	**PhD Lecturer Andreea Vescan**						
2.4. Year of study	**3**	2.5 Semester	**5**	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	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	70
Additional documentation (in libraries, on electronic platforms, field documentation)	12
Preparation for seminars/labs, homework, papers, portfolios and essays	18
Tutorship	3
Evaluations	5
Other activities:	0

3.7 Total individual study hours	108
3.8 Total hours per semester	150
3.9 Number of ECTS credits	6

4. Prerequisites (if necessary)

4.1. curriculum	• Advanced Programming Methods • Object-Oriented Programming
4.2. competencies	• Average programming skills in various high level programming Languages (.NET, Java environment and others)

5. Conditions (if necessary)

5.1. for the course	Video projector, Classroom with network and Internet access and to laboratory

	equipment.
5.2. for the seminar /lab activities	Laboratory with computers; high level programming language environment (.NET and Java and others),

6. Specific competencies acquired

Professional competencies	• Identification of proper methodologies for software systems development; • Identification and explication of proper software systems specification methods; • Using methodologies and tools for development of informatics applications; • Using proper criteria and methods for evaluation of software applications; • Realization of dedicated information projects.
Transversal competencies	• Application of efficient and rigorous working rules, manifest responsible attitudes toward the scientific and didactic fields, respecting the professional and ethical principles. • Use of efficient methods and techniques for learning, information, research and development of abilities for knowledge exploitation, for adapting to the needs of a dynamic society and for communication in Romanian as well as in a widely used foreign language

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

7.1 General objective of the discipline	• know and understand fundamental concepts of Component-based Software Development; • to develop skills in modeling component-based systems systems for various component models;
7.2 Specific objective of the discipline	• will acquire theoretical aspects regarding various component models; • will know similarities and differences between component models. • will know how to model a component-based system using a specific component model.

8. Content

8.1 Course	Teaching methods	Remarks
Lecture 1. 1. Basic concepts (components, composition, component models)	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Lecture 2. 2. The CBD process (component life cycle, system life cycle)	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Lecture 3. 3. The Semantics of Software Components 4. The Syntax of Software Components	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Lecture 4 5. The Composition of Software Components 5.1. An Idealized Component Life Cycle 5.2. Composition in the Design Phase	• Interactive exposure • Explanation • Conversation	

5.3. Composition in the Deployment Phase	• Didactical demonstration	
Lecture 5, 6 6. Software Component Models 6.1. Category 1: Design without Repository - This category includes all simple Acme-like ADLs, UML2.0,PECOS, and Fractal. 6.2. Category 2: Design with Deposit-Only Repository - This category includes EJB, COM, .NET, CCM, and Web Services. The representative example is EJB. 6.3. Category 3: Deployment without Repository - This category contains only JavaBeans. 6.4. Category 4: Design with Repository - This category includes Koala, SOFA, and KobrA.	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Lecture 7 7. A taxonomy 7.1. Categories Based on ComponentSemantics Based on semantics, current component models can be grouped into three categories: 1) component models in which components are classes, 2) models in which components are objects, and 3) those in which components are architectural units 7.2. Categories Based on Component Syntax Based on component syntax, current models fall into three categories: 1) models in which components are defined by object-oriented programming languages, 2) those in which an IDL is used and in which components can be defined in programming languages with mappings from the IDL, and 3) those in which components are defined by ADLs 7.3. A Taxonomy Based on Composition	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Lecture 8 8. Survey of current component models 8.1. categories based on components 8.2 categories based on composition mechanisms	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Lecture 9,10 9. Component models based on objects 9.1. Objects as components 9.2. Method call as a composition mechanism 9.3. Enterprise JavaBeans, JavaBeans	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Lecture 11,12 10. Component models based on architectural units 10.1. Architectural units as components 10.2 Port connection as a composition mechanism 10.3. Acme/ArchJava, UML 2.0	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Lecture 13,14 11. Component models based on encapsulated components	• Interactive exposure • Explanation • Conversation	

11.1. Encapsulated components 11.2. Coordination as a composition mechanism 11.3. Web services, X-MAN	• Didactical demonstration	

Bibliography

[1] K.-K. Lau, Z. Wang, *Software Component Models*, IEEE Trans. on Software Engineering, V 33, n. 10, pp. 709-724, 2007.

[2] Szyperski, C.: *Component Software. Beyond Object-Oriented Programming*, Addison-Wesley (1st ed. 1998, 2nd ed. 2002).

[3] Crnkovic, I., Larsson, M., *Building Reliable Component-Based Software Systems*, Artech House Publisher**,** ISBN 1-58053-327-2, 2002

[4] Heineman, G.T, Councill, W.T., *Component-based software engineering: putting the pieces together,* Addison-Wesley, 2001.

Optional references
Internet resources and conferences

8.2 Seminar / laboratory	Teaching methods	Remarks
		The seminar is structured as 2 hours classes every second week.
		The attendance at seminars is 75% compulsory (5 of 7).
Theme 1 (lab 1-2, weeks 1-4) The first two labs are dedicated to surveying information sources available on Internet and Intranet. In the lab 2 the student must communicate the project title. Live demos are scheduled in the last week.	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Theme 2 (lab 3-4, weeks 5-8) Project design –choose a component model	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	
Theme 3 (lab 5-6, weeks 9-12) Project implementation and presentation.	Presentation, Conversation, Problematizations, Discovery, Individual study, Exercises	

Bibliography

- Students will search and use component model tools suitable for their Project Activity.

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

- This course follows the IEEE and ACM Curriculla Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;

http://www.cs.manchester.ac.uk/study/undergraduate/courses/courseunitsyllabus/?courseunitcode=COMP61521
http://www.idt.mdh.se/kurser/cd5490/

- Course content is considered very important by the software companies for improving advance component-based systems modeling skills.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	• The correctness and completeness of the accumulated knowledge of component models.	Written exam	50%
10.5 Seminar/lab activities	• Class attendance	2 unmotivated absences are accepted, but each unmotivated absence other than those specified above are penalized.	10%
	• Project specification	Evaluation of the project - specification documentation	10%
	• Project development	Evaluation of the project - used component model	20%
	• Project presentation	Evaluation of the project - running the developed application	10%

10.6 Minimum performance standards

- Each student has to prove that:
  - (s)he acquired an acceptable level of knowledge and understanding of the current component models;
  - (s)he has the ability to establish certain connections and to use the knowledge in solving different problems with various component models.
- 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
30.04.2014	Lect. PhD. Andreea Vescan	Lect. PhD. Andreea Vescan

Date of approval

Signature of the head of department

Prof. PhD. Bazil Pârv