

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	Computers and Information Technology
1.5 Study cycle	Bachelor
1.6 Study programme / Qualification	Information Engineering

2. Information regarding the discipline

2.1 Name of the discipline (en) (ro)		Project: Design of algorithms and software systems (Proiect: Proiectarea algoritmilor si sistemelor software)					
2.2 Course coordinator		Lect. PhD. Radu D. Găceanu					
2.3 Seminar coordinator		Lect. PhD. Radu D. Găceanu					
2.4. Year of study	2	2.5 Semester	4	2.6. Type of evaluation	C	2.7 Type of discipline	Compulsory DD
2.8 Code of the discipline	MLE5192						

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

3.1 Hours per week	2	Of which: 3.2 course	-	3.3 seminar/laboratory	2 P
3.4 Total hours in the curriculum	28	Of which: 3.5 course	-	3.6 seminar/laboratory	28
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					6
Additional documentation (in libraries, on electronic platforms, field documentation)					6
Preparation for seminars/labs, homework, papers, portfolios and essays					6
Tutorship					
Evaluations					4
Other activities:					-
3.7 Total individual study hours	22				
3.8 Total hours per semester	50				
3.9 Number of ECTS credits	2				

4. Prerequisites (if necessary)

4.1. curriculum	<ul style="list-style-type: none"> •Advanced Programming Methods •Databases
-----------------	---

	<ul style="list-style-type: none"> •Distributed Operating Systems
4.2. competencies	<ul style="list-style-type: none"> •Average programming skills in a high level programming language •Basic concepts of databases •Basic concepts of networking

5. Conditions (if necessary)

5.1. for the course	Room with projector
5.2. for the seminar /lab activities	Laboratory with internet access and ability to use personal laptops

6. Specific competencies acquired

Professional competencies	<p>C2.1 Describing the structure and operation of hardware, software and communication components</p> <p>C2.2 Explaining the role, interaction and operation of hardware, software and communication components</p> <p>C2.3 Construction of hardware and software components of computing systems using design methods, languages, algorithms, data structures, protocols and technologies</p> <p>C2.4 Metric based evaluation of functional and non-functional characteristics of computing systems</p> <p>C2.5 Implementation of hardware, software components</p> <p>C4.1 Identifying and describing technologies, programming environments and various concepts that are specific to programming engineering</p> <p>C4.2 Explaining the role, interaction and operation patterns of software system components</p> <p>C4.3 Developing specifications and designing information systems using specific methods and tools</p> <p>C4.4 Managing the life cycle of hardware, software and communications systems based on performance evaluation</p> <p>C4.5 Developing, implementing and integrating software solutions</p>
Transversal competencies	<p>CT1 Honorable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation</p> <p>CT2 Identifying, describing and conducting processes in the projects management field, undertaking different team roles and clearly and concisely describing own profesional results, verbally or in writing, in Romanian and in an international language.</p> <p>CT3 Demonstrating initiative and pro-active behavior for updating professional, economical and organizational culture knowledge</p>

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

7.1 General objective of the discipline	<ul style="list-style-type: none"> •To understand distributed software concepts and problems •Improved design and programming skills
---	--

7.2 Specific objective of the discipline	<ul style="list-style-type: none"> •To have a systematic knowledge concerning application development methodologies •To be familiarized with modern concepts and preoccupations in the field of developing application software •To know the use of computer-aided software development tools
--	--

8. Content

8.1 Course	Teaching methods	Remarks
8.2 Seminar / laboratory	Teaching methods	Remarks
1. Build automation, dependency management; version control systems	Presentation, conversation, case studies	
2. JDBC	Presentation, conversation, case studies	
3. Inversion of control containers	Presentation, conversation, case studies	
4. The client server architecture	Presentation, conversation, case studies	
5. Remote procedure call	Presentation, conversation, case studies	
6. Object relational mapping	Presentation, conversation, case studies	
7. Object relational mapping	Presentation, conversation, case studies	
8. Enterprise application integration	Presentation, conversation, case studies	
9. Enterprise application integration	Presentation, conversation, case studies	
10. Web services	Presentation, conversation, case studies	
11. Web applications	Presentation, conversation, case studies	
12. Web sockets	Presentation, conversation, case studies	
13. Web security	Presentation, conversation, case studies	
14. NoSql databases	Presentation, conversation, case studies	
Bibliography <ol style="list-style-type: none"> 1. Joseph Albahari and Ben Albahari, C# 6.0 in a Nutshell, Sixth Edition, O'Reilley, 2015. 2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design and Unified Process, Berlin, Prentice Hall, 2002. 3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002. 4. Hohpe, G., Woolf, B., Enterprise integration patterns, Addison-Wesley, 2003. 5. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/ 6. ***, The Java Tutorial, SUN Microsystems, Inc. http://download.oracle.com/javase/tutorial/ 7. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006 8. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015. 9. Spring http://projects.spring.io/spring-framework 		

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

<ul style="list-style-type: none"> • The course fulfils the IEEE and ACM Curricula Recommendations for Computer Science studies • The content of the course is considered by software companies as being important for average design and advanced programming skills
