

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	Systems for Design and Implementation						
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	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	4	Of which: 3.2 course	2	3.3 seminar/laboratory	2 lab
3.4 Total hours in the curriculum	56	Of which: 3.5 course	28	3.6 seminar/laboratory	28
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					12
Additional documentation (in libraries, on electronic platforms, field documentation)					20
Preparation for seminars/labs, homework, papers, portfolios and essays					40
Tutorship					10
Evaluations					20
Other activities:					-
3.7 Total individual study hours	102				
3.8 Total hours per semester	150				
3.9 Number of ECTS credits	6				

4. Prerequisites (if necessary)

4.1. curriculum	<ul style="list-style-type: none"> • Advanced Programming Methods • Databases • Distributed Operating System
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	•
5.2. for the seminar /lab activities	• Laboratory with computers; high level programming language environment (.NET and Java.), databases

6. Specific competencies acquired

Professional competencies	<ul style="list-style-type: none"> • C2.1 Identification of suitable methodologies for developing software systems. • C2.2 Identification and explanation of suitable mechanism for software systems specification • C2.3 Usage of methodologies, specification mechanisms and development environments for software systems development • C2.4 Usage of suitable criteria and methods for software systems evaluation. • C2.5 Development of specific software systems.
Transversal competencies	<ul style="list-style-type: none"> • 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. • CT2 Efficient development of activities organized in a inter-disciplinary group and the development of emphatic abilities of inter-human communication, relationships and collaboration with different groups • 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	<ul style="list-style-type: none"> • Be able 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 the 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
1. XML documents, DTDs, XML Schema, XML processing in C# and Java	Exposure: description, explanation, examples, discussion of case studies	
2. Object oriented models for accessing relational databases in Java and C#	Exposure: description, explanation, examples, discussion of case studies	
3. Networking in C# and Java	Exposure: description, explanation, examples, debate, dialogue	
4. Remote Procedure Call (RPC), Remoting in C#, RMI in Java	Exposure: description, explanation, examples, discussion of case studies	
5. Reflection in C# and Java	Exposure: description, explanation, examples, proofs	

6. Object/Relational Mapping (ORM), Hibernate (Java), Entity Framework (C#)	Exposure: description, explanation, examples, proofs, debate, dialogue	
7. Inversion of Control based Frameworks, Dependency Injection (Spring, NSpring)	Exposure: description, explanation, examples, discussion of case studies	
8. Enterprise Application Integration - Message Based	Exposure: description, explanation, examples	
9. Web Services - Document Oriented (Contract-First), RPC Based	Exposure: description, explanation, examples, discussion of case studies	
10. Server-side and Client-side Rich Internet Applications (Design and Patterns)	Exposure: description, explanation, examples, debate	
11. Securing web applications and services	Exposure: description, explanation, examples, discussion of case studies	
12. REST Services	Exposure: description, explanation, examples, discussion of case studies	
13. Enterprise Application Integration – SOA	Exposure: description, explanation, examples, discussion of case studies	
14. XPATH, XSLT	Exposure: description, explanation, examples, discussion of case studies	

Bibliography

1. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010.
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, 2004. <http://download.oracle.com/javase/tutorial/>
7. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006
8. David Chappell, Introducing SCA, 2007
9. Walls, Craig, Spring in Action, Third Edition, Ed. O'Reilley, 2011.
10. Spring Documentation <http://www.springframework.org>

8.2 Laboratory	Teaching methods	Remarks
Week1. Assignment of the application. The students have to design and develop a distributed desktop client-server application. Recap of unit testing in Java and C#	Dialogue, case studies	
Week 2: Design the persistence layer. Implementation using XML (Java and C#).	Dialogue, case studies, evaluation	
Week 3. Persistence implementation using relational databases (Java and C#).	Dialogue, case studies, evaluation	
Week 4.: Business layer design and implementation using Java and C#.	Dialogue, case studies, evaluation	
Week 5: Proxy pattern. Design and implementation. (Java and C#)	Dialogue, case studies, evaluation	
Week 6: Distributed Observer Pattern.	Dialogue, case studies,	

Reflection	evaluation	
Week 7. Proxy pattern implementation using protobuff (Java and C#)	Dialogue, case studies, evaluation	
Week 8. Java RMI	Dialogue, case studies, evaluation	
Week 9. .NET Remoting	Dialogue, case studies, evaluation	
Week 10. Dependency Injection (Spring/NSpring Framework)	Dialogue, case studies, evaluation	
Week 11. Thrift/Web Services	Dialogue, case studies, evaluation	
Week 12: Web client for the distributed application	Dialogue, case studies, evaluation	
Bibliography		
<ol style="list-style-type: none"> 1. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010. 2. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/ 3. ***, The Java Tutorial, SUN Microsystems, Inc, 2004. http://download.oracle.com/javase/tutorial/ 4. Walls, Craig, Spring in Action, Third Edition, Ed. O'Reilley, 2011. 5. 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

<ul style="list-style-type: none"> • The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies; • The content of the course is considered by software companies as important for average designing and 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	<ul style="list-style-type: none"> • To know the basic concepts of developing distributed applications • To apply the concepts to design and develop a small client-server application 	Written and Practical exam	60%
10.5 Seminar/lab activities	- be able to design a medium client-server application using different technologies	-Practical examination -documentation -observation	40%
10.6 Minimum performance standards			
At least grade 5 (from a scale of 1 to 10) at practical exam and lab activities, and the final grade at least 5.			

Date	Signature of course coordinator	Signature of seminar coordinator
30.04.2015	Assoc. Prof. PhD. Grigoreta Cojocar	Assoc. Prof. PhD. Grigoreta Cojocar
Date of approval	Signature of the head of department	
.....	