

SYLLABUS

Systems for design and implementation

University year 2025-2026

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		
1.7. Form of education	Full time		

2. Information regarding the discipline

2.1. Name of the discipline	Systems for design and implementation			Discipline code	MLE5013
2.2. Course coordinator	Ioan-Gabriel Mircea				
2.3. Seminar coordinator	Ioan-Gabriel Mircea				
2.4. Year of study	2	2.5. Semester	4	2.6. Type of evaluation	E
				2.7. Discipline regime	Compulsory

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/project	2 LP 1 P
3.4. Total hours in the curriculum	70	of which: 3.5 course	28	3.6 seminar/laboratory/project	28 LP 14P
Time allotment for individual study (ID) and self-study activities (SA)					hours
Learning using manual, course support, bibliography, course notes (SA)					20
Additional documentation (in libraries, on electronic platforms, field documentation)					20
Preparation for seminars/labs, homework, papers, portfolios and essays					21
Tutorship					5
Evaluations					14
Other activities:					
3.7. Total individual study hours	80				
3.8. Total hours per semester	150				
3.9. Number of ECTS credits	6				

4. Prerequisites (if necessary)

4.1. curriculum	Databases Advanced Programming Methods Operating Systems Computer Networks
4.2. competencies	The ability to program in a high-level programming language Basic concepts about databases Basic concepts about computer networks and operating systems

5. Conditions (if necessary)

5.1. for the course	Projector
5.2. for the seminar /lab activities	Projector Laboratories equipped with computers Programming environments for Java, .NET, C++

6.1. Specific competencies acquired ¹

Professional/essential competencies	<ul style="list-style-type: none"> • advanced programming skills in high-level programming languages • development and maintenance of software systems • use of software tools in an interdisciplinary context • use of theoretical foundations of computer science as well as of formal models • design and management of databases • design and administration of computer networks
Transversal competencies	<ul style="list-style-type: none"> • application of organized and efficient work rules, of responsible attitudes towards the didactic-scientific field, to bring creative value to own potential, with respect for professional ethics principles and norms • efficient development of organized activities in an interdisciplinary group and the development of empathetic abilities for • interpersonal communications, to relate to and cooperate with various groups

6.2. Learning outcomes

Knowledge	<ul style="list-style-type: none"> • The student is able to apply architectural patterns, design patterns, and best practices in the field to design highly complex software applications. • The student has appropriate knowledge related to the use of integrated development environments for creating large-scale complex applications. • The student has the ability to understand and use design patterns for application development.
Skills	<ul style="list-style-type: none"> • The student has the ability to develop, design, and create new applications, systems, or products using best practices in the field. • The student has the necessary skills to understand and use object-oriented programming concepts in the development of medium-complexity software applications.
Responsibility and autonomy:	<ul style="list-style-type: none"> • The student has the ability to apply general rules to specific problems and produce relevant solutions. • The student is able to combine diverse information to formulate solutions and generate development ideas for new products and applications. • The student knows multiple programming languages and is capable of writing applications in compiled, interpreted, or dynamic languages, having the ability to choose the appropriate programming language for the specifics of the application being developed.

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
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

¹ One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

8. Content

8.1 Course	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 (I)	Presentation, conversation, case studies	
9. Enterprise application integration (II)	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

Joseph Albahari, C# 10 in a Nutshell, Sixth Edition, O'Reilly, 2022.

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, Inc. <https://docs.oracle.com/javase/tutorial/>

7. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006

8. Walls, Craig, Spring in Action, Fifth Edition, Ed. O'Reilly, 2018

8.2 Seminar / laboratory	Teaching methods	Remarks
Build automation, dependency management; version control systems	Presentation, conversation, case studies	
JDBC	Presentation, conversation, case studies	
Inversion of control containers	Presentation, conversation, case studies	
The client server architecture	Presentation, conversation, case studies	
Remote procedure call	Presentation, conversation, case studies	
Object relational mapping	Presentation, conversation, case studies	
Object relational mapping	Presentation, conversation, case studies	
Enterprise application integration (I)	Presentation, conversation, case studies	

Enterprise application integration (II)	Presentation, conversation, case studies	
Web services	Presentation, conversation, case studies	
Web applications	Presentation, conversation, case studies	
Web sockets	Presentation, conversation, case studies	
Web security	Presentation, conversation, case studies	
NoSql databases	Presentation, conversation, case studies	

Bibliography

- 1.Joseph Albahari, C# 10 in a Nutshell, Sixth Edition, O'Reilly, 2022
- 2.***, Microsoft Developer Network, Microsoft Inc., <http://msdn.microsoft.com/>
- 3.***, The Java Tutorial, <https://docs.oracle.com/javase/tutorial/>
4. Walls, Craig, Spring in Action, Fifth Edition, Ed. O'Reilly, 2018.

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 follows the IEEE and ACM curricular recommendations for studies in computer science.
- Software companies consider the course content useful in developing the students' modeling and programming skills.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	To know the basic concepts of developing distributed applications	Written exam	20
	To apply these concepts to design and implement a small distributed applications	Practical exam	60
10.5 Seminar/laboratory	Being able to design and implement distributed applications using various technologi	Project (iterative development): Practical examination, documentations	20

10.6 Minimum standard of performance

- Students must attend the written and practical exams during the normal or resit session
- The practical exam consists of several iterative stages in which students are given requirements step by step and are required to build an application accordingly
- Each student has to prove that they acquired an acceptable level of knowledge and understanding of the core concepts taught in the class, that they are capable of using knowledge in a coherent form, that they have the ability to establish certain connections and to use the knowledge in solving different problems
- For participating at the examination attendance is compulsory for project and for laboratory activities, as follows: minimum 5 attendances for project and minimum 12 attendances for laboratory activities
- Successfully passing of the examination is conditioned by a minimum grade of 5 for each of the following: practical examination, written examination and project

11. Labels ODD (Sustainable Development Goals)²

Not applicable.

Date: _____ Signature of course coordinator _____ Signature of seminar coordinator _____

3

Lect. 18: D. L. Gabor and M.

Signature of seminar coordinator

Lect. PhD. Ioan-Gabriel Mircea

Lect. PhD. Ioan-Gabriel Mircea

Date of approval: _____ Signature of the head of department _____

...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA

² Keep only the labels that, according to the [*Procedure for applying ODD labels in the academic process*](#), suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable*.”.