COURSE DESCRIPTION

Project: Design of algorithms and software systems

Academic year 2026

1. Programme-related data

1.1. Higher Education Institution	Babeș-Bolyai University Cluj-Napoca
1.2. Faculty	Faculty of Mathematics and Informatics
1.3. Department	Department of Informatics
1.4. Field	Computer Science
1.5. Level of study	Bachelor
1.6. Degree programme / Qualification	Information Engineering (in English)
1.7. Form of education	Full time

2. Course-related data

2.1. Course title	**Project: Design of algorithms and software systems**			Discipline code	**MLE5192**
2.2. Course coordinator	Lect. PhD. Horea-Bogdan Mureșan				
2.3. Seminar coordinator	Lect. PhD. Horea-Bogdan Mureșan				
2.4. Year of study	2	2.5. Semester	4	2.6. Type of assessment	Exam
2.7. Course status	Compulsory			2.8. Course type	Core subject

3. Total estimated time (hours per semester of teaching activities)

3.1. Number of hours per week	**2**	of which: 3.2. course	**-**	3.3. seminar/ laboratory/ project	**0/0/2**
3.4. Total of hours in the curriculum	28	of which: 3.5. course	**-**	3.6. seminar/ laboratory	**28**

Time allocation for individual study (IS) and self-taught activities (ST)	**hours**
Learning from textbooks, course materials, bibliography, and notes (IS)	6
Additional research in the library, on subject-specific electronic platforms, and on-site	6
Preparing seminars/ laboratories/ projects, assignments, reports, portfolios, and essays	6
Tutoring (professional guidance)	-
Examinations	4
Other activities	-
3.7. Total hours of individual study (IS) and self-taught activities (ST)	**22**
3.8. Total hours per semester	**50**
3.9. Number of credits	**2**

4. Prerequisites (where applicable)

4.1. curriculum-related	• Advanced Programming Methods • Databases • Distributed Operating Systems
4.2 skills-related	• Average programming skills in a high-level programming language • Basic concepts of databases • Basic concepts of networking

5. Specific conditions (where applicable)

5.1. course-related	Room with projector
5.2. seminar/laboratory-related	Laboratory with internet access and ability to use personal laptops

6.1. Competencies resulting from the completion of the degree programme *(as referred to in the curriculum)*[1]

Professional competencies	
Competency code	**Competency**
PC1	create software
PC3	analyse software specifications
PC4	define software architecture
PC9	fix software bugs
PC11	use software design patterns
Transversal competencies	
Competency code	**Competency**
TC1	Work independently
TC2	Solve problems
TC3	Think analytically

6.2. Learning outcomes relevant to the degree programme (as referred to in the curriculum)[2]

Learning outcomes targeted by the subject		
Competency code	**Knowledge and comprehension**	**Specific academic skills**
PC1	The student/graduate identifies, explains and justifies fundamental concepts of data structures, algorithms, and programming paradigms, as well as computer architecture.	The student/graduate designs, develops and demonstrates complex software solutions using efficient algorithms and diverse programming paradigms.
PC3	The student/graduate names, gives examples, concludes, specifies, recognizes and critically argues the methods of designing and managing complex IT projects using modern strategies.	The student/graduate initiates, prepares, implements, and proposes methods for developing complex IT projects. The student/graduate produces specific professional reports.
PC4	The student/graduate knows and explains modern programming paradigms, software architectures and software projects development methodologies.	The student/graduate designs, plans, builds, develops scalable software applications, and efficiently uses hardware and software resources.
PC9	The student/graduate selects, describes, analyzes and explains modern programming paradigms, including functional, object-oriented and parallel programming, using current languages and frameworks.	The student/graduate designs, plans, builds, develops scalable software applications, and efficiently uses hardware and software resources.
PC11	The student/graduate knows and software design patterns and user-driven design methodologies for desktop, mobile and web platforms.	The student/graduate designs and develops scalable software applications using modern practices and design patterns largely used in the industry.

[1] The professional and/or transversal skills targeted by the subject for which the course description is prepared will be copied from the curriculum of the degree programme. For each competency, the complete entry, including the competency code, will be copied with the exact wording that appears in the curriculum, without any changes. If no competency is copied from either of the two categories, the row corresponding to that category is deleted from the table.

[2] The learning outcomes relevant for the degree programme and targeted by the subject for which the course description is prepared will be listed. The entries, copied without any changes from the Curriculum by subject type (Core Subject/Specialisation Subject/Complementary Subject), are listed under the corresponding competency.

TC1 **TC2** **TC3**	The student/graduate has the knowledge necessary to understand and solve complex problems, and to plan and organize advanced processes in various fields.	The graduate is able to identify complex problems and examine related issues to develop solving options and implement solutions.The graduate has the ability to apply general rules to specific problems and produce relevant solutions.The graduate is able to combine diverse information to formulate solutions and generate ideas for developing new products and applications.

7. Subject-specific learning outcomes

Knowledge and comprehension
• To understand distributed software concepts and problems
• To understand distributed software concepts and problems
Specific academic skills
• 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. Contents

8.2 Seminar / laboratory	Teaching methods	Remarks
S1. Using an automatic build tool. Choosing an application for the project.	Presentation, conversation, case studies	
S2.Accessing a relational database.	Presentation, conversation, case studies	
S3. Configuring an application using IoC.	Presentation, conversation, case studies	
S4-S5. Designing and implementing services (Proxy Pattern).	Presentation, conversation, case studies	
S7. Enterprise Application Integration (Protobuf, gRPC, Thrift).	Presentation, conversation, case studies	
S8. ORM tools.	Presentation, conversation, case studies	
S9. REST services.	Presentation, conversation, case studies	
S10. Web clients.	Presentation, conversation, case studies	
S11. Asynchronous messaging systems.	Presentation, conversation, case studies	
S12. Websockets.	Presentation, conversation, case studies	
S13. REST Services	Presentation, conversation, case studies	
Bibliography 1.Joseph Albahari and Ben Albahari, C# 6.0 in a Nutshell, Sixth Edition, O'Reilley, 2015. 2.***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/ 3.***, The Java Tutorial, SUN Microsystems, Inc. http://download.oracle.com/javase/tutorial/ 4. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015. 5. Documentatie Spring http://projects.spring.io/spring-framework/		

9. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade

10.4 Course			
10.5 Seminar/laboratory	Being able to design and implement distributed applications using various technologies	Homeworks, developed systems, and documentation	100%
10.6 Minimum standard of performance			
• To pass the subject the student must obtain at least a grade 5 for laboratory work.			

10. SDG labels (Sustainable Development Goals)[3]

	○	Sustainable Development Generic Label						
1 FĂRĂ SĂRĂCIE	2 FOAMETE „ZERO"	3 SĂNĂTATE ȘI BUNĂSTARE	4 EDUCAȚIE DE CALITATE	5 EGALITATE DE GEN	6 APĂ CURATĂ ȘI SANITAȚIE	7 ENERGIE CURATĂ ȘI LA PREȚURI ACCESIBILE	8 MUNCĂ DECENTĂ ȘI CREȘTERE ECONOMICĂ	9 INDUSTRIE, INOVAȚIE ȘI INFRASTRUCTURĂ
○	○	○	○	○	○	○	○	○
10 INEGALITĂȚI REDUSE	11 ORAȘE ȘI COMUNITĂȚI DURABILE	12 CONSUM ȘI PRODUCȚIE RESPONSABILE	13 ACȚIUNE CLIMATICĂ	14 VIAȚA ACVATICĂ	15 VIAȚA TERESTRĂ	16 PACE, JUSTIȚIE ȘI INSTITUȚII EFICIENTE	17 PARTENERIATE PENTRU REALIZAREA OBIECTIVELOR	No label applies
○	○	○	○	○	○	○	○	X

Date of entry:
06.05.2026

Signature of course coordinator

Lect. PhD. Horea-Bogdan Mureșan

Signature of seminar coordinator

Lect. PhD. Horea-Bogdan Mureșan

Date of approval in the department:
...

Signature of the head of department

Assoc. Prof. PhD. Adrian STERCA

[3] Select a single label which, according to the *Implementation of SDG labels in the academic process*, best matches the subject. If the subject addresses sustainable development in a generic manner (i.e. by presenting/introducing the general framework of sustainable development, etc.), then the Sustainable Development generic label may be applied. If none of the labels describe the subject, select the last option: *"No label applies."*