SYLLABUS

Algorithm Designs

University year 2025

1. Information regarding the programme

1.1. Higher education institution	Babeş-Bolyai University Cluj-Napoca
1.2. Faculty	Faculty of Mathematics and Informatics
1.3. Department	Department of Informatics
1.4. Field of study	Computers and Information Technology
1.5. Study cycle	Bachelor
1.6. Study programme/Qualification	Information Engineering
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline	**Algorithm Designs**				Discipline code	**MLE5173**	
2.2. Course coordinator	Assist. PhD. Horea-Bogdan Mureșan						
2.3. Seminar coordinator	Assist. PhD. Horea-Bogdan Mureșan						
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	**4**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**0/2/0**
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	**28**
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					15
Additional documentation (in libraries, on electronic platforms, field documentation)					15
Preparation for seminars/labs, homework, papers, portfolios and essays					5
Tutorship					5
Evaluations					4
Other activities:					-
3.7. Total individual study hours	**44**				
3.8. Total hours per semester	**100**				
3.9. Number of ECTS credits	**4**				

4. Prerequisites (if necessary)

4.1. curriculum	● Advanced Programming Methods ● Databases ● Distributed Operating Systems
4.2. competencies	● 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

6.1. Specific competencies acquired [1]

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

6.2. Learning outcomes

Knowledge	● The graduate knows and understands the concepts and the techniques of knowledge representation and is able to apply them for problem solving. ● The graduate has the necessary knowledge for the use of computers, the development of software programs and applications, and for the information processing.
Skills	● The graduate is able to apply architectural templates, design templates and best practices in the field to design highly complex software applications. ● The graduate is able to combine diverse information to formulate solutions and develop development ideas for new products and applications.
Responsibility and autonomy:	● The graduate has the ability to choose and use programming paradigms (procedural, object-oriented, functional) to create software applications appropriate to the specific field of the developed application. ● The graduate has the necessary knowledge related to the stages of the software life cycle and software process models.

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

7.1 General objective of the discipline	● To understand distributed software concepts and problems ● Improved design and programming skills

[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.

7.2 Specific objective of the discipline	● 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
1. Build automation, dependency management Gradle	Presentation, conversation, case studies	
2. Object-oriented models for accessing databases JDBC	Presentation, conversation, case studies	
3. Object-oriented models for accessing databases ADO.NET	Presentation, conversation, case studies	
4. Inversion of Control Spring	Presentation, conversation, case studies	
5. Client-server applications Proxy pattern	Presentation, conversation, case studies	
6. Client-server applications (cont.) Proxy pattern	Presentation, conversation, case studies	
7. Introduction to Remote Procedure Calls Enterprise Application Integration Protocol buffers	Presentation, conversation, case studies	
8. Enterprise Application Integration gRPC, Thrift	Presentation, conversation, case studies	
9. Object Relational Mapping Strategies. Hibernate, Entity Framework	Presentation, conversation, case studies	
10. REST	Presentation, conversation, case studies	
11. Web application development using frameworks	Presentation, conversation, case studies	
12. Web sockets	Presentation, conversation, case studies	
13. Enterprise Application Integration - Asynchronous messaging systems Activemq, rabbitmq, Jms	Presentation, conversation, case studies	
14. Web security - Role-based access	Presentation, conversation, case studies	

Bibliography
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
9. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015.
10. Documentație Spring http://projects.spring.io/spring-framework/

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. 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 fulfils the IEEE and ACM Curricula Recommendations for Computer Science studies
- The content of the course is considered by software companies to be important for average design and advanced 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; To apply these concepts to design and implement a small distributed application	Examination	60%
10.5 Seminar/laboratory	Being able to design and implement distributed applications using various technologies	Homeworks, developed systems, and documentation	30%
		Laboratory work	10%
10.6 Minimum standard of performance			

- Attendance to this subject and a minimum of 12 attendances are required.
- To pass the subject the student must obtain at least a grade 5 for laboratory work and the exam, and the final grade (calculated according to the weights) is at least 5.

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date: 14.04.2025	Signature of course coordinator Asist. Dr. Horea-Bogdan Mureșan	Signature of seminar coordinator Asist. Dr. Horea-Bogdan Mureșan

Date of approval: ...	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*.".