

# COURSE DESCRIPTION

## *Algorithm Designs*

Academic year 2026

### 1. Programme-related data

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	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	<b>Algorithm Designs</b>			Discipline code	<b>MLE5173</b>
2.2. Course coordinator	Lect. PhD. Horea-Bogdan Mureşan				
2.3. Seminar coordinator	Lect. PhD. Horea-Bogdan Mureşan				
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	4	of which: 3.2. course	2	3.3. seminar/ laboratory/ project	0/2/0
3.4. Total of hours in the curriculum	56	of which: 3.5. course	28	3.6. seminar/ laboratory	42
<b>Time allocation for individual study (IS) and self-taught activities (ST)</b>					<b>hours</b>
Learning from textbooks, course materials, bibliography, and notes (IS)					15
Additional research in the library, on subject-specific electronic platforms, and on-site					15
Preparing seminars/ laboratories/ projects, assignments, reports, portfolios, and essays					5
Tutoring (professional guidance)					5
Examinations					4
Other activities					-
<b>3.7. Total hours of individual study (IS) and self-taught activities (ST)</b>				<b>44</b>	
<b>3.8. Total hours per semester</b>				<b>100</b>	
<b>3.9. Number of credits</b>				<b>4</b>	

### 4. Prerequisites (where applicable)

4.1. curriculum-related	<ul style="list-style-type: none"> <li>• Advanced Programming Methods</li> <li>• Databases</li> <li>• Distributed Operating Systems</li> </ul>
4.2 skills-related	<ul style="list-style-type: none"> <li>• Average programming skills in a high-level programming language</li> <li>• Basic concepts of databases</li> <li>• Basic concepts of networking</li> </ul>

### 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)<sup>1</sup>

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)<sup>2</sup>

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.

<sup>1</sup> 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.

<sup>2</sup> 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.

<p><b>TC1</b> <b>TC2</b> <b>TC3</b></p>	<p>The student/graduate has the knowledge necessary to understand and solve complex problems, and to plan and organize advanced processes in various fields.</p>	<p>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.</p>
---	--	--

### 7. Subject-specific learning outcomes

<b>Knowledge and comprehension</b>
<ul style="list-style-type: none"> <li>● To understand distributed software concepts and problems</li> </ul>
<ul style="list-style-type: none"> <li>● To understand distributed software concepts and problems</li> </ul>
<b>Specific academic skills</b>
<ul style="list-style-type: none"> <li>● To be familiarized with modern concepts and preoccupations in the field of developing application software</li> </ul>
<ul style="list-style-type: none"> <li>● To know the use of computer-aided software development tools</li> </ul>

### 8. Contents

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'Reilly, 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., <a href="http://msdn.microsoft.com/">http://msdn.microsoft.com/</a> 6. ***, The Java Tutorial, SUN Microsystems, Inc. <a href="http://download.oracle.com/javase/tutorial/">http://download.oracle.com/javase/tutorial/</a> 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 <a href="http://projects.spring.io/spring-framework/">http://projects.spring.io/spring-framework/</a>		
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., <a href="http://msdn.microsoft.com/">http://msdn.microsoft.com/</a> 3. ***, The Java Tutorial, SUN Microsystems, Inc. <a href="http://download.oracle.com/javase/tutorial/">http://download.oracle.com/javase/tutorial/</a> 4. Walls, Craig, Spring in Action, Fourth Edition, Ed. O'Reilley, 2015. 5. Documentație Spring <a href="http://projects.spring.io/spring-framework/">http://projects.spring.io/spring-framework/</a>		

## 9. 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			
<ul style="list-style-type: none"> <li>Attendance to this subject and a minimum of 12 attendances are required.</li> </ul>			

- 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.

### 10. SDG labels (Sustainable Development Goals)<sup>3</sup>

	<input type="radio"/>	Sustainable Development Generic Label							
									
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
								No label applies	
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		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

<sup>3</sup> 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."