

COURSE DESCRIPTION

Programming Paradigms

Academic year 2026-2027

1. Programme-related data

1.1. Higher Education Institution	Babes-Bolyai University
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Doctoral School	Mathematics and Computer Science
1.4. Field of study	Computer Science
1.5. Level of study	Doctoral

2. Course-related data

2.1. Course title	Programming Paradigms			Course code	MDE8169
2.2. Course coordinator	Assoc. Prof. Eng. PhD Florin Craciun				
2.3. Seminar coordinator	Assoc. Prof. Eng. PhD Florin Craciun				
2.4. Year of study	1	2.5. Semester	1	2.6. Type of assessment	Exam
2.7. Course status	Optional			2.8. Course type	Core subject

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

3.1. Number of hours per week	3	of which: 3.2. course	2	3.3. seminar/ laboratory/ project	1
3.4. Total of hours in the curriculum	36	of which: 3.5. course	24	3.6. seminar/ laboratory	12
Time allocation for individual study (IS) and self-taught activities (ST)					hours
Learning from textbooks, course materials, bibliography, and notes (IS)					50
Additional research in the library, on subject-specific electronic platforms, and on-site					65
Preparing seminars/ laboratories/ projects, assignments, reports, portfolios, and essays					50
Tutoring (professional guidance)					25
Examinations					24
Other activities					0
3.7. Total hours of individual study (IS) and self-taught activities (ST)				214	
3.8. Total hours per semester				250	
3.9. Number of credits				10	

4. Prerequisites (where applicable)

4.1. curriculum-related	Fundamentals of Programming, Object-Oriented Programming Functional and Logic Programming
4.2 skills-related	Average software development skills

5. Specific conditions (where applicable)

5.1. course-related	Classroom with a projector
5.2. seminar/laboratory-related	

6. Subject-specific learning outcomes

Knowledge
1. Knows advanced concepts, theories, and models in the field of programming languages.
2. Identifies cutting-edge literature in the field of programming languages.
3. Knows methods and techniques for identifying and formulating problems.
4. Understands the criteria for relevance and originality of scientific problems.
Skills
1. Applies knowledge to analyze complex problems.
2. Integrates theoretical and practical concepts for innovative solutions.
3. Formulates original problems and proposes creative solutions.
4. Applies innovative methods to solve identified problems.
Responsibility and autonomy
1. Works autonomously in investigating and expanding knowledge.
2. Takes responsibility for the validity of conclusions.
3. Takes responsibility for the quality and impact of solutions.
4. Works autonomously in defining research directions.

7. Contents

7.1. Course	Teaching and learning methods	Remarks¹
1. Oz syntax, data structures	Interactive exposure, Explanation, Conversation, Didactical demonstration	
2. Statements, Kernel Language, Abstract Machine	Interactive exposure, Explanation, Conversation, Didactical demonstration	
3. Higher-Order Programming	Interactive exposure, Explanation, Conversation, Didactical demonstration	
4. Lambda Calculus	Interactive exposure, Explanation, Conversation, Didactical demonstration	
5. Tupled Recursion and Exceptions	Interactive exposure, Explanation, Conversation, Didactical demonstration	
6. Types, ADT, Components	Interactive exposure, Explanation, Conversation, Didactical demonstration	
7-9. Declarative Concurrency	Interactive exposure, Explanation, Conversation, Didactical demonstration	
10. Stateful Programming	Interactive exposure, Explanation, Conversation, Didactical demonstration	

¹ For example, organisational aspects, recommendations for students, specific aspects relating to the course/seminar, such as inviting experts in the field, etc.

11. Relational Programming	Interactive exposure, Explanation, Conversation, Didactical demonstration	
12. Constraint Programming	Interactive exposure, Explanation, Conversation, Didactical demonstration	

Bibliography

1. SCOTT, MICHAEL L.: Programming Language Pragmatics, 4th ed, Morgan-Kaufmann, 2016
2. SEBESTA, ROBERT W.: Concepts of Programming Languages, 10th ed, Pearson Education, 2012
3. SZYPERSKI, CLEMENS: Component Software. Beyond Object-Oriented Programming, Addison Wesley (1st ed. 1998, 2nd ed. 2002 with GRUNTZ, DOMINIK and MURER, STEFAN).
4. STROUSTRUP, BJARNE: The C++ Programming Language Special Edition, Addison-Wesley, 2000 chapter 2
5. VAN ROY, PETER; HARIDI, SEIF: Concepts, Techniques and Models of Computer Programming, MIT Press, 2004
6. WATT, David A.: Programming Language Design Concepts, Wiley, 2004

7.2. Seminar/ laboratory	Teaching and learning methods	Remarks
1. Research papers allocation and First programming assignment	Interactive exposure, Conversation	
2. Second programming assignment	Interactive exposure, Conversation	
3. First research papers discussion	Interactive exposure, Conversation	
4. Third programming assignment	Interactive exposure, Conversation	
5. Second research papers discussion	Interactive exposure, Conversation	
6. Third Research papers discussion	Interactive exposure, Conversation	

Bibliography

Recent research papers from the conferences POPL, OOPSLA, PLDI
Mozart System

8. Evaluation

Type of activity	8.1 Evaluation criteria ²	8.2 Evaluation methods ³	8.3 Percentage in the final grade
8.4. Course	know the basic principle of the domain;	Final written exam	60%
	apply the course concepts problem solving		
8.5. Seminar/ laboratory	apply the course concepts in problem solving	Oral assesment	40%
	Ability to critically analyse research papers		
8.6 Minimum standard for passing			

² The evaluation criteria must directly reflect the learning outcomes targeted at the level of the degree programme respectively at the level of the subject. More specifically, the learning outcomes set out in the expected learning outcomes are assessed.

³ Both final evaluation methods and ongoing evaluation strategies should be established.

At least grade 5 (from a scale of 1 to 10) at both final written exam and seminar work.

9. SDG labels (Sustainable Development Goals)⁴

	<input type="radio"/>	Sustainable Development Generic Label						
								
<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"/>					

Date of entry:
16.02.2026

Signature of course coordinator
Assoc. Prof. Eng. PhD Florin Craciun

FCraciun

Signature of seminar coordinator
Assoc. Prof. Eng. PhD Florin Craciun

FCraciun

Date of approval in the department:
...

Signature of the head of department
.....

⁴ 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."