syllabus

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	**Computers and Information Technology**
1.5 Study cycle	**Bachelor**
1.6 Study programme / Qualification	**Information Engineering**

2. Information regarding the discipline

2.1 Name of the discipline (en) (ro)			**Programming Paradigms** **Paradigme de Programare**				
2.2 Course coordinator			**Assoc. Prof. Eng. Florin Craciun**				
2.3 Seminar coordinator			**Assoc. Prof. Eng. Florin Craciun**				
2.4. Year of study	**2**	2.5 Semester	**3**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Compulsory DD**
2.8 Code of the discipline		MLE5172					

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	1 S 2 LP
3.4 Total hours in the curriculum	70	Of which: 3.5 course		28	3.6 seminar/laboratory	42
Time allotment:						hours
Learning using manual, course support, bibliography, course notes						25
Additional documentation (in libraries, on electronic platforms, field documentation)						5
Preparation for seminars/labs, homework, papers, portfolios and essays						30

Tutorship						10
Evaluations						10
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	· Object-oriented programming, Data Structures and Algorithms
4.2. competencies	· Programming skills

5. Conditions (if necessary)

5.1. for the course	· Video projector
5.2. for the seminar /lab activities	· Video projector

6. Specific competencies acquired

Professional competencies	· C3.1 Identifying classes of problems and solving methods that are specific to computing systems · C3.2 Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results · C3.3 Applying solution patterns using specific engineering tools and mehods · C3.4 Comparatively and experimentaly evaluation of the alternative solutions for performance optimization · C3.5 Developing and implementing information system solutions for concrete problems · 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 · C4.5 Developing, implementing and integrating software solutions
Transversal competencies	· CT1 Honorable, responsible, ethical behavior, in the spirit of the law, to ensure the professional reputation · CT3 Demonstrating initiative and pro-active behavior for updating professional, economical and organizational culture knowledge

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

7.1 General objective of the	· Each student has to prove that (s)he acquired an acceptable level of

discipline	knowledge and understanding of the subject, that (s)he is capable of stating these knowledge in a coherent form, that (s)he has correct habits of analysis, design, and implementation based on design patterns and programming paradigms
7.2 Specific objective of the discipline	· The students should have the ability to use Java and C# language, design patterns, and to create GUI for their applications. Also they have to be able to use different programming paradigms concepts in program analysis and design.

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction to Java platform: platform, language syntax, primitive data types, arrays, classes, interfaces, packages, enums, overriding, overloading, exceptions	Exposure, description, explanation, debate and dialogue, discussion of case studies	
2. Collections and Generic Types: anonymous classes, polymorphism, casting	Exposure, description, explanation, debate and dialogue, discussion of case studies	
3. IO,NIO: binary and character oriented streams, files, channels and buffers	Exposure, description, explanation, debate and dialogue, discussion of case studies	
4. Functional programming: lambda expressions, streams	Exposure, description, explanation, debate and dialogue, discussion of case studies	
5. GUI: Java FX components, event handling	Exposure, description, explanation, debate and dialogue, discussion of case studies	
6. Concurrency: threads, executors, futures, exception handling	Exposure, description, explanation, debate and dialogue, discussion of case studies	
7. Concurrency: sync vs async methods,	Exposure,	

callback methods, cancellation	description, explanation, debate and dialogue, discussion of case studies	
8. XML: schema, documents	Exposure, description, explanation, debate and dialogue, discussion of case studies	
9. GUI (cont.):FXML, CSS. Metaprogramming: reflection, serialization	Exposure, description, explanation, debate and dialogue, discussion of case studies	
10. Introduction in C# and .Net	Exposure, description, explanation, debate and dialogue, discussion of case studies	
11. Collections in C#	Exposure, description, explanation, debate and dialogue, discussion of case studies	
12. IO operations in C#	Exposure, description, explanation, debate and dialogue, discussion of case studies	
13. GUI in C#	Exposure, description, explanation, debate and dialogue, discussion of case studies	
14. LINQ	Exposure, description, explanation, debate and dialogue, discussion of case studies	
Bibliography		

1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition. 2. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006 3. Eckel, B.: Thinking in Patterns with Java, 2004. MindView, Inc 4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994 5. ***, The Java Tutorial, 2022. http://download.oracle.com/javase/tutorial/ 6. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010 7. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/, 2023 8. Michael L. Scott, Programming Language Pragmatics, Morgan Kaufmann, 4th Edition, 2019. 9. Aleksandar Prokopec, Learning Concurrent Programming in Scala, Packt Publishing, 2nd Edition, 2020. 10. Steve Klabnik and Carol Nichols, The Rust Programming Language, 2021.		
8.2 Laboratory	Teaching methods	Remarks
1. Java basic project	Conversation, debate, case studies, examples	
2. Java project: Collections, Generics	Conversation, debate, case studies, examples	
3. Java project: Generics	Conversation, debate, case studies, examples	
4. Java project: IO	Conversation, debate, case studies, examples	
5. Java project: Functional programming	Conversation, debate, case studies, examples	
6. Java project: concurrency	Conversation, debate, case studies, examples	

7. Java project: GUI	Conversation, debate, case studies, examples	
8. Java project:xml	Conversation, debate, case studies, examples	
9. C# project basics	Conversation, debate, case studies, examples	
10. C# project collections	Conversation, debate, case studies, examples	
11. C# project io	Conversation, debate, case studies, examples	
12. C# project linq	Conversation, debate, case studies, examples	
13. C# project concurenta	Conversation, debate, case studies, examples	
14. C# project GUI	Conversation, debate, case studies, examples	
Bibliography 1. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition. 2. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006 3. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994 4. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010 5. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/, 2023 6. ***, The Java Tutorial, 2022. http://download.oracle.com/javase/tutorial/ 7. Michael L. Scott, Programming Language		

Pragmatics, Morgan Kaufmann, 4th Edition, 2019. 8. Aleksandar Prokopec, Learning Concurrent Programming in Scala, Packt Publishing, 2nd Edition, 2020. 9. Steve Klabnik and Carol Nichols, The Rust Programming Language, 2021.		
8.3 Seminar	Teaching methods	Remarks
1. Java: basics, Collections, Generics.	Conversation, debate, case studies, examples	
2. Java: IO and functional programming	Conversation, debate, case studies, examples	
3. Java: concurrency, GUI and xml.	Conversation, debate, case studies, examples	
4. C# basics, collections.	Conversation, debate, case studies, examples	
5. C# : IO, linq	Conversation, debate, case studies, examples	
6. C#: concurency	Conversation, debate, case studies, examples	
7. C#: GUI	Conversation, debate, case studies, examples	
Bibliography 10. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition. 11. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006 12. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994 13. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O’Reilley, 2010		

14. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/, 2023 15. ***, The Java Tutorial, 2022. http://download.oracle.com/javase/tutorial/ 16. Michael L. Scott, Programming Language Pragmatics, Morgan Kaufmann, 4th Edition, 2019. 17. Aleksandar Prokopec, Learning Concurrent Programming in Scala, Packt Publishing, 2nd Edition, 2020. 18. Steve Klabnik and Carol Nichols, The Rust Programming Language, 2021.		

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 respects the IEEE and ACM Curriculla Recommendations for Computer Science studies;
- The content of the course is considered by the software companies as important for average software development skills

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	know the basic principle of the domain; - apply the course concepts - problem solving	Written final exam	20%
		Practical final exam	30%

10.5 Seminar/lab activities	be able to use course concepts in solving the real problems	Practical Assignments	50%
10.6 Minimum performance standards			
□ At least grade 5 (from a scale of 1 to 10) at written final exam and practical final exam. At least grade 5 for the final grade.			

Date Signature of course coordinator Signature of seminar coordinator

05.05.2022 Assoc. Prof. Eng. Florin Craciun Assoc. Prof. Eng. Florin Craciun

FCraciun FCraciun

Date of approval Signature of the head of department

Prof. dr. Laura Dioşan

24.05.2022 Diosan