

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	<ul style="list-style-type: none"> · 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 methods · C3.4 Comparatively and experimentally 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 Developing specifications and designing information systems using specific methods and tools · C4.5 Developing, implementing and integrating software solutions
Transversal competencies	<ul style="list-style-type: none"> · 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		

<ol style="list-style-type: none"> 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’Reilly, 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		

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

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

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

- | |
|--|
| <ul style="list-style-type: none"> · The course respects the IEEE and ACM Curricula 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	25%
		Practical final exam	35%

10.5 Seminar/lab activities	be able to use course concepts in solving the real problems	Laboratories Assignments	35%
		Seminar Activity	5%
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.
- Rules:
- **You can change your subgroup for the lab only once at the first lab. You have to announce the lab teacher about this. After the first lab you cannot change your subgroup lab time.**
- **you have to present each lab assignment at its deadline**
- **- for each lab assignment you will get a grade between 1 to 10**
- **- the deadline for each lab assignment is clearly specified in the assignment text file**
- **- if you delay an assignment 1 week you can get maximum 7 on that assignment**
- **- if you delay an assignment more than 1 week you will automatically get the grade 0 for it and you cannot submit it anymore**
- **- the final grade for the lab activity is the arithmetic average of the lab assignments grades**
- **you have to implement all the assignments since the problems of the final practical exam are extensions of the lab assignments**
- - the lab assignments mainly consist of a big project to implement an interpreter (virtual machine) of an imperative concurrent toy language
- - at each lab assignment (almost each week) you will add the rules and the data structures required to execute one or more new instructions of the toy language
- - the toy language interpreter will be implemented in Java
-
- - a schedule of the lab assignments (periodically updated) can be found at LabAssignmentsSchedule.pdf
- **The first condition to get into the final exam is to attend minimum 90% of the labs and minimum 70% of the seminars. That means you must attend minimum 5 seminars and minimum 12 laboratories. Please read the following document:**
- <http://www.cs.ubbcluj.ro/wp-content/uploads/Hotarare-CDI-15.03.2017.pdf>
- **- Holydays and first week are considered by default attended**
- **The second condition to get into the final exam is to get minimum grade 5 at the lab activity.**
- **Rules for the Students from previous years (“Restantieri”): the students must attend the labs and the seminars, must do the lab assignments, and must pass the final exam**
- **- in order to pass the final exam you must have:**
- **-- at least 5 at the final theoretical exam and**
- **-- at least 5 at the final practical exam and**
- **-- the final grade must be at least 5**
- - you can pass either both the final theoretical exam and the final practical exam or nothing
- **Rules for the second exam (“restanta”): The first condition to get into the final exam is to attend minimum 90% of the labs and minimum 70% of the seminars. That means you must attend minimum 5 seminars and minimum 12 laboratories. Please read the following document:**
- <http://www.cs.ubbcluj.ro/wp-content/uploads/Hotarare-CDI-15.03.2017.pdf>
- Holydays and first week are considered by default attended**
- The second condition to get into the final exam is to get minimum grade 5 at the lab activity.**
- in order to pass the final second exam you must have:**
- at least 5 at the final theoretical exam and**
- at least 5 at the final practical exam and**
- the final grade is 5**
- you can pass either both the second final theoretical exam and the second final practical exam or

nothing

Date

.....

Signature of course coordinator

Assoc. Prof. Eng. Florin Craciun

FCraciun

Signature of seminar coordinator

Assoc. Prof. Eng. Florin Craciun

FCraciun

Date of approval

.....

Signature of the head of department