SYLLABUS

Programming Paradigms

University year 2025-2026

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	**Undergraduate**
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	**Programming Paradigms**					Discipline code	**MLE5172**
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. Discipline regime	**Compulsory**

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/project	**3**
3.4. Total hours in the curriculum	70	of which: 3.5 course	28	3.6 seminar/laboratory/project	**42**

Time allotment for individual study (ID) and self-study activities (SA)	**hours**
Learning using manual, course support, bibliography, course notes (SA)	25
Additional documentation (in libraries, on electronic platforms, field documentation)	10
Preparation for seminars/labs, homework, papers, portfolios and essays	22
Tutorship	5
Evaluations	15
Other activities:	3

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, Algorithmics, Data structures
4.2. competencies	Basic notions and programming skills

5. Conditions (if necessary)

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

6.1. Specific competencies acquired [1]

[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.

Professional/essential competencies	• advanced programming skills in high-level programming languages • development and maintenance of software systems
Transversal competencies	• application of organized and efficient work rules, of responsible attitudes towards the didactic-scientific field, to bring creative value to own potential, with respect for professional ethics principles and norms • efficient development of organized activities in an interdisciplinary group and the development of empathetic abilities for interpersonal communications, to relate to and cooperate with various groups

6.2. Learning outcomes

Knowledge	The student knows: The graduate has the necessary knowledge for using computers, developing software programs and applications, information processing. • The graduate has the ability to develop, design and create new applications, systems or products using best practices of the field. • The graduate has the necessary skills for computer program design and software systems analysis. • 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. • The graduate has knowledge related to programming, mathematics, engineering and technology and has the skills to use them to create complex information technology systems.
Skills	• The graduate is able to combine diverse information to formulate solutions and generate ideas for developing new products and applications. • The graduate is able to apply architectural styles, design patterns and best practices in the field to design software applications of high complexity.
Responsibility and autonomy:	The student has the ability to work independently to obtain" • The graduate has the ability to evaluate different architectures and possible solutions to a problem and choose the right one for the specific requirements and constraints of the application to be developed. • • The graduate has the ability to choose and use programming paradigms (procedural, object-oriented, functional) to develop software applications appropriate for the specific domain of the application being developed.

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

7.1 General objective of the discipline	• Each student has to prove that (s)he acquired an acceptable level ofknowledge 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 general object oriented paradigms

7.2 Specific objective of the discipline	• The students should have the ability to use Java language, design patterns, and to create GUI for their applications. Also they have to be able to use object-oriented concepts in program analysis and design.

8. Content

8.1 Course	Teaching methods	Remarks
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	
Collections and Generic Types: anonymous classes, polymorphism, casting	Exposure,description, explanation, debate and dialogue, discussion of case studies	
IO,NIO: binary and character oriented streams, files, channels and buffers	Exposure,description, explanation, debate and dialogue, discussion of case studies	
Functional programming: lambda expressions, streams	Exposure,description, explanation, debate and dialogue, discussion of case studies	
GUI: Java FX components, event handling	Exposure,description, explanation, debate and dialogue, discussion of case studies	
Concurrency: threads, executors, futures, exception handling		
Concurrency: sync vs async methods, callback methods, cancellation		
GUI (cont.):FXML, CSS. Metaprogramming: reflection, serialization		
Introduction in C# and .Net		
Collections in C#		
IO operations in C#		
GUI in C#		
Introduction in Scala		
Introduction in Rust		

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, 2013. http://download.oracle.com/javase/tutorial/
6. Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O O'Reilley, 2022
7. ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/

8.2 Seminar / laboratory	Teaching methods	Remarks
Java basic project	Conversation, debate, case studies, examples	
Java project: Collections, Generics		
Java project: IO		
Project evaluation		
Java project: Functional programming		
Java project: memory management		
Project evaluation		
Java project: GUI		
Project evaluation		
Java project: concurrency		
Java project: concurrency		
Project evaluation		
Java project: type system		
Project evaluation		
Project evaluation		

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/

***, The Java Tutorial, 2013. http://download.oracle.com/javase/tutorial/

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

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	- know the basic principle of the domain;	Written final exam	25%
	apply the course concepts in problem solving	Practical final exam	35%
10.5 Seminar/laboratory	- concepts understanding	Seminar tests	5%
	-concept implementation	Lab project	35%
10.6 Minimum standard of performance			

- 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 10 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 2 October 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 10 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 2 October 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

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

Not applicable.

Date:
...

Signature of course coordinator

Assoc.Prof.Eng Phd. Florin Craciun

Signature of seminar coordinator

Assoc.Prof.Eng Phd. Florin Craciun

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.*".