SYLLABUS

Principles of Performance Oriented Coding

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	Computer Science
1.5. Study cycle	Bachelor
1.6. Study programme/Qualification	Computer Science
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline	**Principles of Performance Oriented Coding**					Discipline code	**MLE5109**
2.2. Course coordinator				Boris Vleju, PhD			
2.3. Seminar coordinator				Boris Vleju, PhD			
2.4. Year of study	3	2.5. Semester	5	2.6. Type of evaluation	C	2.7. Discipline regime	Optional

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	**1 lab + 2 pr**
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)	5
Additional documentation (in libraries, on electronic platforms, field documentation)	6
Preparation for seminars/labs, homework, papers, portfolios and essays	10
Tutorship	4
Evaluations	5
Other activities:	0
3.7. Total individual study hours	**30**
3.8. Total hours per semester	**100**
3.9. Number of ECTS credits	**4**

4. Prerequisites (if necessary)

4.1. curriculum	Object Oriented Programming Advanced Programming Methods
4.2. competencies	Average Java programming skills

5. Conditions (if necessary)

5.1. for the course	Projector
5.2. for the seminar /lab activities	Laboratory with internet access and ability to use personal laptops

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	• 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 graduate is able to identify complex problems and examine related issues to develop solving options and implement solutions.
Skills	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 graduate is familiar with tools used for testing, debugging, validating software applications. The graduate is familiar with project management tools, version control systems, and continuous integration/continuous delivery (CI/CD) concepts, methods, tools.

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

7.1 General objective of the discipline	• To acquire a deeper insight of Java Core Technologies
7.2 Specific objective of the discipline	• To attain an enhanced level of object oriented design principles (in Java) • To get a good grasp of Concurrent Programming in Java • To be a solid base for preparing to become a Java certified programmer

8. Content

8.1 Course	Teaching methods	Remarks
1. Requirements and overview	Exposition: presentation, explanations, practical examples, demonstrations, case-study	
2. Application and process performance		
3. High performance collections (1)		

4. High performance collections (2)	discussions	
5. Lambdas – good and bad		
6. String processing and regular expressions in practice		
7. Java I/O performance		
8. Profiling in practice		
9. Concurrency in practice (1)		
10. Concurrency in practice (2)		
11. Concurrency in practice (3)		
12. Performance traps in design patterns (1)		
13. Performance traps in design patterns (2)		
14. Exam		

Bibliography
1.Jeanne Boyarsky, Scott Selikoff, OCA: Oracle Certified Associate Java SE 8 Programmer I Study Guide, John Wiley & Sons, Dec 11, 2014
2. Jeanne Boyarsky, Scott Selikoff, OCP: Oracle Certified Professional Java SE 8 Programmer II Study Guide, John Wiley & Sons, Dec 14, 2015
3. Joshua Bloch, Effective Java (3nd Edition), Addison-Wesley Professional, 2017
4. Joshua Bloch, Neal Gafter, Java puzzlers: traps, pitfalls, and corner cases, Addison-Wesley, 2005
5. Tim Peierls, Brian Goetz, Joshua Bloch, Joseph Bowbeer, Doug Lea, David Holmes, Java Concurrency in Practice, Pearson Education, May 9, 2006

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Requirements and overview. Application and process performance	Explanation, examples, dialog, case-studies	The lab is structured as 2 hours classes every second week.
2. High performance collections – profiling in practice		
3. Lambdas, Java I/O performance		
4. Concurrency in practice (1)		
5. Concurrency in practice (2)		
6. Performance traps in design patterns		
7. Exam		

Bibliography
1. Jeanne Boyarsky, Scott Selikoff, OCA: Oracle Certified Associate Java SE 8 Programmer I Study Guide, John Wiley & Sons, Dec 11, 2014
2.Jeanne Boyarsky, Scott Selikoff, OCP: Oracle Certified Professional Java SE 8 Programmer II Study Guide, John Wiley & Sons, Dec 14, 2015
3 .Joshua Bloch, Effective Java (3nd Edition), Addison-Wesley Professional, 2017
4. Joshua Bloch, Neal Gafter, Java puzzlers: traps, pitfalls, and corner cases, Addison-Wesley, 2005
5. Tim Peierls, Brian Goetz, Joshua Bloch, Joseph Bowbeer, Doug Lea, David Holmes, Java Concurrency in Practice, Pearson Education, May 9, 2006

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 Curricula Recommendations for Computer Science studies
- The course is very well appreciated by the software industry the content being set up in very close collaborations with various software companies

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	- understanding the concepts and language features presented in the course	Written exam or quizzes during the course	40%
10.5 Seminar/laboratory	Implementing course concepts and algorithms	Lab assignments	60%
10.6 Minimum standard of performance			
• At least grade 5 (1 to 10 scale) at all activities seminar/lab, written exam. • The final grade must be at least 5.			

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

Not applicable.

Date:
...

Signature of course coordinator

Boris Vleju, Phd

Signature of seminar coordinator

Boris Vleju, Phd

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*.".