S

YLLABUS

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	**Artificial Intelligence**

2. Information regarding the discipline

2.1 Name of the discipline (en) / (ro)			**Advanced methods for solving Computer Science problems**				
2.2 Course coordinator			**Lect. PhD. Mircea Ioan-Gabriel**				
2.3 Seminar coordinator			**Lect. PhD. Mircea Ioan-Gabriel**				
2.4. Year of study	**1**	2.5 Semester	**2**	2.6. Type of evaluation	**C**	2.7 Type of discipline	**F**
2.8 Code of the discipline		MLE5199	**FACULTATIVE**				

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	1/2
3.4 Total hours in the curriculum	7	Of which: 3.5 course	28	3.6 seminar/laboratory	14 / 28

Time allotment:	hours
Learning using manual, course support, bibliography, course notes	20
Additional documentation (in libraries, on electronic platforms, field documentation)	0
Preparation for seminars/labs, homework, papers, portfolios and essays	30
Tutorship	28
Evaluations	2
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	•
4.2. competencies	•

5. Conditions (if necessary)

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

6. Specific competencies acquired

Professional competencies	C1.1 Recognizing and describing specific concepts to calculability, complexity, programming paradigms and modeling of computing and communication systems C1.2 Using specific theories and tools (algorithms, schemes, models, protocols, etc.) for explaining the structure and the functioning of hardware, software and communication systems C1.3 Building models for various components of computing systems C1.4 Formal evaluation of the functional and non-functional characteristics of computing systems C1.5 Providing theoretical background for the characteristics of the designed systems 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

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 discipline	• The theoretical and practical training of student teams for computer science competitions
7.2 Specific objective of the discipline	• Understanding and properly applying • specific data structures • fundamental algorithms and solving methods • team work • in-competition time management • in the context of student computer science problem solving competitions

8. Content

8.1 Course	Teaching methods	Remarks
1. (weeks 1-2) : Searching and sorting - binary search - quick sort - merge sort - heap sort - counting sort - KMP	Interactive exposure Explanation Conversation Didactical demonstration	
2. (weeks 2-4) : Relevant data structures - Binary Indexed Tree - Interval Tree - Tries	Interactive exposure Explanation Conversation Didactical	

- Finite Automata - AVL Trees - Disjoint Sets	demonstration	
3. (weeks 5-6) : Graph Algorithms - BFS, DFS - Dijkstra, Floyd-Warshall - Prim, Kruskal - DAGs, Topological sorting - Bridges in graphs, Strongly-connected components	Interactive exposure Explanation Conversation Didactical demonstration	
4. (weeks 7-8) : Dynamic Programming - longest common subsequence - edit distance - Needleman-Wunsch	Interactive exposure Explanation Conversation Didactical demonstration	
5. (weeks 9-10) : Mathematical methods applied in computer science - Number theory - Combinatorics - Approximation methods	Interactive exposure Explanation Conversation Didactical demonstration	
6. (weeks 11-13) : Geometry and Networks - convex hull - Ford Fulkerson - bipartite graphs - LCA and RMQ - Hopcroft-Karp	Interactive exposure Explanation Conversation Didactical demonstration	
7. Review and Evaluation		

Bibliography

1. DONALD E. KNUTH, The Art of Computer Programming, Addison-Wesley, 1998
2. DONALD E. KNUTH, The Stanford GraphBase : A Platform for Combinatorial Computing, ACM Press / Addison-Wesley, 1993.
3. STEVEN SKIENA and MIGUEL REVILLA, Programming Challenges : The Programming Contest Training Manual, Springer-Verlag, 2003.
4. DAVID HAREL, Algorithmics : The Spirit of Computing, 3rd edition, Addison-Wesley, 2004.

8.2 Seminar / laboratory	Teaching methods	Remarks
The labs will follow the same content taught at the lecture by applying the newly taught notions for solving actual programming contest problems. The students may propose interesting problems as well especially if they bring up new areas that have not been tackled so far. If the online rounds of the major programming competitions synchronize with the lab the teams will be competing in those online stages	Lab assignment Explanation Conversation	
Lab 1-2: Searching and sorting Lab 3-4: Advanced data structures Lab 5-6: Graph Algorithms Lab 7-8:Dynamic Programming Lab 9-10: Mathematical methods applied in computer science	Lab assignment Explanation Conversation	

Lab 11-13: Geometry and Networks Lab 14: Review and Evaluation		
Bibliography http://www.infoarena.ro/ http://codeforces.com/ https://www.hackerrank.com/ https://www.hackerearth.com/challenges/		

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 complies to the IEEE and ACM Curiculla Recommendations for Computer Science Studies The coure compiles to the requirements of ACM-ICPC competitions

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Proper understanding of the theoretical and practical aspects of the subject	Written paper	25%
	Properly solving the proposed problems		
10.5 Seminar / lab activities	Properly solving programming problems during labs and in competitions		75%
10.6 Minimum performance standards			
• Minimum 5 grade for the lab activity			

Date	Signature of course coordinator	Signature of seminar coordinator
16.04.2023	Lect. Dr. Mircea Ioan-Gabriel	Lect. Dr. Mircea Ioan-Gabriel

Date of approval	Signature of the head of department
.....	Prof. Dr. Diosan Laura