COURSE DESCRIPTION

Data Structures and Algorithms

Academic year 2026 - 2027

1. Programme-related data

1.1. Higher Education Institution	Babeş – Bolyai University
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Department of Computer Science
1.4. Field	Computer Science
1.5. Level of study	Bachelor
1.6. Degree programme / Qualification	Artificial Intelligence
1.7. Form of education	Full time

2. Course-related data

2.1. Course title	**Data Structures and Algorithms**			Course code	**MLE5022**
2.2. Course coordinator	Lect. PhD. Hotea Diana – Lucia				
2.3. Seminar coordinator	Lect. PhD. Hotea Diana – Lucia				
2.4. Year of study	1	2.5. Semester	2	2.6. Type of assessment	Exam
2.7. Course status	Compulsory			2.8. Course type	Core subject

3. Total estimated time (hours per semester of teaching activities)

3.1. Number of hours per week	**4**	of which: 3.2. course	**2**	3.3. seminar/ laboratory/ project	**1S + 1P**
3.4. Total of hours in the curriculum	56	of which: 3.5. course	28	3.6. seminar/ laboratory	**28**
Time allocation for individual study (IS) and self-taught activities (ST)					**hours**
Learning from textbooks, course materials, bibliography, and notes (IS)					17
Additional research in the library, on subject-specific electronic platforms, and on-site					6
Preparing seminars/ laboratories/ projects, assignments, reports, portfolios, and essays					30
Tutoring (professional guidance)					6
Examinations					10
Other activities					
3.7. Total hours of individual study (IS) and self-taught activities (ST)				**69**	
3.8. Total hours per semester				**125**	
3.9. Number of credits				**5**	

4. Prerequisites (where applicable)

4.1. curriculum-related	Fundamentals of programming
4.2 skills-related	Medium programming skills

5. Specific conditions (where applicable)

5.1. course-related	Class room with projector
5.2. seminar/laboratory-related	For the seminar, a class-room with a whiteboard/blackboard

6.1. Competencies resulting from the completion of the degree programme *(as referred to in the curriculum)*[1]

[1] The professional and/or transversal skills targeted by the subject for which the course description is prepared will be copied from the curriculum of the degree programme. For each competency, the complete entry, including the competency code, will be copied with the exact wording that appears in the curriculum, without any changes.

Professional competencies	
Competency code	**Competency**
CP7	Design information system
CP17	Create data models
CP1	Create software
Transversal competencies	
Competency code	**Competency**
CT3	Think analytically
CT2	Solve problems

6.2. Learning outcomes relevant to the degree programme (as referred to in the curriculum)[2]

Learning outcomes targeted by the subject		
Competency code	**Knowledge and comprehension**	**Specific academic skills**
CP1	1. The student/graduate identifies, explains and justifies fundamental concepts of data structures, algorithms, and programming paradigms, as well as computer architecture.	1. The student/graduate designs, develops and demonstrates complex software solutions using efficient algorithms and diverse programming paradigms.

7. Subject-specific learning outcomes

Knowledge and comprehension
1. The student/graduate identifies, explains and justifies fundamental concepts of data structures, abstract data types and algorithms.
Specific academic skills
1. The student/graduate designs, develops and demonstrates complex software solutions using efficient algorithms and suitable data structures.

8. Contents

8.1. Course	**Teaching and learning methods**	**Remarks[3]**
1. Introduction. Data structures. Abstract Data Types. • Abstract Data Types and Data Structures • Pseudocode conventions • Complexities	• Exposure • Description • Conversation • Didactical demonstration	
2. **Arrays. Iterators** • Dynamic array • Amortized complexity analysis • Interface of an iterator		
3. Abstract Data Types		

If no competency is copied from either of the two categories, the row corresponding to that category is deleted from the table.

[2] The learning outcomes relevant for the degree programme and targeted by the subject for which the course description is prepared will be listed. The entries, copied without any changes from the Curriculum by subject type (Core Subject/Specialisation Subject/Complementary Subject), are listed under the corresponding competency.

[3] For example, organisational aspects, recommendations for students, specific aspects relating to the course/seminar, such as inviting experts in the field, etc.

• ADT Set: description, domain, interface and possible representations • ADT Map: description, domain, interface and possible representations • ADT Matrix: description, domain, interface and possible representations • ADT MultiMap: description, domain, interface and possible representations		
4. Abstract Data Types II • ADT Stack: description, domain, interface and possible representations • ADT Queue: description, domain, interface and possible representations • ADT PriorityQueue: description, domain, interface and possible representations • ADT Deque: description, domain, interface and possible representations • ADT List : description, domain, interface and possible representations		
5. Linked Lists • Singly linked list: representation and operations • Doubly linked list: representation and operations • Iterator for linked lists		
6. **Linked Lists II** • Sorted linked lists: representation and operations • Circular linked lists: representation and operations • Linked lists on arrays: representation and operations		
7. **Binary Heap** • Representations, specific operations. • HeapSort		
8. **Hash Table** • Direct address tables • Hash tables: description, properties • Collision resolution through separate chaining		
9. **Hash Table II** • Collision resolution through coalesced chaining • Collision resolution through open addressing		
10. **Hash Table III** • Perfect hashing • Linked hash tables		

• Containers represented over hash tables		
11. **Trees. Binary Trees** • Concepts related to trees • Applications of trees • Description and properties of binary trees • Domain and interface of ADT Binary Tree • Possible representations of ADT Binary Tree • Binary tree traversals: recursive/non-recursive algorithms		
12. Binary Search Trees • Description, properties • Representation • Operations: recursive and non-recursive algorithms • Containers represented over binary search trees		
13. Balanced Binary Search Trees • AVL Trees		
14. Applications and data structure libraries in different programming languages		
Bibliography 1. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT Press, 2009 2. S. Skiena: The algorithms design manual, Second edition, Springer, 2008 3. N. Karumanchi: Data structures and algorithms made easy, CareerMonk Publications, 2016 4. M. A. Weiss: Data structures and algorithm analysis in Java, Third Edition, Pearson, 2012 5. R. Sedgewick: Algorithms, Addison-Wesley Publishing, 1984		
8.2. Laboratory	**Teaching and learning methods**	**Remarks**
Lab1. Example of a solved lab assignment (Demo)	• Exposure • Examples • Conversation	Laboratory is structured as 2 hour classes every second week. Laboratory problems assigned at a lab have to be presented in the next lab (exception is Lab1). Every assignment focuses on a given data structure. Students will receive a container (ADT) that has to be implemented using the given data structure.
Lab2. Discussion about the Demo. Example of an extra operation. A1- Dynamic array		
Lab 3. A2 - Linked lists with dynamic allocation		
Lab 4. A3 - Linked lists on array		
Lab 5. A4 - Hash table		
Lab 6. A5 - Binary search tree		
Lab 7. Presentation of problem from Lab 6		
Bibliography 1. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT Press, 2009 2. S. Skiena: The algorithms design manual, Second edition, Springer, 2008 3. N. Karumanchi: Data structures and algorithms made easy, CareerMonk Publications, 2016 4. M. A. Weiss: Data structures and algorithm analysis in Java, Third Edition, Pearson, 2012 5. R. Sedgewick: Algorithms, Addison-Wesley Publishing, 1984		
8.2. Seminar	**Teaching and learning methods**	**Remarks**
1. ADT Bag with generic elements. Representations and implementation on an array. Iterator for ADT Bag	• Exposure • Conversation • Examples • Debate	Seminar is structured as 2 hour classes every second week.
2. Complexities		

3. Bucket sort, Lexicographic sort, radix sort. Merging two sorted singly linked lists.		
4. Sorted MultiMap – representation and implementation on a singly linked list		
5. Evaluating an arithmetic expression. Problems solved with binary heap.		
6. Hash tables		
7. Binary trees		

Bibliography

1. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT Press, 2009
2. S. Skiena: The algorithms design manual, Second edition, Springer, 2008
3. N. Karumanchi: Data structures and algorithms made easy, CareerMonk Publications, 2016
4. M. A. Weiss: Data structures and algorithm analysis in Java, Third Edition, Pearson, 2012
5. R. Sedgewick: Algorithms, Addison-Wesley Publishing, 1984

9. Evaluation

Type of activity	9.1 Evaluation criteria[4]	9.2 Evaluation methods[5]	9.3 Percentage in the final grade
9.4. Course	• Correctness and completeness of the assimilated knowledge • Knowledge of applying the concepts	Written evaluation (in the exam session): written exam	60%
9.5. Laboratory	• C++ implementation of the concepts and algorithms presented at the lectures • Lab assignment documentation • Respecting the deadlines for lab presentation	Correctness of the implementation and documentation (representation, specifications, algorithms, complexities).	40%
9.6. Seminar	• Seminar activity	Active participation at the discussions during the seminar (asking and answering questions, volunteering to solve problems, etc.)	Maximum 0.5 points bonus, added to the final grade

9.7 Minimum standard for passing

- Knowledge of the basic concepts. Each student has to prove that he/she has acquired an acceptable level of knowledge and understanding of the domain, that he/she is capable of expressing the acquired knowledge in a coherent form, that he/she has the ability of using this knowledge for problem solving.
- For participating at the written exam, a student must have at least 6 lab attendances and 5 seminar attendances.
- For successfully passing the examination, a student must have at least 5 for the written exam, and minimum 5 as a final grade.

[4] The evaluation criteria must directly reflect the learning outcomes targeted at the level of the degree programme respectively at the level of the subject. More specifically, the learning outcomes set out in the expected learning outcomes are assessed.

[5] Both final evaluation methods and ongoing evaluation strategies should be established.

10. SDG labels (Sustainable Development Goals)[6]

	○	Sustainable Development Generic Label

								
○	○	○	○	○	○	○	○	○
								No label applies
○	○	○	○	○	○	○	○	X

Date of entry:
22.05.2026

Signature of course coordinator

Lect. PhD. HOTEA Diana – Lucia

Signature of seminar coordinator

Lect. PhD. HOTEA Diana – Lucia

Date of approval in the department:
...

Signature of the head of department

Assoc. Prof. PhD. STERCA Adrian

[6] Select a single label which, according to the *Implementation of SDG labels in the academic process*, best matches the subject. If the subject addresses sustainable development in a generic manner (i.e. by presenting/introducing the general framework of sustainable development, etc.), then the Sustainable Development generic label may be applied. If none of the labels describe the subject, select the last option: *"No label applies."*