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	Computer Science
1.5 Study cycle	Bachelor
1.6 Study programme / Qualification	Computer Science

2. Information regarding the discipline

2.1 Name of the discipline		Data Structures and Algorithms			
2.2 Course coordinator		Lecturer PhD. Dana Lupsa			
2.3 Seminar coordinator		Lecturer PhD. Dana Lupsa			
2.4. Year of study	**1**	2.5 Semester **2**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Compulsory**

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	3	Of which: 3.2 course	2	3.3 seminar/laboratory	1 sem
3.4 Total hours in the curriculum	42	Of which: 3.5 course	28	3.6 seminar/laboratory	14

Time allotment:	hours
Learning using manual, course support, bibliography, course notes	14
Additional documentation (in libraries, on electronic platforms, field documentation)	10
Preparation for seminars/labs, homework, papers, portfolios and essays	14
Tutorship	6
Evaluations	14
Other activities:	

3.7 Total individual study hours	58
3.8 Total hours per semester	100
3.9 Number of ECTS credits	4

4. Prerequisites (if necessary)

4.1. curriculum	●
4.2. competencies	●

5. Conditions (if necessary)

5.1. for the course	● Class room with projector
5.2. for the seminar /lab activities	●

6. Specific competencies acquired

Professional competencies	C 4.1 Definirea conceptelor şi principiilor de bază ale informaticii, precum şi a teoriilor şi modelelor matematice C 4.3 Identificarea modelelor si metodelor adecvate pentru rezolvarea unor probleme reale C 4.5 Încorporarea de modele formale în aplicaţii specifice din diverse domenii
Transversal competencies	CT1 Aplicarea regulilor de muncă organizată şi eficientă, a unor atitudini responsabile faţă de domeniul didactic-ştiinţific, pentru valorificarea creativă a propriului potenţial, cu respectarea principiilor şi a normelor de etică profesională CT3 Utilizarea unor metode şi tehnici eficiente de învăţare, informare, cercetare şi dezvoltare a capacităţilor de valorificare a cunoştinţelor, de adaptare la cerinţele unei societăţi dinamice şi de comunicare în limba română şi într-o limbă de circulaţie internaţională

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

7.1 General objective of the discipline	- Knowledge, understanding and use of data structure concepts and their algorithms; - Improved programming skills
7.2 Specific objective of the discipline	- Understand data structure design, algorithms and their complexities - Acquire knowledge necessary for working with data structure libraries - Ability to choose appropriate data structure in order to model and solve real world problems

8. Content

8.1 Course	Teaching methods	Remarks

1. Introduction. Data Structures.
    Abstract Data Types: domain, operations
    Data type.
        Data representation.
        Operation design and implementation.
        Complexity.
    Examples. Design issues
	Exposure: description, examples	

2.
3. Lists
    Linear List (Sequence)
     - Essential properties
     - ADT
    Arrays. Vectors
    Lists. Linked Lists
        Types of linked list: singly linked, doubly linked, circular
    Representations & operations design
    Examples. Implementation issues
	- Interactive exposure - Explanation - Conversation - Didactical demonstration	

4. Collection containers. Iterators
    Set, Bag
    Stack ; Queue ; Deque
	Exposure: description, examples, case	

Topic	Method	

Map, MMap
Sorted Containers. Priority Queue
   - Essential properties
   - ADT design issues. Examples.
	studies	

5. How to choose a data structure? Where to look for advantages / disadvantages (time/space analysis) Examples over linear data structure	Examples, case studies	
6. Hash: Hash table, hash function. Collisions.	Exposure: description, examples, case studies	

7. Hash: Hash table & collision resolution
   - Separate chaining
   - Open adressing
   - Coalesced hashing
Performance analysis
Containers represented over Hash.
	Exposure: description, explanation, examples, discussion of case studies	

8. Trees - Concepts related to trees - ADT - representation - tree traversals; recursive / non recursive algorithms	Exposure: description, explanation, examples, discussion of case studies	
9. Binary tree - representation - oprations: search, add, remove an element - tree traversals; recursive / non recursive algorithms	Exposure: description, explanation, examples, discussion of case studies	

10. Heap. Binary heap
   - Representation
   - Priority Queues represented over heaps
HeapSort
	Exposure: description, examples, discussion of case studies	

11. Binary Search Trees Balanced trees. Terminology. Examples	Exposure: explanation, examples, discussion of case studies	
12. 13. Balanced trees. Red-black trees, AVL trees Balanced trees as representation of sorted containers	Exposure: description, explanation, examples	
14. Applications	Examples, discussion of case studies	

Bibliography
1. CORMEN, THOMAS H. - LEISERSON, CHARLES - RIVEST, RONALD R.: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000.
2. FRENTIU M., POP H.F., SERBAN G., Programming Fundamentals, Ed.Presa Universitara Clujeana, Cluj-Napoca, 2006

3. HOROWITZ, E.: Fundamentals of Data Structures in C++. Computer Science Press, 1995.
4. MOUNT, DAVID M.: Data Structures. University of Maryland, 1993.
5. NICULESCU V., CZIBULA G., Structuri fundamentale de date. O perspectiva orientata obiect. Editura Casa Cartii de Stiinta, Cluj-Napoca, 2011
6. SIMONAS SALTENIS, Algorithms and Data Structures, 2002.
7. STANDISH, T.A.: Data Structures, Algorithms & Software Principles in C, Addison-Wesley, 1995
8. Java™ Platform, Standard Edition 7, API Specification, http://docs.oracle.com/javase/7/docs/api/
9. Java™ Tutorial. Trail: Collections. http://docs.oracle.com/javase/tutorial/collections/index.html
10. STLProgrammer's Guide http://www.sgi.com/tech/stl/index.html
11. STL Containers - C++ Reference, http://www.cplusplus.com/reference/stl/
12. http://www.cs.ubbcluj.ro/~dana/2013-2014/DSA/Notes/
13. http://www.cs.ubbcluj.ro/~dana/2013-2014/DSA/Project/

8.2 Seminar	Teaching methods	Remarks
1. DS as a problem of data representation. Algorithms and complexities	Dialogue, debate, case studies, examples	The seminar is structured as 2 hours classes every second week
2. Representation and operation design for different kind of lists	Dialogue, debate, case studies, examples	

3,4. Container collections
    ADT, representation under some given
    restrictions, operations design
	Dialogue, debate, case studies, examples	

5. Iterator design. Iterators vs. element access container operation.	Dialogue, debate, case studies, examples	

6,7. Trees
    - Operation design
      (recursive/ non recursive)
    - Problems with trees
  Binary trees
    - Operation design
      (recursive/ non recursive)
    - Iterators over binary trees
	Dialogue, debate, case studies, examples	

Bibliography

1. CORMEN, THOMAS H. - LEISERSON, CHARLES - RIVEST, RONALD R.: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000.
2. FRENTIU M., POP H.F., SERBAN G., Programming Fundamentals, Ed.Presa Universitara Clujeana, Cluj-Napoca, 2006
3. HOROWITZ, E.: Fundamentals of Data Structures in C++. Computer Science Press, 1995.
4. STANDISH, T.A.: Data Structures, Algorithms & Software Principles in C, Addison-Wesley, 1995
5. SIMONAS SALTENIS, Algorithms and Data Structures, 2002.
6. http://www.cs.ubbcluj.ro/~dana/2013-2014/DSA/Seminary/

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 ACM Curriculla Recommendations for Computer Science studies
- The course exists in the studying program of all major universities in Romania and abroad;
- *"Many of the top software companies like Google and Facebook hire experienced computer scientists who have extensive knowledge of algorithms and data structures. These areas are also a topic in software development interviews at both startups and large companies"*

	Sam Snyder (He works for Motorola Mobility (a division of Google)) http://samsnyder.com/2011/05/18/algorithms-and-data-structures/
•	

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	Written exam	60%
10.5 Seminar/lab activities	Realization of a project - apply the course concepts - problem solving	Project evaluation	20%
	- Homework assignments, including intermediate delivery of (parts of) the project - Homework discussion - Seminar activity	Oral evaluation	20%
10.6 Minimum performance standards			
➢ At least grade 5 (from a scale of 1 to 10) at written exam, homework evaluation and final grade			

Date   Signature of course coordinator   Signature of seminar coordinator

....   lecturer PhD Dana Lupsa   lecturer PhD Dana Lupsa

Date of approval   Signature of the head of department

....…...........................