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	20
Additional documentation (in libraries, on electronic platforms, field documentation)	10
Preparation for seminars/labs, homework, papers, portfolios and essays	26
Tutorship	7
Evaluations	20
Other activities:	

3.7 Total individual study hours	83
3.8 Total hours per semester	125
3.9 Number of ECTS credits	5

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	- Knowledge, understanding and use of data structure concepts and their algorithms; data structures and their algorithms are basic concepts of theoretical Computer Science.
Transversal competencies	- Ability to choose appropriate data structure in order to model and solve real world problems - Improved programming abilities

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

7.1 General objective of the discipline	- Be able to understand basic data structures - 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

8. Content

8.1 Course	Teaching methods	Remarks
1. Introduction. Data Structures. Complexity.	Exposure: description, examples	
2. Data Types: domain, operations and data representation	Exposure: description, explanation, examples	
3. Collection containers. Iterators Set, Bag, Map, Stack ; Queue ; Priority Queue - Essential properties: general& specific issues	Exposure: description, examples, discussion of case studies	
4. Arrays. Vectors	Exposure: description, examples, case studies	
5. Linked Lists Types of linked list: singly linked, doubly linked, circular Representation & operations design	Exposure: description, examples, case studies	
6. Hash: Hash table, hash function - Properties & operations	Exposure: description, explanation, examples, discussion of case studies	
7. Trees	Exposure:	

- Concepts related to trees - ADT - reprezentation - tree traversals; recursive / non recursive algorithms	description, explanation, examples, discussion of case studies	
8. Heaps ; HeapSort	Exposure: description, explanation, examples, discussion of case studies	
9. Binary Search Trees Balanced trees. Terminology. Examples	Exposure: description, examples, discussion of case studies	
10. Binary Search Trees Balanced trees. Terminology. Examples	Exposure: description, explanation, examples	
11. Balanced trees. Red-black trees	Exposure: description, explanation, examples, discussion of case studies	
12. Balanced trees. AVL trees	Exposure: description, explanation, examples	
13. Issues in choosing an ADT to solve a problem; issues in choosing DS for an ADT - Advantages / disadvantages : time/space analysis	Exposure: examples, case studies	
14. Examples and 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. STANDISH, T.A.: Data Structures, Algorithms & Software Principles in C, Addison-Wesley, 1995
7. SIMONAS SALTENIS, Algorithms and Data Structures, 2002.
8. Generic and JCF Java, http://download.oracle.com/javase/1.5.0/docs/guide/
9. STLProgrammer's Guide http://www.sgi.com/tech/stl/index.html.
10. STL Containers - C++ Reference, http://www.cplusplus.com/reference/stl/

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. DS and operation design for different kind of	Dialogue, debate,	

lists	case studies, examples	
3. Concrete problems to be solved by using studied collections containers	Dialogue, debate, case studies, examples	
4. Define a DS for an ADT under some given restrictions	Dialogue, debate, case studies, examples	
5. Trees. Operation design Recursive/ non recursive	Dialogue, debate, case studies, examples	
6. Binary trees - Iterators over binary trees	Dialogue, debate, case studies, examples	
7. Choose DS for concrete problems. Examples	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.		

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	Project evaluation	30%

	concepts - problem solving		
	- Homework assignments, including intermediate delivery of (parts of) the project	Homework verification	10%
10.6 Minimum performance standards			
➢ At least grade 5 (from a scale of 1 to 10) at both written exam 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

....…...........................