SYLLABUS

1. Information on the study programme

1.1. Higher education institution	Universitatea de Vest din Timisoara
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Study program field	Computer Science - in English
1.5. Study cycle	Licence
1.6. Study programme / Qualification	Computer Science/ Database Administrator- *252101; Network Administrator - 252301; Analist - 251201; Research assistant in Computer Science- 214918; Research assistant in mathematics-computer science - 212024; Highschool teacher- 233002; Programmer - 251202; System designer - 251101*

2. Information on the course

2.1. Course title			Algorithms and Data Structures I				
2.2. Lecture instructor			Gabriel Istrate				
2.3. Seminar / laboratory instructor			Gabriel Istrate, Anca Vulpe				
2.4. Study year	1	2.5. Semester	1	2.6. Examination type	E	2.7. Course type	DI

3. Estimated study time (number of hours per semester)

3.1. Attendance hours per week	4	out of which: 3.2 lecture	2	3.3. seminar / laboratory	2
3.4. Attendance hours per semester	56	out of which: 3.5 lecture	28	3.6. seminar / laboratory	28
Distribution of the allocated amount of time*					**hours**
Study of literature, course handbook and personal notes					35
Supplementary documentation at library or using electronic repositories					15
Preparing for laboratories, homework, reports etc.					40
Exams					6
Tutoring					8

Other activities…		
3.7. Total number of hours of individual study	104	
3.8. Total number of hours per semester	160	
3.9. Number of credits (ECTS)	6	

4. Prerequisites (if it is the case)

4.1. curriculum	N/A
4.2. competences	Elementary knowledge of mathematics. Elementary problem-solving abilities.

5. Requirements (if it is the case)

5.1. for the lecture	Amphitheater with whiteboard and projector
5.2. for the seminar / laboratory	Lab with appropriate tools - Python installed on lab computers.

6. Specific acquired competences

Professional competences	□ capacity to design, analyze and implement simple algorithms and data structures.
Transversal competences	□ capacity to communicate algorithms and data structures

7. Course objectives

7.1. General objective	To become familiar with the design, analysis and implementation of basic algorithms and datastructures
7.2. Specific objectives	Capacity to communicate knowledge pertaining to domain-specific algorithms.

8. Content

8.1. Lecture	**Teaching methods**	**Remarks, details**

Course 1. Introduction to algorihtmics. Notion of algorithms. Properties of algorithms: correctness, Basic examples.		
Course 2. Analysis of Algorithms, Insertion Sort, Mergesort		
Course 3. Asymptotic Notation; Recurrences; Substitution, Master Theorem.		
Course 4. Divide-and-Conquer: analysis of mergesort, binary search, number powering, computing Fibonacci, Strassen's method for matrix multiplication.		
Course 5. Quicksort, Worst-case and average case analysis.		
Course 6. Linear-time Sorting: Lower Bounds, Counting Sort, Radix Sort		
Course 7. Order Statistics, Median		
Course 8. Introduction to data structures. Lists. Stacks, queues.		
Course 9. Binary Search Trees. Relation of BSTs to Quicksort - Analysis of Random BST		
Course 10. Heaps. Heapsort.		
Course 11. Dynamic programming. The Longest Common Subsequence Problem.		

Course 12. Greedy Algorithm. Minimum Spanning Tree.		
Course 13: Backtracking. Applications.		
Course 14: Revision for the fall semester exam.		
Recommended literature **T.H. Cormen, C.E.Leiserson, R.R. Rivest, C. Stein – Introduction to Algorithms, MIT Press. 2009.** **J. Kleinberg, E. Tardos – Algorithm Design. Addison Wesley 2005.** **Various course notes: Daniela Zaharie (UVT), Jeff Edmonds (York U, Canada), etc.**		

8.2. Seminar / laboratory	**Teaching methods**	**Remarks, details**
problem solving related to coursework	problem solving/seminar	
Intro to Programming in Python	**laboratory**	
programming in python - various algorithms pertaining to the course	**laboratory**	
Recommended literature		

9. Correlations between the content of the course and the requirements of the professional field and relevant employers.

This is a basic course in computer science programs all around the world. Questions from this course appear both in the graduation exam, as well as in hiring interviews.

10. Evaluation

Activity	10.1. Assessment criteria	10.2. Assessment methods	10.3. Weight in the final mark
10.4. Lecture	final	written exam	75 %
10.5. Seminar / laboratory	exam	practical exam	25 %
	homework	homework	extra credit
10.6. Minimum needed performance for passing			
pass both sections: Course/Seminar-lab. - abilities to descign simple algorithms and describe them in pseudocode - basic abilities to analyse the complexity of simple algorithms - basic familiarity with Python programming (lab section)			

Date of completion Signature (lecture instructor) Signature (seminar instructor)

1.10.2016

Date of approval Signature (director of the department)