ERASMUS+
EU programme for education, training, youth and sport

Incoming student mobility

Name of UNIOS University Unit: DEPARTMENT OF MATHEMATICS

COURSES OFFERED IN FOREIGN LANGUAGE
FOR ERASMUS+ INDIVIDUAL INCOMING STUDENTS

Department or Chair within the UNIOS Unit	**Department of Mathematics**

Study program	• ***Undergraduate university study programme in Mathematics and Computer Science*** • ***Undergraduate university study programme in Mathematics***

Study level	Undergraduate (Bachelor)

Course title	**Data Structures and Algorithms II**
Course code	I054
Language of instruction	English
Brief course description	**Syllabus.** 1. **Advanced Data Structures.** Red-Black trees. Augmenting Data Structures. Fenwick trees. B-Trees. Binomial Heaps. Fibonacci Heaps. Data structures for Disjoint Sets. 2. **Graph Theory Fundamentals.** Basic terms and definitions. Paths, walks and cycles. Connectivity. Trees and forests. Bipartite graphs. Euler tours. Hamiltonian cycles. Matchings. Planar graphs. Colouring. Flows and circulations. 3. **Elementary Graph Algorithms.** Representations of graphs. Breadth-first search. Depth-first search. Topological sort. Connected components. Minimum Spanning Trees. The algorithms of Prim and Kruskal. Single-Source Shortest Path problem. The Bellman-Ford algorithm. Single-source shortest paths in acyclic graphs. Dijkstra's algorithm. 4. **Advanced Graph Algorithms.** All-Pairs Shortest path problem. The Floyd-Warshall algorithm. Johnson's algorithm for sparse graphs. Maximum Flow problem. Flow networks. The Ford-Fulkerson method. Edmonds Karp's algorithm. Minimum Cut problem. Maximum bipartite matching. 5. **Selected Topics.** Linear Programming. Parallel algorithms. Polynomial and the Fast Fourier Transform. Number-Theoretic algorithms. String Matching algorithms. Computational

	Geometry algorithms.
Form of teaching	Consultative teaching.
Form of assessment	Lectures contain a deep and systematic overview of advanced data structures and algorithms. During exercises student are expected to solve given programming problems by using acquired knowledge. The correctness, time and space complexity of implemented algorithms are the most important elements. At the end of each practice session students individually solve short quizzes. During the semester, students solve homework assignments that contain programming problems. The assessment of theoretical knowledge is done by written examinations. If students achieve satisfactory results in homework and written exams, they are not obliged to take final written and oral exams. A seminar or final project can contribute to the final grade.
Number of ECTS	**7**
Class hours per week	**3+2+0**
Minimum number of students	
Period of realization	**Summer semester**
Lecturer	Domagoj Ševerdija