COURSE OUTLINE

GENERAL

SCHOOL	Sciences and Engineering		
ACADEMIC UNIT	Computer Science		
LEVEL OF STUDIES	1st Cycle		
COURSE CODE	COMP-270	**SEMESTER**	Spring
COURSE TITLE	Data Structures and Algorithms		

INDEPENDENT TEACHING ACTIVITIES *if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits*	**WEEKLY TEACHING HOURS**	**CREDITS**
	2.5	6
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	Special background
PREREQUISITE COURSES:	COMP-113
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	English
IS THE COURSE OFFERED TO ERASMUS STUDENTS	
COURSE WEBSITE (URL)	

LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.

Consult Appendix A

- *Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B*
- *Guidelines for writing Learning Outcomes*

After completion of the course students are expected to be able to:

- Explain the Big-Oh and Big-Omega notation, calculate the running times of programs and categorize algorithms based on complexity classes.
- Perform algorithm analysis for different data structures including lists, stacks, queues, trees, sets and graphs; and compare the various approaches towards element insertion/deletion/updating/traversing and space requirements.
- Perform mathematical proofing for basic algorithmic problem descriptions.
- Understand the mechanics behind major sorting algorithms and implement them.
- Design and implement appropriate hashing functions.
- Design and implement collision-resolution algorithms for a hash table.

- Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing.
- Solve problems using the fundamental graph algorithms, including graph traversal, single-source and all-pairs shortest paths, transitive closure, topological sorting, and spanning tree algorithm.
- Demonstrate the following capabilities: evaluate algorithms, select the appropriate algorithm for solving a particular problem and justify the choice, and implement the algorithm in a programming context.

General Competences

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

Search for, analysis and synthesis of data and information, with the use of the necessary technology	*Project planning and management*
Adapting to new situations	*Respect for difference and multiculturalism*
Decision-making	*Respect for the natural environment*
Working independently	*Showing social, professional and ethical responsibility and sensitivity to gender issues*
Team work	*Criticism and self-criticism*
Working in an international environment	*Production of free, creative and inductive thinking*
Working in an interdisciplinary environment	*……*
Production of new research ideas	*Others…*
	…….

Search for, analysis and synthesis of data and information, with the use of the necessary technology
Adapting to new situations
Decision-making
Working independently
Production of free, creative and inductive thinking

SYLLABUS

1. Introduction to Data Structures and Algorithms
    a. Algorithms and their Properties
    b. Data Types and Data Structures
    c. Algorithm Pre-Execution Criteria and Evaluation Measures
2. Mathematical Proof Techniques
    a. Proof by Induction
    b. Proof by Counterexample
    c. Proof by Contradiction
3. Algorithm Analysis
    a. Empirical Analysis
    b. Algorithm Growth Notation
    c. Scientific Framework for Running Time Calculation
    d. Complexity Classes
4. Algorithm Analysis for Linear Data Structures
    a. Linear search vs Binary Search
    b. Recursion Running Time Calculation
    c. Master Theorem Arrays and Linked Lists

5. Sorting Algorithms
    a. Internal Sorting (incl. BubbleSort, SelectionSort, InsertionSort)
    b. Mergesort, QuickSort and BucketSort
    c. Shuffling
    d. External Sorting
6. Algorithm Analysis for Tree Data Structures
    a. Tree Representation and Implementations
    b. Binary Search Trees
    c. AVL Trees
    d. B-Trees
7. Algorithm Analysis for Heap Data Structures
    a. Priority Queues
    b. Binary Heaps
    c. Skewed Heaps
    d. Huffman Encoding
8. Algorithm Analysis for Sets and Hashing Data Structures
    a. Bit Vectors
    b. Hash functions
    c. Collision Management with Chaining and Probing
    d. Funneling and Rehashing
9. Algorithm Analysis for Graph Data Structures
    a. Directed and Undirected Graphs
    b. Graph Traversal and Cycle Management
    c. Shortest Path Problems
    d. Spanning Trees
    e. Topological Sorting
    f. Flow Problems and Euler Paths
10. Algorithm Design Strategies
    a. Divide and Conquer
    b. Greedy Algorithms
    c. Dynamic Programming

TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	Face-to-face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	*Use of ICT in teaching / Χρήση ΤΠΕ* *Communication with students / Επικοινωνία με Φοιτητές*
TEACHING METHODS *The manner and methods of teaching are described in detail.* *Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art*	

Activity	***Semester workload***
Lectures	35
Preparation	26

workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc. *The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS*	Coursework: 40 Exam Preparation: 45 Examination: 4 Course total: ***150***
STUDENT PERFORMANCE EVALUATION *Description of the evaluation procedure* *Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other* *Specifically-defined evaluation criteria are given, and if and where they are accessible to students.*	Final Exam, Homework Assignments, Midterm, Participation

ATTACHED BIBLIOGRAPHY

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Data Structures and Algorithm Analysis* (3rd edition)	Mark Weiss	Pearson	2014	978-1-292-01415-9

* Note: the textbook is available with coded snippets for the presented algorithms and data structures in both Java and C++ with only the coded examples changing. For Python, a git repository with the coded snippets is also publicly maintained.

Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Algorithms (4th ed.)	Robert Sedgewick and Kevin Wayne	Addison-Wesley	2011	978-0-321-57351-3
Data Structures and Algorithm Analysis* (3rd ed.)*	Clifford A. Shaffer	Open-Source	2013	-

* Made freely available online by the author https://people.cs.vt.edu/~shaffer/Book/