COURSE OUTLINE

1. GENERAL

SCHOOL	BUSINESS ADMINISTRATION		
ACADEMIC UNIT	DEPARTMENT OF CULTURAL HERITAGE MANAGEMENT AND NEW TECHNOLOGIES		
LEVEL OF STUDIES	UNDERGRADUATE		
COURSE CODE	**Y204**	**SEMESTER**	**2nd**
COURSE TITLE	INTRODUCTION TO 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**
Lectures (theoretical part - presentation, study of algorithms for practical problems)	3	5
Lectures (laboratory part – analysis of foundational algorithms and algorithmic techniques)	2	
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*	Background course
PREREQUISITE COURSES:	DISCRETE MATHEMATICS (1st SEMESTER)
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek
IS THE COURSE OFFERED TO ERASMUS STUDENTS	No
COURSE WEBSITE (URL)	goo.gl/geY1Ga

2. 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*

An algorithm can be considered as an exact "recipe" that determines the sequence of steps required to solve a problem.

The aim of this course is to let students have an introductory experience with fundamental algorithms and basic - yet powerful - problem solving methods and techniques.

In the background of all, rather routine, activities of modern culture - such as web browsing, data storage and transmission, management of database systems, cryptography and secure internet transactions, data compression, artificial intelligence - simple, fascinating "ideas", i.e., "algorithms",

exist!

The objectives of this course include (1) the description and analysis of fundamental ideas (i.e., algorithms) that modern computers use many times every day while we hardly realize it, and (2) an introduction to the formal analysis of algorithms (correctness and performance issues).

Students who regularly participate in course activities and successfully complete the course:

- have knowledge and understanding for (1) fundamental algorithmic issues related to web search, data storage and transmission, management of database systems, cryptography and secure Internet transactions, data compression, artificial intelligence as well as (2) analysis of algorithms in terms of correctness and performance; students are therefore able to keep track of current developments at the cutting edge of their field of knowledge
- are able to use knowledge and understanding they have acquired in a way that shows a professional approach to their work or profession, and appropriately skilled to develop and support algorithmic approaches for various problems within their field
- have the ability to collect and interpret relevant data (typically within their field) to form judgments that include reflection on relevant social, scientific or ethical issues
- are able to communicate information, ideas, problems and solutions to specialized and non-specialized audience
- have developed knowledge acquisition skills necessary to further continue their studies with a high degree of autonomy
- have become familiar with computational thinking and are able to exploit its advantages in scientific, professional and practical issues

In particular, students who regularly participate in course activities and successfully complete the course:

1. have knowledge of fundamental principles and techniques in logic, proofs, set theory, basic and advanced counting methods
2. understand algorithmic problems
3. are able to apply algorithmic techniques for computing solutions to corresponding problems
4. analyze problems / questions in order to gain understanding of their structure and components
5. suggest solutions to these problems by applying existing or new algorithms and methods
6. evaluate findings (solutions or hardness results) through analysis in terms of correctness and performance
7. are familiar with computational thinking

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…*
	…….

Familiarity with computational thinking

Search for, analysis and synthesis of data and information, with the use of the necessary technology

Adapting to new situations

Decision-making

Working independently

Team work

Working in an international environment

Working in an interdisciplinary environment

Production of new research ideas

Project planning and management

Respect for difference and multiculturalism

Showing social, professional and ethical responsibility and sensitivity to gender issues

Criticism and self-criticism

Production of free, creative and inductive thinking

3. SYLLABUS

An algorithm can be considered as an exact "recipe" that determines the sequence of steps required to solve a problem.

The aim of this course is to let students have an introductory experience with fundamental algorithms and basic - yet powerful - problem solving methods and techniques.

In particular:

- Have you, perhaps, searched in billions of documents in the web, choosing a couple of them that are most relevant to your needs?
- Have you stored or transmitted extremely large amounts of information, without making a single error, in presence of electromagnetic interference that affects all electronic devices?
- Did you successfully complete an online bank transaction, even though many thousands of other customers were simultaneously using the same server?
- Did you communicate some confidential information (e.g., your credit card number) securely over wires that can be actually “seen” by many other computers?
- Did you use compression to reduce a picture of several megabytes to a more manageable size for sending it by e-mail?
- Or did you, without even thinking about it, exploit the artificial intelligence in a hand-held device that self-corrects your typing on its tiny keyboard?

In the background of all these - rather routine - activities there exist simple, fascinating "ideas", i.e., "algorithms"!

The objectives of this course include (1) the description and analysis of fundamental ideas (i.e., algorithms) that modern computers use many times every day while we hardly realize it, and (2) an introduction to the formal analysis of algorithms (correctness and performance issues).

Summing up, it is worth emphasizing the necessity and usefulness of inclusion of courses focusing on the design and analysis of algorithms in undergraduate and graduate curricula. Why?

Donald Knuth, one of the most prominent computer scientists in the history of algorithmics, put it as follows:

"A person well-trained in computer science knows how to deal with algorithms: how to construct them, manipulate them, understand them, and analyze them. This knowledge is preparation for much more than writing good computer programs; it is a general-purpose mental tool that will be a definite aid to the understanding of other subjects, whether they be chemistry, linguistics, or music, etc. The reason for this may be understood in the following way: It has often been said that a person does not really understand something until after teaching it to someone else. Actually, a person does not really understand something until after teaching it to a computer, i.e., expressing it as an algorithm. . An attempt to formalize things as algorithms leads to a much deeper understanding than if we simply try

to comprehend things in the traditional way."

Lectures are scheduled as follows:

Introduction: Course outline, objectives and role in the curriculum

Theoretical part:

- Introduction/Overview
- PageRank algorithm
- Diffie-Hellman public key-exchange algorithm
- Error correction algorithms
- Data compression algorithms
- Pattern recognition algorithms
- Database Systems management algorithms
- Digital Signatures
- Limits of computation

Laboratory part:

- Asymptotic analysis
- Addition: algorithms and complexity
- Multiplication: algorithms and complexity
- Graph traversal: algorithms and complexity
- Shortest paths: algorithms and complexity
- Classification of algorithms
- Binary Search
- Mergesort

4. TEACHING and LEARNING METHODS - EVALUATION

<table>
<tr><td>DELIVERY
Face-to-face, Distance learning, etc.</td><td>Face to face, Distance learning</td></tr>
<tr><td>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
Use of ICT in teaching, laboratory education, communication with students</td><td>Use of ICT in teaching (online lectures, course website, extensive use of Web resources), in communication/collaboration with students (mailing lists, social networks (Facebook), course website, Doodles) and in the process of progress monitoring and evaluation (use of specialized software for the monitoring and evaluation of student progress)</td></tr>
<tr><td>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 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</td><td>
<table>
<tr><th>Activity</th><th>Semester Workload</th></tr>
<tr><td>Lectures (theoretical part)</td><td>39</td></tr>
<tr><td>Lectures (laboratory part)</td><td>26</td></tr>
<tr><td>Intense cooperation among professor and students also using ICT</td><td>8</td></tr>
<tr><td>Independent study</td><td>52</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Course total
(25 hours per credit)</td><td>125</td></tr>
</table>
</td></tr>
<tr><td>STUDENT PERFORMANCE EVALUATION
Description of the evaluation procedure
Language of evaluation, methods of evaluation, summative or conclusive, multiple</td><td>Assessment - Grading Process (it is announced on the course website before the beginning of the semester and remains available throughout the semester)</td></tr>
</table>

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.*	The final score is obtained as a function of: (A) 2 intermediate computer-based multiple choice examinations. They contribute by 40% to the final score. - All students attending the course can participate in the intermediate examinations. - Scores are valid only for the current academic year. - Participation in the intermediate exams is not mandatory: students who decide not to participate in intermediate examinations are not excluded from the final examination in February. However, the 2 intermediate examinations contribute to the final score (by 40%). (B) a final, computer-based, multiple choice examination. It contributes by 60% to the final score.

5. ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

9 ALGORITHMS THAT CHANGED THE FUTURE, John MacCormick

INTRODUCTION TO ALGORITHMS, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.

FOUNDATIONS OF COMPUTER SCIENCE, Behrouz A. Forouzan, Firouz Mosharraf

ALGORITHM DESIGN, Jon Kleinberg, Eva Tardos

- Related academic journals:

Theoretical Computer Science, Elsevier

Theory of Computing Systems, Springer