COURSE OUTLINE

(1) GENERAL

SCHOOL	Engineering		
DEPARTMENT	Electrical and Computer Engineering		
LEVEL OF STUDY	Undergraduate		
COURSE UNIT CODE	5.004	**SEMESTER**	5th
COURSE TITLE	Analysis and Design of Algorithms		
COURSEWORK BREAKDOWN		**TEACHING WEEKLY HOURS**	**ECTS Credits**
Theory (Lectures)		2	2
Practice		2	2
Lab		1	2
TOTAL		**5**	**6**
COURSE UNIT TYPE	Specialized general knowledge/Skills development		
PREREQUISITES	None		
LANGUAGE OF INSTRUCTION/EXAMS	Greek		
COURSE DELIVERED TO ERASMUS STUDENTS	Yes		
WEB PAGE (URL)	https://eclass.hmu.gr/courses/		

(2) LEARNING OUTCOMES

Learning Outcomes

The knowledge which students will acquire upon successful completion of the course relates to the design of solutions for computational problems, the implementation of these solutions using a specific programming language, the deep understanding of standard algorithmic techniques that are available for the solution of computational problems, and last but not least, the ability to assess a proposed computational solution utilizing the standard theoretic framework that has been developed for this means and to be able to design, select and implement an efficient solution for a computational problem. The programming language that will be used for the practical sessions is the C programming language, thus students will be able to enhance, and advance the programming skill they acquired in previous courses.

To summarize upon successful completion of the course, students will be able to:

- Understand the theoretical framework that governs the analysis of algorithms and to be able to assess the time and space complexity of a computational solution.
- To have a deep understanding of specific and well established computational techniques, such as divide and conquer, greedy solutions, dynamic programming.
- To be able to design and propose a new algorithmic solution for a problem and to assess its efficiency.
- To be able to implement various algorithmic solutions for a problem, compare them computationally, and select the most efficient one.

General Skills

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adapt solutions to new situations
- Autonomous work
- Teamwork
- Decision making
- Work in an interdisciplinary environment

(3) SYLLABUS

Course description

The course has a central role in computer science in both theoretical and practical levels. It strives for students to understand fundamental ways of problem solving and data organization in computer memory and to learn and implement computational techniques for the solution of the problems and for the efficient handling of the data. Special attention is placed on creating new data algorithms with an obvious result on the dexterities of students to cope with computational problems.

More specifically, the course covers the following topics:

- Introduction: Algorithms description and analysis. Algorithm design methodology. Asymptotic notation. Growth of functions. Complexity analysis. Brief mathematical review.
- Search: Linear search. Binary search. Interpolation search. Fibonacci search.
- Sorting: Selection sort. Insertion sort. Bubble sort. Interpolation sort. Recursive sorting algorithms. Merge sort and merge. Quicksort and in-place quicksort. Linear time sorting (radix sort, bucket sort, etc.) Heap sort. Minimum, maximum and median finding.
- Graphs: Introduction and definitions. Data structures for graphs. Graph traversal (Breadth First Search, Depth First Search). Topological sorting. Strongly connected components. Minimum spanning trees (Prim and Kruskal algorithms). Weighed graphs. Single source shortest paths (Bellman-Ford and Dijskstra algorithms). All-pairs shortest paths. Euler tour.
- Algorithmic techniques: Divide and conquer. Greedy algorithms (i.e., fractional knapsack). Dynamic programming (0-1 knapsack, matrix chain multiplications). Randomized algorithms (selection of topics).

In addition to the above topics, selected from the following list will be covered:

- String Matching
- Network Flow and Matching
- Number Theory and Cryptography
- Computational Geometry

The practical training includes programming exercises based on the material covered in the lectures using the C programming language. The instructor evaluates the design, development and results of the programming exercises for all students during the course of the semester. Additionally, there is a short to moderate scale project, which is performed in groups and on which the students are examined individually at the end of the semester.

(4) TEACHING METHODS - ASSESSMENT

MODE OF DELIVERY	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATION TECHNOLOGY	▪ Use of ICTs in lecturing and lab ▪ Use of ICTs for the communication with students via the e-class platform
TEACHING ORGANIZATION	

Method description/Activity	Semester Workload
Lectures	40
Demos/Labs	40
Individual Study & Programming (on Server or Laptop/PC using VM or Dual Boot)	40
Total Contact Hours	**120**

	Language for Evaluation: Greek/English (Erasmus)

ASSESSMENT METHODS	All announcements related to the syllabus, including grading, and complementary reading material are permanently posted in the course web page (ECLASS). The course grade incorporates the following evaluation procedures: 1. Final exam (60%) 2. Midterm Exam (20%) 3. Programming exercises (20%)

(5) RECOMMENDED BIBLIOGRAPHY

- *"Algorithm Design and Applications", Michael T. Goodrich, Roberto Tamassia, John Wiley & Sons, Inc., 2015.*
- *"Introduction to Algorithms", Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Third Edition, MIT Press, 2009.*
- *"Algorithms", S. Dasgupta, C. Papadimitriou, U. Vazirani, MC Graw, Hill Higher Education Edition, 2008.*
- *"Algorithms in C", R. Sedgewick, Addison-Wesley, 1990.*