INSTITUTION	NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS					
SCHOOL	SCHOOL OF SCIENCE					
DEPARTMENT	INFORMATICS AND TELECOMMUNICATIONS					
COURSE LEVEL	UNDERGRADUATE					
COURSE TITLE	**Computational Geometry**					
COURSE CODE	**C06**		**Semester**	**8**	**ECTS**	**6**
TEACHING HOURS per week	**THEORY**	3	**SEMINAR.**	**1**	**LABORATORY**	
URL	https://eclass.uoa.gr/courses/D42/					

COURSE CONTENT

Introduction to Geometric Algorithms and Extensions to the Research Areas of Computational Geometry.
In particular:
• Convexity in two dimensions and in three dimensions, but also in a general dimension.
• Volume of curved polyhedra and Minkowski Sum of Polyhedra.
• Triangulation and Delaunay Triangulation / Voronoi Diagram.
• Geometric Search Algorithms, Geometric Data Structures, Nearest Neighbor, clustering.
Also, topics on geometric algorithms and the use of libraries (eg CGAL) for the development of geometric software in programming languages like C / C ++, and Python are presented.

STUDENT LEARNING OBJECTIVES

Teaching-Learning Goals-Expected Learning Outcomes:
Introduction to Geometric Algorithms in an understandable way and using basic mathematical knowledge and data structures so that students can see modern applications and extensions to active research areas of Computational Geometry. It is based on both on theory and applications.

Upon successful completion of the course the student will be able to:
• Explain the notion of Convexity in two dimensions and in three dimensions
• Describe the basic concept of Delaunay Triangulation and Triangulation / Voronoi Diagram
• Handle Minkowski sum and Polyhedra
• Address Basic Geometric Search algorithms
• Design, develop and evaluate Python applications

TEACHING AND LEARNING METHODS - ASSESSMENT

TEACHING METHOD	In Class (Face to Face)
USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES	Learning process supported by the e-class platform (specifically Course Description, Class Material, Discussions, Announcements, Task assignments, Student groups, Questionnaires, External links). Email communication
TEACHING ORGANIZATION *Describe in detail the way and methods of teaching:* *Enhanced Lectures,* *Online Lectures,* *Seminars,* *Tutorial,* *Laboratory,* *Laboratory Exercise,* *Study & analysis of literature,* *Practice (Positioning),* *Interactive teaching,* *Developing a project,* *Individual / group work* *Telework (reference to tools) etc.* *Details of the student's study hours for each learning activity and hours of non-guided study are shown to ensure that the total workload at the semester corresponds to the ECTS*	see workload table below Theory is presented with slide projection. The programming environments are presented in demonstration workshops in real-time parallel machines. Assignment is given in groups of 1-2 students. Support with discussions in eclass.
ASSESSMENT OF STUDENTS *Description of the assessment process* *Assessment Methods, Formative or Concluding, Multiple Choice Test, Quick Response Questions, Test Development Questions, Problem Solving, Written Work, Report / Report, Oral Examination, Public Presentation, Laboratory Work, Other / Other* *Fully defined evaluation criteria are mentioned and if and where they are accessible to students.*	Students are assessed with intermediate work and a final assignment. Intermediate works cover the theoretical and programmatic part of the course while in the final assignment the students choose the topic that they are most interested in. Each assignment is evaluated with classified criteria communicated to the students. In order to pass the course, 50% of the theoretical (1) and programming assignments (2) and 50% of the final assignment (3) must be obtained. For "perfect grade" one needs to get perfect grade in (1) or in (2), and in (3).

Activity	***Student Workload*** *(hours)*
Lectures	39
Studying and Slides	30
Studying Python	20
Small individual exercises	30
Independent Studying for final assignment	31
Total Course (25 hours of workload per unit of credit)	***150***

Assessment methods	***Number***	***Percentage***
assignments	5	50%
Final project	1	50%

LITERATURE AND STUDY MATERIALS / READING LIST

Ioannis Emiris. Computational geometry: a modern algorithmic approach. Kleidarithmos Publishing, 2008.
Mark Overmars et al. Computational geometry and Applications. Springer.
Notes and Slides for Python.