<table>
<tr><td><b>INSTITUTION</b></td><td colspan="6">NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS</td></tr>
<tr><td><b>SCHOOL</b></td><td colspan="6">SCHOOL OF SCIENCE</td></tr>
<tr><td><b>DEPARTMENT</b></td><td colspan="6">INFORMATICS AND TELECOMMUNICATIONS</td></tr>
<tr><td><b>COURSE LEVEL</b></td><td colspan="6">UNDERGRADUATE</td></tr>
<tr><td><b>COURSE TITLE</b></td><td colspan="6"><b>Software Development for Algorithmic Problems</b></td></tr>
<tr><td><b>COURSE CODE</b></td><td colspan="2"><b>K23γ</b></td><td><b>Semester</b></td><td><b>7</b></td><td><b>ECTS</b></td><td><b>8</b></td></tr>
<tr><td><b>TEACHING HOURS per week</b></td><td><b>THEORY</b></td><td><b>1</b></td><td><b>SEMINAR.</b></td><td></td><td><b>LABORATORY</b></td><td><b>3</b></td></tr>
<tr><td><b>COURSE TYPE</b></td><td colspan="6"><b>Select one of the following and delete the rest</b>
Project
<table><tr><th>K</th><th>E1</th><th>E2</th><th>E3</th><th>E4</th><th>E5</th><th>E6</th></tr><tr><td>A</td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table><i>Fill the table as in the curriculum:</i> Track (A-Computer Science, B- Computer Engineering) / Specialization Compulsory (Y) / Core Specialization (B)/ Elective Specialization (E)</td></tr>
<tr><td><b>URL</b></td><td colspan="6">https://eclass.uoa.gr/courses/DI352/</td></tr>
<tr><td><b>EXPECTED PRIOR KNOWLEDGE/ PREREQUISITES AND PREPARATION:</b></td><td colspan="6">K17- Algorithms and Complexity (Αλγόριθμοι και Πολυπλοκότητα)
Recommended: K25 – Theory of Computation (Θεωρία Υπολογισμού)</td></tr>
<tr><td><b>TEACHING AND EXAMINATIONS LANGUAGE:</b></td><td colspan="6">GREEK</td></tr>
<tr><td><b>THE COURSE IS OFFERED TO ERASMUS STUDENTS</b></td><td colspan="6">NO</td></tr>
</table>

COURSE CONTENT
The course focuses on the development of software for solving an important problem in Computer Science, on the use of the software in a real application and on its experimental evaluation. The followed approach aims at the familiarization of the students with large scale programming, often with the use of open source software libraries through team work as well as at the development of skills for the implementation and practical utilization of established, efficient algorithms. C/C++ is used as a programming language in the course. Topics covered include the implementation of efficient algorithms, their experimental evaluation, software organization and design principles, the use of open source libraries and software tools such as: Qt, Boost, Gnuplot, LAPACK, Eigen, Unit testing frameworks, collaboration and version control systems such as git and SVN as well as collaborative large scale implementation. Three assignments for the design and development of

software for solving hard algorithmic problems are given with emphasis on the fields of data science and computational geometry.

STUDENT LEARNING OBJECTIVES
Expected Learning Outcomes Upon successful completion of the course the student will be able to: • Describe and explain hard algorithmic problems. • Select appropriate programming techniques and algorithms for solving difficult problems in Computer Science. • Install and use open source libraries and software tools such as Qt, Boost, Gnuplot, LAPACK, Eigen, Unit testing frameworks, collaboration and version control systems (git and SVN). • Implement algorithms according to design principles that permit their easy reuse. • Execute an experimental evaluation of the algorithmic techniques and of the software, in general, as well as produce reports presenting accurately and clearly the experimental results.

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 (Course description, Material Provision, Announcements, Calendar, Student Teams, Assignments, Discussion forum, External links) Email communication Live transmission of lectures Ability to track recorded lectures
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*	Theory is presented in lectures through the use of powerpoint/pdf presentations. Open source libraries and software tools are presented in laboratory demonstrations. Tutorials are organized for answering questions regarding the algorithms, for the resolution of technical issues that arise during their implementation as well as for the execution of the evaluation experiments. 3 assignments for the design and development of software for solving hard algorithmic problems are assigned to teams of 1-2 persons. The third assignment focuses on developing a real application utilizing real or simulated data. Support is provided for the implementation of the programming assignments through the discussion forum of the eclass platform.

Activity	***Student Workload*** *(hours)*
Lectures	13

Tutorial	26
Laboratory	13
1st assignment	40
2nd assignment	40
3rd assignment	40
Independent study in order to understand the algorithms and to be familiarized with the libraries and software tools.	28
Total Course (25 hours of workload per unit of credit)	***200***

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 evaluated through an oral examination of each consecutive assignment. The assignments are assessed on the basis of graded criteria and have equal weight in the final evaluation. Students are given the opportunity to request an analysis of the evaluation of their assignments. The evaluation is corrected in case of an error.

Assessment methods	***Number***	***Percentage***
Assignments	3	100%

LITERATURE AND STUDY MATERIALS / READING LIST

Textbook for general C/C++ programming (Εύδοξος):

- Bruce Eckel, *Thinking in C++, vol. 2*, εκδόσεις Μ. Γκιούρδας, 2009

Scientific publication that change from year to year according to the examined algorithmic problems.
PDF/powerpoint presentations for the description of the examined problems, the algorithmic techniques, software organization and design principles as well as for all the libraries and software tools which are used in the course.