COURSE OUTLINE

(1) GENERAL

SCHOOL	ENGINEERING		
ACADEMIC UNIT	Department of Computer Engineering and Informatics		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE		SEMESTER	WINTER
COURSE TITLE	DECENTRALIZED COMPUTING AND MODELING		

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, Tutorials, Laboratory	2(L), 1(T), 2(Lab)	5
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*	Specialized general knowledge, skills development
PREREQUISITE COURSES:	Recommended prerequisite knowledge: “Object-Oriented Programming” (NY134), “Introduction to Algorithms” (NY205), “Probability and Statistics” (NY204), “Graph Theory and Applications” (NY202), “Distributed Systems” (NE4117) or equivalent.
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek (English if there are Erasmus students)
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes
COURSE WEBSITE (URL)	

(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*

Upon conclusion of the course the students ought to be able to:

- Understand techniques, properties, implementations and applications of fundamental and advanced decentralized algorithms.
- Model complex systems and apply techniques for their evaluation.
- Apply the scientific experimental methodology in empirically and comparatively assessing decentralized algorithms and models.
- Use algorithmic software platforms and libraries for developing new efficient implementations of decentralized algorithms and models.
- Develop implementations of complex models with practical usability and applicability.
- Understand the process of modelling a complex decentralized system aiming at predicting its behavior as well as its improvement through emulation.

Upon conclusion of the course the students are expected to have the following skills/competences:

- Considerably improve their skills in modeling complex systems.
- Use advanced modeling techniques and analysis tools for decentralized systems.
- Understand and apply properly the scientific experimental methodology in empirically and comparatively assessing different models for the same complex system.
- Develop efficient and effective implementations of decentralized algorithms.
- Apply effectively the multi-agent modeling methodology on complex natural/artificial systems.

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
- Team Work
- Criticism and self-criticism
- Production of free, creative and inductive thinking

(3) SYLLABUS

Models: Distributed and Decentralize: LOCAL-CONGEST, Population Protocols, Population Dynamics, Opinion Dynamics, Cellular Automata, Network Dynamics, Agent-based Modeling.

Distributed Algorithms:

- Maximal Independent Set
- Coloring (deterministic and randomized)
- Approximation Algorithm for Dominating Sets
- Self-Stabilization Algorithms
- Computation based on Physarum Polycephalum

Decentralized Algorithms/Systems:

- Boolean Networks, Hopfield Networks
- Cellular Automata
- Network Systems
- Opinion Dynamics (De Groot Model)
- Population Dynamics (Disease Models, Predator-Prey Models)

Agent-based Modeling – Lab:

- Simple Agent-based Models
- Agent Properties and Actions
- Environment of Agents
- Agent Interactions
- Model implementation in NETLOGO

(4) TEACHING and LEARNING METHODS - EVALUATION

<table>
<tr><td>DELIVERY
Face-to-face, Distance learning, etc.</td><td>Face-to-face and distance learning. Tutorials and laboratory sessions with exemplary solutions of exercises.</td></tr>
<tr><td>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
Use of ICT in teaching, laboratory education, communication with students</td><td>ICT methods are used in both teaching and communication with the students. Lecture slides and supplementary material are uploaded in the course’s web site.</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</td><td>2*13=26</td></tr>
<tr><td>Tutorials (exercises)</td><td>1*13=13</td></tr>
<tr><td>Laboratory practice</td><td>2*13=26</td></tr>
<tr><td>Individual study, preparation and problem solving</td><td>4*13=52</td></tr>
<tr><td>Weekend study</td><td>1*13=13</td></tr>
<tr><td>Study during the 3 “empty weeks” (2 weeks of vacation and 1 week of exam preparation)</td><td>6*3=18</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Course total (25-30 hours per ECTS unit)</td><td>148</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 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.</td><td>The language of instruction and examination is Greek. Special provisions (lecture notes and examinations in English) can be made for foreign students.

Evaluation (criteria can be found in the web site of the course):
• Theoretical/Programming exercises (30% - 40% of final mark).
• Final examination (60% - 70% of final mark).

Final examination: oral and written examination on a theoretical/programming project assigned individually to each student as well as in a team project. Examination on the code of the implementation as well as on the written report containing the details of the implementation and the results of the experimental evaluation.

Series of theoretical/programming exercises aiming at familiarizing students with:
• The use of the algorithmic software platforms and libraries in NETLOGO.
• The use of the algorithm techniques taught in the course.
• The proper experimental evaluation of implemented models.
• The proper interpretation of experimental results and the errors they may contain.</td></tr>
</table>

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- D. Peleg. Distributed computing: a locality-sensitive approach. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

- J. Aspnes. Notes on Theory of Distributed Systems. Lectures Notes, 2002. https://www.cs.yale.edu/homes/aspnes/classes/465/notes.pdf
- S.Schmid and P.S. Mandal. Distributed Network Algorithms. Lecture Notes, 2016. https://www.univie.ac.at/ct/stefan/GIAN-Lecture-Notes-NetAlg.pdf
- J.L. Schiff. Introduction to Cellular Automata, 2007.
- S. Wolfram. A New Kind of Science. https://www.wolframscience.com/nks/, 2002.
- U. Wilensky and W. Rand. An Introduction to Agent-Based Modeling: Modeling Natural, Social and Engineered Complex Systems with NetLogo, 2015.
- F. Bullo. Lectures on Network Systems. http://motion.me.ucsb.edu/book-lns. 2020.
- A.V. Proskurnikov and R Tempo. A Tutorial on Modeling and Analysis of Dynamic Social Networks Part I. 2018.
- A.V. Proskurnikov and R Tempo. A Tutorial on Modeling and Analysis of Dynamic Social Networks Part II. 2018.
- C. Altafini. Notes for a course on Opinion Dynamics on Social Networks. 2020.
- M.J. Keeling. Notes on Population Dynamics
- S. Dolev. *Self-Stabilization*. Cambridge, MA: MIT Press, 2000.
- Editor: A. Adamatzky. Advances in Physarum Machines. 2016.
- J. Brownlee, Clever Algorithms: Nature Inspired Programming Recipes. 2011.
- Algorithmic Game Theory. Editors: N. Nisan, T. Roughgarden, E. Tardos and V.V. Vazirani. 2007.
- Federico Rossi, Saptarshi Bandyopadhyay, Michael T. Wolf and Marco Pavone. Multi-Agent Algorithms for Collective Behavior A structural and application-focused atlas:, 2021.
- N.A. Lynch. Distributed Algorithms. 1996.
- Διδακτικές σημειώσεις και διαφάνειες που αναρτώνται στην ιστοσελίδα του μαθήματος.

-Συναφή επιστημονικά περιοδικά:

- Nature
- Science
- Journal of the ACM