ERASMUS

HELLENIC MEDITERRANEAN UNIVERSITY

COURSE OUTLINES

Department of Electrical and Computer Engineering

School of Engineering

Academic Year 2026-2027

TABLE OF CONTENTS

ADVANCED TOPICS IN ARTIFICIAL INTELLIGENCE

COURSE OUTLINE

Responsible: Emmanouil Marakakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Postgraduate		
COURSE CODE	8000.1.121.0	**SEMESTER**	2nd
COURSE TITLE	Advanced Topics in Artificial Intelligence		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	5	7.5
Total	**5**	**7.5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	
PREREQUISITE COURSES	All students are expected to have background from the following undergraduate courses: Algorithms, Data Structures, Discrete maths, Logic and Introduction to AI.
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/TP281/

2. LEARNING OUTCOMES

Learning outcomes

The students are expected to get the required knowledge in order to be able to develop projects and to carry out research in selected, state of the art topics of AI. That is, in *Machine Learning* and in particular in Statistical relational learning.

General Competences

The primary aim of this course is to teach students advanced techniques of modern AI. In addition, it equips students with the appropriate programming tools for developing AI applications. Moreover, the course fosters an appreciation for the engineering issues underlying the design and development of AI systems.

3. SYLLABUS

Overview of Machine Learning. Statistical Relational Learning. Probability Theory & Bayes' Rule. Probability & Random Variables. Reasoning under Uncertainty I. Reasoning under Uncertainty II. Probabilistic Graphical Models - Bayesian Networks. Markov Networks. Probabilistic Inference. Probabilistic Logic Programming: ProbLog, Cplint. Implementation of Markov Models & HMM.

<table>
<tr><th colspan="2">4. TEACHING and LEARNING METHODS - EVALUATION</th></tr>
<tr><td>DELIVERY
Face-to-face, Distance learning, etc.</td><td>Lectures using power-point slides.</td></tr>
<tr><td>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
Use of ICT in teaching, laboratory education, communication with students</td><td>Programming, Word and Power-point are used for developing assignments. Internet is used for assignments and lectures. For example, eClass is used for uploading lectures, assignments, bibliography etc.</td></tr>
<tr><td>TEACHING METHODS
The manner and methods of teaching are described in detail.</td><td>
<table>
<tr><th>Activity</th><th>Semester workload</th></tr>
<tr><td>Assignments and projects</td><td>7</td></tr>
<tr><td>class attendance</td><td>5</td></tr>
<tr><td>Course total</td><td>12</td></tr>
</table>
</td></tr>
<tr><td>STUDENT PERFORMANCE EVALUATION
Description of the evaluation procedure</td><td>• 4 assignments 40% (each assignment is graded 10%).
• 1 project 60%.</td></tr>
</table>

5. ATTACHED BIBLIOGRAPHY

1. D. Poole, A. Mackworth, *Artificial Intelligence – Foundations of Computational Agents*, Cambridge University Press, Third Edition, 2023.
2. S. Russell, P. Norving, *Artificial Intelligence – A Modern Approach*, 4th edition, Pearson, 2022.
3. M. P. Deisenroth, A. A. Faisal, C. S. Ong, *Mathematics for Machine Learning*, Cambridge University Press, 2020.
4. G. Bontempi, *Handbook of Statistical Foundations for machine Learning*, Comp. Science Department, Universite Libre de Bruxelles, Belgique, 2017.
5. L. De Raedt, K. Kersting, S. Natarajan, D. Poole, *Statistical Relational Artificial Intelligence*, Morgan & Claypool Publ, 2016.
6. D. Koller, N. Friedman, *Probabilistic Graphical Models – Principles and Techniques*, The MIT Press, 2009.
7. L. De Raedt, *Logical and Relational Learning*, Springer, 2008.
8. D. Bertsekas, J. Tsitsiklis, *Introduction to Probability*, 2nd Edition, Athena Scientific, 2008.
9. L. Getoor, B. Taskar, *Introduction to statistical relational learning*, The MIT Press, 2007.

POWER ELECTRONICS

COURSE OUTLINE

Responsible: Georgios Orfanoudakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.7.003.0	**SEMESTER**	1st
COURSE TITLE	Power Electronics		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	4	4
Total	**4**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Special background / Core
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE141/

2. LEARNING OUTCOMES

Learning outcomes

The course "Power Electronics I" aims to provide students with basic knowledge on the semiconductor power modules and power converters built based on them. More specifically, it refers to the structure, operation, special features and applications of different types of power converters. It also covers elements of Fourier analysis and electric power quality.

Upon successful completion of the course, students will be able to:

1. describe and explain the structure, the characteristics, the capabilities and operation of basic power semiconductor elements
2. identify and describe the basic topologies of power converters (rectifiers, AC regulators, choppers, inverters)
3. identify and describe the basic control techniques of the above converters
4. examine and analyze the operation of converters and explain the characteristics of voltage-current waveforms at their input and output
5. explain the principles of Fourier analysis and apply it to the calculation of harmonic components, distortion, etc. of voltage-current waveforms
6. select the appropriate power converter for a given application.

General Competences

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adaptation to new situations
- Autonomous work
- Teamwork
- Work in an interdisciplinary environment
- Production of new research ideas

3. SYLLABUS

Theoretical Lecture Sections

- Introduction to power electronics - relation to other scientific fields. Classification of electronic power converters and their applications.
- Structure and functional characteristics of semiconductor power components (power diode, thyristor, BJT, MOSFET, GTO, IGBT, …).
- Uncontrolled rectifier circuits (using power diodes): Single-phase and three-phase rectifier topologies. Effect of the network's internal inductance (transition).
- Controlled rectifiers (using thyristors): Topologies of single-phase and three-phase fully controlled inverters, voltage and current waveforms, calculation of active and reactive power.
- AC converters: AC regulators with anti-parallel thyristors, reference to cycloconverters.
- DC to DC converters: Basic topologies of DC to DC converters (step-down, step-up). Analysis of the Pulse Width Modulation (PWM) technique and its application to them.
- AC to DC converters: Topology of single-phase (half-full bridge) and three-phase switching inverter. Analysis of operation with PWM.
- Principles of Fourier analysis and calculation of harmonic components. Derivation of spectrum. Calculation of active / reactive power, RMS value, total harmonic distortion and application to AC converters.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Project-based, Individual study
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	
TEACHING METHODS The manner and methods of teaching are described in detail.	
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Project evaluation (60%) Oral exam on given Power electronics syllabus (40%)

Activity	Semester workload
Lectures	0
Laboratory Exercises	10
Study and report writing	50
Independent study	50
Study and literature review	10
Course total	**120**

5. ATTACHED BIBLIOGRAPHY

- Suggested bibliography

- Mohan N., Undeland T. M., Robbins W. P., “Introduction to Power Electronics”, 3rd edition, Publisher: Tziola, Thessaloniki, 2010.
- Rashid M., “Power Electronics”, 1st edition, Publisher: "ION" Publishing Group, Athens, 2010.
- Manias St. “Power Electronics”, 2d edition, Publisher: Symeon, Athens, 2017.
- Kioskeridis I., “Power Electronics”, 1st edition, Publisher: Tziola, Thessaloniki, 2008.

- Relevant scientific journals:

- IEEE Transactions on Power Electronics
- IEEE Transactions on Industrial Electronics
- IEEE Transactions on Industry Applications
- IEEE Journal of Emerging and Selected Topics in Power Electronics
- IET Power Electronics

ADVANCED TECHNOLOGY ELECTRONIC DEVICES

COURSE OUTLINE

Responsible: Emmanouil Kymakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.7.016.0	**SEMESTER**	1st
COURSE TITLE	Advanced Technology Electronic Devices		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	4
Total	**0**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	

2. LEARNING OUTCOMES

Learning outcomes

The aim of the course is to familiarize students with the physical and electrical properties of organic semiconductors and the corresponding devices. Upon successful completion of the course, the student will be able to:

- Assess the physical and electrical properties of organic semiconductors.
- Understand the mechanisms of current conduction through these materials.
- Understand the procedures and methodology for studying the structure and properties of organic materials, and be acquainted with the modern fabrication methods currently used in the design of flexible materials and devices.
- Describe and analyze the operating principles of the basic organic electronic devices: from transistors to photovoltaic cells & systems, organic light-emitting diodes (OLEDs) and lasers.
- Know and apply the appropriate characterization protocols for the electrical and optical characterization of the various electronic devices for the reliable evaluation of their performance.
- Be able to use bibliographic databases to find and evaluate the most relevant articles in the field of their work.

General Competences

The course aims at the acquisition, by the graduate, of the following general competences:

- Understand the procedures and methodology for studying the structure and properties of organic materials, and be familiar with the modern fabrication methods currently used in the design of flexible materials and devices.
- Describe and analyze the operating principles of the basic organic electronic devices.
- Know and apply the appropriate protocols for electrical characterization in order to evaluate the performance of the various electronic devices.
- In the context of the literature searches they will be asked to carry out, be able to independently evaluate and extract the most useful information, thereby developing critical thinking both as a reader and as a writer.
- In the context of presenting their results, be able to function as a member of a research team and within an interdisciplinary environment.

3. SYLLABUS

Here's the English translation:

The aim of the course is familiarization with "printable" electronic devices that are not based on inorganic semiconductors, such as silicon, but on organic semiconductors fabricated using printing technologies, which constitute a pioneering category of electronics with enormous market potential in four key application areas: displays, photovoltaics, lighting and bio-electronic systems. To achieve this goal, the course is structured as follows:

Theory

A. Organic Semiconductors

- Conductive Conjugated Polymers and Small Organic Molecules.
- Electronic Structure and Electronic Properties.
- Correlation between chemical structure and optoelectronic properties.
- Synthesis and Characterization Techniques for Organic Semiconductors

B. Organic optoelectronic devices

- Photovoltaic Cells (OPVs)
- Light-Emitting Diodes (OLEDs)
- Organic Semiconductor Lasers
- Optical displays
- Colorimetry and cathode ray tube displays
- Field-Effect Transistors (OFETs)

C. Flexible electronic devices

- Printing techniques for electronic devices
- Flexible sensors
- Printed organic thin-film transistors
- Flexible bio-electronic devices

D. Characterization Techniques for Semiconductor Devices

- Morphological Characterization (AFM)
- Structural Characterization (X-ray spectroscopy)
- Elemental Characterization (Absorption, Raman and FTIR)
- Electrical Characterization (Hall characterization)

Laboratory: Fabrication and characterization of OPVs

- Processing of organic electronic thin-film layers by spin coating
- Thermal deposition of metals
- Optical characterization of organic electronic thin films
- Fabrication of PVs (perovskite)
- PV performance analysis (J/V, EQE, IQE)
- PV stability analysis

4. TEACHING and LEARNING METHODS - EVALUATION	
DELIVERY Face-to-face, Distance learning, etc.	Πρόσωπο με πρόσωπο στην τάξη
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICT in laboratory education • Use of ICT in communication with students via the e-class electronic platform • Specialized software in laboratory exercises • Support of the learning process via the e-class electronic platform
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** | **Semester workload** **Course total** |
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Students are assessed through: • Laboratory work: performance and participation in the laboratory exercises • Written laboratory report on the fabrication and characterization of PV devices • Oral presentation of the results

5. ATTACHED BIBLIOGRAPHY

1. E. Cantatore (ed.), *Applications of Organic and Printed Electronics: A Technology-Enabled Revolution*, Springer, 2013, ISBN 978-1-4614-3160-2
2. H. Klauk (ed.), *Organic Electronics II: More Materials and Applications*, Wiley-VCH, 2012, ISBN 978-3-527-32647-1
3. C. Brabec, U. Scherf, V. Dyakonov (eds.), *Organic Photovoltaics: Materials, Device Physics, and Manufacturing Technologies*, 2nd Edition, Wiley-VCH, 2014, ISBN 978-3-527-65693-6
4. S. R. Forrest, *Organic Electronics: Foundations to Applications*, Oxford University Press, 2020, ISBN 978-0-19-852953-3
5. P. Cosseddu, M. Caironi (eds.), *Organic Flexible Electronics: Fundamentals, Devices, and Applications*, Woodhead Publishing (Elsevier), 2021, ISBN 978-0-12-818890-3
6. G. Nisato, D. Lupo, S. Ganz (eds.), *Organic and Printed Electronics: Fundamentals and Applications*, Jenny Stanford Publishing, 2016, ISBN 978-981-4669-74-0

ADVANCED PROGRAMMING TECHNIQUES

COURSE OUTLINE

Responsible: Nikolaos Vidakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.7.022.0	**SEMESTER**	1st
COURSE TITLE	Advanced Programming Techniques		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	3	2
	1	1
	1	1
Total	**5**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialized knowledge/Core
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	Greek
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE203/

2. LEARNING OUTCOMES

Learning outcomes

The course "Advanced Programming Techniques" aims to provide students with the necessary knowledge in medium and large-scale programming, and to deepen their understanding of the C programming language, by examining specific language topics. Special functions of the language will be examined, a brief review of C memory management techniques will be provided, extensive reference to files will be made, and specific topics of compilation, linking multiple files, and elements of multithreaded programming will be discussed.

Upon successful completion of the course, the student will be able to:

1. Create and compile multi-file programs.
2. Implement and manipulate complex data structures.
3. Practice using files in computer storage peripheral units.

Develop parallel processing programs using multithreaded programming.

General Competences

- Searching, analyzing, and synthesizing data and information using the necessary technologies.
- Adaptation to new situations.
- Autonomous work.

- Teamwork.
- Generating new research ideas.

3. SYLLABUS

Units of Theoretical Lectures

- C preprocessor (directives, macros).
- Special operators and types (bit operators, enumerated types).
- Pointers and functions (pointers to arrays, arrays of pointers, pointers to pointers, pointers to functions, functions with variable number of arguments, recursion).
- Dynamic memory allocation and applications (functions, dynamic arrays, lists, trees, graphs).
- Special compilation topics (command line parameters, programs with multiple files, error handling).
- Input-output, files (streams, accessing text and binary files).
- C libraries (string, time, stdlib, etc.).
- Multithreaded programming (thread creation and usage, mutex, semaphores, etc.).

Laboratory Exercises

Exercises using the C programming language and Dev C++ software and Linux (gcc).

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICT in teaching • Use of ICT in laboratory training • Use of ICT in communication with students via the e-class electronic platform
TEACHING METHODS The manner and methods of teaching are described in detail.	
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	**Language of Assessment:** Greek **Assessment methods:** • Written final exam with problem solving (75%) • Group laboratory work (report and oral examination) (15%) Weekly homework exercises (10%) Assessment criteria are announced to students at the beginning of the semester and are posted on the course website on eClass.

Activity	Semester workload
Weekly homework exercises	7
Non-guided personal study	40
Laboratory	13
Group workshop activities	8
Lectures	52
Course total	**120**

5. ATTACHED BIBLIOGRAPHY

- “C: From Theory to Implementation”, G. S. Tselikis and N. D. Tselikas, ISBN: 978-960-93-1961-4.
- "C Programming in Depth" by N. M. Chatzigiannakis, Klidarithmos Publications, ISBN 978-960-461-715-9. (In Greek)

- ”Teach Yourself C Programming in 21 Days” by P. Aitken & B. L. Jones, Sams Publications, ISBN 978-067-230-736-2.
- "Introduction to Data Structures and Algorithms. Implementation in C" by I. Papoutsis, Ath. Stamoulis Publications, ISBN 978-960-351-832-7. (In Greek)

MULTIMEDIA TECHNOLOGIES: AUDIO, IMAGE, VIDEO

COURSE OUTLINE

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.7.026.0	**SEMESTER**	1st
COURSE TITLE	Multimedia Technologies: Audio, Image, Video		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	3	2
	1	1
	1	1
Total	**5**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Special background / Core
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English)
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE199/

2. LEARNING OUTCOMES

Learning outcomes

The aim of the course is the critical understanding of multimedia collection, representation, processing and management techniques, as well as the acquisition of basic and advanced knowledge in matters of digitization and compression of audio, image and video. Main learning outcomes include:

1. Knowledge of the basic terminology found in scientific texts in the field of multimedia.
2. Critical understanding of the fundamental principles of multimedia content compression (audio, image and video).
3. Knowledge of multimedia information creation & processing techniques and skills of handling specialized software packages for its processing.
4. Familiarity with technical specifications (e.g., ISO, ITU)

General Competences

- Search, analysis and synthesis of data and information, using the necessary technologies
- Decision making
- Autonomous work
- Promoting creative and inductive/deductive thinking

3. SYLLABUS

Theoretical Lecture Units

- Multimedia - Text, audio, image, video, animation. Categories and characteristics of multimedia applications.
- Information digitization - sampling, quantization, compression / encoding.
- General principles of compression. Lossy vs Lossless compression. Entropy and source encoding, RLE, differential or predictive coding. Vector quantization, Transformation encoding.
- Symmetry and asymmetry of compression techniques.
- Compression / encoding standards: PCM, DPCM, APCM, linear and logarithmic coding.
- Sound. Capture and audio compression techniques, MIDI, DCT, MP3.
- Image. Capture and image compression techniques (GIF, JPEG). Vector image.
- Video. Capture and video compression techniques (MPEG, MPEG2, MPEG4, H.264). Animation.
- Large capacity storage media and storage systems.

Laboratory Exercises

- Audio modulation and editing techniques.
- Image modulation, editing techniques, filtering tools.
- Video editing and tools.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICTs in lecturing • Use of ICTs for the communication with students via the e-class platform
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lectures: 39 Exercises: 13 personal study: 68 **Course total: 120**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Assessment Methods: • Individual laboratory exercises that require completion of concepts and combination of techniques taught (20%) • Written mid-term with short answer questions and problem solving (20%) • Written final exam with short answer questions and problem solving (60 %) Current course assessment details are posted in eclass.

5. ATTACHED BIBLIOGRAPHY

Relevant English Texts:

- Fundamentals of Multimedia, Ze-Nian Li, M.S. Drew, J. Liu, 2nd ed., Springer Berlin Heidelberg, 2014, ISBN:9783319052892.
- Multimedia Systems: Algorithms, Standards, and Industry Practices, P. Havaldar and G. Medioni, Cengage

Learning, 2009, ISBN: 9781418835941.
- Multimedia Systems, R. Steinmetz and K. Nahrstedt, Springer Berlin Heidelberg, 2014, ISBN: 9783662088791.

Internet Sources:

- http://www.jpeg.org
- http://www.mpeg.org

DATA AND INFORMATION VISUALIZATION SYSTEMS

COURSE OUTLINE

Responsible: Nikolaos Vidakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.7.027.0	**SEMESTER**	1st
COURSE TITLE	Data and Information Visualization Systems		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
Lectures	4	4
Total	**4**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Deepening / Consolidation of specialty knowledge
PREREQUISITE COURSES	Procedural Programming (1.004: 1st semester course), Object Oriented Programming (2.003: 2nd semester course) Introduction to data Bases (3.005: 3nd semester course)
LANGUAGE OF INSTRUCTION and EXAMINATIONS	Greek
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE176/

2. LEARNING OUTCOMES

Learning outcomes

The main goal of the course is to understand and learn techniques for creating and editing visualizations, reports and graphics.

The course teaches and uses two basic different environments: JasperSoft Studio and D3. At the same time, Gephi and Refuse are presented and used.

Upon successful completion of the course the student will be able to:

- Understand basic concepts of visualizations and graphics
- Know basic concepts of techniques for creating and editing visualizations, reports and graphics
- Analyse, designs and implements simple graphics
- Know and apply methods and tools of visualizations, reports and graphics
- Evaluate and ensure the quality of visualizations, reports and graphics

General Competences

- Autonomous & Independent work
- Teamwork
- Search, analysis and synthesis of data and information, using the necessary technologies

- Decision making
- Promoting liberal, creative and inductive/deductive thinking
- Work in an interdisciplinary environment
- Adapt to new situations
- Project Planning and Management

3. SYLLABUS

Theoretical Lecture Units

- Introduction to data and information visualization
- Report Generators

1. Data connection
2. Design Principles
3. Design elements

- Data Sources

1. XML, JSON, CSV, Databases

- Types of Reports

1. aPrints
2. Graphs
3. Grouping

- Patterns and their relationship with visualization

1. Outlines
2. Textures
3. Formulation Principles

- The colors and their role

1. Processing, design and placement of colors in the space

- Mind - Recognition - Conception

1. Principles of visual intellect
2. Highlights of the image
3. Understanding scene
4. Long Term Memory

- Interaction I.

1. Overview & Details
2. Zoom in
3. Focus & Content

- Interaction II

1. Dynamic Questions
2. Movement
3. Off-the-Desktop Interaction.

- Visualization applications (In art, science, etc.)

- Maps
- Trees & Visualization Networks
- Visualization Tools (Prefuse, JfreeChart, Gephi etc).

Laboratory Exercises

- In the laboratory part of the course students have the opportunity to practice the concepts of theory by using exercises that cover the material extensively and cultivate correct skills for the development of visualizations, reports and graphics.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICTs in lecturing • Specialized Software for analysis, design and implementation of software such as Gephi & Prefuse. • Use of ICTs for the communication with students via the e-class platform
TEACHING METHODS The manner and methods of teaching are described in detail.	

Activity	**Semester workload**
Lectures	45
Small individual exercises	30
Non-guided personal study	40
Written Final Exam	5
Course total	**120**

STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Assessment Language: Greek All announcements for the course regulations and complementary reading material are permanently posted in the course web page. The course grade incorporates the following evaluation procedures: Theory: Final written examination in the whole material (100%). The exam includes theory questions (from 3 to 5) and practice exercises (from 1 to 2). Laboratory: The final grade consists of written laboratory work (30%), project preparation (50%) and final exam (20%) The evaluation criteria are announced to the students at the beginning of each semester and are posted on the course website in the open e-class LMS.

5. ATTACHED BIBLIOGRAPHY

Recommended Bibliography:

- Few, Stephen (2009): Now You See It: Simple Visualization Techniques for Quantitative Analysis. Analytics Press
- Ware, Colin (2008): Visual Thinking: for Design. Morgan Kaufmann
- Card, Stuart K., Mackinlay, Jock D. and Shneiderman, Ben (eds.) (1999): Readings in Information Visualization: Using Vision to Think. Academic Press
- Tufte, Edward R. (1983): The Visual Display of Quantitative Information. Cheshire, CT, Graphics Press

Relevant Scientific Journals:

- Journal: Information Visualization Journal, published quarterly by Palgrave Macmillan,

- Conferences:

1. ο IEEE's VisWeek, η οποία περιέχει τις InfoVis και VAST (Visual Analytics Science and Technology)
2. ο CHI (Computer-Human Interaction), SIGGRAPH.

HUMAN COMPUTER INTERACTION

COURSE OUTLINE

Responsible: Dimosthenis Akoumianakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.7.029.0	**SEMESTER**	1st
COURSE TITLE	Human Computer Interaction		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	4
Total	**0**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Special background / Core
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	Greek / English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE157/

2. LEARNING OUTCOMES

Learning outcomes

The course aims to introduce students to the theory of Human Computer Interaction and the engineering practices of interactive systems and user interfaces. This is attempted by blending concepts from design theories and practice, engineering methods and techniques and evaluation of interactive software. Specific modules are further exposed in laboratory settings where students become acquainted with a) interactive technologies and their physical, syntactic and semantic analysis b) the user-centred approach to designing interactive systems and c) the architectural models, programming techniques and methodology for developing user interfaces.

General Competences

Successful completion of the course will promote general skills including

- Teamwork in framing, understanding and tackling HCI design challenges
- Searching, analysing and conceptualizing solutions
- Decision making given options in a design space
- Development of presentation skills and argumentative discourse

3. SYLLABUS

Topics include the following:

- Introduction to the principles of interaction and the multi-disciplinary nature of the underlying domain of discourse
- Review of interactive technologies and their physical, syntactic and semantic layers of analysis
- Detailed treatment of physical – lexical level analysis of user interfaces covering input / output devices and models, morphological analysis of input devices and interaction objects
- Dialogue models covering early efforts such as BNF Grammars, Task Action Grammar, UAN, as well as more recent approaches such as event-models
- Semantic analysis of user interfaces and interaction metaphors
- Theory-based HCI and models (including Ergonomic analysis, Fitts law, cognitive models (KLM, GOMS, NL-GOMS) and architectures
- User centered design and methods (premises and concepts)
- Scenario-based design and techniques
- Design space analysis
- Prototyping methods and tools
- User interface software and technologies (toolkits, IDEs, scripting)
- Implementation methods and tools

In the laboratory students engage in individual exercises and a case study which addresses most of the issues raised in the theory part.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Face to face / distance learning
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	e-class
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lecturing / 40 Mini-projects / case study / 80 **Course total** / **120**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	The course grade is based on project-based assessments of written (30 %), presentation (30 %) and practical work (40 %).

5. ATTACHED BIBLIOGRAPHY

D. Akoumianakis (2006): Designing the user Interface, A modern approach, Athens: Kleidarithmos

J. Jacko & A. Sears Eds., (2003): The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, Routledge.

N. Avouris (2000): Human Computer Interaction, Diavlos Publishing.

Instructors' notes

Selected papers from ACM Transactions on Computer Human Interaction, Human Computer Interaction and International Journal of Human Computer Interaction

ELECTRICAL MACHINES II

COURSE OUTLINE

Responsible: Georgios Orfanoudakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.8.002.0	**SEMESTER**	2nd
COURSE TITLE	Electrical Machines II		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	6
Total	**0**	**6**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Special background / Core
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE143/

2. LEARNING OUTCOMES

Learning outcomes

The course "Electric Machines II" aims to give students the necessary knowledge on AC electric motors. More specifically, it refers to the structure, operation, special features and applications of different types of AC motors.

Upon successful conclusion of this course, the students will be able to:

1. describe and explain the structure, structural characteristics and basic design principles of AC machines,
2. to experimentally determine and numerically calculate the parameters of the equivalent circuit of each machine,
3. examine and analyze the operation of AC motors through the corresponding equivalent circuits,
4. identify and distinguish the different types and variants of AC motors,
5. to create and reconstruct corresponding wiring connections of AC motors in the laboratory,
6. compare the alternatives and suggest the appropriate AC motor for a given application.

General Competences

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adaptation to new situations
- Autonomous work
- Teamwork

- Work in an interdisciplinary environment
- Production of new research ideas

3. SYLLABUS

Theoretical Lecture Units

- General: Creation of a rotating magnetic field.
- Synchronous machines: Structure, principle of operation, characteristics, types, excitation, equivalent circuits, vector equation - vector diagram, power flow diagram, losses and efficiency. Experimental determination of equivalent circuit parameters. Study of a synchronous machine for motor/generator operation. For generator operation: Autonomous operation - synchronization, behavior in changes of load and excitation current, specifications. For motor operation: Starting, behavior in changes of load and excitation current.
- Three phase asynchronous motors: Structure, structural characteristics, types (squirrel cage, wound rotor), principle of operation, concept of slip, equivalent circuit, vector equation - vector diagram, power flow diagram, losses and efficiency, torque-speed characteristic, changes on the characteristic.
- Single phase asynchronous motor: Structure and principle of operation, equivalent circuit, starting.
- Reference to special types of motors (Universal, Synchronous reluctance, Switched reluctance).

Laboratory Exercises

1. Study of synchronous AC machines.
2. Study of parallel operation of alternators and synchronous motor loading.
3. Starting of three-phase asynchronous motors, of squirrel cage and slip-ring type.
4. Study of power balance and loading of an asynchronous squirrel cage motor.
5. Study of single-phase asynchronous motor.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Project-based, Individual study
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	
TEACHING METHODS The manner and methods of teaching are described in detail.	
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	a) Project evaluation, b) Oral exam on given syllabus

Activity	Semester workload
Lectures	0
Laboratory Exercises	20
Συγγραφή εργ. αναφορών	20
Independent study	80
Study and literature review	60
Course total	**180**

5. ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- Stephen Chapman, 'AC-DC Electric Machines'
- Hubert I. Charles,'Electric Machines': Theory, Operation, Applications, Adjustment, and Control

- *Relevant scientific journals:*

- IEEE Transactions on Energy Conversion
- IEEE Transactions on Industry applications
- IET Electric Power Applications

OPERATING SYSTEMS

COURSE OUTLINE

Responsible: Miltiadis Grammatikakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.8.009.0	**SEMESTER**	2nd
COURSE TITLE	Operating Systems		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	4
Total	**0**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Compulsory Elective
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE147

2. LEARNING OUTCOMES

Learning outcomes

The **knowledge** which students acquire upon successful completion of the course relates to design and implementation of modern operating systems (OSes). More specifically, the students are taught concepts related to the lifecycle of processes and threads, SystemV/POSIX shared memory, and resource sharing, focusing on inter-process communication and synchronization primitives (IPC message queues, pipes, UNIX signals, and POSIX locks, semaphores, barriers, and condition variables). They are also exposed to techniques that can detect or avoid hazards, such as data race and protocol deadlock, and guided to examine cost-efficient solutions of classical OS problems, such as producer-consumer, readers-writers, dining philosophers, and sleeping barber. Students are finally introduced to complex OS kernel functions and high-level services, related to job scheduling, virtual-to-physical address translation, memory management, file system operations, and I/O device management.

The **skills**, which students develop upon successful course completion, relate to:

- Understanding the design, implementation, operation, and high-level services of modern OSes
- Understanding fundamental issues related to resource management in Linux and real-time OSes
- Programming in consistent and efficient manner concurrent systems utilizing IPC
- Embedding OS API functions (IPC, POSIX, SystemV)

The **abilities**, which students develop upon successful course completion, relate to:

- Enabling new problem-solving abilities, e.g. for avoiding hazards in concurrent systems

- Learning shell & systems programming (C/C++ & GNU/Linux) and GNU software development

General Competences

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adapt solutions to new situations (resource sharing, congestion, contention etc)
- Autonomous work
- Teamwork
- Decision making
- Promoting liberal, creative and inductive/deductive thinking\
- Work in an interdisciplinary environment

3. SYLLABUS

Theoretical Lectures

The theoretical part concentrates on the following topics:

- Introduction to OS Design, Architecture and Services, Examples, Open vs Free Software
- Processes & Threads, IPC, Shared Memory, Hazards (Data Races, Deadlocks), IPC functions
- Classical OS Problems and Solutions
- Job Scheduling Algorithms, Implementations
- Virtual-to-Physical Address Translation
- Memory Management, Paging Systems, Replacement Algorithms, Implementations
- File Systems, Kernel support, Examples
- I/O Device Management, Device Drivers Types
- Special OS Topics

Lab

Students are introduced to Linux OS implementation. Hands-on activities relate to shell programming and systems programming, focusing on task management and efficient use of different IPC functions. In addition through simple demos, students are exposed to sophisticated OS topics, such as kernel scheduling policies, paging and address translation (GNU/Linux page maps, analysis), file systems (simpleFS), and I/O device management (UART-to-SPI TTY driver of an LCD).

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Eclass for Optional Exercises. Project Presentations/Demonstration in the Lab
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Using Eclass
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** | **Semester workload** **Course total** |

STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	All announcements related to the syllabus, including complementary reading material, solved exercises, and optional homework problems, are permanently posted in the course web page (ECLASS). The course grade incorporates the following evaluation procedures: 1. Final Oral Exam on theoretical/practical problems (50%) 2. Project (50%) or Project

5. ATTACHED BIBLIOGRAPHY

Bibliography:

- Andrew S. Tanenbaum, and H. Bos, “Modern Operating Systems”, 3rd Edition, Prentice Hall, 2018.
- W.R. Stevens and S.A. Rago, "Advanced Programming in the UNIX Environment", 3rd edition, Addison-Wesley (2013), ISBN 978-0321637734.
- A. Silberschatz, P. B. Galvin, and G. Gagne, “Operating System Concepts”, 9th Edition, Wiley (2013), ISBN 978-1118093757, http://os-book.com

Other Important Sources

- Eclass - http://eclass.hmu.gr (notes, examples, open source code)

Relevant Scientific Journals:

- ACM Symposium on Operating Systems Principles (SOSP)
- Operating Systems Design and Implementation (OSDI)
- Workshop on Hot Topics in Operating Systems (HotOS)
- Architectural Support for Programming Languages and Operating Systems (ASPLOS)
- Linux Foundation conferences, e.g., Open Source Forum
- Linux Plumbers Conference

ADVANCED TOPICS IN DATABASES

COURSE OUTLINE

Responsible: Dimosthenis Akoumianakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.8.023.0	**SEMESTER**	2nd
COURSE TITLE	Advanced Topics in DataBases		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	4
Total	**0**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Special background / Core
PREREQUISITE COURSES	Sucessful completeion of an Introductory course on Databases
LANGUAGE OF INSTRUCTION and EXAMINATIONS	Greek and English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE158/

2. LEARNING OUTCOMES

Learning outcomes

The course aims to present current and emerging approaches to the modeling, design and development of database applications. The course builds on the foundations of the introductory third-semester course “Introduction to databases” which is considered a prerequisite. The specific targets of the course cover four thematic areas, namely (a) review of classical data models and database management systems (b) theory of database design (c) advanced data models and (d) databases and the internet. Accordingly, the course outline is formed around these four thematic areas.

General Competences

Successful completion of the course will promote general skills including

- Analysing an information system and record the logical structure of the data used in alternative forms and using alternative tools
- Designing relational schemata to obey to advanced normal forms
- Transforming data across data modeling frameworks
- Understand the basics of alternative approaches to database development with emphasis on object-orientation, graph databases and deductive systems

3. SYLLABUS

The theoretical part will cover:

- Introduction to data modeling and relational database management systems: Overview of classical data models, The relational approach to database development, Overview of SQL and Relational Algebra, Extensions to the basic relational data model, summary of advanced data modeling approaches.
- Relational design theory: Functional dependency theory, Schema decomposition, Desirable properties of well-formed schema, Assessment of decomposition, closure of a set of attributes, Applications of functional dependency theory, Normal forms
- Alternative data models and approaches to database development: Database modelling and the Unified Modelling Language (UML), Transformation of UML models to relational schema and vice versa, Object-oriented concepts and object orientation to databases, (e.g., abstract data types, inheritance, isa- and part-of hierarchies), object-relational databases (SQL3), Graph databases, Deductive databases.
- NoSQL systems and data model mappings (e.g., relations to property graph and vice versa)

In the laboratory students engage in individual exercises and a case study which addresses most of the issues raised in the theory part.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Face to face / distance learning
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	e-class and PostgreSQL
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lecturing / 40 Project-based work / case study / 80 **Course total** / **120**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	The course grade is based on project-based assessments of written (30 %), presentation (30 %) and practical work (40 %).

5. ATTACHED BIBLIOGRAPHY

H. Garcia-Molina, J. Ullman, J. Widom (2020): Database Systems (single volume), Crete University Publishing.

A. Silberschatz, H. F. Korth & S. Sudarshan (2001): Database System Concepts (4th Edition), McGraw-Hill ISBN 0-07-255481-9.

R. Elmasri & S. Navathe (1996): Fundamentals of Database Systems, Μετάφραση στα Ελληνικά από τις εκδόσεις "ΔΙΑΥΛΟΣ".

Instructor's notes and papers

GAME DESIGN & DEVELOPMENT

COURSE OUTLINE

Responsible: Nikolaos Vidakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.8.025.0	**SEMESTER**	2nd
COURSE TITLE	Game Design & Development		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	4	3
	1	1
Total	**5**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Deepening / Consolidation of specialty knowledge
PREREQUISITE COURSES	None.
LANGUAGE OF INSTRUCTION and EXAMINATIONS	Greek
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE173/

2. LEARNING OUTCOMES

Learning outcomes

The course deals with the design and development of games and especially serious games. Serious Games are games that do not have entertainment as their primary purpose but rather educational, awareness, training, advertising or other "serious" purposes that benefit from the engaging environment provided by the games to motivate users. Serious games are commonly used in defense, health, training, education, energy and other areas in which an engineer is active.

Students will go through all the steps of developing a game, selecting the idea and engaging it with learning objectives in developing and testing the original game in an environment that requires not only programming skills. The curriculum will allow students to understand the process of idealizing, designing, developing, testing, and delivering a game.

The aim of the course is to enable students to understand the process of conceiving, designing, developing, testing and delivering a serious game.

At the end of the course, students will be able to:

- Understand the basic concepts of game culture and digital game theory,
- Analyze game classifications and identify the specific characteristics of each type of game,
- Analyze, perform tests and record user needs and turn them into serious games,

• Understand and apply game methodologies,

• Understand the concepts related to the game, the flow of games, interactive storytelling, narration and their application in practice with the aim of developing an "addictive" game,

• They can refer to the main game tools (game editors, game engines) that are available for game development and identifying the best ones for a specific purpose,

? Conceive the idea, design, development, test and deliver a game.

General Competences

Adaptation to new situations,

- Decision making,
- Autonomous Work,
- Teamwork,
- Project Planning and Management,
- Work in an interdisciplinary environment,

Promoting free, creative and inductive thinking

3. SYLLABUS

Theoretical Lecture Units

Module 1: Introduction

- Chronology. Brief history of audiovisual narration (theater, painting, photography, comics, cinema, animation, television). Aesthetic movements. The evolution of storytelling in the digital world (theme parks, role playing games, multimedia games)
- Basic Principles of Audiovisual Narration. Brief history of audiovisual narration (theater, painting, photography, comics, cinema, animation, television). Aesthetic movements. The evolution of storytelling in the digital world (theme parks, role playing games, multimedia games)

Module 2: Analysis - Design - Art-Production

- GAME WORLD DESIGN

1. World Design,
2. Environment Design,
3. Multi-player Design,
4. Concept Artists
5. Creative Directors

- STORYLINE DESIGN

1. Level Design
2. Mission Design

- CHARACTERS DESIGN

1. character artist,
2. combat systems design,
3. animator,
4. motion capture artist, and
5. character rigger.

- LOOK AND FEEL

1. Cinematics Design or FX artist (effects artist)

- Game Manual
- CONTENT – GAME – SCRIPT WRITING

1. understanding the narrative and incidental writing needs of the game,
2. collaborating with mission designers to fuse their ideas with the storyline,
3. proofreading and rehearsing with actors and directors, and
4. communicating directly with the cinematics department.

- SOUND OR AUDIO DESIGN (recording and crafting audio to sync with animations in a game)

1. sound effects,
2. music,
3. ambient sound, and
4. voices

- QUALITY ASSURANCE DESIGN

Performed by:

- game testers,
- design analysts,
- software quality assurance engineers,
- beta game testers, or
- video game testers

To discover and document:

- defects,
- bugs, or glitches with game software.

Module 3: Programming and Development

- Game Algorithms and Logic Programming
- Tools and Languages for making games
- Game engines,
- Operating systems and game implementation
- Development of diffuse computing (AR) games
- Game Oriented Programming
- Artificial Intelligence
- Multiplayer & Computer Networked Gaming

Module 4: Verification & Validation

- Validation and Verification: The concepts of software validation and verification and their differences. Description of the program control process Explanation of static analysis as a verification technique,
- Game Testing: The control techniques used to find program errors,
- Quality Assurance Testing

Laboratory Exercises

- In the laboratory part of the course students have the opportunity to practice the concepts of theory by using exercises that cover the material extensively and cultivate correct programming skills for flexible & agile software development.

4. TEACHING and LEARNING METHODS - EVALUATION	
DELIVERY Face-to-face, Distance learning, etc.	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICTs in lecturing • Specialized Software for analysis, design and implementation of games • Use of ICTs for the communication with students via the e-class platform
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Non-guided personal study: 30 Small individual exercises: 20 Lectures: 24 Tutoring: 11 Teamwork Project with case study: 35 **Course total: 120**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Assessment Language: Greek All announcements for the course regulations and complementary reading material are permanently posted in the course web page. The course grade incorporates the following evaluation procedures: Theory: Final written examination in the whole material (100%). The exam includes theory questions (from 3 to 5) and practice exercises (from 1 to 2). Laboratory: The final grade consists of written laboratory work (10%), project preparation (50%) and final exam (40%) The evaluation criteria are announced to the students at the beginning of each semester and are posted on the course website in the open e-class LMS.

5. ATTACHED BIBLIOGRAPHY

Recommended Bibliography:

- Learn Unity for Windows 10 Game Development, Κωδικός Βιβλίου στον Εύδοξο: 75488229, Έκδοση: /2016, Συγγραφείς: Sue Blackman / Adam Tuliper, ISBN: 9781430267577
- Learn Unity for Android Game Development, Κωδικός Βιβλίου στον Εύδοξο: 75488228, Έκδοση: 1st ed./2017, Συγγραφείς: Adam Sinicki, ISBN: 9781484227046,
- Learn Unity 2017 for iOS Game Development [electronic resource], Κωδικός Βιβλίου στον Εύδοξο: 75488227, Έκδοση: 2nd ed./2017, Συγγραφείς: Allan Fowler / Philip Chu, ISBN: 9781484231746
- Polished Game Development, Κωδικός Βιβλίου στον Εύδοξο: 75490719, Έκδοση: /2016, Συγγραφείς: Steven Goodwin, ISBN: 9781484221228
- Mostly Codeless Game Development, Κωδικός Βιβλίου στον Εύδοξο: 75489336, Έκδοση: 1st ed./2017, Συγγραφείς: Robert Ciesla, ISBN: 9781484229705
- Evolutionary Optimization and Game Strategies for Advanced Multi-Disciplinary Design, Κωδικός Βιβλίου στον Εύδοξο: 73263415, Αριθμός τόμου: 75, Έκδοση: /2015, Συγγραφείς: Jacques Periaux / Felipe Gonzalez / Dong Seop Chris Lee, ISBN: 9789401795203
- Advanced Game Design with HTML5 and JavaScript, Κωδικός Βιβλίου στον Εύδοξο: 73261389, Έκδοση: /2015 Συγγραφείς: Rex Spuy ISBN: 9781430258018
- Serious Games Analytics, Κωδικός Βιβλίου στον Εύδοξο: 73266598, Έκδοση: /2015, Συγγραφείς: Christian Sebastian Loh / Yanyan Sheng / Dirk Ifenthaler, ISBN: 9783319058344

- Serious Games Interaction and Simulation, Κωδικός Βιβλίου στον Εύδοξο: 75492361, Αριθμός τόμου: 176, Έκδοση: /2017, Συγγραφείς: Carlos Vaz de Carvalho / Paula Escudeiro / Ant?nio Coelho, ISBN: 9783319510552
- Serious Games and Edutainment Applications, Κωδικός Βιβλίου στον Εύδοξο: 75492360, Έκδοση: /2017, Συγγραφείς: Minhua Ma / Andreas Oikonomou, ISBN: 9783319516455.

Relevant Scientific Journals:

- Schell, J. (2014). The Art of Game Design: A book of lenses. AK Peters/CRC Press.
- Salen, K., Tekinbas, K. S., & Zimmerman, E. (2004). Rules of play: Game design fundamentals. MIT press.
- Jenkins, H. (2004). Game design as narrative. Computer, 44(53), 118-130.
- El-Nasr, M. S., Drachen, A., & Canossa, A. (2016). Game analytics. Springer London Limited.
- Bethke, E. (2003). Game development and production. Wordware Publishing, Inc..
- Michael, D. R., & Chen, S. L. (2005). Serious games: Games that educate, train, and inform. Muska & Lipman/Premier-Trade.

DISTRIBUTED SYSTEMS AND CLOUD COMPUTING

COURSE OUTLINE

Responsible: Charalampos Papadakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.8.027.0	**SEMESTER**	2nd
COURSE TITLE	Distributed Systems and Cloud Computing		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	5	4
Total	**5**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Special background / Core
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE196/

2. LEARNING OUTCOMES

Learning outcomes

The aim of the course is to provide the necessary knowledge to understand the specifics involved in the design of Distributed Algorithms and Systems. The course will cover theoretical and practical issues of Distributed Systems in terms of their requirements, design approaches and available technologies. Upon successful completion of the course, the student will be able to:

Know the basic principles governing the operation of Distributed Systems.

Understand the structure, capabilities and limitations of Cloud Systems.

Design Distributed Algorithms to solve practical problems.

Evaluate the performance of Distributed Algorithms.

Propose solutions for the development of integrated Distributed Systems.

General Competences

Search, analysis and synthesis of data and information, using the necessary technologies

Adaptation to new situations

Autonomous work

Work in an interdisciplinary environment

Generation of new research ideas

Promotion of free, creative and inductive thinking

3. SYLLABUS

Lectures 1. Introduction 1.1. What is a distributed system. 1.2. Basic characteristics of distributed systems (Segmentation, Heterogeneity, Errors, Trust, Synchronization) 1.3. Categories of Distributed Systems 1.4. Distributed Computing Systems Distributed Information Systems Distributed Pervasive Systems 1.5. Comparison with related systems 1.6. Parallel Processing, Cluster Computing, Grid Computing 1.7. Basic Design Principles - Objectives 1.8. Transparency (of access, location, migration, migration, replication, concurrency, failure). Open mode. System composition from heterogeneous pieces of software and hardware from different sources. It includes the concepts of interoperability, portability, flexibility. Scalability – Scaling (Size, geographic, management). Techniques for achieving scaling (hiding delays, distribution, etc.), reliability. 1.9. Concepts of Distributed Operating Systems 1.10. Distributed Operating Systems (tightly coupled systems). Multiprocessor and Multicomputer Distributed Operating Systems. Networked Operating Systems (loosely coupled systems). Heterogeneity, reduced transparency. Conceptual transition to the concept of Middleware. 1.11. Middleware 1.12. Definition – Examples of middleware services (high-level communication, remoting, names, files, objects, distributed transactions, replication, security, process synchronization, resource locking). Description of examples. Analysis of how these paradigms obey and promote the desired characteristics of Distributed Systems described in the Introduction. 2. Models of Distributed Systems 2.1. Communication Models 2.2. Client/Server Model. Description, request/response protocol, server classes, model advantages and disadvantages. Three-Tier Client/Server Model Description - Layered Architectures. Model of Peer-to-Peer Systems. Basic concepts, model advantages and disadvantages, comparison with the Client/Server model, Examples of Peer-to-Peer Systems. How they achieve the goals of Distributed Systems. 2.3. Theoretical Models of Distributed Systems Modern Systems. Processes, states, execution steps, calculation and receipt events, security and liveness conditions. Asynchronous systems. Model description, clocks and delivery events not synchronized. 3. Basic Distributed Algorithms 3.1. Introduction to Distributed Algorithms. 3.2. Evaluation methods (time complexity and number of messages) 3.3. Message Propagation in a Spanning Tree. 3.4. Description and analysis of algorithm complexity in synchronous and asynchronous systems. 3.5. Collection of information in a Spanning Tree. 3.6. Description and analysis of algorithm complexity in synchronous and asynchronous systems. 3.7. Creating a Spanning Tree. 3.8. Description and complexity analysis of a Spanning Tree structure creation algorithm in synchronous and asynchronous systems with the flooding method. 3.9. Chief Selection in Distributed Systems. 3.10. Problem description. Description and analysis of named, uniform and non-uniform algorithms for Leader Selection in synchronous and asynchronous systems (use of ring topology, general topology, etc.). 4. Causality 4.1. Introduction to the concept of causality. 4.2. Definition of the happens-before relation. Definition of the concept of casual shuffles. 4.3. Logical Clocks. 4.4. Introduction and description of the concept of logic clocks. Definition, advantages and disadvantages. 4.5. Vector clocks. 4.6. Introduction and description of the concept of vector clocks. Their definition and function. Usage examples and conceptual transition to the message layout concept. 5. Arrangement of Messages 5.1. Introduction to the concept of message layout. 5.2. Description of reliable message propagation and requirements (Integrity, Validity, Agreement). 5.3. FIFO order of messages. 5.4. Description of the concept. Description and analysis of a layout FIFO algorithm using a message disbursement queue. 5.5. Causal arrangement of messages. 5.6. Description of the concept. Description and analysis of a causal message ordering algorithm using vector clocks. 5.7. Ultimate message layout. 5.8. Description of the concept. Description and analysis of a causal message ordering algorithm using Successor. Description and analysis of a causal message ordering algorithm using timestamps and a message buffer. 6. Fault Tolerance Algorithms 6.1. Introduction to Fault tolerance. 6.2. Analysis of fault-tolerant algorithms in the context of the Consensus problem. Introduction to error types. Description of it

4. TEACHING and LEARNING METHODS - EVALUATION	
DELIVERY Face-to-face, Distance learning, etc.	Face to Face in classroom
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Use of ICT in laboratory training Use of ICT in communication with students through the electronic platform e-class
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lectures: 52 Εργαστήριο: 13 Independent study: 52 Laboratory Exercises: 33 **Course total: 150**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	1. Written exam with Short Answer Questions (60%) 2. Oral exam of laboratory exercises (40%) The evaluation criteria are announced to students at the beginning of the semester and are posted on the course website in eClass.

5. ATTACHED BIBLIOGRAPHY

- Distributed Systems: Principles and Paradigms, Andrew S. Tanenbaum, Maarten van Steen, Publisher: Prentice Hall (2003), ISBN: 0131217860
- Journal of Parallel and Distributed Computing - Elsevier
- IEEE Transactions on Parallel and Distributed Systems - IEEE
- Journal of Parallel and Distributed Computing - ACM

ELECTRICAL DRIVE SYSTEMS

COURSE OUTLINE

Responsible: Georgios Orfanoudakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.9.005.0	**SEMESTER**	1st
COURSE TITLE	Electrical Drive Systems		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	4
Total	**0**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialised general knowledge
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE144/

2. LEARNING OUTCOMES

Learning outcomes

The course aims to give students basic knowledge on the structure and operation of Electric Motor Drive Systems, i.e. the control and power devices used to drive electric motors. The course presents the general principles of Electric Drive Systems and analyzes the specific characteristics of systems for driving DC and AC motors.

Upon successful conclusion of this course, the students should be able to:

1. identify and explain which electrical quantities and in what way they should be adjusted in order to apply the desired control on the driven motor - load,
2. describe and explain the structure, characteristics, capabilities and operation of the basic devices currently used to control DC and AC electric motors,
3. distinguish the differences and possibilities of the different control devices, as well as the requirements of the different types of mechanical load,
4. compose or propose the appropriate system for driving a given type of motor or load.

General Competences

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adaptation to new situations
- Autonomous work

- Teamwork
- Work in an interdisciplinary environment
- Production of new research ideas

3. SYLLABUS

Theoretical Lecture Units

1. Structural elements and basic characteristics of electric motor drive systems. System structure, understanding of the effect of the load's torque-speed characteristic on the selection and stability of the system. Drive system selection criteria. Analysis of operation in the four quadrants.
2. Systems with DC motors: Methods of controlling the speed-torque of direct current motors. Starting, braking, speed control with field weakening. Analysis of the operation of power converters (semi / fully controlled rectifiers and choppers) for the control of DC motors.
3. Systems with AC motors (asynchronous): Speed-torque control methods for three-phase asynchronous squirrel cage and wound rotor motors. Effect of supply voltage and frequency, and rotor resistance. Starting, braking, speed control with constant V/f ratio. Speed ??control of slip ring motors with slip power recovery. Analysis of the operation of power converters for soft start (soft starters) and speed regulation (inverters) of three-phase asynchronous motors.

Laboratory Exercises

They include laboratory exercises and simulations using MATLAB-Simulink.

1. DC motor starting.
2. DC motor speed control.
3. Start and adjustment of speed and torque of separately excited DC motor.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Project-based, Individual study
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	
TEACHING METHODS The manner and methods of teaching are described in detail.	
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	a) Project evaluation, b) Oral exam on given syllabus

Activity	**Semester workload**
Lectures	0
Laboratory Exercises	20
Study and report writing	20
Independent study	30
Study and literature review	50
Course total	**120**

5. ATTACHED BIBLIOGRAPHY

- Recommended bibliography:

- Krishnan R., “Electric Motor Systems: Modeling, Analysis and Control”, 1st edition, Klidarithmos Publications, Athens, 2009.
- Mohan N., Undeland TM, Robbins W. P., "Introduction to Power Electronics", 3rd edition, Tziola Publications, Thessaloniki, 2010.
- Rashid M., “Power Electronics”, 1st edition, Ion Publications, Athens, 2010.

- Relevant scientific journals:

- IEEE Transactions on Power Electronics
- IEEE Transactions on Industrial Electronics
- IEEE Transactions on Industry Applications

ADVANCED PHOTOVOLTAIC DEVICES

COURSE OUTLINE

Responsible: Emmanouil Kymakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.9.008.0	**SEMESTER**	1st
COURSE TITLE	Advanced Photovoltaic Devices		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	4
Total	**0**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialised general knowledge
PREREQUISITE COURSES	Ηλεκτροτεχνικά Υλικά Ι
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE116

2. LEARNING OUTCOMES

Learning outcomes

The purpose of the course is to familiarize students with the operating principles and development of new technologies in modern photovoltaic systems, as well as their design and evaluation. Upon successful completion of the course, the student will be able to:

- Understand the operating principle of new-technology PV devices based on classical semiconductor theory.
- Explore the growth techniques of organic/hybrid materials and the fabrication stages of the corresponding PV devices at both small and large scale.
- Design electrical characterization techniques for the calculation of the PV parameters of the devices, along with the corresponding protocols for measuring PV lifetime.
- Use bibliographic databases to find and evaluate the most relevant articles in the field of their work.
- Orally present and explain in detail their research/literature project.

General Competences

The course aims at the acquisition, by the graduate, of the following general competences:

- Search for, analysis and synthesis of data and information, with the use of the necessary technologies.
- Adapting to new situations.
- Working in an interdisciplinary environment.
- Team work.

- Promotion of free, creative and inductive thinking.
- Production of new research ideas.
- Search for, analysis and synthesis of data and information, with the use of the necessary technologies.

3. SYLLABUS

Here's the English translation:

The aim of the course is familiarization with the operating principles, fabrication methods and electrical characterization of third-generation photovoltaic devices, which are not based on silicon, but on organic and hybrid semiconductors that can be fabricated using printing technologies. To achieve this goal, the course is structured as follows:

- Introduction and Historical Review of third-generation PVs
- Operating Principles of organic photovoltaics (OPVs)
- Methods for OPV performance characterization
- Characterization methods and stability protocols for OPVs
- Introduction to hybrid perovskite photovoltaics (HPVs)
- Operating principles of HPVs
- The role of two-dimensional materials in HPVs
- Tandem devices
- Printing-based production technology for OPVs & HPVs
- Laboratory: Fabrication and electrical characterization of PV devices.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Πρόσωπο με πρόσωπο στην τάξη
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICT in laboratory education • Use of ICT in communication with students via the e-class electronic platform • Specialized software in laboratory exercises • Support of the learning process via the e-class electronic platform
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** | **Semester workload** **Course total** |
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Students are assessed through: • Laboratory work: performance and participation in the laboratory exercises • Written laboratory report on the fabrication and characterization of PV devices • Oral presentation of the results

5. ATTACHED BIBLIOGRAPHY

1. J. Bisquert, *The Physics of Solar Cells: Perovskites, Organics, and Photovoltaic Fundamentals*, CRC Press, 2017, ISBN 9781138099968
2. M. A. Green, *Third Generation Photovoltaics: Advanced Solar Energy Conversion*, Springer, 2003, ISBN 978-3-540-26562-7
3. C. Brabec, U. Scherf, V. Dyakonov (eds.), *Organic Photovoltaics: Materials, Device Physics, and Manufacturing Technologies*, 2nd Edition, Wiley-VCH, 2014, ISBN 978-3-527-65693-6

4. N.-G. Park, M. Gratzel, T. Miyasaka (eds.), *Organic-Inorganic Halide Perovskite Photovoltaics: From Fundamentals to Device Architectures*, Springer, 2016, ISBN 978-3-319-35112-4
5. T. Miyasaka (ed.), *Perovskite Photovoltaics and Optoelectronics: From Fundamentals to Advanced Applications*, Wiley-VCH, 2021, ISBN 978-3-527-34748-3
6. M. Pazoki, A. Hagfeldt, T. Edvinsson (eds.), *Characterization Techniques for Perovskite Solar Cell Materials*, Elsevier, 2020, ISBN 978-0-12-814727-6
7. F. C. Krebs (ed.), *Stability and Degradation of Organic and Polymer Solar Cells*, Wiley, 2012, ISBN 978-1-119-95251-
8. M. V. Khenkin et al., "Consensus statement for stability assessment and reporting for perovskite photovoltaics based on ISOS procedures," *Nature Energy* 5, 35–49 (2020) — directly supports the stability protocols unit

COMPUTER SYSTEMS SECURITY

COURSE OUTLINE

Responsible: Miltiadis Grammatikakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.9.015.0	**SEMESTER**	1st
COURSE TITLE	Computer Systems Security		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	4
Total	**0**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialization
PREREQUISITE COURSES	Operating Systems
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE150

2. LEARNING OUTCOMES

Learning outcomes

The **knowledge** which students acquire upon successful completion of the course relates to understanding the design of multilayer protection mechanisms for computing systems, with an emphasis on embedded systems security. Security primitives are examined in detail, including lightweight cryptographic software libraries and hardware security devices (programmable crypto engines, crypto ICs). In addition, security patterns/protocols for efficient access control, data privacy, anonymity, confidentiality, integrity, and availability are examined. Case studies range from device security (cryptos), to memory protection/isolation (ARM Trustzone), to operating system kernel and file system support, to application and system/network security, including high-level security event tracing, correlation, and visualization.

The **skills**, which students develop upon successful course completion, relate to:

- Understanding the design and use of public key and symmetric cryptography (lightweight
- Understanding the design and use of digital certificates and signatures
- Designing and implementing protocols and techniques for security and data privacy at device, system/network, and application level

The **abilities**, which students develop upon successful course completion, enable problem-solving abilities that relate to

- Integrating security/trust in system/platform design and implementation
- Implementing secure embedded systems using lightweight security primitives/protocols

- Validating security functions and evaluating overheads of at device-, system-, and network-level

General Competences

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adapt solutions to new situations (resource sharing, congestion, contention etc)
- Autonomous work
- Teamwork
- Decision making
- Work in an interdisciplinary environment
- Promoting liberal, creative and inductive/deductive thinking

3. SYLLABUS

Theoretical Lectures

- History – Classical Cryptography- Mathematical Preliminaries
- Authentication, Authorization
- SW Vulnerabilities - Memory Errors/Buffer Overflows, Viruses, Worms
- Side Channel, Energy Profiling, Covert Channel
- Buffer Overflows
- Operating System and Network Security (DDoS, Firewall, IPSec, OpenSSL/TLS, OpenVPN, syslog/IDPS)
- Symmetric Cryptography, NIST-approved Operating Modes
- Public Key Cryptography (RSA, Diffie Hellmann) & Elliptic Cryptography
- Security Primitives, Protocols, and Services
- Digital Certificates & Signatures
- Message Authenticity - Merkle Trees
- Application Security – Web/Ηλεκτρονικό Ταχυδρομείο (HTTPS, SMTP)
- Embedded Security, Cybersecurity & Safety, e.g. Smart Vehicles, e-Health platforms
- Special Topics (e.g. Blockchains, Steganography, Kerberos, Secret Sharing, Zero-Knowledge Proofs, Oblivious Transfers, Commit Protocols, HW Security (Crypto ICs, ARM Trustzone, Secure Boot/File Systems), Homomorphic Security, Quantum Cryptography, Data Privacy, Anonymity, Onion Routing/Tor, Legal Framework, GDPR, HIPAA etc)

Lab

The student lab focuses on open source hardware/software and Linux system security. Students gain experience in cryptographic mechanisms (AES encryption/decryption, integrity), authentication (SHA3, one-way hash functions), domain isolation, data privacy and anonymity by applying well-established security patterns for device, system/network, and application security. The lab also examines practical use of software tools, cryptographic security libraries, programmable crypto engines, and crypto ICs in experimental platforms and real embedded systems, such as healthcare and automotive.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Eclass for optional exercises. Project presentations and demonstrations in the Lab.

USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Eclass
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** | **Semester workload** **Course total** |
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	All announcements related to the syllabus, including grading, and complementary reading material, solved exercises, and optional homeworks, are permanently posted in the course web page (ECLASS). The course grade incorporates the following evaluation procedures: 1. Final Oral Exam on theoretical/practical problems (50%) 2. Term Project Presentation and Demonstration (50%) The project usually relates to systems/network programming, Linux drivers & kernel modules, RTOS, real-time systems or small software stacks. Students provide weekly reports on their progress, and a final presentation and demonstration at the end of their project.

5. ATTACHED BIBLIOGRAPHY

***Recommended Bibliography*:**

- P. C. Pfleeger, S. L. Pfleeger, J. Margulies, "Security in Computing", 5th edition, 2018. Prentice Hall, 2018.
- D. Basin, P. Schaller, M. Schlaepfer, "A Hands-on Approach", Springer, 2011.

Other Important Sources

- Eclass - http://eclass.hmu.gr (notes, examples, open source coce)
- Development boards, pirate devices, virtual machines accompanied with open source software and manuals for examining attack and devising protection mechanisms

Relevant Scientific Journals & Conferences

- ACM Transactions on Privacy and Security
- IEEE Transactions on Dependable and Secure Computing
- IEEE Security & Privacy
- IEEE Transactions on Information Forensics & Security
- IEEE Transactions on Intelligent Transportation Systems
- IEEE Transactions on Vehicular Technology
- USENIX Security Symposium
- IEEE Symposium on Security and Privacy
- DEFCON and BLACKHAT conferences
- Embedded Security-related conferences, e.g. Embedded Security in Cars (ESCAR), Linux Security Summit, Automotive Linux Summit, Automotive Manufacturing Summit, Automotive World

INTERNET MULTIMEDIA AND COMPUTER GRAPHICS

COURSE OUTLINE

Responsible: Athanasios Malamos

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.9.021.0	**SEMESTER**	1st
COURSE TITLE	Internet Multimedia and Computer Graphics		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	5	4
Total	**5**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialized general knowledge/Skills development
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE186/

2. LEARNING OUTCOMES

Learning outcomes

The course "Multimedia and Graphics on the Internet" aims to give students the cutting edge knowledge on creating web applications enhanced with multimedia (video, audio, text) and graphics. The course covers in theory and practice the modern technology of multimedia in an internet environment. The course studies important topics related to platforms and protocols streaming technologies and web3D technologies. In the course laboratory there is an internship in graphics and multimedia on the internet and in the use of cutting-edge technologies. Upon successful completion of the course the student: 1. Knows the methodologies of design and development of multimedia and graphic applications on the internet. 2. Uses cutting-edge technologies and tools used to develop multimedia and computer graphics applications in a browser environment and the Internet. 3. Integrates knowledge from different areas, such as usability and user-machine interface, internet technologies and computer systems, etc. 4. Designs and develops innovative applications required to serve the specialized needs of companies operating in the field of online games, telecommunications, information, entertainment, etc

General Competences

Search, analysis and synthesis of data and information, using the necessary technologies

? Decision making

? Autonomous work

? Promoting creative and inductive/deductive thinking

? Creation of new research ideas

3. SYLLABUS

Theory Lecture Units ? Introduction. Internet, protocols and servers. ? Introduction to web programming on the browser side ? Multimedia technology with emphasis on the internet ? Real-time applications using streaming technology ? Three dimensional technology on the internet ? Sound and virtuality. ? Rigid body technology on the internet ? Physics in graphics and multimedia applications ? Coexistence of graphics and multimedia in applications ? Applications in education, entertainment, social networks and telecommunications, etc.

Laboratory Exercise Modules ? Web3D platforms ? Internet data exchange protocols ? Designing models and worlds in an internet environment ? Design and development of multimedia applications in web and browser environments (Browser) ? Individual work of designing and developing multimedia and graphic applications based on state-of-the-art technologies.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Lectures and Lab projects
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	The course is in the area of Informatics and Computer Engineering
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lectures / 50 Lab Exercises / 30 Lab Projects / 40 **Course total** / **120**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	In classroom tests (10%) 2. Individual laboratory exercises that require completion of concepts and combination of techniques taught (30%) 3. Written mid-term with short answer questions and problem solving (20%) 4. Written final exam with short answer questions and problem solving (40 %) Current course assessment details are posted in eclass platform.

5. ATTACHED BIBLIOGRAPHY

"HTML5 and Javascript Projects," Jeanine Meyer, APRESS ? "Foundation Game Design with HTML5 and Javascript" , Rex Van Der Spuy, APRESS ? www.w3.org SVG, Declarative 3D, WEBRTC ? www.x3dom.org ? www.web3d.org ? www.threejs.org ? Multimedia Tools and Applications, Journal, Editor Springer ? International Journal of Interactive Multimedia and Artificial Intelligence (UNIR), open access

REALISITIC MULTIMEDIA AND ANIMATION

COURSE OUTLINE

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.9.022.0	**SEMESTER**	2nd
COURSE TITLE	Realisitic Multimedia and Animation		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	3	2
	1	1
	1	1
Total	**5**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialised general knowledge
PREREQUISITE COURSES	Object-Oriented programming (recommended)
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English)
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE200/

2. LEARNING OUTCOMES

Learning outcomes

The aim of the course is the critical application of fundamental engineering and mathematics concepts onto game engine programming environments. Main learning outcomes include:

1. Understanding key features of gaming engines.
2. Practice fundamental mathematics and physics concepts found in game engines.
3. Mathematical transformations and motion of a solid body.
4. Study, prediction and detection of collisions. What follows collisions.
5. Numerical solution of equations of motion in fields of forces under constraints.

General Competences

- Search, analysis and synthesis of data and information, using the necessary technologies
- Decision making
- Autonomous work
- Promoting creative and inductive/deductive thinking

3. SYLLABUS

Theoretical Lecture Units

- Main ingredients in games and game engines.
- Mathematical background relevant to game engines: points and lines: definition of point and line, straight line properties, applications in collision detection - geometry: distances, parabolas, circles and spheres with applications in collision detection - trigonometry: degrees and radians, identities. Scalars - Cartesian and polar coordinates - Vectors: addition/subtraction, internal and external product – arrays.
- Common transformations and applications in game engines: 2D/3D translation, scaling and rotation and their combinations.
- Description of 1D, 2D 3D motion in gaming engines, speed and acceleration, equations of motion, missiles and explosions.
- Description of forces and collisions in game engines: Newton's laws, effect of forces on the motion of bodies - work, kinetic energy, dynamic energy and conservation of energy. Collisions among fixed and moving objects, elastic and inelastic collisions - conservation of energy and momentum, modeling, prediction and detection of collisions. Rotational motion.

Laboratory Exercises

- Programming in a game engine environment to demonstrate theoretical concepts as well as the capabilities and limitations of gaming machines.
- Build games from scratch and / or convert / augment existing games.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICTs in lecturing • Use of ICTs for the communication with students via the e-class platform
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lectures / 39 Exercises / 13 personal study / 68 **Course total** / **120**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	• Individual laboratory exercises that require completion of concepts and combination of techniques taught (20%) • Written mid-term with short answer questions and problem solving (20%) • Written final exam with short answer questions and problem solving (60 %) Current course assessment details are posted in eclass.

5. ATTACHED BIBLIOGRAPHY

Relevant English Texts:

- Physics for Game Programmers, G. Palmer, APress, 2005, ISBN: 1-59059-472-X.
- Unity Game Physics (https://unity3d.com/learn/tutorials/s/physics).

AGILE SOFTWARE DEVELOPMENT

COURSE OUTLINE

Responsible: Nikolaos Vidakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.9.024.0	**SEMESTER**	1st
COURSE TITLE	Agile Software Development		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	4	3
	1	1
Total	**5**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Deepening / Consolidation of specialty knowledge
PREREQUISITE COURSES	Procedural Programming (1.004: 1st semester course), Object Oriented Programming (2.003: 2nd semester course)
LANGUAGE OF INSTRUCTION and EXAMINATIONS	Greek
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE175/

2. LEARNING OUTCOMES

Learning outcomes

The course deals with contemporary issues and concepts of software implementation and compares new methods and practices of development (flexible programming) with traditional (procedural programming)

The aim of the course is to acquire knowledge in modern techniques and methods of software implementation (agile programming, extreme programming, scrum development etc.). Using examples of both software implementation philosophies (flexible and pre-designed), their key features are presented in order to improve students' skills in professional software production.

Upon successful completion of the course the students will be able to:

- Know the basic concepts of Object Oriented & Procedural Software Engineering.
- Can implement Object Oriented & Procedural Software
- Know Quality Standards and Metrics of Procedural Systems
- Know Agile Programming Methods and Practices
- Compare and evaluate Planned & Agile Planning methods and practices.
- Compare and evaluate implementations with a pre-planned and agile implementation philosophy
- Adapt methods and practices to the specifics of the software

General Competences

- Autonomous & Independent work
- Teamwork
- Search, analysis and synthesis of data and information, using the necessary technologies
- Decision making
- Promoting liberal, creative and inductive/deductive thinking
- Work in an interdisciplinary environment Adapt to new situations
- Project Planning and Management

3. SYLLABUS

Theoretical Lecture Units

- Introduction and Historical Review

1. Procedural Programming
2. Object Oriented Software Implementation
3. Agile Implementation

- Principles of Analysis & Design:

1. SRP - Single Responsibility Principle
2. OCP - Open-Closed Principle (Open-Closed Design Principle)
3. LSP - Liskov Substitution Principle (Liskov Substitution Principle)
4. DIP - Dependency Inversion Principle
5. ISP - Interface Segregation Principle

- Basic Development Concepts
- Design Patterns:

1. Record
2. Standards
3. Construction standards
4. Structural standards
5. Behavioural standards

- Quality Systems of Pre-designed Systems:

1. Weighted Methods per Class (WMC)
2. Depth of Inheritance Tree (DIT)
3. Number of Children (NOC)
4. Coupling Between Object Classes (CBO)
5. Response for a Class (RFC and RFC ')
6. LCOM1, LCOM2, LCOM3

- Introduction to Flexible Programming:

1. The Manifesto for Agile Planning
2. Values and principles

- Agile Planning Perspective:

1. Comparison with traditional software implementation,

2. Adaptive vs. Predictive,
3. Iterative vs. Waterfall,
4. Code vs. Documentation

- Methods and practices used in flexible programming:

1. Methods such as ASD, RUP, AUP, DSDM etc.
2. Practices such as ATDD, IID klp.

- Customization of Methods: Customization of methods and practices to the specifics of the software.
- Overview and comparison with other methods:

1. RAD (Papid Application Development)
2. CMMI (Capability Maturity Model Integration)

- Implementation Evaluation with the philosophy of Agile Programming:

1. Implementation evaluation based on quantitative characteristics
2. Implementation evaluation based on measurable objectives

- Comparison of Planned and Flexible Programming

Laboratory Exercises

- In the laboratory part of the course students have the opportunity to practice the concepts of theory by using exercises that cover the material extensively and cultivate correct programming skills for flexible & agile software development.

4. TEACHING and LEARNING METHODS - EVALUATION	
DELIVERY Face-to-face, Distance learning, etc.	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICTs in lecturing • Specialized Software for analysis, design and • implementation of software. • Use of ICTs for the communication with students via the e-class platform
TEACHING METHODS The manner and methods of teaching are described in detail.	

Activity	**Semester workload**
Lectures	24
Tutoring	11
Small individual exercises	20
Teamwork Project with case study	35
Non-guided personal study	30
Course total	**120**

STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Assessment Language: Greek All announcements for the course regulations and complementary reading material are permanently posted in the course web page. The course grade incorporates the following evaluation procedures: Theory: Final written examination in the whole material (100%). The exam includes theory questions (from 3 to 5) and practice exercises (from 1 to 2). Laboratory: The final grade consists of written laboratory work (10%), project preparation (50%) and final exam (40%) The evaluation criteria are announced to the students at the beginning of each semester and are posted on the course website in the open e-class LMS.

5. ATTACHED BIBLIOGRAPHY

Recommended Bibliography:

- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development.
- Bruce Eckel:Thinking in Java, Prentice Hall, 1999, ISBN 0-13659-723-8
- Bruegge, B., & Dutoit, A. H. (2009). Object-?Oriented Software Engineering. Using UML, Patterns, and Java. Learning, 5(6), 7.
- Cockburn, A. (2006). Agile software development: the cooperative game. Pearson Education.
- Cockburn, Alistair, Agile Software Development, Addison Wesley, 2002.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. Computer, 34(9), 120-127.
- Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum (Vol. 1). Upper Saddle River: Prentice Hall.

Relevant Scientific Journals:

- Agile Alliance/Manifesto ? Principles of the Agile Alliance
- The New Methodology (Martin Fowler)
- The Agile Manifesto: Where It Came from and Where It May Go (Martin Fowler)
- Pairprogramming.com
- Agile Modeling
- www.AmbySoft.com/javaCodingStandards.pdf

ADVANCED EMBEDDED SYSTEMS (MSC)

COURSE OUTLINE

Responsible: Miltiadis Grammatikakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	MSc in Informatics Engineering		
LEVEL OF STUDIES	Postgraduate		
COURSE CODE	МП1007	**SEMESTER**	2nd
COURSE TITLE	Advanced Embedded Systems		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	7.5
	3	7.5
Total	**3**	**15**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialization
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/TP264

2. LEARNING OUTCOMES

Learning outcomes

Upon successful course completion, the students will be able to:

- identify problems that can be solved using low-cost embedded systems, or real-time embedded systems,
- understand the design flow and the use of EDA tools for implementing systems-on-chip (develop, FPGA),
- design, implement and optimize embedded systems related to key application domains (e.g., transportation, E-health).

General Competences

The course aims towards improving the following student abilities,

- understanding the foundations of complex system design, development, and analysis,
- mastering new technology challenges,
- doing individual work,
- doing teamwork,
- criticize and/or self-criticize,
- perform reasonable R&D related to system engineering.

3. SYLLABUS

Unit 1: Embedded Systems: Design Languages and Methodologies - Examples

- System-on-chip Modeling, design flow, abstraction levels
- Top-down (platform-based), and bottom-up (component-based) design
- System-level design, High-level synthesis, Model-driven design engineering, π.χ. μέσω UML
- Intellectual Property cores and related standardization

Unit 2: RTL Design: Languages and Methodologies - Examples

- IP Modeling using HDLs (Verilog and VHDL)
- Reconfigurable technology and rapid prototyping
- EDA tools for ISA (ISS, caching)
- Clocks, Critical Paths, Time closure
- Co-design, co-simulation, co-validation, verification
- RTL synthesis from system-level design using SystemC/C/C++ libraries.

Unit 3: Electronics System-Level Design

- System-level modeling and simulation (processor, routers, network-on-chip or bus, memory interfaces, peripherals)
- Transaction-Level Modeling (TLM)
- Analog-Mixed Signal Modeling
- Sequential and parallel simulation (simulation kernels, event lists)
- System-level power estimation: tools and methodologies
- RTOS Modeling
- Stochastic approaches to system-level optimization, e.g., simulated annealing
- SystemC modeling & simulation

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Lectures & Optional Exercises
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Use of ICTs in • lecturing and lab • for the communication with students via the e-class platform
TEACHING METHODS The manner and methods of teaching are described in detail.	
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Project that relates to the design, implementation, and analysis of an embedded system (100%). The methods and criteria for evaluation are explained in eclass and involve slide presentation and demo.

Activity	**Semester workload**
Lectures and Lab Demonstrations	65
Independent Student Projects	155
Final Oral Presentation and Demonstration	5
Course total	**225**

5. ATTACHED BIBLIOGRAPHY

Open-source software & tools (Manuals, papers).

Sample Books

- Peter Marwedel, ``Embedded System Design', Kluwer Academic Publisher, 2003.
- W. Wolf, ``Computers as Components: Principles of Embedded Computing Systems Design', Morgan Kaufman Publisher, 2001, ISBN 1-55860-541- X (case), ISBN 1-55860-693-9.
- Qing Li and Carolyn Yao, “Real-Time Concepts for Embedded Systems”, ISBN:1578201241.

Relevant Scientific Literature

From international conferences, e.g., DAC and DATE or journals, e.g., IEEE IoT Journal, IEEE Transactions on Computers, ACM Transactions on Embedded Computing Systems

APPLIED MATHEMATICS (MSC)

COURSE OUTLINE

Responsible: Emmanouil Mageiropoulos

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Postgraduate		
COURSE CODE	ΜΠ100Α	**SEMESTER**	1st
COURSE TITLE	Applied Mathematics		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	5	
Total	**5**	**7.5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	
PREREQUISITE COURSES	Basic Calculus and Basic Linear Algebra
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/TP370/

2. LEARNING OUTCOMES

Learning outcomes

Ability to use mathematical tools that are necessary for signal analysis and for the study of the electrical circuit

General Competences

Abstract thinking

3. SYLLABUS

Elements of Linear Algebra: Vector Spaces and Linear Maps, Mattrices over Real and Complex Numbers (Operations, Determinants, Eigenvalues and Eigenstates, Little Spectral Theorem, Gauss-Jordan Elimination Process. The Discreet Fourier Transform.

Laplace and Fourier Transforms: The L1 and L2 Spaces. Definitions of the Laplace and Fourier Transforms as particular cases of Integral Transforms, and properties of them.

Probability Theory: Basic Definitions. The concept of a Continuous Distribution. Some typical examples.

-

4. TEACHING and LEARNING METHODS - EVALUATION

<table>
<tr><td>DELIVERY
Face-to-face, Distance learning, etc.</td><td>On board</td></tr>
<tr><td>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
Use of ICT in teaching, laboratory education, communication with students</td><td>Yes</td></tr>
<tr><td>TEACHING METHODS
The manner and methods of teaching are described in detail.</td><td><table>
<tr><th>Activity</th><th>Semester workload</th></tr>
<tr><td>Four (4) hours per week (3 hours theory and 1 hour Exercises)</td><td></td></tr>
<tr><td>Course total</td><td></td></tr>
</table></td></tr>
<tr><td>STUDENT PERFORMANCE EVALUATION
Description of the evaluation procedure</td><td>•A midterm project (on Linear Algebra and Laplace Transform) measuring 30% of the final grade and a final exam measuring 70% of the final grade</td></tr>
</table>

5. ATTACHED BIBLIOGRAPHY

ADVANCED SOFTWARE ENGINEERING & BIG DATA MODELLING (MSC)

COURSE OUTLINE

Responsible: Nikolaos Vidakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Postgraduate		
COURSE CODE	ΜΠ100Η	**SEMESTER**	1st
COURSE TITLE	Advanced Software Engineering & Big Data Modelling		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
Total		**7.5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialized knowledge/Core
PREREQUISITE COURSES	None.
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE106/

2. LEARNING OUTCOMES

Learning outcomes

The course aims to present the principles, techniques, and methods for professional and systematic software development. The Unified Modeling Language (UML), CASE tools like Visual Paradigm and programming languages like Python and JAVA, will be used in the context of this course. Furthermore, the course shows how to handle the volume, speed and variety of big data of SQL and noSQL databases. It also looks at issues related to data management and data quality. In order for students to deepen in Software engineering and big data modelling, several software examples will be examined during the course lectures.

After completing the course the student will have the necessary knowledge to:

- Be able to design software,
- Use CASE tools for analyzing and designing software,
- Be able to administer the software development,
- Be able to use language modeling systems,
- Be able to implement professional software,
- Identify different data items,
- Select a data model that matches its data characteristics,
- Apply techniques for managing flow data,
- Know differences between a traditional database management system and a large data management system,

Recognize why there are so many data management systems.

General Competences

- Searching, analyzing, and synthesizing data and information using the necessary technologies.
- Adaptation to new situations.
- Autonomous work.
- Teamwork.
- Generating new research ideas.

3. SYLLABUS

System description languages – Unified Modelling Language (UML)

- Designing software
- Requirements analysis
- Software design engineering
- Software design trade-offs
- Software implementation planning
- Software design techniques

Software development process management

- Software development methods
- Rapid application development
- Agile software development
- Capability Maturity Model
- Waterfall model
- Spiral model
- V-model
- Design patterns
- Risk management

Software development techniques

- Software prototyping
- Object oriented development
- Flowcharts
- Reusability
- Software product lines
- Error handling and recovery

Big Data Modeling

- Introduction to Big Data Modelling
- Introduction to Data Models (Structures, Operations & Constraints)
- Introduction to Relational Data Models and Semi-structured Data Models
- XML versus JSON on the Internet
- Exploring the Relational Data Model of Comma Separated Values (CSV)
- Exploring the Semi-structured Data Model of JSON data
- Vector Space Model, Graph Data Model, Other Data Models

- Working With Data Models
- Data Model vs. Data Format
- Data Streams
- Data Lakes
- Streaming data applications
- Exploring Streaming Sensor Data
- Big Data Management: The “M” in DBMS
- DBMS-based and non-DBMS-based Approaches to Big Data
- From DBMS to BDMS
- Laboratory Exercises

Exercises using the C programming language and Dev C++ software and Linux (gcc).

4. TEACHING and LEARNING METHODS - EVALUATION	
DELIVERY Face-to-face, Distance learning, etc.	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Use of ICT in teaching • Use of ICT in laboratory training • Use of ICT in communication with students via the e-class electronic platform
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lectures / 52 Laboratory / 13 Non-guided personal study / 40 Group workshop activities / 8 Weekly homework exercises / 7 **Course total** / **120**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	**Language of Assessment:** English **Assessment methods:** • Oral final exam with problem solving (60%) • Group semester project (30%) Weekly homework exercises (10%) Assessment criteria are announced to students at the beginning of the semester and are posted on the course website on eClass.

5. ATTACHED BIBLIOGRAPHY

- Ian Sommerville, “Software Engineering” (8th edition), Addison-Wesley, May 25, 2006
- Pfleeger Shari Lawrence, “Τεχνολογία λογισμικού Θεωρία και Πράξη”, Α’ & Β’ ΤΟΜΟΣ, Κλειδάριθμος, 2003
- Grady Booch, James Rumbaugh, “The Unified Modeling Language User Guide”, (2nd Edition), Pearson Education Limited, 2005
- Grady Booch, Robert A. Maksimchuk, J. Newkirk et.al., “Object Oriented Analysis and Design with Applications”, Pearson Education Limited, 2007
- Martin Fowler, “UML Distilled: A Brief Guide to the Standard Object Modeling Language” (3rd Edition) (The Addison-Wesley Object Technology Series) (Paperback), Pearson Education Limited, 2003
- Alan Dennis, “Systems Analysis & Design with UML”, 2nd Edition with Visio 2007 Set, John Wiley and Sons Ltd, 2008

- Ivar Jacobson, Grady Booch, James Rumbaugh: “The Unified Software Development Process”, Addison-Wesley, 1999
- Ivar Jacobson, Magnus Christerson, Patrik Jonsson, Gunnar Oevergaard: “Object-Oriented Software Engineering (A use case Driven Approach)”, Addison-Wesley, 1995
- Rational Rose Home Page: http://www.rational.com/products/rose/ index.jtmpl
- Object Management Group – UML: http://www.uml.org
- The Carnegie Mellon Software Engineering Home: http:// www.sei.cmu.edu /sei-home.html
- RabbitMQ in Action: Distributed Messaging for Everyone, Manning Pubs Co Series, Alvaro Videla, Jason J. W. Williams, Manning Publications Company, 2012, ISBN 1935182978, 9781935182979
- Evan Stubbs, Big data, big innovation: enabling competitive differentiation through business analytics, 2018 ISBN 978-1-118-72464-4
- Pete Warden, Big Data Glossary, 2011, O’Reilly, ISBN: 978-1-449-31459-0

ADVANCED TOPICS IN MULTIMEDIA AND COMPUTER GRAPHICS (MSC)

COURSE OUTLINE

Responsible: Athanasios Malamos

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	MSc in Informatics Engineering		
LEVEL OF STUDIES	Postgraduate		
COURSE CODE	ΜΠ100Ι	**SEMESTER**	1st
COURSE TITLE	Advanced Topics in Multimedia and Computer Graphics		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	5	7.5
Total	**5**	**7.5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Ειδικότητας
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE108/

2. LEARNING OUTCOMES

Learning outcomes

The course aims to give students the cutting edge knowledge on creating applications enhanced with multimedia (video, audio, text) and graphics. The course covers in theory and practice of the novel technologies of multimedia and especially on the internet environment. The course studies important topics related to platforms and protocols streaming technologies and web3D technologies. In the course laboratory there is an internship in graphics and multimedia on the internet and in the use of cutting-edge technologies.

General Competences

Upon successful completion of the course the student:

1. Knows the methodologies of design and development of multimedia and graphic applications on the internet.
2. Uses cutting-edge technologies and tools used to develop multimedia and computer graphics applications in a browser environment and the Internet.
3. Integrates knowledge from different areas, such as usability and user-machine interface, internet technologies and computer systems, etc.
4. Designs and develops innovative applications required to serve the specialized needs of companies operating in the field of online games, telecommunications, information, entertainment, etc.

3. SYLLABUS

Theory Lecture Units

- Introduction. Internet, protocols and servers.
- Introduction to web programming on the browser side
- Multimedia technology with emphasis on the internet
- Real-time applications using streaming technology
- Three dimensional technology on the internet
- Sound and virtuality.
- Advanced topics in graphics
- Physics in graphics and multimedia applications
- Coexistence of graphics and multimedia in applications
- Applications in education, entertainment, social networks and telecommunications, etc.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Lectures and Lab projects
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	The course belongs in the area of Informatics & Computer Engineering
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lectures / 50 Lab Exercises / 40 Lab Projects / 85 **Course total** / **175**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	The course evaluation is mainly based on small project reports and exercises during the semester and a final project where you will develop an application or make a literature research of your choice.

5. ATTACHED BIBLIOGRAPHY

- "HTML5 and Javascript Projects," Jeanine Meyer, APRESS
- "Foundation Game Design with HTML5 and Javascript" , Rex Van Der Spuy, **APRESS**
- www.w3.org SVG, Declarative 3D, WEBRTC
- www.x3dom.org
- www.web3d.org
- www.threejs.org
- Multimedia Tools and Applications, Journal, Editor Springer
- International Journal of Interactive Multimedia and Artificial Intelligence (UNIR), open access

DISTRIBUTED SYSTEMS AND APPLICATIONS (MSC)

COURSE OUTLINE

Responsible: Charalampos Papadakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	MSc in Informatics Engineering		
LEVEL OF STUDIES	Postgraduate		
COURSE CODE	ΜΠ2000	**SEMESTER**	1st
COURSE TITLE	Distributed Systems and Applications		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	5	7.5
Total	**5**	**7.5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Winter Semester
COURSE WEBSITE (URL)	

2. LEARNING OUTCOMES

Learning outcomes

The aim of the course is to provide the necessary knowledge to understand the specifics involved in the design of Distributed Algorithms and Systems. The course will cover theoretical and practical issues of Distributed Systems in terms of their requirements, design approaches and available technologies. Upon successful completion of the course, the student will be able to:

Know the basic principles governing the operation of Distributed Systems.

Understand the structure, capabilities and limitations of Cloud Systems.

Design Distributed Algorithms to solve practical problems.

Evaluate the performance of Distributed Algorithms.

Propose solutions for the development of integrated Distributed Systems.

General Competences

Search, analysis and synthesis of data and information, using the necessary technologies

Adaptation to new situations

Autonomous work

Work in an interdisciplinary environment

Generation of new research ideas

Promotion of free, creative and inductive thinking

3. SYLLABUS

Lectures 1. Introduction 1.1. What is a distributed system. 1.2. Basic characteristics of distributed systems (Segmentation, Heterogeneity, Errors, Trust, Synchronization) 1.3. Categories of Distributed Systems 1.4. Distributed Computing Systems Distributed Information Systems Distributed Pervasive Systems 1.5. Comparison with related systems 1.6. Parallel Processing, Cluster Computing, Grid Computing 1.7. Basic Design Principles - Objectives 1.8. Transparency (of access, location, migration, migration, replication, concurrency, failure). Open mode. System composition from heterogeneous pieces of software and hardware from different sources. It includes the concepts of interoperability, portability, flexibility. Scalability – Scaling (Size, geographic, management). Techniques for achieving scaling (hiding delays, distribution, etc.), reliability. 1.9. Concepts of Distributed Operating Systems 1.10. Distributed Operating Systems (tightly coupled systems). Multiprocessor and Multicomputer Distributed Operating Systems. Networked Operating Systems (loosely coupled systems). Heterogeneity, reduced transparency. Conceptual transition to the concept of Middleware. 1.11. Middleware 1.12. Definition – Examples of middleware services (high-level communication, remoting, names, files, objects, distributed transactions, replication, security, process synchronization, resource locking). Description of examples. Analysis of how these paradigms obey and promote the desired characteristics of Distributed Systems described in the Introduction. 2. Models of Distributed Systems 2.1. Communication Models 2.2. Client/Server Model. Description, request/response protocol, server classes, model advantages and disadvantages. Three-Tier Client/Server Model Description - Layered Architectures. Model of Peer-to-Peer Systems. Basic concepts, model advantages and disadvantages, comparison with the Client/Server model, Examples of Peer-to-Peer Systems. How they achieve the goals of Distributed Systems. 2.3. Theoretical Models of Distributed Systems Modern Systems. Processes, states, execution steps, calculation and receipt events, security and liveness conditions. Asynchronous systems. Model description, clocks and delivery events not synchronized. 3. Basic Distributed Algorithms 3.1. Introduction to Distributed Algorithms. 3.2. Evaluation methods (time complexity and number of messages) 3.3. Message Propagation in a Spanning Tree. 3.4. Description and analysis of algorithm complexity in synchronous and asynchronous systems. 3.5. Collection of information in a Spanning Tree. 3.6. Description and analysis of algorithm complexity in synchronous and asynchronous systems. 3.7. Creating a Spanning Tree. 3.8. Description and complexity analysis of a Spanning Tree structure creation algorithm in synchronous and asynchronous systems with the flooding method. 3.9. Chief Selection in Distributed Systems. 3.10. Problem description. Description and analysis of named, uniform and non-uniform algorithms for Leader Selection in synchronous and asynchronous systems (use of ring topology, general topology, etc.). 4. Causality 4.1. Introduction to the concept of causality. 4.2. Definition of the happens-before relation. Definition of the concept of casual shuffles. 4.3. Logical Clocks. 4.4. Introduction and description of the concept of logic clocks. Definition, advantages and disadvantages. 4.5. Vector clocks. 4.6. Introduction and description of the concept of vector clocks. Their definition and function. Usage examples and conceptual transition to the message layout concept. 5. Arrangement of Messages 5.1. Introduction to the concept of message layout. 5.2. Description of reliable message propagation and requirements (Integrity, Validity, Agreement). 5.3. FIFO order of messages. 5.4. Description of the concept. Description and analysis of a layout FIFO algorithm using a message disbursement queue. 5.5. Causal arrangement of messages. 5.6. Description of the concept. Description and analysis of a causal message ordering algorithm using vector clocks. 5.7. Ultimate message layout. 5.8. Description of the concept. Description and analysis of a causal message ordering algorithm using Successor. Description and analysis of a causal message ordering algorithm using timestamps and a message buffer. 6. Fault Tolerance Algorithms 6.1. Introduction to Fault tolerance. 6.2. Analysis of fault-tolerant algorithms in the context of the Consensus problem. Introduction to error types. Description of it

4. TEACHING and LEARNING METHODS - EVALUATION	
DELIVERY Face-to-face, Distance learning, etc.	Face to Face in classroom
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Use of ICT in laboratory training Use of ICT in communication with students through the electronic platform e-class
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** / **Semester workload** Lectures: 52 Stidy: 52 Project: 46 **Course total**: **150**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Written Exam (40%) Implementation Project (60%)

5. ATTACHED BIBLIOGRAPHY
Distributed Systems: Principles and Paradigms, Andrew S. Tanenbaum, Maarten van Steen, Publisher: Prentice Hall (2003), ISBN: 0131217860 Journal of Parallel and Distributed Computing - Elsevier IEEE Transactions on Parallel and Distributed Systems - IEEE Journal of Parallel and Distributed Computing - ACM

REALISTIC MULTIMEDIA AND ANIMATION (MSC)

COURSE OUTLINE

Responsible: Ioannis Pachoulakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	MSc in Informatics Engineering		
LEVEL OF STUDIES	Postgraduate		
COURSE CODE	ΜΠ200Λ	**SEMESTER**	2nd
COURSE TITLE	Realistic Multimedia and Animation		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	3	4
	2	3.5
Total	**5**	**7.5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialised general knowledge
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/TP274/

2. LEARNING OUTCOMES

Learning outcomes

The course combines three fundamental ingredients in game programming: (a) physics, basically mechanics – kinematics and dynamics, (b) related mathematics and (c) numerical methods to describe and follow up the evolution of a specific problem. Additional game elements such as user interaction, guidance and control, will be implemented for a user-centered experience. Finally, students will have the opportunity to develop games of increasing complexity using popular/free game engines.

Due to the material mix and focus, attending the course requires an avid interest in game development. Ideal candidates are:

- Natural Sciences graduates (they have a good understanding of physics/math, but may need work on programming and SW engineering).
- Informatics graduates (they have a solid background in programming/SW engineering, but may need to build physics/math knowledge).

General Competences

- Describe and follow up the evolution of a specific problem
- User interaction and guidance and control
- Support a user-centered experience.

- Development of games of increasing complexity using popular/free game engines

3. SYLLABUS

Theoretical Lecture Units

- Recap: Basic Game Ingredients: Physics, Math, Numerical Methods, Design and Implementation. Points and Lines, Coordinate Systems, Distances between points and shapes, Applications to Collision Detection, Trigonometry.
- Vector and Matrix Operations: Scalar and Vector physical quantities; Common Geometrical Transformations (Translation, Scaling, Rotation) and their combinations.
- Rigid Body Motion: Speed, Velocity, Acceleration, Newton's Laws, Forces (gravitation, spring, friction, torque, Momenta (Linear & Angular), Center of Mass, Inertia Tensor, Work, Kinetic & Potential Energies, Conservation Laws.
- Rotational Motion and Rotational Dynamics.
- Rigid Body Motion: Newtonian & Lagrangian Dynamics – Equations of Motion for a Particle and for Particle Systems, the effect of Conservative and Dissipative Forces.
- Detecting and Handling Collisions.
- Relevant topics in Linear Algebra and Calculus.
- Selected Numerical Methods: Euler & Taylor methods, Runge-Kutta Methods, Extrapolation Methods, Numerical Stability.
- Topics in Game Engines: Workflow, Models, Environment, Shaders, Textures, Effects, Lighting Models, Transparency, Scripting, etc.

Laboratory Exercises

- Programming in a game engine environment to demonstrate theoretical concepts as well as the capabilities and limitations of gaming machines.
- Build games from scratch and / or convert / augment existing games

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Face-to-face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	• Powerpoint, Simulation Packages • e-class
TEACHING METHODS The manner and methods of teaching are described in detail.	

Activity	**Semester workload**
Lectures	39
Homework	26
Self-study	160
Course total	**225**

STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	• [30%] Weekly HWK assignments (theory or lab problems): These are individual - not group - assignments, intended for you to practice new material and hone your skills. Unless otherwise stated in the assignment itself, hwks will be due the following Tuesday @ 13:00 after they are announced in class (late HWK penalty 20%/day). • [30%] Final Written Exam: Two weeks before the end of classes • [40%] Term Project: Develop your own game using a game engine of your choice – kicks off around week #5. Term projects will be presented in class in the last 2 weeks of the semester. Project-level credit break up is as follows: - 10% : Project planning and execution: (make a plan, discuss it in class and follow up diligently) - 20% : Live Demo (quality of work, professionalism, oral presentation) - 10% : Packaging-Documentation: you will hand in a software package with source code, executables and detailed report detailing representative test run cases - also your code must be appropriately commented.

5. ATTACHED BIBLIOGRAPHY

Bibliography:

- "Beginning Math and Physics for Game Programmers – 3rd ed", W. Stahler, New Riders, 2004, ISBN 978-0735713901
- "Physics for Game Developers – 2nd ed", D. M. Bourg, O'Reilly, 2013, ISBN 978-1-449-39251-2
- "Physics for Game Programmers", G. Palmer, A.Press, 2005, ISBN 1-59059-472-X
- "Mathematics for Game Developers", C. Tremblay, Thomson, 2004, ISBN 1-59200-038-X
- "Game Physics, Second Edition", D. H. Eberly, Morgan Kaufmann, 2010, ISBN 978-0123749031
- Selected Articles & other readings from the current bibliography

HELLENIC MEDITERRANEAN UNIVERSITY

International Relations Office

Estavromenos, 71410 Herakleion, Crete, Greece

iro.hmu.gr

Generated on August 25, 2026