COURSE OUTLINE

(1) GENERAL

SCHOOL	ENGINEERING		
ACADEMIC UNIT	INFORMATICS AND COMPUTER ENGINEERING		
LEVEL OF STUDIES	UNDERGRADUATE		
COURSE CODE		**SEMESTER**	**1st**
COURSE TITLE	Introduction to Computer Science		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits	**WEEKLY TEACHING HOURS**	**CREDITS**
Lectures	3	
Tutorials	2	
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).	5	5

COURSE TYPE general background, special background, specialised general knowledge, skills development	GENERAL BACKGROUND.
PREREQUISITE COURSES:	-
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	GREEK (Instruction and Examination)
IS THE COURSE OFFERED TO ERASMUS STUDENTS	NO
COURSE WEBSITE (URL)	

(2)

(2) LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.

Consult Appendix A

- Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area
- Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B
- Guidelines for writing Learning Outcomes

The Introduction in Computer Science course is a background course for Information Technology and Computer Science. The purpose of the course is to provide introductory and fundamental knowledge of Computer Science, introduce the elementary background understanding, so that the student is properly prepared to attend the majority of the courses included in the curriculum of the education program.

Upon successful completion of this course, the student:

- **will recognize and be able to describe much of the terminology of Computer Science,**
- **will be able to adequately use knowledge from Architecture and the operation of a machine in other subjects during his studies,**
- **will know the way the information is represented and stored in it,**
- **will have a thorough understanding of logic gates and be able to design logic circuits**
- **will understand data processing,**
- **will have basic knowledge of Operating Systems and their role,**
- **will know Computer Networks and their elementary functioning,**
- **will understand the operation and importance of the internet, the internet of Things, Sensor Networks and Cloud Computing.**
- **will have acquired knowledge of communications and communication systems,**
- **will know the concept of algorithm and algorithmic thinking, will understand the algorithmic solution of a problem, its connection with coding, and will be able to code this solution with a programming language.**
- **will be able to separate the different programming paradigms and corresponding elements such as iteration structures and control structures from different programming languages so that he can proceed to learn new programming languages, non-procedural,**
- **will have general knowledge of Software Technology for the development of applications with procedural or object-oriented programming,**
- **will have acquired knowledge of Theory of Computation, Turing machines, and complexity problems.**
- **will have acquired elementary knowledge of Information Theory and its applications in the Theory of Computation.**

General Competences

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

Search for, analysis and synthesis of data and information, with the use of the necessary technology
Adapting to new situations
Decision-making
Working independently
Team work
Working in an international environment
Working in an interdisciplinary environment
Production of new research ideas

Project planning and management
Respect for difference and multiculturalism
Respect for the natural environment
Showing social, professional and ethical responsibility and sensitivity to gender issues
Criticism and self-criticism
Production of free, creative and inductive thinking
……
Others…
…….

- Examine, retrieve, analyze and synthesize data and information by utilizing necessary technologies
- Work independently
- Adapting to new situations.

(3)

(3) SYLLABUS

The course includes the topics described in the following list:

1. Introduction
    1.1. The Science of Information and Computers
    1.2. Algorithms and algorithmic thinking
    1.3. The origin and evolution of Information Technology and Computers

Part 1 MACHINE ARCHITECTURE

2. Data Storage
    2.1. Logical operations and gates
    2.2. Logic circuits (FLIP FLOPS)
    2.3. Storing bits
    2.4. Main Memory (MB)
    2.5. Mass Storage
    2.6. Representation of information in the form of bits
    2.7. The binary system
    2.8. Storing integers
    2.9. Storing fractions
    2.10. Data compression
    2.11. Communication errors
3. Data Management
    3.1. Computer Architecture
    3.2. Machine language
    3.3. Program Execution
    3.4. Arithmetic/Logical Commands
    3.5. Communication with other devices
    3.6. Other Architectures (parallel systems, etc.)

Part 2 SOFTWARE

4. Operating Systems and Networks
    4.1. Evolution of Operating Systems
    4.2. Operating System Architecture
    4.3. Coordination of Activities
    4.4. Handling of Process Competition

    4.5. Networks
    4.6. The Internet
    4.7. The Internet of Things
    4.8. Cloud Computing
    4.9. Network Protocols
    4.10. Security

5. Programming Languages
    5.1. Programming Languages through time
    5.2. Traditional Programming
    5.3. Procedural Units
    5.4. Object-Oriented Programming
    5.5. Declarative Programming
    5.6. Algorithmic problem solving
    5.7. From algorithm to program

6. Software Engineering
    6.1. The software life cycle.
    6.2. Software engineering methodologies.
    6.3. Organize software into modules.
    6.4. Design methodologies.

Part 3 MACHINES AND COMPUTATIONS

7. Theory of Computation
    7.1. Functions and their Computation
    7.2. Turing machines
    7.3. Universal Programming Languages
    7.4. Non-Computable Function
    7.5. Complexity of Problems
    7.6. Public Key Encryption

(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	• Purpose-built Software for the construction of digital circuits, a virtual machine simulator with a specific machine language (the VirtualMachineSimulator - VMS), different OS platforms (Windows, Linux), programming languages, tools for algorithms. • Use of ICT in Course Teaching • Use of the Open eClass course management system, for distributing lecture notes and exercises for practice, and for communication with students.
TEACHING METHODS The manner and methods of teaching are described in detail. Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc. The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS	**ActivitySemester workload** Lectures 39 Tutorials 26 Essays / Project 10 Independent Study 50 **Course total 125**
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other Specifically-defined evaluation criteria are given, and if and where they are accessible to students.	I. Final written exam (100%), which includes: * Questions that combine knowledge and criticism, with complete justification and description of arguments, through which it is established the level of understanding of the topics. - Separation of Computer Science terms - Solving problems with quantitative data for the memory, the measurement of data, transfer rates, etc. - machine limit issues. - Questions about networks, the internet. - Questions about the machine and its operation. - Problems with language programs machine. - Questions about programming languages and programming paradigms. - Questions about Software Engineering. - Questions on the Theory of Computation.

	II. Additional (10%) score from individual optional practical tasks over the grade of the final theory written exam, when this is at least 5.

(5)

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

1. Βογιατζής Ι., Αντωνοπούλου Η., ΥΛΙΚΟ, ΛΟΓΙΣΜΙΚΟ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΕΣ ΥΠΟΛΟΓΙΣΤΩΝ, 4η έκδοση, 2021, ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ, ISBN 978-960-578-030-2 (in greek).

2. Glenn Brookshear J. and Dennis Brylow, Computer Science an Overview, 13th edition, 2018, Pearson, ISBN-13: 978-0134875460.

3. Καλαφατούδης, Δροσίτης, Κοίλιας, Εισαγωγή στις Τεχνολογίες Πληροφορίας και Επικοινωνίας, ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ ΜΟΝ. ΕΠΕ, 2011, ISBN: 978-960-6759-69-7 (in greek).

4. Behrouz Forouzan, Foundations of Computer Science, 4η έκδοση, 2018, Cengage, ISBN: 9781473751040.

5. John V. Guttag, Introduction to Computation and Programming Using Python: With Application to Understanding Data, 2nd edition, 2016, MIT Press, ISBN 978-0262529624.

6. John Hennessy and David Patterson, COMPUTER ARCHITECTURE - A Quantitative Approach, 5th Edition, 2011, Morgan Kaufmann, eBook ISBN: 9780123838735.

7. Martin Erwig, ONCE UPON AN ALGORITHM: HOW STORIES EXPLAIN COMPUTING, 2017, The MIT Press, ISBN: 978-026-203-663-4.

(6)