COURSE OUTLINE

(1) GENERAL

SCHOOL	SCHOOL OF ENGINEERING		
ACADEMIC UNIT	COMPUTER ENGINEERING AND INFORMATICS		
LEVEL OF STUDIES	UNDERGRADUATE, OBLIGATORY		
COURSE CODE	CEID_NY131	SEMESTER	WINTER (1th)
COURSE TITLE	INTRODUCTION TO COMPUTERS AND PROGRAMMING		

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	CREITS
Lectures	4	4
Laboratory Exercises	2	2
Recitation sections	3	3
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).	9	9

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	general background, specialised general knowledge, skills development
PREREQUISITE COURSES:	There are no prerequisite courses. It is however recommended that students have at least a basic knowledge of mathematics.
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek. Instruction may be given in English if foreign students attend the course.
IS THE COURSE OFFERED TO ERASMUS STUDENTS	YES
COURSE WEBSITE (URL)	http://software.hpclab.ceid.upatras.gr/ https://www.ceid.upatras.gr/webpages/faculty/alexiou/eis_sys/ https://eclass.upatras.gr/courses/CEID1124/

(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*
Learning Outcomes: At the end of this course the student will: 1. have at an initial level been introduced into the world of computer science and computers 2. have understood the computer science topics he will face in the following years

3. have created the basic knowledge infrastructure, necessary for starting and successfully continuing his studies
4. have understood the applications of the various knowledge topics
5. be able design, implement, debug and document programs in C language
6. be able to apply methodologically and cooperative programming skills to C to solve difficult algorithmic problems.

At the end of the course the student will have further developed the following skills/competences:

1. ability to demonstrate knowledge and understanding of basic computing concepts
2. ability to apply this knowledge methodically to understand and solve practical problems.
3. ability to demonstrate knowledge and understanding of basic principles and structures of structured programming
4. ability to collaborate with others to solve programmatic problems.

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…
…….

Search for, analysis and synthesis of data and information, with the use of the necessary technology
Decision-making
Working independently
Team work
Production of new research ideas
Project planning and management
Criticism and self-criticism
Production of free, creative and inductive thinking

(3) SYLLABUS

Basic concepts of computer science. Information processing. Algorithm, Computer commands. Computer program. Computer systems structure. Organization and Operation of PCs. Ways of Representing Information. Basic Operations in Numeric Data. Types of addressing modes. Memory system. Peripheral devices. Introduction to assembly. Computer networks. Introduction to programming concepts (problem

algorithm program). Program execution procedure. Language programming definition. C a programming language. C program structure. Alphabet, vocabulary, constants, variables. Basic data types. Basic input-output commands. Operators and expression categories. Program flow control and structure of selection and repeat. Table and pointer type. Subprograms and functions in C. Advanced topics in functions (range, variable passing). Structures in C and file access. Introduction to python. basic programming concepts of Python. The course also contains a set of exercises in the programming language C and Python.
These exercises are carried out by students in order to consolidate what has been taught in theory.

(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*	We use Information and Communications Technology in communication with students. We use e_class, e_mail, forum. The course has a web site.
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*	*Activity* / *Semester workload* Lectures: 120 Laboratory Exercises: 60 Recitation sections: 80 Course total: *260*
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.*	(1) Written examination (2/3 of the final grade, in two separate exams one in the middle of semester and one final) (2) Exams in the laboratory in the part of C (1/3 of the total grade).

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

1. Behrouz A. Forouzan, Foundations of Computer Science (3rd Edition), Cengage Learning EMEA; 3rd edition edition (6 December 2013)

2. Herbert Schildt, C: The Complete Reference, May 17th 2000 by McGraw-Hill Education

3.Brian W. Kernighan, Dennis M. Ritchie, The C programming language Prentice Hall Press Upper Saddle River, NJ, USA ©1988

4. Haridimos Vergos, Lectures in Introduction in Computer Systems, University of Patras Department of Computer Engineering and Informatics, October 2006

- Related academic journals: