COURSE OUTLINE

1. GENERAL

SCHOOL	SCHOOL OF ENGINEERING		
DEPARTMENT	DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATICS		
LEVEL OF COURSE	UNDERGRADUATE, OBLIGATORY		
COURSE CODE	CEID_NNY103	**SEMESTER OF STUDIES**	WINTER (1h)
COURSE TITLE	INTRODUCTION TO 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 thewhole of the course, give the weekly teaching hours and the total credits*	**TEACHING HOURS PER WEEK**	**ECTS CREDITS**
Lectures	3	3
Laboratory Exercises	2	2
Recitation sections	2	4
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	General background, specialised general knowledge, skills development
PREREQUISITE COURSES:	There are no prerequisite courses.
TEACHING AND ASSESSMENT LANGUAGE:	Greek. Instruction may be given in English if foreign students attend the course.
THE COURSE IS OFFERED TO ERASMUS STUDENTS	YES
COURSE WEBPAGE (URL)	https://eclass.upatras.gr/courses/CEID1247/

2. LEARNING OUTCOMES

Leraning 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 been introduced into the world of computer science and computers
2. be familiar with the basic concepts of information representation and processing
3. have understood the basic concepts of procedural programming and how they are implemented by the C programming language
4. be able to design, implement, debug and document programs in the C language
5. be able to apply programming skills in C in order to solve algorithmic problems of non-trivial complexity methodologically and cooperatively

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 Abilities

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

Search for, analysis and synthesis of data and information, with the use of the necessary technology	*Project planning and management*
Adapting to new situations	*Respect for difference and multiculturalism*
Decision-making	*Respect for the natural environment*
Working independently	*Showing social, professional and ethical responsibility and sensitivity to gender issues*
Team work	*Criticism and self-criticism*
Working in an international environment	*Production of free, creative and inductive thinking*
Working in an interdisciplinary environment	*.....*
Production of new research ideas	*Others…*

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

3. COURSE CONTENT

Introductory programming concepts (problem, algorithm, program).
Definition a programming language. The programming language C.
Program structure in C.
Alphabet, vocabulary, constants, variables.
Basic data types.
Abstraction in processes, functions
Basic input / output processes - input / output functions
Program development and execution.
Operators and categories of expressions.
Program execution flow control. Instruction for program flow control.
Array and pointers data type
Subroutines and functions in C.
Advanced function issues (scope, passing arguments with reference). Recursive functions
C data structures and file access
Examples of simple data structures in C (stacks, lists etc.)
Special topics (C preprocessor, macros, libraries, Unix Programming Environment)

4. TEACHING AND LEARNING METHODS - ASSESSMENT

TEACHING METHOD *Face-to-face, Distance learning, etc.*	Face-to-face
USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES *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.

TEACHING ORGANIZATION

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.

Δραστηριότητα	***Φόρτος Εργασίας Εξαμήνου***
Lectures	75
Recitation sections	50
Laboratory exercises	100
Total number of hours for the Course (25 hours of work-load per ECTS credit)	225

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	
STUDENT ASSESSEMNT *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.*	• Written examination (60% of the total grade) • Laboratory Exercices (3-4 obligatory lab exercises, 40% of the total grade)

5. RECOMMENDED LITERATURE

1.Brian W. Kernighan, Dennis M. Ritchie, The C programming language. 2η/2008, Εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ ΕΠΕ

2. Abbey Deitel,Harvey Deitel, C Programming, 7η Έκδοση/2014, Χ. ΓΚΙΟΥΡΔΑ & ΣΙΑ ΕΕ

3.Γ. Σ. Τσελίκης - Ν. Δ. Τσελίκας, C: Από τη Θεωρία στην Εφαρμογή, ΓΕΩΡΓΙΟΣ ΤΣΕΛΙΚΗΣ, 3/2016

4. Θραμπουλίδης Κλεάνθης Χ. Διαδικαστικός προγραμματισμός – C ΕΚΔΟΣΕΙΣ Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε. 2η έκδ./2003

5.Νίκος Μ. Χατζηγιαννάκης, Η γλώσσα C σε βάθος. 5η/2017 ΕΚΔΟΣΕΙΣ ΚΛΕΙΔΑΡΙΘΜΟΣ ΕΠΕ

6. Abbey Deitel,Harvey Deitel, C Programming, 8th Edition /2016, Pearson