

COURSE OUTLINE

1. GENERAL

SCHOOL		SCHOOL OF ENGINEERING, UNIVERSITY OF PATRAS	
DEPARTMENT		DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATICS	
LEVEL OF COURSE		UNDERGRADUATE, COMPULSORY	
COURSE CODE		CEID_22Y103	SEMESTER OF STUDIES WINTER (1h)
COURSE TITLE		INTRODUCTION TO PROGRAMMING	
INDEPENDENT TEACHING ACTIVITIES <i>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</i>		TEACHING HOURS PER WEEK	ECTS CREDITS
Lectures		3	3
Laboratory Exercises		2	2
Recitation sections		2	4
<i>Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).</i>			
COURSE TYPE <i>general background, special background, specialised general knowledge, skills development</i>	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/CEID1504/ (2025_2026) https://eclass.upatras.gr/courses/CEID1456/ (2024_2025)		

2. LEARNING OUTCOMES

Learning outcomes <i>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.</i> <i>Consult Appendix A</i> • 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
Decision-making	Respect for difference and multiculturalism Adapting to new situations
Working independently	Respect for the natural environment
Team work	Showing social, professional and ethical responsibility and sensitivity to gender issues
Working in an international environment	Criticism and self-criticism
interdisciplinary environment	Production of free, creative and inductive thinking Working in an
Production of new research ideas	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. COURSE CONTENT

Introduction to basic concepts of imperative procedural planning. The process of assigning work to the machine. The program. Process of developing and executing a program through simple programs
 Computational Processes and Information.
 Description of computational processes using Structured Greek. Procedural abstraction. Algorithm.
 Basic concepts of Programming Languages.
 Data abstraction. Variables. Constants. Basic data types. Representation of information in C. Data types in C.
 Representation of computational processes in C. Simple computational processes - Operators. Advanced Computational Processes – Expressions and Functions. The C. Declaration, Definition, and Function Library Calls. Basic functions of the C library.
 Program structure C. The C preprocessor
 Program development and execution process. Incremental development.
 Program execution flow. Execution flow and control statements. Execution flow control statements in C.
 Arrays. Introduction to Indices and Structures in C.
 Scope. Pass parameters with reference. Store and recall information in/from files. Examples of simple data structures in C. Recursion

4. TEACHING AND LEARNING METHODS - ASSESSMENT

TEACHING METHOD <i>Face-to-face, Distance learning, etc.</i>	Face-to-face	
USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES <i>Use of ICT in teaching, laboratory education, communication with students</i>	We use Information and Communications Technology in communication with students. We use e_class, e_mail.	
TEACHING ORGANIZATION <i>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.</i>	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

<i>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</i>	
STUDENT ASSESSEMENT <i>Description of the evaluation procedure</i> <i>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</i> <i>Specifically-defined evaluation criteria are given, and if and where they are accessible to students.</i>	• Written examination (60% of the total grade) • Laboratory Exercises (3-4 obligatory lab exercises, 40% of the total grade)

5. RECOMMENDED LITERATURE

1. Μία Εισαγωγή στην Τέχνη του Προγραμματισμού (Με την C και το AI Chatbot), C , Θραμπουλίδης Κλεάνθης Χ. ΕΚΔΟΣΕΙΣ Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε
2. Διαδικαστικός προγραμματισμός– C , Θραμπουλίδης Κλεάνθης Χ. ΕΚΔΟΣΕΙΣ Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε. 2η έκδ./2003
3. C Προγραμματισμός, Abbey Deitel, Harvey Deitel, 9η Έκδοση/2023, Χ. ΓΚΙΟΥΡΔΑ & ΣΙΑ ΕΕ
4. Η Γλώσσα Προγραμματισμού C. 2η/2008, Brian W. Kernighan, Dennis M. Ritchie, Εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ ΕΠΕ
5. Από τη Θεωρία στην Εφαρμογή Έκδοση: 4/2023 Συγγραφείς: Γ. Σ. Τσελίκης - Ν. Δ. Τσελίκας