STRUCTURED PROGRAMMING

COURSE OUTLINE

Responsible: Nikolaos Spanoudakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electronic Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0806.1.001.0	**SEMESTER**	1st
COURSE TITLE	Structured Programming		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	5	5
Total	**5**	**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)	https://eclass.hmu.gr/courses/EE369/

2. LEARNING OUTCOMES

Learning outcomes

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

- Find solutions to problems of moderate difficulty, describe the algorithmic solutions in pseudo-code and/or in a flowchart, and write the C code.
- Evaluate algorithmic solutions.
- Design and implement software applications that access text files.
- Design and write code for programs that require the use of vectors or arrays whose elements are structured items.
- Use a sorting or searching technique appropriate to the situation.

General Competences

- Searching, analysis and synthesis of data and information, with the use of the necessary technologies
- Decision-making
- Team work
- Time management
- Project planning and management
- Oral communication
- Criticism and self-criticism

- Promotion of free, creative and inductive thinking
- Use of computers

3. SYLLABUS

Introduction to Informatics and Computers.

Introduction to the parts of the computer (hardware) and the computer programs (software).

Numbering systems and conversions from one system to another.

The concept of the algorithm. Algorithm structures. Logic diagrams (flowcharts).

Programming in the C language. Data types and sizes.

Control statements. Relational and logical operators.

Loop statements.

Functions and program structuring into blocks.

One-dimensional/multi-dimensional arrays. Address pointers. Recursion and recursive functions.

Ways of defining data structures and data unions as well as the operators used to access their members.

Introduction to search techniques (Sequential Search, Binary Search) and sorting (Sort by Selection, Bubble Sort).

File access.

Learning the basic principles of designing and implementing programs in an Integrated Development Environment (e.g. CODE::BLOCKS, Dev-C++, VS code).

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Face-to-face theoretical teaching. Laboratory training in groups of students (20 per group). Tutorial exercises in small groups of students.
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Use of slide presentation software. Use of an integrated software development environment (IDE) such as CODE::BLOCKS or VS code. Communication with students via an asynchronous e-learning platform (eclass).
TEACHING METHODS The manner and methods of teaching are described in detail.	

Activity	**Semester workload**
Lectures	26
Laboratory exercises	13
Tutorial Exercises	26
Group assignment carried out in small groups	35
Self-study	50
Course total	**150**

STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	I. Written final examination (FE) (40-80%) - Problem solving - Multiple-choice questions - Short-answer questions II. Laboratory examination (LE) (20%) - Multiple-choice questions - Short-Answer Questions - Problem Solving III. Written mid-term test (MT) (20%) - Problem solving - Multiple-choice questions - Multiple-choice questions IV. Group assignment (GA) (20%) - Marking of the Submitted Assignment - Oral Examination - Public Presentation The course grade (FE*0.4 + LE*0.2 + MT*0.2 + GA*0.2) must be at least five (5). Students must have no more than one absence from the laboratory exercises. The FE can range from 40 to 80 percent in cases where students have missed an item (MT or GA) with the needed justification. Laboratory examination (LE) is compulsory and cannot be ommitted (20%). The assessment criteria are available to students on the course website and are announced in the first lecture.

5. ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

Greek or translated textbooks:

- N. Hatzigiannakis, The C language in depth, Kleidarithmos Publications, 5th revised edition, 2017.
- G. S. Tselikis and N. D. Tselikas, C: From Theory to Practice, 4th edition, 2023.
- Bradley L. Jones, Peter Aitken,Dean Miller, Complete Manual of C, 7th Ed., X. GKIOURDAS & CO. LP, 2015Paul Deitel, Harvey Deitel, C Programming, 9th Edition, X. GKIOURDAS & CO. LP, 2023.
- Paris Mastorokostas. Procedural programming. The C language. KALLIPOS Open Academic Editions, 2015

Foreign-language textbooks:

- Israel Gbati. Bare-Metal Embedded C Programming: Develop high-performance embedded systems with C for Arm microcontrollers. Packt Publishing, 2024
- Robert C. Seacord. Effective C, 2nd Edition: An Introduction to Professional C Programming. No Starch Press, 2024