STRUCTURED PROGRAMMING

COURSE OUTLINE

Responsible: Maximos Kaliakatsos-Papakostas

1. GENERAL			
SCHOOL	School of Music and Optoacoustic Technologies		
ACADEMIC UNIT	Department of Music Technology and Acoustics		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0807.2.004.1	**SEMESTER**	2nd
COURSE TITLE	Structured Programming		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	7
Total	**0**	**7**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Background
PREREQUISITE COURSES	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/SMOT115/

2. LEARNING OUTCOMES

Learning outcomes

The aim of the course is to introduce the design and development of computer programs based on the principles of structured programming. The course content is intended to familiarize students with the fundamental concepts of computer programming, provide an understanding of how computer programs are executed, and develop proficiency in the C programming language.

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

Develop programming skills and implement software using the C programming language.

Understand the fundamental principles of designing and implementing programs based on the structured programming paradigm.

Solve computational problems using computers.

Successfully undertake advanced courses in the curriculum that require computer programming knowledge and skills.

General Competences

The course aims to develop the following general competencies:

Ability to search for, analyze, and synthesize data and information using appropriate technologies.

Ability to adapt to new situations.

Ability to work independently.

Ability to work effectively as a member of a team.

Ability to foster free, creative, and inductive thinking.

3. SYLLABUS

Computer operation. Computer architecture and memory organization. Data flow within a computer system. Instruction execution.

Software development. Software engineering. Software project life cycle. The phases of analysis, design, testing, and maintenance.

Software and programming languages. Source code and executable programs. The programming environment. Program compilation, debugging, and execution.

Structured programming. The importance of program structure. Fundamental principles and techniques of structured programming.

Algorithms: general concepts. Stepwise algorithms. Flowcharts. Pseudocode. Algorithmic problem solving. Searching and sorting algorithms.

The C programming language: characteristics and capabilities. Structure of C programs. Functions in C. Mathematical functions in C. Introductory concepts.

Data representation: characters, integers, and floating-point numbers. Basic data types, constants, variables, and the assignment operator. Number systems.

Input/output functions.

Operators: arithmetic, relational, and bitwise operators. Boolean expressions, relational expressions, logical expressions, and operator precedence. Compound operators. Pointers and memory addresses.

Program flow control structures. Nested control structures.

Iterative control structures (loops). Nested loops.

Functions in C: definition, declaration, and invocation. Returning values from functions. Function types. Passing addresses to functions. Storage classes. Automatic, external, and static variables. Variable scope and lifetime. Recursive functions.

One-dimensional arrays: declaration, initialization, input, and output. Array processing techniques. Character strings. String manipulation. Multidimensional arrays. Pointers and arrays. Arrays as function arguments.

Enumerations, structures, and unions.

File handling. File access functions.

Functions for dynamic memory management.

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	Eclass. Discussion Forum.
TEACHING METHODS The manner and methods of teaching are described in detail.	**Activity** | **Semester workload** **Course total** |

STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Intermediate exercises (50%), intermediate tests (30%), final project (20%).

5. ATTACHED BIBLIOGRAPHY