COURSE OUTLINE

(1) GENERAL

SCHOOL	SCHOOL OF ENGINEERING		
ACADEMIC UNIT	DEPARTMENT OF ELECTRONICS ENGINEERING		
LEVEL OF STUDIES	UNDERGRADUATE		
COURSE CODE	2601005	**SEMESTER**	1
COURSE TITLE	Structured 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**	**CREDITS (ECTS)**
Lectures	2	4
Laboratory	2	
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 Course
PREREQUISITE COURSES:	None
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek
IS THE COURSE OFFERED TO ERASMUS STUDENTS	YES (in English)
COURSE WEBSITE (URL)	http://eclass.teipir.gr/openeclass/courses/ENGI126/

(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*

The course objective is to introduce students in the algorithmic way of thinking and problem solving by computers. Issues addressed in class are: the notion of algorithm, data representations, algorithm design methods, algorithmic problem solving. Students learn the fundamental principles of structured programming. Typical characteristics and mechanisms of a structured programming language are introduced and students are introduced to the design and development of structured programs in this language. C programming language is used as the course basis. Lectures are completed by lab practice where theoretical knowledge is applied in an appropriate software environment.

Upon successful completion of this course, the students possess advanced knowledge, skills and competences in Structured Programming that enable them to:

- Understand and explain the basic design principles for algorithms,

- Understand basic computer programming principles, distinguish them and classify them,
- Know a substantial number of basic algorithms and use them in problem solving,
- Know the C programming language and use it to write original code for problem solving,
- Know the tools for software development in C and use them to analyse complex problems, to construct solutions (algorithms) and to code them in C,
- Collaborate within a team that develops algorithms and application in C.

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	*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 the necessary technology
- Working independently
- Team work

(3) COURSE CONTENT

Lectures:

1. Introduction to programming electronic systems.
2. Introduction to “C” lang.
3. Control structures and loops.
4. Arrays.
5. Pointers.
6. Strings.
7. Functions.
8. Algorithms I
9. Structures.
10. Algorithms II.
11. Recursion
12. Algorithms III.

Laboratory Experiments:

1. Control structures and loops, programs I
2. Control structures and loops, programs II
3. Arrays, programs I
4. Arrays, programs II
5. Pointers, programs
6. Strings, programs
7. Functions, programs I
8. Functions, programs II
9. Structures, programs I

10. Structures, programs II
11. Recursion, programs I.
12. Recursion, programs II.
13. Advanced programs.

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	Face to face lectures
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	• Use of electronic presentation with multimedia content in class, • Student support through the course webpage and the departmental e-learning platform (e-class), • Electronic communication of instructors and students, through the course webpage and by e-mail. • Use of C / C++ programming environment.
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*	Lectures, laboratory practice, assignments, study. ***Activity*** / ***Semester workload (hours)*** Lectures: 26 Study for lectures - assignments: 26 Laboratory practice: 26 Reports on lab practice: 26 Study and preparation for exams: 16 **Course Total**: ***120***
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.*	**Final grade** = Theory part grade x 60% + Lab part grade x 40% ***Theory Part grade:*** Final written exam (100%) ***Lab part grade:*** Average of all grades received at each weekly Lab Experiment

(5) ATTACHED BIBLIOGRAPHY

Essential reading

1. The “C” programming language, B. W. Kernighan, D. M. Ritchie
2. The Art of Computer Programming, D. E. Knuth

Recommended Books

1. C, from theory to practice, G. Tselikis and N. Tselikas, (in Greek)
2. C language in depth, N. Chatzigiannakis (in Greek).
3. Learn C language, D. Karolidis (in Greek).