COURSE OUTLINE

(1) GENERAL

SCHOOL	ENGINEERING				
ACADEMIC UNIT	ELECTRICAL AND COMPUTER ENGINEERING DEPT.				
LEVEL OF STUDIES	Undergraduate				
COURSE CODE	ECE_K150 SEMESTER		1		
COURSE TITLE	COMPUTER PROGRAMMING TECHNIQUES				
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	
Lectures		tures	2		
Seminars / Practice exercises		rcises	-		
Laboratory		atory	2		
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (4).		1	4		5
COURSE TYPE general background, special background, specialised, general knowledge, skills development	General background, S	Skills D	evelopment		
PREREQUISITE COURSES:					
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek				
IS THE COURSE OFFERED TO ERASMUS STUDENTS	No				
COURSE WEBSITE (URL)	https://eclass.uop.gr/courses/ECE101/				

(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

Course content aims to provide students with an understanding of the usefulness of different programming languages and their philosophy. During the course, students will get familiarized with the basic principles of computer programming and the various programming paradigms through their practical involvement with different programming languages, and they will acquire the necessary background for the relevant courses that follow in the next semesters.

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

- At the knowledge level:
 - have a clear picture of the usefulness of computer programming and the skills that they will develop while dealing with it.
 - understand the different categories of programming languages, the classification criteria, and the choice of their use to solve a particular problem.
 - recognize the various programming paradigms, as well as acquire the ability to serve them using different programming languages.

- understand in-depth the basic programming principles and create programs in the programming languages taught, intending to solve a specific problem.
- Understand the great usefulness of testing in the programs they create and be able to write the corresponding code to automate this provess.
- At the skill level:
 - make good use of variables, understand their relationship to computer memory, recognize their scope, and consistently use the data types of different programming languages.
 - use top-down design while programming and correctly use the ability to comment while coding.
 - organize the program code it writes into separate files.
 - At the level of abilities:
 - use branching commands and loops in the programs they develop.
 - create functions when programming and become familiar with the concept of objectorientation and how this model is applied in different programming languages while in parallel being familiar with the concepts of abstraction, inheritance, encapsulation, polymorphism, and synthesis.
 - create World Wide Web documents in HTML 5 language and format them using CSS3, as well as develop WWW applications.

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? Project planning and management Search for, analysis and synthesis of data and information, with the use of the necessary technology Respect for difference and multiculturalism Adapting to new situations Respect for the natural environment Decision-making Showing social, professional and ethical responsibility and Working independently 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
- Production of free, creative and inductive thinking
- Creation of new research ideas

(3) SYLLABUS

This course is an introduction to the basic concepts of computer programming for the department's students.

The course is developed in 13 lectures.

- i. An Introduction to Programming: Why to write software, Programming Language Categories, Selection Criteria.
- ii. An introduction to the Scratch Programming Language: Programming with Visual Tools, Troubleshooting – Choosing Alternatives, Logic Diagram, and Program Structure, Command Sequence, Repetition, Coordinate Systems, and Direction.
- iii. Utilizing Branch Commands, Conditions, Repetition, and Loops (forever, counting, and conditional loops),
- iv. Memory and Variables, Commenting the Code, Testing, and Design Patterns.
- An Introduction to JavaScript Programming Language: Why JavaScript, Utilizing Node.js,
 Programming Languages Syntax, and Best Practices in Code Writing, Variables & Data Types,
 An Introduction to Functions.
- vi. Functions, Parameters, Variable Scope, Program Flow Control with Branch Commands and Loops, Operators.

vii.	Objects, Organizing code in files, Testing through NodeUnit, Utilizing JavaScript for Web
	Applications – Express.js.
viii.	An Introduction to Python Programming Language: Abstraction, Why Python, Variables $\&$
	Data Types, Branch Commands, Loops, Functions, A Comparison with other Programming
	Languages.
ix.	Top-Down Design, Functions, Lists, Strings, Tuples, Sets, and Dictionaries, Utilizing Data
	Structures in Programming.
х.	Object-Oriented Programming Concepts, Objects, Creating Objects – Instantiation, Classes,
	Methods, Attributes, Variables, and Objects.
xi.	Inheritance, Encapsulation, Polymorphism, Synthesis, Modules, and Packages.
xii.	An Introduction to HTML5 and CSS3: The World Wide Web, Markup Languages, Web Page
	Structure, HTML Syntax, HTML Elements and Tags, Attributes, Cascading Style Sheets,
	Syntax, CSS Box Model.
xiii.	New Elements, New JavaScript APIs, Web Applications.

(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	 Learning process support via the electronic e-class platform. PowerPoint presentation with examples and practice exercises in the classroom. Laboratory Exercises using a PC. 		
TEACHING METHODS	Activity	Semester workload	
aescribea in aetail. 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 Study of lectures and bibliography Preparation of laboratory exercises Course Total	26 26 52 21 125 hours (5 ECTS)	
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	Written final exam (up to60% -80% of the final grade) and lab project (20% -40% depending on the degree of difficulty) which includes:		
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	 Theory evaluation. Short answer questions. Multiple choice questions. Problem-solving related to: Coding in the programming languages taught to achieve the desired result. Proper use of variables and their scope, as well as data types. 		

Specifically-defined evaluation criteria are given, and if and where they are accessible to students.	•	Correction of syntactic and logical errors within
	•	programs. The validity of the correct use of objects in
		different programming languages.
	•	Utilization of programming languages for the
		development of World Wide Web applications.

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

- Scott, Michael L. (2015) [2000]. Programming Language Pragmatics (4th ed.). Morgan Kaufmann Publishers. ISBN 978-0-12-410409-9.
- Brooks Webber Adam, «Modern Programming Languages: A Practical Introduction», ISBN-13: 978-1887902762

- Related academic journals: